NASA/TP—2000-207428

Reliability and Maintainability (RAM) Training

Edited by
Vincent R. Lalli
Glenn Research Center, Cleveland, Ohio

Henry A. Malec
Siemens Stromberg-Carlson, Albuquerque, New Mexico

Michael H. Packard
Ratheon Engineers and Constructors, Cleveland, Ohio

National Aeronautics and
Space Administration

Glenn Research Center

July 2000

The NASA STI Program Office . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program Office plays a key part in helping NASA maintain this important role.

The NASA STI Program Office is operated by Langley Research Center, the Lead Center for NASA's scientific and technical information. The NASA STI Program Office provides access to the NASA STI Database, the largest collection of aeronautical and space science STI in the world. The Program Office is also NASA's institutional mechanism for disseminating the results of its research and development activities. These results are published by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA's counterpart of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or cosponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services that complement the STI Program Office's diverse offerings include creating custom thesauri, building customized data bases, organizing and publishing research results . . . even providing videos.

For more information about the NASA STI Program Office, see the following:

- Access the NASA STI Program Home Page at ***http://www.sti.nasa.gov***

- E-mail your question via the Internet to help@sti.nasa.gov

- Fax your question to the NASA Access Help Desk at (301) 621-0134

- Telephone the NASA Access Help Desk at (301) 621-0390

- Write to:
  NASA Access Help Desk
  NASA Center for AeroSpace Information
  7121 Standard Drive
  Hanover, MD 21076

Acknowledgments

In 1993 the Orlando Division of the Martin Marietta Company recognized the need to provide its engineers, especially its design engineers, with a practical understanding of the principles and applications of reliability engineering. To this end, a short, informative reliability training program was prepared. The author of this company-sponsored effort was Richard B. Dillard, who was also the principal instructor.

In response to the students' enthusiasm, their quest for additional information, and the support of their supervisors and managers, Mr. Dillard researched and wrote chapters 2 to 6 and appendix A of this text. Robert E. Kastner and Henry N. Hartt were coauthors for our training manual on interface definition and control.

Credit is also due to William L. Hadley, who was the stimulus for many of the ideas presented, and to Dr. D.C. Schiavone and William P. Wood, who directed and supported the efforts that went into preparing this material.

Thanks are extended to Frank E. Croxton and Prentice-Hall, Inc. for the use of two-tail and one-tail tables of the normal distribution and to Arthur Wald and John Wiley & Sons, Inc. for the use of tables of the cumulative normal distribution.

In recognition of the need to help project managers better understand safety and assurance technologies, Gary G. Kelm, Frank J. Barber, and Frank J. Barina prepared appendix B.

Kam L. Wong, using information and concepts from Charles Ryerson and Irwin Quart, prepared chapter 1 in our previous workbook, RP–1253; thanks are extended to North-Holland, Inc. for permission to reprint some of the figures and text. Thanks to Fredrick D. Gregory, Dr. Michael A. Greenfield, and Dr. Peter Rutledge, Vernon W. Wessel, and Frank Robinson, Jr. for their encouragement and support in allowing the Professional Development Team to develop this new workbook for our NASA Safety Training Course 017.

Henry A. Malec has passed away and will be missed. He will be remembered for his hard work in promoting the Reliability Society. He prepared chapters 7, 10, and 11 of the original version of this text. Martha Wetherholt and Tom Ziemianski prepared chapters 8 and 9. Thanks are extended to the Digital Press of Digital Equipment Corporation for the software evaluation materials contained in chapter 7. Vincent R. Lalli, presently a risk management consultant at the NASA Glenn Research Center in Cleveland, Ohio, prepared some of the new sections and appendix C, added some of the problems, and edited and cared for the final NASA printing of this revised version of the manual.

E.A. Winsa and Gary G. Kelm served as the final NASA project office reviewers. Their suggestions improved the usefulness of the text for flight projects.

To supplement the material presented herein, the bibliography at the end of this manual will enable the reader to select other authoritative material in specific reliability areas.

The editors, Vincent R. Lalli, Henry A. Malec, and Michael H. Packard would like to thank the many members of the IEEE Reliability Society Administrative Committee for their help in developing this text.

Available from

NASA Center for Aerospace Information
7121 Standard Drive
Hanover, MD 21076
Price Code: A16

National Technical Information Service
5287 Port Royal Road
Springfield, VA 22100
Price Code: A16

Preface

What Does Reliability Mean?

Systems . . .

The word "reliability" applies to systems that consist of people, machines, and written information.

A system is reliable—that is, has good reliability—if the people who need it can depend on it over a reasonable period of time. People can depend on a system if it reasonably satisfies their needs.

People . . .

The views of the people involved in a system are different and depend on their responsibilities; some rely on it, others keep it reliable, and others do both. Consider an automatic grocery checkout system and the people involved:

- The owners, who are the buyers
- The store manager, who is responsible for its operation
- The clerk, who operates it
- The repair person, who maintains it in working condition
- The customer, who buys the products

Machines . . .

A grocery checkout system may comprise several types of machines. It has mechanical (conveyor belt), electrical (conveyor belt motor, wiring), electronic (grocery and credit card scanners, display screen, and cash register), and structural (checkout counter, bag holder) parts.

Written Information . . .

Several types of written information contribute to the way people rely on a system:

- The sales literature
- The specifications
- The detailed manufacturing drawings
- The software user's manual, programs, and procedures
- The operating instructions
- The parts and repair manual
- The inventory control

Reliability . . .

People rely on systems to

- Do work or provide entertainment
- Do no unintentional harm to users, bystanders, property, or the environment
- Be reasonably economical to own and to repair
- Be safe to store or dispose of
- Accomplish their purposes without failure

What Does Reliability Engineering Mean?

Reliability engineering means accomplishing specific tasks while a system is being planned, designed and developed, manufactured, used, and improved. These tasks are not the usual engineering and management tasks but are those that ensure that the system meet the users' expectations—not only when it is new but as it ages and requires repeated repairs.

Why Do We Need Reliability Engineering?

Technology users have always needed reliability engineering, but it has only developed since the 1940's as a separate discipline. Before the Industrial Revolution, most of the reliability details were the individual worker's responsibility because the machines, products, and tools were relatively simple. However, shoddy goods were produced—wheels that broke easily, farming implements that were not dependable, lumber that rotted prematurely.

As technology rapidly changed, systems became large and complex. Companies that produce these systems must likewise be large and complex. In such situations, many important details that affect reliability are often relegated to a lower priority than completing a project on time and at an affordable cost. Among the first to see the need for a separate reliability discipline were the telephone and electric power utilities and the military.

Contents

Chapter 1
Historical Perspective of Space System Reliability

Summary

The NASA Strategic Plan (ref. 1–1) is the backbone of our new Strategic Management System, an important aspect of which is risk management. Coincident with a decreasing NASA budget is the new working environment that demands a better, faster, and cheaper way to conduct business. In such an environment where risk is considered a knowledge-based resource, mission assurance has come to play an important role in our understanding of risk.

Through the years, much of mission assurance has been aimed at increasing independent systems engineering and further refining basic design approaches. Now the time has come to direct our attention to managing the risks that come from system interactions during a mission. To understand such risks, we must bring to bear all the engineering techniques at our disposal. Mission assurance engineers are entering the era of interaction in which engineering and system engineering must work closely to achieve better performance on time and within cost.

A structured risk management approach is critical to a successful project. This is nothing new. A risk policy must be integral to the program as part of a concurrent engineering process, and risk and risk drivers must be monitored throughout. Risk may also be managed as a resource: the new way of managing better, faster, cheaper programs encompasses up-front, knowledge-based risk assessment. The safety and mission assurance (S&MA) community can provide valuable support as risk management consultants.

Past Space System Reliability

Ever since the need for improved reliability in space systems was recognized, it has been difficult to establish an identity for mission assurance engineering. Attempts to delineate an independent set of tasks for mission assurance engineering in the 1970's and 1980's resulted in the development of applied statistics for mission assurance and a large group of tasks for the project. Mission failures in a well-developed system come from necessary risks that remain in the system for the mission. Risk management is the key to mission assurance. The traditional tasks of applied statistics, reliability, maintainability, system safety, quality assurance, logistics support, human factors, software assurance, and system effectiveness for a project are still important and should still be performed.

In the past, mission assurance activities were weakly structured. Often they were decoupled from the project planning activity. When a project had a problem (e.g., a spacecraft would not fit on the launch vehicle adapter ring), the mission assurance people were involved to help solve it. Often problems were caused by poorly communicated overall mission needs, a limited data base available to the project, tight funding, and a limited launch window. These factors resulted in much risk that was not recognized until it happened. The rule-based management method used by NASA recognized risk as a consequence and classified four types of payloads: A, B, C, and D. These were characterized as high priority, minimum risk; high priority, medium risk; medium priority, medium-high risk; and high risk, minimum cost. Guidelines for system safety, reliability, maintainability and quality assurance (SRM&QA) project requirements for class A-D payloads were also spelled out. An example is the treatment of single failure points (SFP): class A, success-critical SFP's were not permitted; class B, success-critical SFP's were allowed without a waiver but were minimized; class C, success critical SFP's were allowed without a formal waiver; class D, the same as class C.

Often risk came as a consequence of the mission. In an attempt to minimize risk, extensive tests and analyses were conducted. The residual risk was a consequence of deficiencies in the tradable resources of mass, power, cost, performance, and schedule. NASA tried to allocate resources, develop the system, verify and validate risk, launch the system, and accomplish the mission with minimal risk. Using these methods resulted in a few failures.

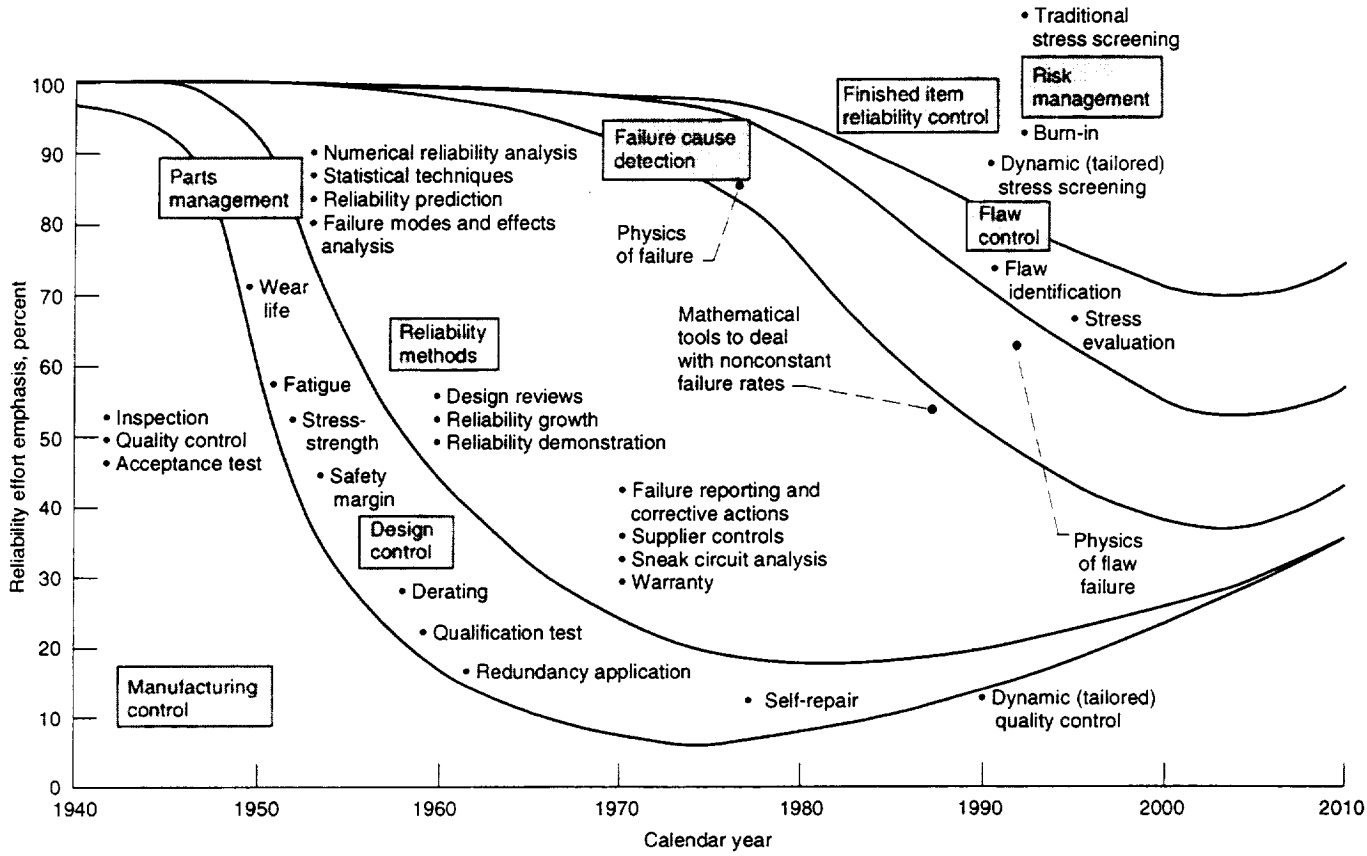


Figure 1–1.—Distribution of reliability emphasis with respect to calendar years (updated from ref. 6; original figure prepared by Kam L. Wong).

Various reliability efforts were grouped into categories: manufacturing control, design control, reliability methods, failure-cause detection, finished item reliability, flaw control, and risk management. Figure 1–1 illustrates how these categories have been emphasized through the years. The construction of figure 1–1 is approximate because its purpose is to identify activities, not to classify efforts precisely. Note that specific mission assurance activities are changing and that the amount of effort expended in these may not be proportional to the emphasis given them. A good parts management program is always important. The decrease in the use of reliability methods does not mean that parts management is unimportant; it only reflects that the importance of parts management has been well established and that parts management has become a standard design control task as part of a project.

Risk Management in the Revised NASA

The new NASA handbook on the Management of Major Systems and Programs is divided according to the four parts of the program life cycle: formulation, approval, implementation, and evaluation. It stresses risk management as an integral part of project management. The Formulation section defines a risk management-risk assessment process and requires that all projects use it. All risks must be dispositioned before flight.

The definition of risk management (ref. 1–2) is "An organized, systematic decision-making process that efficiently identifies risks, assesses or analyzes risks, and effectively reduces or eliminates risks to achieving the program goals." It also explains that effective project management depends on a thorough understanding of the concept of risk, the principles of risk management, and the establishment of a disciplined risk management process, which is shown in figure 1–2. The figure also explains the risk management plan requirements. A completed risk management plan is required at the end of the formulation phase and must include risk management responsibilities: resources, schedules, and milestones; methodologies: processes and tools to be used for risk identification, risk analysis, assessment, and mitigation; criteria for categorizing or ranking risks according to probability and consequences; the role of decisionmaking, formal reviews, and status reporting with respect to risk management; and documentation requirements for risk management products and actions.

A new direction for mission assurance engineers should be to provide dynamic, synthesizing feedback to those responsible for design, manufacturing, and mission operations. The feedback should take the form of identifying and ranking risk,

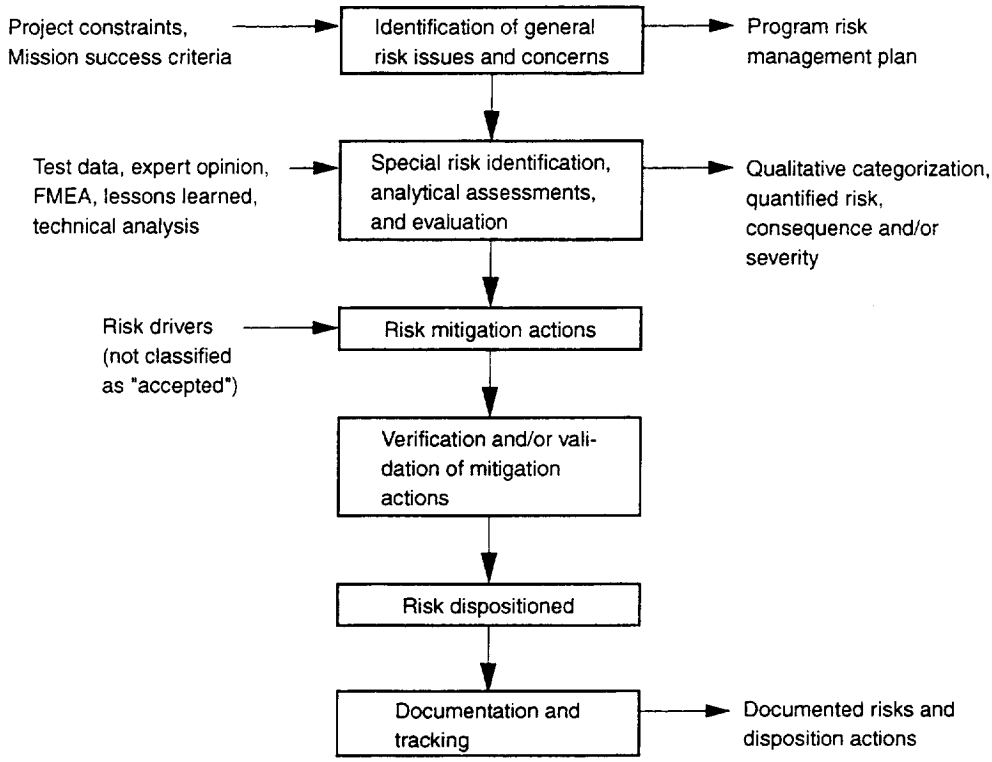


Figure 1–2.—Risk management process (ref. 3).

determining risk mechanisms, and explaining risk management techniques. Mission assurance and the project should work together to achieve mission success.

The Challenge of NASA's Brave New World

NASA and many other Government agencies have been forced to face a new workplace environment. With the NASA budget shrinking, the nature of projects has changed: many are fast track and have fixed prices, which means that they must be completed in a better, faster, and cheaper manner. The dollars once put into facilities are very limited; the spacecraft budgets are smaller so the development cycle time has been reduced to save money. NASA's solution to these constraints is to emphasize proactive risk management processes. The paradigm has to change from rule-based to knowledge-based decisions and new methods that will improve productivity. Figure 1–3 shows the total NASA Earth and Space Science project budgets that reflect the slogan "better, faster, and cheaper."

Risk as a Resource

NASA's new paradigm (ref. 1–3) requires that risks be identified and traded as a resource with an appropriate level of mitigation. The tradable resources have increased by one: risk, mass, power, schedule, performance, and cost. The resources are hardware allocated during development, and at the same time risks are addressed and traded off. When the adequacy is demonstrated, the spacecraft is launched, and the flight performance is accomplished with a recognized risk. As seen for rule-based activities, there may be some failures but there will be more spacecraft launches to learn from. Thus, the risk has been used as a resource process. The goal is to optimize the overall risk posture by accepting risk in one area to benefit another. A strategy to recover from the occurrence of risk must also be considered. Risk trades will be made (best incremental return), possible risk consequences evaluated and developed, and decision or recovery options accepted and tracked. How is the cost of risk reduced? Here it is important to consider its marginal cost. When the cost per "unit of risk reduction" in a given component or subsystem increases significantly—stop. It would be better to buy down risk somewhere else.

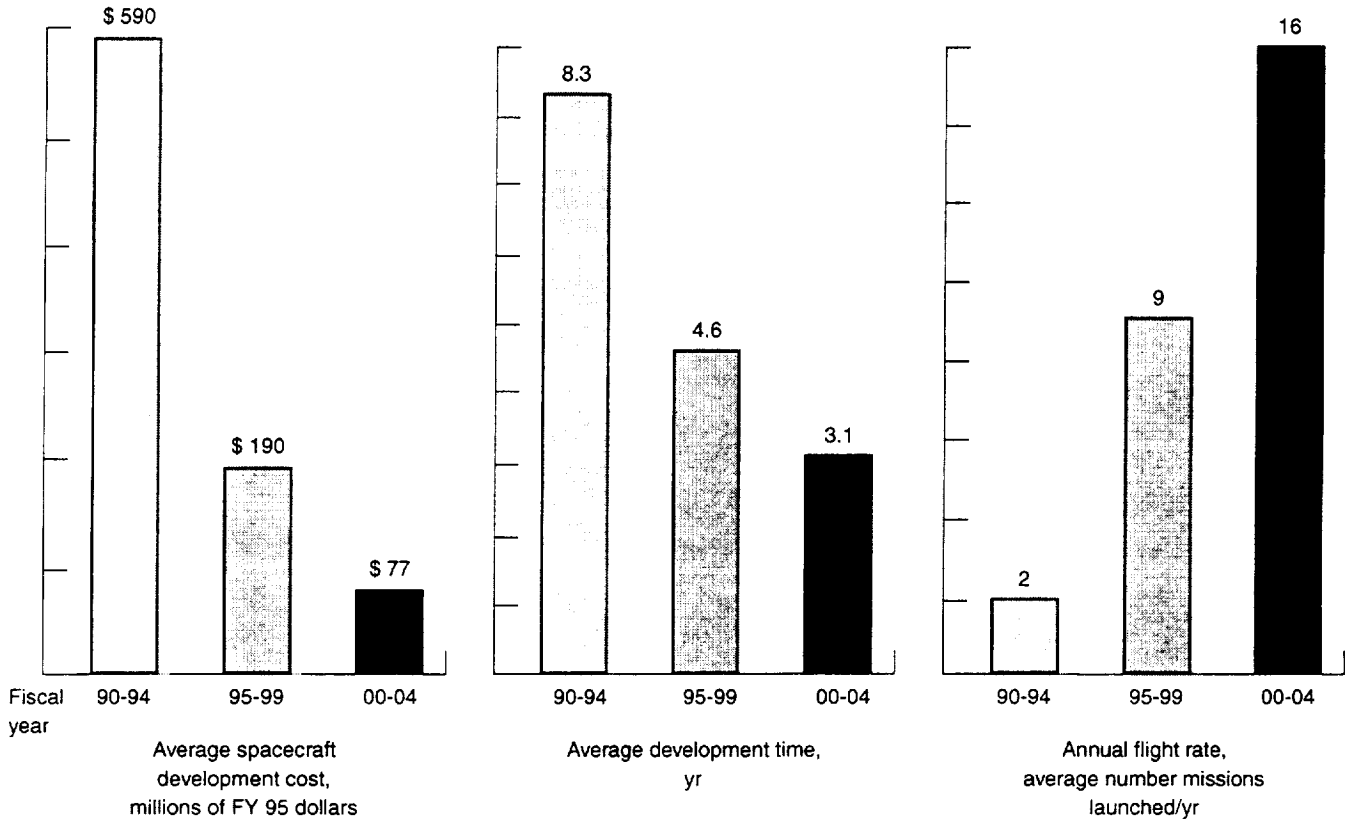


Figure 1–3.—Total NASA Earth and space science projects completed in better, faster, and cheaper environment (ref. 3).

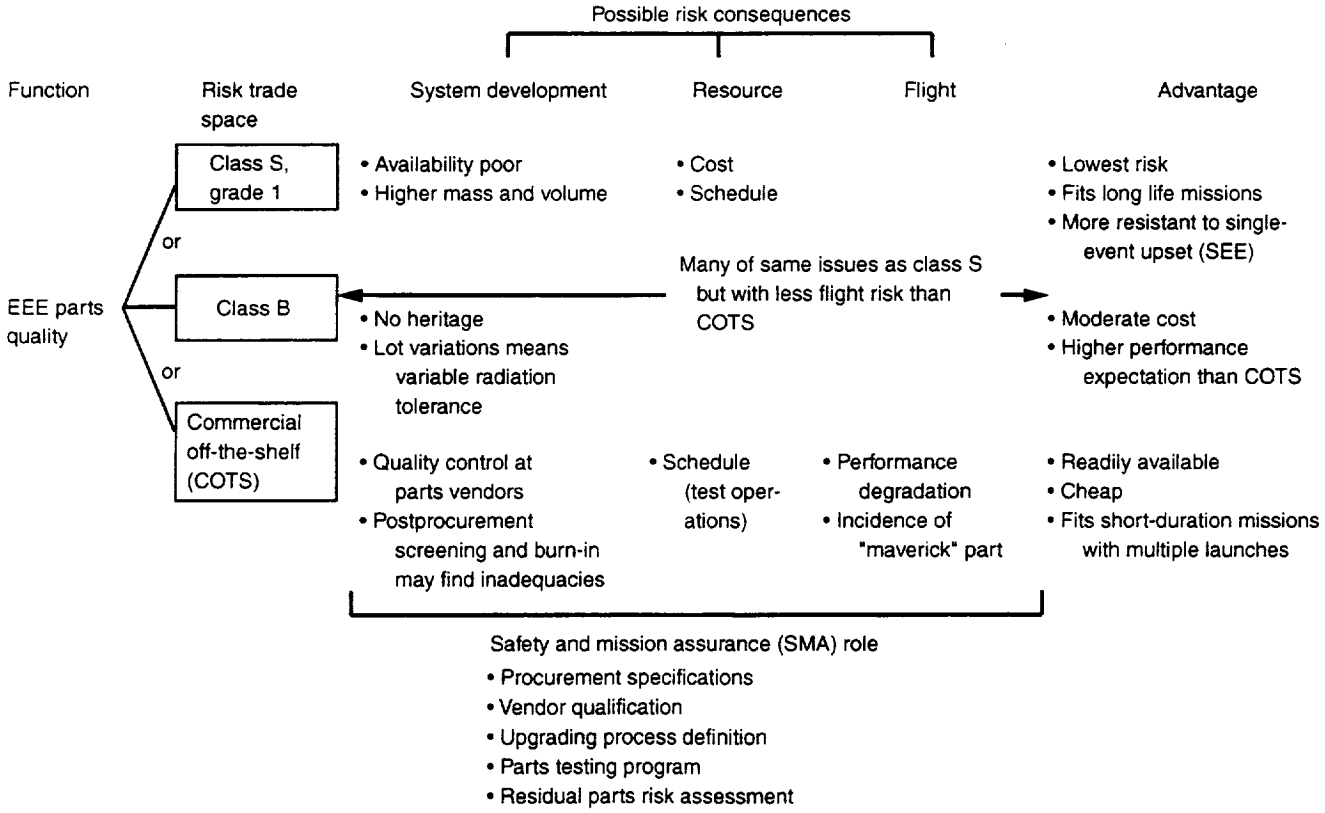


Figure 1–4.—Risk analysis for class EEE parts (ref. 3).

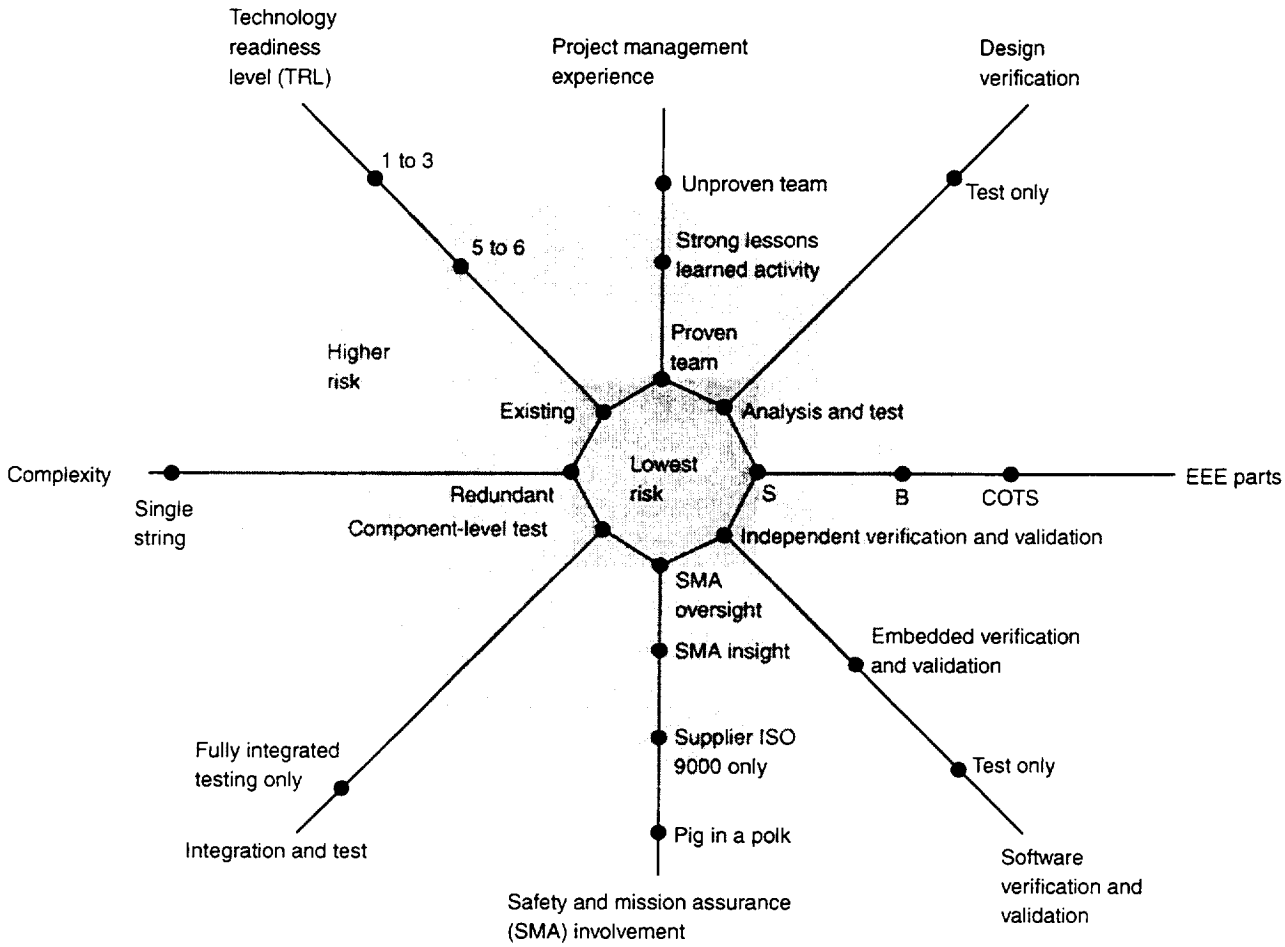


Figure 1–5.—Notational risk surface (ref. 3).

Dr. Greenfield, the Deputy Associate Administrator in the Office of Safety and Mission Assurance at NASA Headquarters, gave a risk management presentation and illustrated through six examples how to use risk as a resource (ref. 1–4). One of his examples dealt with the class of electrical, electronic, and electromagnetic (EEE) parts (ref. 1–5). Figure 1–4 shows the function, risk trade, possible risk consequence, and advantages for the class of parts to be used in a spacecraft. The risk trade that a project needs to make is the type of parts to use: class S, grade 1, class B, or commercial off-the-shelf (COTS) parts. Each has possible risk consequences. For example, class S, grade 1 parts have poor availability and are usually older technology, which means higher mass and volume. The advantages are that they are low risk, fit long-life missions, and are more resistant to single-event upset (SEE).

A measure of risk exists for a project that chooses to use a new technology, and it is now termed the technology infusion risk. The technology readiness level (TRL) scale ranges from 1 to 9. A TRL of 9 is used for existing, well-established, proven (very low-risk) technology. A TRL of 1 is used for unproven, very high-risk technology at the basic research stage. New technology can save time and money so there is a critical point at which it should be put to use. The diagram of figure 1–5 shows areas of high to low risk for the various risk elements. Called a risk surface (notational), if one looks along the EEE parts line, the commercial off-the-shelf parts (COTS) have more risk than B parts and B parts have more risk than S parts. Other risk elements are also shown in this figure.

The Role of Safety and Mission Assurance (SMA) in Risk Management

NASA's Safety and Mission Assurance (SMA) Office has the core competencies to serve as a risk management consultant to the projects and is supporting the risk management plan development. It provides projects with risk-resource tradeoffs: strategies, consequences, benefits, and mitigation approaches. Its role is to interact in all phases of the project decision process (planning, design, development, and operations). It provides projects with residual risk assessment during the project life cycle. Figure 1–6 shows the mission failure modes that cause

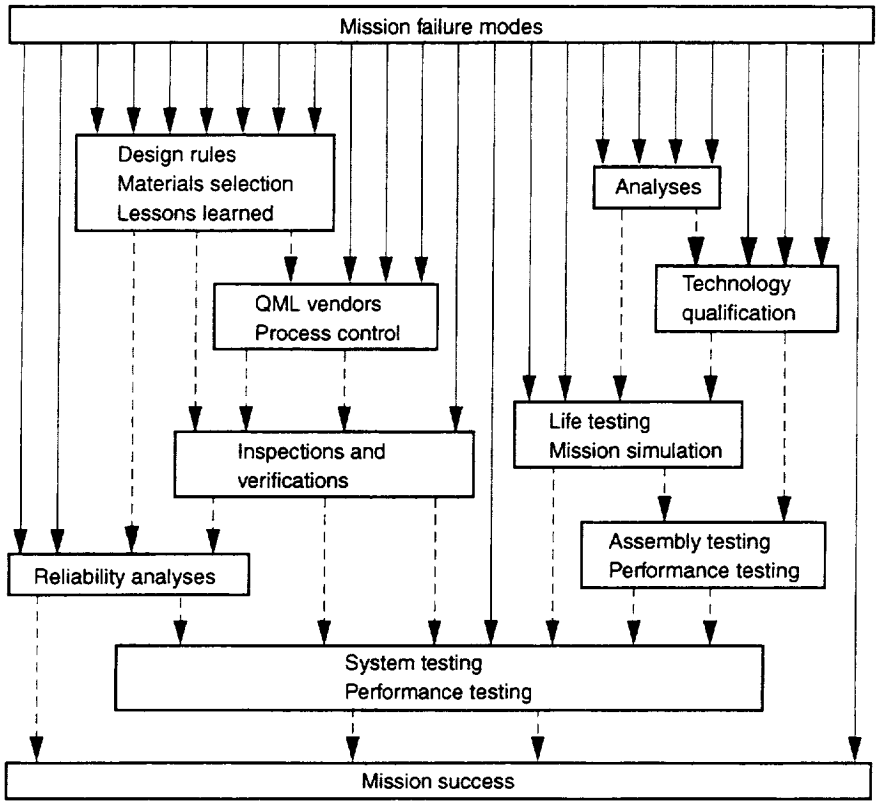


Figure 1–6.—Some mission failure modes and methods leading to mission success (ref. 3).

TABLE 1–1.—SAFETY AND MISSION ASSURANCE (SMA) ROLE IN RISK MANAGEMENT

SMA area	Typical areas involved in tradeoffs
Quality assurance	Documentation, surveillance, inspection, certification, audit, materials review board
Configuration control	Drawings, equipment lists, delivery schedules, approval authority, freeze control, as-built documentation
Environmental requirements	Design and test requirements, documentation, approvals, functional and environment tests, programmatics (component, subsystem, system), analysis
EEE parts	Parts lists, parts class, policy, nonstandard parts, traceability, derating, failure analysis, burn-in, selection, acquisition, upgrades, lot control, screening, destructive physical analysis, vendor control
Reliability	Single-failure-point policy, problem and failure reporting and disposition, design performance analysis (failure modes and effects criticality analysis, fault tree analysis, part stress, redundancy switching, worst case, single-event upset, reviews, redundancy
Systems safety	Documentation, hazard identification and/or impact, analysis (fault tree analysis, hazard, failure modes and effects criticality analysis, sneak circuit), structures and materials reviews, electrostatic discharge (ESD) control, tests, inspections, surveys
Software product assurance	Initiation, problem and failure reporting and disposition, simulations, independent verification and validation (IVV), tests

risk and some of the methods used to manage them so that mission success can be achieved. The SMA role in risk management is presented in table 1–1, which shows the SMA area and other typical areas involved in project tradeoffs. For example, with EEE parts, 16 tradeoff areas are identified to help the project understand parts management risks. SMA must take the lead to answer some very important questions: Where are the problems? What has been done about them? Have all the risks been mitigated? Are we ready to fly?

References

1–1. NASA Strategic Plan (NASA Policy Directive 1000.1), 1998 available at http://www.hq.nasa.gov/office/codeq/qdoc.pdf

1–2. Greenfield, Dr. Michael A.: Risk Management Risk as a Resource. NASA HQ, Washington, DC, 1998.

1–3. Lalli, Vincent R.: Reliability Training. NASA RP–1253, 1992.

1–4. Hoffman, Dr. Edward J.: Issues in NASA Program and Project Management. NASA SP–6101(11), 1996.

1–5. Recommended Practice for Parts Management: ANS BSR/AIAA R–100–1996.

1–6. Management of Major Systems and Programs: NASA NHB–\ 7120.5A, 1998 available at: http://www.hq.nasa.gov/office/codeq/qdoc.pdf

Reliability Training[1]

1. Which NASA Policy Guide explains risk management?

   A. 8701.draft 1 B. 7120.5A C. 2820.1

2. What challenge is NASA facing?

   A. The NASA budget is shrinking.
   B. Many projects are being done faster, cheaper, and better.
   C. Dollars are very limited for facilities.
   D. All of the above.

3. What are the tradeable resources that projects can use?

   A. Performance, cost, and schedule
   B. Mass, power, performance, cost, and schedule
   C. Risk, mass, power, performance, cost, and schedule

4. How should the projects use the Safety and Mission Assurance Office?

   A. Design consultants
   B. Systems consultants
   C. Risk management consultants

[1]Answers are given at the end of this manual.

Chapter 2
Reliability Mathematics and Failure Physics

Mathematics Review

Readers should have a good working knowledge of algebra and a familiarity with integral and differential calculus. However, for those who feel rusty, the following review includes solved examples for every mathematical manipulation used in this manual.

Notation

The Greek symbol Σ (sigma) means "take the sum of," and the notation

$$\sum_{i=1}^{n} x_i$$

means to take the sum of the x_i's from $i = 1$ to $i = n$.

The symbol $\sqrt[n]{x}$ means "take the n^{th} root of x." The square root $\sqrt[2]{x}$ is usually written as $\sqrt{x}$ without the radicand (the 2).

The Greek symbol Π (pi) means "take the product of," and the notation

$$\prod_{i=1}^{n} x_i$$

means to take the product of the x_i's from $i = 1$ to $i = n$.

The notation $x!$ is referred to as a factorial and is a shorthand method of writing $1 \times 2 \times 3 \times 4 \times 5 \times 6 \times \ldots \times x$ or in general as $x! = x(x - 1(x - 2) \ldots (1)$. However, 0! is defined as unity.

Manipulation of Exponential Functions

An exponential function is the Napierian base of the natural logarithms, e = 2.71828 . . ., raised to some power. For example, e^2 is an exponential function and has the value 7.3891. This value can be calculated on most calculators.

Rules that must be followed when manipulating these functions are given next.

Rule 1:

$$e^x \times e^y = e^{x+y}$$

Rule 2:

$$e^{-x} = \frac{1}{e^x}$$

Rule 3:

$$\frac{e^x}{e^y} = e^{x-y}$$

Rounding Data

Reliability calculations are made by using failure rate data. If the failure rate data base is accurate to three places, calculations using these data can be made to three places. Use should be made of the commonly accepted rule (computer's rule) to round the computational results to the proper number of significant figures. The *Mathematics Dictionary* (ref. 2-1) defines rounding off:

> When the first digit dropped is less than 5, the preceding digit is not changed; when the first digit dropped is greater than 5 or 5 and some succeeding digit is not zero, the preceding digit is increased by 1; when the first digit dropped is 5 and all succeeding digits are zero, the commonly accepted rule is to make the preceding digit even, i.e., add 1 to it if it is odd, and leave it alone if it is already even.

For example, if the reliability of a system is 0.8324, 0.8316, or 0.8315, it would take the form 0.832 if rounded off to three places.

Integration Formulas

Only the following integration formulas are used in this manual:

$$\int_a^b x^n \, dx = \frac{x^{n+1}}{n+1}\Big|_a^b = \frac{b^{n+1} - a^{n+1}}{n+1} \tag{1}$$

$$\int_a^b e^{-x} dx = -e^{-x}\Big|_a^b = e^{-b} + e^{-a} = e^{-a} - e^{-b} \tag{2}$$

$$\int_p^q e^{-ax} \, dx = \frac{-e^{-ax}}{a}\Big|_p^q = \frac{e^{-ap} - e^{-aq}}{a} \tag{3}$$

Example 1:

$$\int x^2 \, dx = \frac{x^{2+1}}{2+1} = \frac{x^3}{3}$$

$$\int_2^3 x \, dx = \frac{x^2}{2}\Big|_2^3 = \frac{(3)^2 - (2)^2}{2} = \frac{9-4}{2} = \frac{5}{2}$$

Example 2:

$$\int_3^4 e^{-x} \, dx = -e^{-x}\Big|_3^4 = e^{-3} - e^{-4}$$

Example 3:

$$\int_3^4 e^{-2x} = \frac{-e^{-2x}}{2}\Big|_3^4 = \frac{e^{-8} - e^{-6}}{2}$$

Differential Formulas

Only the following differential formulas are used in this manual:

$$\frac{d(ax)}{dx} = a \tag{4}$$

$$\frac{d\left(ax^n\right)}{dx} = nax^{n-1} \tag{5}$$

Example 4:

$$\frac{d(x)}{dx} = 1$$

$$\frac{d(4x)}{dx} = 4$$

Example 5:

$$\frac{d\left(x^2\right)}{dx} = 2x^{2-1} = 2x$$

$$\frac{d\left(4x^3\right)}{dx} = (3)4x^{3-1} = 12x^2$$

Partial Derivatives

This manual uses the following partial derivative formula:

$$\frac{\partial v}{\partial x_1} = \frac{\partial(xyz)}{\partial x} = yz \tag{6}$$

TABLE 2-1.—BINOMIAL COEFFICIENTS

n	Coefficient of each term of $(a + b)^n$										
	1	2	3	4	5	6	7	8	9	10	11
0	1										
1	1	1									
2	1	2	1								
3	1	3	3	1							
4	1	4	6	4	1						
5	1	5	10	10	5	1					
6	1	6	15	20	15	6	1				
7	1	7	21	35	35	21	7	1			
8	1	8	28	56	70	56	28	8	1		
9	1	9	36	84	126	126	84	36	9	1	
10	1	10	45	120	210	252	210	120	45	10	1

Example 6:

$$v = 2\ \text{ft} \times 3\ \text{ft} \times 4\ \text{ft} = 24\ \text{ft}^3 \quad \begin{cases} x = 2\ \text{ft} \\ y = 3\ \text{ft} \\ z = 4\ \text{ft} \end{cases}$$

$$\frac{\partial v}{\partial x} = yz = 12\ \text{ft}^2$$

Expansion of $(a + b)^n$

It will be necessary to know how to transform the expression $(a + b)^n$ into a binomial expansion. This type of problem is easily solved by using table 2–1 and recalling that

$$(a+b)^n = a^n + na^{n-1}b + \frac{(n-1)(n)}{2!}a^{n-2}b^2$$

$$+ \frac{(n-2)(n-1)(n)}{3!}a^{n-3}b^3 + \ldots$$

$$+ \frac{(n-1)(n-2)\ldots(n-m+1)}{m!}$$

$$\times a^{n-m}b^m + \ldots + b^n \qquad (7)$$

Example 7:

Expand $(a+b)^4$. From table 2–1 with $n = 4$,

$$(a+b)^4 = a^4 + 4a^3b + 6a^2b^2 + 4ab^3 + b^4$$

Failure Physics

When we consider reliability, we think of all the parts or components of a system continuing to operate correctly. Therefore a reliable system or product must have reliable parts. But what makes a part reliable? When asked, many people would say a reliable part is one purchased according to a certain source control document and bought from an approved vendor. Unfortunately, these two qualifications are not always guarantees of reliability. The following case illustrates this problem.

A clock purchased according to PD 4600008, procured from an approved vendor for use in the ground support equipment of a missile system, was subjected to qualification tests as part of the reliability program. These tests consisted of high- and low-temperature, mechanical shock, temperature shock, vibration, and humidity. The clocks from the then sole-source vendor failed two of the tests: low temperature and humidity. A failure analysis revealed that lubricants in the clock's mechanism froze and that the seals were not adequate to protect the mechanism from humidity. A second approved vendor was selected. His clocks failed the high-temperature test. In the process, the dial hands and numerals turned black, making readings impossible from a distance of 2 ft. A third approved vendor's clocks passed all the tests except mechanical shock, which cracked two of the cases. Ironically, the fourth approved vendor's clocks, though less expensive, passed all the tests.

The point of this illustration is that four clocks, each designed to the same specification and procured from a qualified vendor, all performed differently in the same environments. Why did this happen? The specification did not include the gear lubricant or the type of coating on the hands and numerals or the type of case material.

Many similar examples could be cited, ranging from requirements for glue and paint to complete assemblies and systems. The key to solving these problems is best stated as follows: To know how reliable a product is or how to design a reliable product, you must know all the ways its parts could fail and the types and magnitude of stresses that cause such failures. Think about this: if you knew every conceivable way a missile could fail and if you knew the type and level of stress required to produce each failure, you could build a missile that would never fail because you could eliminate

(1) As many types of failure as possible
(2) As many stresses as possible
(3) The remaining potential failures by controlling the level of the remaining stresses

TABLE 2–2.—RESULTS OF QUALIFICATION TESTS ON SOURCE CONTROL DOCUMENT CLOCK

Vendor	High temperature	Low temperature	Mechanical shock	Temperature shock	Vibration	Humidity
1		Fail				Fail
2	Fail					
3			Fail			
4						

Sound simple? Well, it would be except that despite the thousands of failures observed in industry each day, we still know very little about why things fail and even less about how to control the failures. However, through systematic data accumulation and study, we continue to learn more.

As stated, this manual introduces some basic concepts of failure physics: failure modes (how failures are revealed); failure mechanisms (what produces the failure mode); and failure stresses (what activates the failure mechanisms). The theory of and the practical tools for controlling failures are also presented.

Probability Theory

Fundamentals

Because reliability values are probabilities, every student of reliability disciplines should know the fundamentals of probability theory, which is used in chapter 3 to develop models that represent how failures occur in products.

Probability defined.—Probability can be defined as follows: If an event can occur in A different ways, all of which are considered equally likely, and if a certain number B of these events are considered successful or favorable, the ratio B/A is called the probability of the event. A probability, according to this definition, is also called an *a priori* (beforehand) probability because its value is determined without experimentation. It follows that reliability predictions of the success of missile flights that are made before the flights occur are a priori reliabilities. In other words, a priori reliabilities are estimates of what may happen and are not observed facts.

After an experiment has been conducted, an *a posteriori* probability, or an observed reliability, can be defined as follows: If $f(n)$is the number of favorable or successful events observed in a total number of n trials or attempts, the relative frequency $f(n)/n$ is called the statistical probability, the a posteriori probability, the empirical probability, or the observed reliability. Note that the number of favorable events $f(n)$ is a function of the total number of trials or attempts n. Therefore, as the number of trials or attempts changes, $f(n)$ may also change, and consequently the statistical probability (or observed reliability) may change.

Reliability of a coin.—To apply this theory, consider the physics of a coin. Assume that it has two sides, is thin, and is made of homogeneous material. If the coin is tossed, one of two possible landings may occur: with the head side up or tail side up. If landing heads up is considered more favorable than landing tails up, a prediction of success can be made by using the a priori theory. From the a priori definition, the probability of success is calculated as

$$\frac{1 \text{ favorable event}}{2 \text{ possible events}} = \frac{1}{2}, \text{ or 50 percent}$$

TABLE 2–3.—OBSERVED PROBABILITY OF SUCCESS

Number of tosses, n	1	10	100	1000	10 000
Number of heads observed, $f(n)$	0	7	55	464	5080
Relative frequency of probability of success, $f(n)/n$	0	0.70	0.55	0.464	0.508

This is an estimate of what should be observed if the coin is tossed but is not yet an observed fact. After the coin is tossed, however, the probability of success could be much more specific as shown in table 2–3.

The table shows two important phenomena:

(1) As the number of trials changes, the number of favorable events observed also changes. An observed probability of success (or observed reliability) may also change with each additional trial.

(2) If the assumptions made in calculating the a priori probability (reliability prediction) are correct, the a posteriori (observed) probability will approach the predicted probability as the number of trials increases. Mathematically, the relative frequency $f(n)/n$ approaches the a priori probability B/A as the number of trials n increases, or

$$\lim_{n \to \infty} \frac{f(n)}{n} = \frac{B}{A}$$

In the coin toss example, the predicted reliability was 0.50. The observed reliability of 0.508 indicates that the initial assumptions about the physics of the coin were probably correct. If, as a result of 10 000 tosses, heads turned up 90 percent of the time, this could indicate that the coin was incorrectly assumed to be homogeneous and that, in fact, it was "loaded." Inconsistency in the actual act of tossing the coin, a variable that was not considered in the initial assumptions, could also be indicated. Here again, even with a simple coin problem, it is necessary to consider all the ways the coin may "fail" in order to predict confidently how it will perform.

Reliability of missiles.—In the aerospace industry, a priori probabilities (reliability predictions) are calculated for missiles in an effort to estimate the probability of flight success. Inherent in the estimate are many assumptions based on the physics of the missile, such as the number of its critical parts, its response to environments, and its trajectory. As in the coin problem, the ultimate test of the missile's reliability prediction is whether or not the prediction agrees with later observations.

If during flight tests, the observations do not approach the predictions as the number of flights increases, the initial assumptions must be evaluated and corrected. An alternative approach is to modify the missile to match the initial assumptions. This approach is usually pursued when the reliability

prediction represents a level of success stated by the customer or when the predicted value is mandatory for the missile to be effective. This subject of reliability predictions is discussed again in chapter 4.

In practice, reliability testing yields the knowledge needed to verify and improve initial assumptions. As experience is gained, the assumptions undergo refinements that make it possible to develop more accurate reliability predictions on new missiles and systems not yet tested or operated. This information also provides design engineers and management with data to guide design decisions toward maximum missile or system reliability. Some reliability problems require the use of Bayes or Markovian probability theorems. Additional information on other topics is available in references 2–2 to 2–5 and in IEEE Reliability Society publications and other documents listed in the reference sections for chapters 3 to 9 and in the bibliography at the end of this manual.

Probability Theorems

The three probability theorems presented here are fundamental and easy to understand. In these theorems and examples, the probability of success (reliability) is represented with an R and the probability of failure (unreliability) with a Q. The following section (Concept of Reliability) examines what contributes to the reliability and unreliability of products.

Theorem 1.—If the probability of success is R, the probability of failure Q is equal to $1 - R$. In other words, the probability that all possible events will occur is $Q + R = 1$.

Example 1: If the probability of a missile flight success is 0.81, the probability of flight failure is $1 - 0.81 = 0.19$. Therefore, the probability that the flight will succeed or fail is $0.19 + 0.81 = 1.0$.

Theorem 2.—If R_1 is the probability that a first event will occur and R_2 is the probability that a second independent event will occur, the probability that both events will occur is R_1R_2. A similar statement can be made for more than two independent events.

Example 2: If the probability of completing one countdown without a failure R_1 is 0.9, the probability of completing two countdowns without failure is $R_1R_2 = (0.9)(0.9) = 0.81$. The probability that at least one of the two countdowns will fail is $1 - R_1R_2 = 1 - 0.81 = 0.19$ (from theorem 1). We say that at least one will fail because the unreliability term Q includes all possible failure modes, which in this case is two: one or both countdowns fail.

Example 3: If the probability of failure Q_1 during one countdown is 0.1, the probability of failure during two countdowns is $Q_1Q_2 = (0.1)(0.1) = 0.01$. Therefore, the probability that at least one countdown will succeed is $1 - Q_1Q_2 = 1 - 0.01 = 0.99$. We say that at least one will succeed because the value 0.99 includes the probability of one countdown succeeding and the probability of both countdowns succeeding.

Example 4: If the probability of completing one countdown without failure R_1 is 0.9 and the probability of a second countdown failing is $Q_2 = 0.1$, the probability that the first will succeed and the second will fail is $R_1Q_2 = (0.9)(0.1) = 0.09$.

Theorem 3.—If the probability that one event will occur is R_1 and the probability that a second event will occur is R_2 and if not more than one of the events can occur (i.e., the events are mutually exclusive), the probability that either the first or second event, not both, will occur is $R_1 + R_2$. A similar theorem can be stated for more than two events.

Example 5 (true event method): Consider now the probability of completing two countdowns without a failure. Let the probabilities of success for the first and second countdowns be R_1 and R_2 and the probabilities of failure be Q_1 and Q_2. To solve the problem using theorem 3, it is best to diagram the possible events as shown in figure 2–1. The mutually exclusive events are

Q_1	first countdown fails
R_1Q_2	first countdown succeeds and second fails
R_1R_2	both countdowns succeed

From theorem 3, the probability that one of the three events will occur is

$$Q_1 + R_1Q_2 + R_1R_2$$

But because these three events represent all possible events that can occur, their sum equals 1 (from theorem 1). Therefore,

$$Q_1 + R_1Q_2 + R_1R_2 = 1$$

The probability of completing both countdowns without one failure R_1R_2 is the solution to the proposed problem; therefore,

$$R_1R_2 = 1 - (R_1Q_2 + Q_1)$$

If $R_1 = 0.9$, $Q_1 = 0.1$, $R_2 = 0.9$, and $Q_2 = 0.1$, then

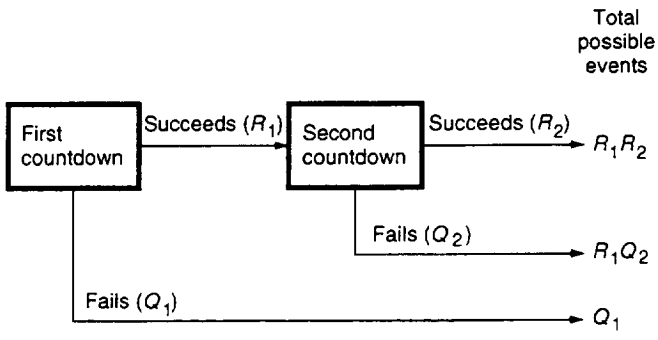


Figure 2–1.—Diagram of possible events—probability of completing two countdowns without a failure.

$$R_1R_2 = 1-[(0.9)(0.1)+0.1]$$
$$= 1-(0.9+0.1) = 1-0.19 = 0.81$$

which agrees with the answer found in example 2 by using theorem 2. The expression for R_1R_2 can also be written

$$R_1R_2 = 1-(R_1Q_2+Q_1) = 1-[(1-Q_1)Q_2+Q_1]$$
$$= 1-(Q_1+Q_2-Q_1Q_2)$$

which is the usual form given for the probability of both events succeeding. However, note that in this expression, the event indicated by Q_1Q_2 (both countdowns fail) is not a true possible event because we stipulated in the problem that only one countdown could fail. The term Q_1Q_2 is only a mathematical event with no relation to observable events. In other words, if the first countdown fails, we have lost our game with chance.

Example 6 (mathematical event method): Now consider the problem of example 5, ignoring for the time being the restriction on the number of failures allowed. In this case, the diagram of the possible events looks like that shown in figure 2–2. In this case the mutually exclusive events are

R_1R_2	both countdowns succeed
R_1Q_2	first countdown succeeds and second fails
Q_1R_2	first countdown fails and second succeeds
Q_1Q_2	both countdowns fail

Keep in mind that in this example both countdowns may fail. From theorem 3, the probability that one of the four events will occur is

$$R_1R_2 + R_1Q_2 + Q_1R_2 + Q_1Q_2$$

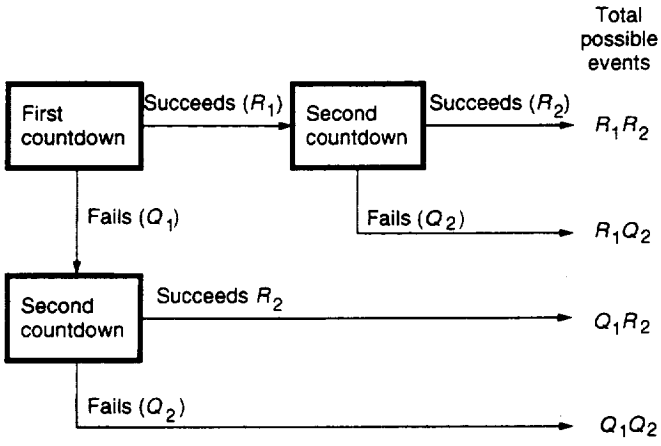


Figure 2–2.—Diagram of possible events—number of failures not restricted.

Again, because the four events represent all possible events that can occur, their sum equals unity (from theorem 1); that is,

$$R_1R_2 + R_1Q_2 + Q_1R_2 + Q_1Q_2 = 1$$

Solving for the probability that both countdowns will succeed is

$$R_1R_2 = 1-(R_1Q_2 + Q_1R_2 + Q_1Q_2)$$

Substituting $1-Q_1$ for R_1 and $1-Q_2$ for R_2 on the right side of the equation yields the answer given in example 5:

$$R_1R_2 = 1-[(1-Q_1)Q_2 + Q_1(1-Q_2) + Q_1Q_2]$$
$$= 1-(Q_2 - Q_1Q_2 + Q_1 - Q_1Q_2 + Q_1Q_2)$$
$$= 1-(Q_1+Q_2-Q_1Q_2)$$

This countdown problem has been solved in two ways to acquaint you with both methods of determining probability diagrams, the true event and the mathematical event. The exercises at the end of this chapter may be solved by using the method you prefer. We suggest that you work the problems before continuing to the next section because they help you to gain a working knowledge of the three theorems presented.

Concept of Reliability

Now that you understand the concepts of probability and failure physics, you are ready to consider the concept of reliability. First, we will discuss the most common definition of reliability—in terms of the successful operation of a device. This definition, to fit the general theme of the manual, is then modified to consider reliability in terms of the absence of failure modes.

Reliability as Probability of Success

The classical definition of reliability is generally expressed as follows: Reliability is the probability that a device will operate successfully for a specified period of time and under specified conditions when used in the manner and for the purpose intended. This definition has many implications. The first is that when we say that reliability is a probability, we mean that reliability is a variable, not an absolute value. Therefore, if a device is 90 percent reliable, there is a 10 percent chance that it will fail. And because the failure is a chance, it may or may not occur. As in the coin example, as more and more of the devices are tested or operated, the ratio of total success to total attempts should approach the stated reliability of 90 percent. The next implication concerns the statement ". . . will

operate successfully . . ." This means that failures that keep the device from performing its intended mission will not occur. From this comes a more general definition of reliability: it is the probability of success.

It should be obvious then that a definition of what constitutes the success of a device or a system is necessary before a statement of its reliability is possible. One definition of success for a missile flight might be that the missile leaves the launching pad; another, that the missile hits the target. Either way, a probability of success, or reliability, can be determined, but it will not be the same for each definition of success. The importance of defining success cannot be overemphasized. Without it, a contractor and a customer will never reach an agreement on whether or not a device has met its reliability requirements (i.e., the mission).

The latter part of the classical definition indicates that a definition of success must specify the operating time, the operating conditions, and the intended use. Operating time is defined as the time period in which the device is expected to meet its reliability requirements. The time period may be expressed in seconds, minutes, hours, years, or any other unit of time. Operating conditions are defined as the environment in which the device is expected to operate; they specify the electrical, mechanical, and environmental levels of operation and their durations. Intended use is defined as the purpose of the device and the manner in which it will be used. For example, a missile designed to hit targets 1000 miles away should not be considered unreliable if it fails to hit targets 1100 miles away. Similarly, a set of ground checkout equipment designed to be 90 percent reliable for a 1-hour tactical countdown should not be considered unreliable if it fails during 10 consecutive countdowns or training exercises. The probability of success in this case is $(0.9)^{10} = 0.35$ (from probability theorem 2).

In addition to these specified requirements, we must also consider other factors. As explained in the inherent product reliability section of this chapter, these areas have a marked effect on the reliability of any device.

Reliability as Absence of Failure

Although the classical definition of reliability is adequate for most purposes, we are going to modify it somewhat and examine reliability from a slightly different viewpoint. Consider this definition: Reliability is the probability that the critical failure modes of a device will not occur during a specified period of time and under specified conditions when used in the manner and for the purpose intended. Essentially, this modification replaces the words "a device will operate successfully" with the words "critical failure modes . . . will not occur." This means that if all the possible failure modes of a device (ways the device can fail) and their probabilities of occurrence are known, the probability of success (or the reliability of a device) can be stated. It can be stated in terms of the probability that those failure modes critical to the performance of the device will not occur. Just as we needed a clear definition of success when using the classical definition, we must also have a clear definition of failure when using the modified definition.

For example, let a system have two subsystems, A and B, whose states are statistically independent and whose separate reliabilities are known to be $R_A = 0.990$ and $R_B = 0.900$. The system fails if and only if at least one subsystem fails. The appropriate formula for system reliability is

$$R_{\text{system}} = R_A \cdot R_B$$

$$R_{\text{system}} = 0.990 \cdot 0.900 = 0.891$$

Product Application

This section relates reliability (or the probability of success) to product failures.

What are the types of product failure modes? In general, critical equipment failures may be classified as catastrophic, tolerance, or wearout. The expression for reliability then becomes

$$R_D = \text{Probability}\{C \times t \times W\}$$

where

R_D design-stage reliability of a product
C event that catastrophic failure does not occur
t event that tolerance failure does not occur
W event that physical wearout does not occur

This is the design-stage reliability of a product as described by its documentation (Note that R_i, the inherent reliability, is a term often used in place of R_D). The documentation specifies the product itself and states the conditions of use and operation. This design-stage reliability is predicated on the decisions and actions of many people. If they change, the design-stage reliability could change.

Why do we consider design-stage reliability? Because the facts of failure are these: When a design comes off the drawing board, the parts and materials have been selected; the tolerance, error, stress, and other performance analyses have been performed; the type of packaging is firm; the manufacturing processes and fabrication techniques have been decided; and usually the test methods and the quality acceptance criteria have been selected. The design documentation represents some potential reliability that can never be increased except by a design or manufacturing change or good maintenance. However, the possibility exists that the observed reliability will be much less than the potential reliability.

To understand why this is true, consider the hardware as a black box with a hole in both the top and bottom. Inside are potential failures that limit the design-stage reliability of the design. When the hardware is operated, these potential failures fall out the bottom (i.e., operating failures are observed). The rate at which the failures fall out depends on how the box or hardware is operated. Unfortunately, we never have just the design-stage failures to worry about because other types of failures are being added to the box through the hole in the top. These other failures are generated by the manufacturing, software, quality, and logistics functions, by the user or customer, and even by the reliability organization itself. We discuss these added failures and their contributors in the following paragraphs, but it is important to understand that because of the added failures, the observed failures could be greater than the design-stage failures.

K-Factors

The other contributors to product failure just mentioned are called *K*-factors; they have a value between 0 and 1 and modify the design-stage reliability:

$$R_{\text{product}} = R_D \times \left(K_q \times K_m \times K_s \times K_r \times K_l \times K_u\right)$$

K-factors denote probabilities that design-stage reliability will not be degraded by

- K_q quality test methods and acceptance criteria
- K_m manufacturing, fabrication, and assembly techniques
- K_s software
- K_r reliability engineering activities
- K_l logistics activities
- K_u user or customer

Any *K*-factor can cause reliability to go to zero. If each *K*-factor equals 1 (the goal), $R_{\text{product}} = R_D$.

Interface Definition and Control

This section is a training manual describing the elements of interface definition and control (ref. 2–7).

This technical manual was developed as part of the Office of Safety and Mission Assurance continuous training initiative. The structured information contained herein will enable the reader to efficiently and effectively identify and control the technical detail needed to ensure that flight system elements mate properly during assembly operations (on the ground and in space).

Techniques used throughout the Federal Government to define and control technical interfaces for hardware and software were investigated. The proportion of technical information actually needed to effectively define and control the essential dimensions and tolerances of system interfaces rarely exceeded 50 percent of any interface control document. Also, the current government process for interface control is very paper intensive. Streamlining this process can improve communication, provide significant cost savings, and improve overall mission safety and assurance.

The objective of this manual is to ensure that the format, information, and control of interfaces between equipment are clear and understandable and contain only the information needed to guarantee interface compatibility. The emphasis is on controlling the engineering design of the interface and is not on the functional performance requirements of the system or on the internal workings of the interfacing equipment. Interface control should take place, with rare exception, at the interfacing elements and not further.

Two essential sections of the manual are Principles of Interface Control and The Process: Through the Design Phases. The first discusses how interfaces are defined, describes the types of interfaces to be considered, and recommends a format for the documentation necessary to adequately control the interface. The second provides tailored guidance for interface definition and control.

This manual can be used to improve planned or existing interface control processes during system design and development and also to refresh and update the corporate knowledge base. The information presented will reduce the amount of paper and data required in interface definition and control processes by as much as 50 percent and will shorten the time required to prepare an interface control document. It also highlights the essential technical parameters that ensure that flight subsystems will indeed fit together and function as intended after assembly and checkout. Please contact the NASA Center for Aerospace Information, (301) 621–0390 to obtain a copy.

Appendix A contains tables and figures that provide reference data to support chapters 2 to 6. Appendix B is a practical product assurance guide for project managers.

Concluding Remarks

Chapter 2 explained two principal concepts:

1. To design a reliable product or to improve a product, you must understand first how the product can fail and then how to control the occurrence of the failures.
2. There is an upper limit to a product's reliability when a traditional method of design and fabrication is used. This limit is the inherent reliability. Therefore, the most effective reliability engineer is the designer because all his decisions directly affect the product's reliability.

The three probability theorems were also illustrated.

References

2–1. James, G.: Mathematics Dictionary. Fourth Edition, Van Nostrand Reinhold, 1976.

2–2. Bazousky, I.: Reliability Theory and Practice. Prentice Hall, 1961.

2–3. Earles, D.R.; and Eddins, U.F.: Reliability Physics, The Physics of Failure. AVCO Corp., Wilmington, MA, 1962.

2–4. Calabro, S.: Reliability Principles and Practices. McGraw-Hill, 1962.

2–5. Electronic Reliability Design Handbook, MIL–HDBK–338, vols. 1 and 2, Oct. 1988.

2–6. Lalli, Vincent R.; and Packard, Michael H.: Design for Reliability: Failure Physics and Testing. AR & MS Tutorial Notes, 1994.

2.7. Lalli, Vincent R.; Kastner, Robert E.; and Hartt, Henry N.: Training Manual for Elements of Interface Definition and Control. NASA RP–1370, Cleveland, OH, 1997.

Reliability Training[1]

1a. What notation means to take the sum of the x_i's from $i = 1$ to $i = n$?

A. $\sum x's$ B. $\sum_{i=1}^{\infty} x_k$ C. $\sum_{i=1}^{n} x_i$

1b. If $\bar{x} = 100$, $x_1 = 90$, $x_2 = 70$, and $x_3 = 50$, what is $\sum_{i=1}^{n} (\bar{x} - x_i)^2$?

A. 350 B. 35×10^2 C. 35 000

2a. What notation means to take the n^{th} root of x?

A. x^n B. $\in^n$ C. $n\sqrt{x}$

2b. If $\bar{x} = 100$, $x_1 = 90$, $x_2 = 70$, and $x_3 = 50$, what is $\sqrt{\sum_{i=1}^{n} (\bar{x} - x_i)^2}$?

A. 3.6 B. 59.2 C. 640

3a. What notation means to take the product of the x_i's from $i = 1$ to n?

A. $\prod x's$ B. $\prod_{i=0}^{\infty} x_k$ C. $\prod_{i=1}^{n} x_i$

3b. If $x_1 = 0.9$, $x_2 = 0.99$, and $x_3 = 0.999$, what is $\prod_{i=1}^{3} x_i$?

A. 0.890 B. 0.800 C. 0.991

4a. The notation $x!$ refers to what shorthand method of writing?

A. Poles B. Factorial C. Polynomials

4b. What does 10!/8! equal?

A. 800 B. 900 C. 90

5a. Describe the three rules for manipulation of exponential functions.

i. Products

A. Subtract exponents B. Add exponents C. Multiply exponents

ii. Negative exponent

A. Cancel exponents B. Balance exponents C. 1/Exponent

iii. Division

A. Add exponents B. Subtract exponents C. Multiply exponents

[1] Answers are given at the end of this manual.

5b. Simplify, $\epsilon^6\epsilon^3\epsilon^4$.

A. ϵ^2 B. ϵ^4 C. ϵ^5

6. What is the integral of the following functions?

a. $\int_{x_1}^{x_2} x^3\,dx$

A. $x^4/4$ B. $x^4/4\Big|_0^{x_2}$ C. $[(x_2)^4-(x_1)^4]/4$

b. $\int_{x_1}^{x_2} \epsilon^{-ax}\,dx$

A. $-\epsilon^{-ax}/a$ B. $\left[\epsilon^{-ax_1}-\epsilon^{-ax_2}\right]/a$ C. $-\epsilon^{-ax}/a\Big|_0^{x_2}$

7. What is the derivative of the following functions?

a. $10x^4$

A. $40x^2$ B. $40x^3$ C. $10x^3$

b. ϵ^{2x}

A. ϵ^{2x} B. $\epsilon^{2x/2}$ C. $2\epsilon^{2x}$

8a. Write the first two terms of the binomial expansion $(a + b)^n$.

A. $a^n + (n-1)a^{n-1}b + \ldots$ B. $a^n - na^{n-1}b + \ldots$ C. $a^n + na^{n-1}b + \ldots$

8b. Expand $(a + b)^3$ by using table 2–1.

A. $a^3 + 2a^2b + b^3$ B. $a^3 - 3a^2b - 3ab^2 + b^3$ C. $a^3 + 3a^2b + 3ab^2 + b^3$

9. What needs to be done to design a reliable product?

A. Test and fix it
B. Know how its parts fail
C. Know the type and magnitude of stresses that cause such failures
D. Both B and C

10. What are a priori reliabilities estimates of?

A. What may happen B. What will happen C. What has happened

11. What are a posteriori reliabilities observing?

A. What may happen B. What has happened C. What will happen

12. If the probability of success is R, what is the probability of failure Q?

A. $1 + R$ B. $1 - R^2$ C. $1 - R$

13. If R_1, R_2, and R_3 are the probabilities that three independent events will occur, what is the probability that all three will occur?

A. $R_1 + R_2 + R_3$ B. $R_1 (R_2 + R_3)$ C. $\prod_{i=1}^{3} R_i$

14. If R_1, R_2, and R_3 are the probabilities that three independent events will occur and not more than one of the events can occur, what is the probability that one of these events will occur?

A. $R_1 R_2 R_3$ B. $R_3 (R_1 + R_2)$ C. $\sum_{i=1}^{3} R_i$

15. What do we need to know if a device is to perform with classical reliability?

A. Operating time and conditions
B. How it will be used
C. The intended purpose
D. All of the above

16. What do we need to know if a device is to perform with reliability defined as the absence of failure?

A. Critical failure modes
B. Operating time and conditions
C. How it will be used
D. The intended purpose
E. All of the above

17. What is the inherent reliability R_i of the product you are working on?

A. P_C (the probability that catastrophic part failures will not occur)
B. P_t (the probability that tolerance failures will not occur)
C. P_w (the probability that wearout failures will not occur)
D. The product of all the above

18. What is the reliability of your product?

A. K_q (the probability that quality test methods will not degrade R_i)
B. K_m (the probability that manufacturing processes will not degrade R_i)
C. K_r (the probability that reliability activities will not degrade R_i)
D. K_l (the probability that logistic activities will not degrade R_i)
E. K_u (the probability that the user will not degrade R_i)
F. The product of all of the above and R_i

Chapter 3
Exponential Distribution and Reliability Models

An expression for the inherent reliability of a product was given in chapter 2 as (ref. 3–1)

$$R_i = P_c P_t P_w$$

where

P_c probability that catastrophic part failures will not occur
P_t probability that tolerance failures will not occur
P_w probability that wearout failures will not occur

In chapter 3, we discuss the term P_c and develop and explain its mathematical representation in detail. We then use the probability theorems to establish methods of writing and solving equations for product reliability in terms of series and redundant elements.

Exponential Distribution

To understand what is meant by exponential distribution, first examine a statistical function called the Poisson distribution, which is expressed as (ref. 3–2)

$$P(x,t) = \frac{(\lambda t)^x e^{-\lambda t}}{x!}$$

where

x observed number of failures
t operating time
λ average failure rate

This distribution states that if an observed average failure rate λ is known for a device, it is possible to calculate the probability $P(x,t)$ of observing $x = 0,1,2,3,\ldots$, number of failures when the device is operated for any period of time t.

To illustrate, consider a computer that has been observed to make 10 arithmetic errors (or catastrophic failures) for every hour of operation. Suppose that we want to know the probability of observing 0, 1, and 2 failures during a 0.01-hr program. From the data given,

x (observed failures) = 0, 1, and 2
t (operating time) = 0.01 hr
λ (failure rate) = 10 failures/hr

The probability of observing no failures $P(0, 0.01)$ is then

$$P(0, 0.01) = \frac{(10 \times 0.01)^0 e^{-(10 \times 0.01)}}{0!}$$

$$= \frac{1 \times e^{-0.1}}{1} = e^{-0.1} = 0.905$$

The probability of observing one failure $P(1, 0.01)$ is

$$P(1, 0.01) = \frac{(10 \times 0.01)^1 e^{-(10 \times 0.01)}}{1!}$$

$$= \frac{(0.1)^1 e^{-0.1}}{1} = 0.1 \times 0.905 = 0.091$$

The probability of observing two failures $P(2, 0.01)$ is

$$P(2, 0.01) = \frac{(10 \times 0.01)^2 e^{-(10 \times 0.01)}}{2!}$$

$$= \frac{(0.1)^2 e^{-0.1}}{2 \times 1} = \frac{0.01 \times 0.905}{2}$$

$$= \frac{0.00905}{2} = 0.0045$$

Remember that the definition of P_c is the probability that no catastrophic failures will occur. So, for the computer, $P_c = P(0, 0.01) = 0.905$. In other words, there is a 90.5-percent chance that no arithmetic errors will occur during the 0.01-hr program. This is the reliability of the computer for that particular program.

Again the Poisson distribution for $x = 0$ (i.e., no observed failures) is

$$P(0, t) = \frac{(\lambda t)^0 e^{-\lambda t}}{0!} = e^{-\lambda t}$$

The term $e^{-\lambda t}$ is called the exponential distribution and is the simplest form of P_c. Consequently, for a device that has an average failure rate λ, the probability of observing no failures for a period of time t is (ref. 3–3)

$$P_c = e^{-\lambda t}$$

The expression for inherent reliability now takes the form

$$R_i = e^{-\lambda t} P_t P_w$$

or in the more general expression for total product reliability,

$$R = e^{-\lambda t} P_t P_w \left(K_q K_m K_r K_t K_u\right)$$

At this point it is probably a good idea to digress for a moment to explain why these expressions for reliability may differ from those used elsewhere. During the conceptual and early research and development phases of a program, it is common practice (and sometimes necessary because of a lack of information) to assume that $P_t = 1$ (the design is perfect), that $P_w = 1$ (no wearout failures will occur), and that the K-factors all equal 1 (there will be no degradation of inherent reliability). These assumptions reduce the inherent reliability and product reliability expressions to

$$R_i = R = e^{-\lambda t}$$

Frequently, these assumptions are not realistic and the resultant reliability predictions are usually high. They may bear little resemblance to the reliability finally observed when the product is tested. Later in this manual, we will let

$$P_c = R = e^{-\lambda t}$$

to keep the notation simple.

On the other hand, it is also common to use $e^{-\lambda t}$ to represent the observed product reliability. In this case the observed average failure rate λ represents the combination of all types of failures including catastrophic, tolerance, and wearout. If the total product failure rate is λ', then

$$R = e^{-\lambda' t} = e^{-\lambda t} P_t P_w \left(K_q K_m K_r K_t K_u\right)$$

Failure Rate Definition

The failure rate λ as used in the exponential distribution $e^{-\lambda t}$ represents random catastrophic part failures that occur in so short a time that they cannot be prevented by scheduled maintenance (ref. 3–4). Random means that the failures occur randomly in time (not necessarily from random causes as many people interpret random failure) and randomly from part to part. For example, suppose a contractor uses 1 million integrated circuits in a computer. Over a period of time he may observe an average of one circuit failure every 100 operating hrs. Even though he knows the failure rate, he cannot say which one of the million circuits will fail. All he knows is that on the average, one will fail every 100 hrs. In fact, if a failed circuit is replaced with a new one, the new one, theoretically, has the same probability of failure as any other circuit in the computer. In addition, if the contractor performs a failure analysis on each of the failed circuits, he may find that every failure is caused by the same mechanism, such as poorly welded joints. Unless he takes some appropriate corrective action, he will continue to observe the same random failures even though he knows the failure cause.

A catastrophic failure is an electrical open or short, a mechanical or structural defect, or an extreme deviation from an initial setting or tolerance (a 5-percent-tolerance resistor that deviated beyond its end-of-life tolerance, say to 20 percent, would be considered to have failed catastrophically).

The latter portion of the failure rate definition refers to the circumstance under which a failure is revealed. If a potential operating failure is corrected by a maintenance function, such as scheduled preventive maintenance where an out-of-tolerance part could be replaced, that replacement cannot be represented by λ because it did not cause an operating or unscheduled failure. Here we see one of the many variables that affect the operating failure rate of a product: the maintenance philosophy.

TABLE 3–1.—COMMON FAILURE RATE DIMENSIONS

Failures/hr, percent	Failures/ 10^6 hr	Failures/ 10^9 hr
10.0	100.0	100 000.0
1.0	10.0	10 000.0
.1	1.0	1 000.0
.01	.1	100.0
.001	.01	10.0
.0001	.001	1.0
.00001	.0001	.1
.000001	.00001	.01
.0000001	.000001	.001

Failure Rate Dimensions

Failure rate has the dimension of failure per unit of time, where the time is usually expressed in 10^x hours or cycles. Some government documents express λ in percent failures per 10^3 hours. Table 3–1 shows the most common usage. Generally, the form that permits calculations using whole numbers rather than decimal fractions is chosen.

"Bathtub" Curve

In the Poisson distribution, λ was referred to as an average failure rate, indicating that λ may be a function of time $\lambda(t)$.

Figure 3–1 shows three general curves representing $\lambda(t)$ possibilities. Curve A shows that as operating time increases, the failure rate also increases. This type of failure rate is found where wearout or age is a dominant failure mode stress (e.g., slipped clutches or tires). Curve B shows that as operating time increases, the failure rate decreases. This type of failure rate has been observed in some electronic parts, especially semiconductors. Curve C shows that as operating time increases, the failure rate remains constant. This type of failure rate has been observed in many complex systems and subsystems. In a complex system (i.e., one with a large number of parts), parts having decreasing failure rates reduce the effect of those having increasing failure rates. The net result is an observed near-constant failure rate for the system. Therefore, part failure rates are usually given as a constant although in reality they may not be. This manual deals only with constant part failure rates because they are related to system operation. Even if the failure rates might be changing over a period of time, the constant-failure-rate approximation is used.

If the failure rate for a typical system or complex subsystem is plotted against operating life, a curve such as that shown in figure 3–2 results. The curve is commonly referred to as a "bathtub" curve. The time t_0 represents the time at which the system is first put together. The interval from t_0 to t_1 represents a period during which assembly errors, defective parts, and compatibility problems are found and corrected. As shown, the system failure rate decreases during this debugging, or burn-in, interval as these gross errors are eliminated. The interval from t_1 to t_2 represents the useful operating life of the equipment and is generally considered to have a constant failure rate. During this time, the expression $P_c = e^{-\lambda t}$ is used. Therefore, when using $e^{-\lambda t}$, we assume that the system has been properly debugged. In practice, this assumption may not be true, but we may still obtain an adequate picture of the expected operating reliability by accepting the assumption. The interval from t_2 to t_3 represents the wearout period during which age and deterioration cause the failure rate to increase and render the system inoperative or extremely inefficient and costly to maintain.

The following analogy should help to summarize the concepts of failure and failure rate. A company picnic is planned to be held on the edge of a high cliff. Because families will be invited, there will be various types of people involved: large, small, young, and old, each type with its own personality and problems. Picnic officials are worried about someone's falling over the cliff. The question is, What can be done about it? Four possible solutions are presented:

(1) Move the picnic farther back from the cliff. The farther back, the less the chance someone will fall over.

(2) Shorten the picnic time. The shorter the picnic, the less time someone has to walk to the cliff.

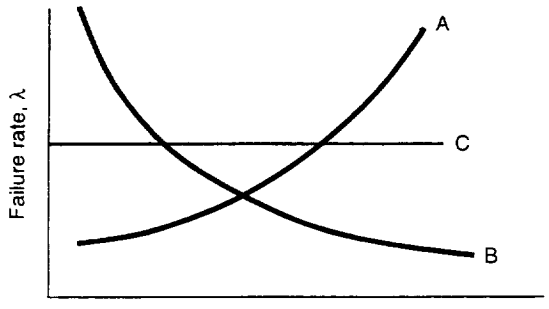


Figure 3–1.—Failure rate curves.

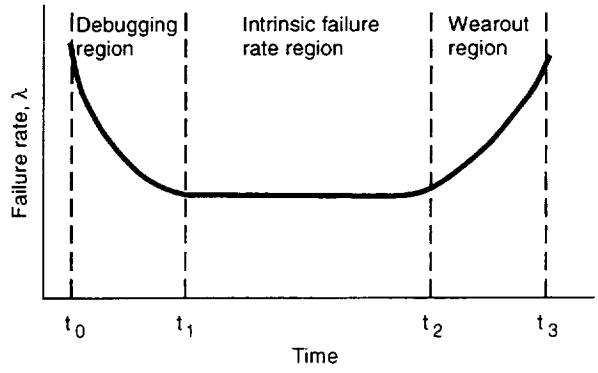


Figure 3–2.—Failure rate versus operating time.

(3) Look over the cliff to see if anyone has fallen. A good idea because people would know when to call the ambulance. Unfortunately, looking over the cliff does not keep others from falling. It is possible, however, that going to the bottom of the cliff to see who has fallen over might reveal that every 15 minutes one person over the age of 99 falls over the cliff. Knowing this, all persons over 99 could be sent home and the picnic saved from further tragedy.

(4) Build a high fence to separate the cliff edge from the picnickers. Obviously, this is the best solution because it is doubtful that anyone would climb the fence just to get to the cliff.

Now, let us look at this picnic-to-failure rate analogy. Say that we are building a system (picnic) made of many parts (people) and that there are many types of parts; some large, some small, and some new and untried, such as integrated circuits. Some of these parts, the composition resistors for instance, are old and mature. Each part has its own personality (the way it was fabricated). Our problem is how to keep these parts from failing (falling over the cliff). Again we have four possible solutions:

(1) Reduce the stresses on the parts (move the picnic back from the cliff); the lower the stresses, the fewer the failures.

(2) Reduce the operating time (the picnic); the shorter the operating time, the less chance a part has to fail. Part failure rates can be established (look over the cliff to see if anyone has fallen), but this only helps if we know what parts (people) are failing. Once we know this, we can eliminate those parts from our system.

(4) Eliminate the failure mechanisms of the part (build a fence to separate the cliff edge from the picnic). This is the best answer, of course, because if we eliminate the cause of part failures, we cannot have any system failures.

Mean Time Between Failures

For the exponential distribution, the reciprocal of the failure rate is the mean time between failures (MTBF) and is the integral of the exponential distribution:

$$\text{MTBF} = \frac{1}{\lambda}\int_0^\infty e^{-\lambda t}\,dt = -\frac{1}{\lambda}\left(e^{-\lambda t}\right)_0^\infty$$

$$= -\frac{1}{\lambda}\left(\frac{1}{e^\infty} - \frac{1}{e^0}\right) = -\frac{1}{\lambda}(0-1) = \frac{1}{\lambda}$$

Therefore, if a device has a failure rate of one failure per 100 hrs, its MTBF is 100 hrs.

If the time dimension is given in cycles, the MTBF becomes mean cycles between failures (MCBF), a term also in common use. For a nonrepairable device, mean time to failure (MTTF) is used instead of MTBF. For a repairable device MTBF, is usually equal to MTTF.

For example, if a device has an MTBF of 200 hrs, this neither means that the device will not fail until 200 operating hours have accumulated nor that the device will fail automatically at 200 hrs. MTBF is exactly what is says: a mean or average value, which can be seen from

$$e^{-\lambda t} = e^{-t/\text{MTBF}}$$

When the operating time t equals the MTBF, the probability of no failure is (using exponential tables or a slide rule)

$$e^{-\text{MTBF}/\text{MTBF}} = e^{-1} = 0.368$$

which means that there is a chance of $1 - 0.368 = 0.632$ that the device will fail before its MTBF is reached. In other words, if a device has an MTBF of 1000 hrs, replacing the device after 999 hrs of operation will not improve reliability. To show the concept of a mean value in another way, consider the following empirical definition of MTBF:

$$\text{MTBF} = \frac{\text{Total test hours}}{\text{Total observed failures}}$$

Note that the time when the failures were observed is not indicated. The assumption of a constant failure rate leads to a constant time between failures, or MTBF.

Calculations of P_c for Single Devices

If a failure rate for a device is known, the probability of observing no failures for any operating period t can be calculated.

Example 1: A control computer in a missile has a failure rate of 1 per 10^2 hrs. Find P_c for a flight time of 0.1 hr.

Solution 1:

$$P_c = e^{-\lambda t} = e^{-(1/10^2)(0.1)} = e^{-1\times10^{-3}} = e^{-0.001} = 0.999$$

Therefore, there is one chance in a thousand that the control computer will fail. (Note: if λt or t/MTBF is less than 0.01, $P_c \cong 1 - \lambda t$, or $1 - t$/MTBF.) For example,

$$P_c = e^{-0.001} \cong 1 - 0.001 = 0.999$$

If λt, or t/MTBF, is greater than 0.01, use exponential tables to find P_c, as shown here:

$$P_c = e^{-0.08} = 0.923$$

Example 2: The same type of problem can be solved if the MTBF is known. The MTBF of a tape reader used in ground support equipment is 100 hrs. Find P_c for a 2-hr operation.

Solution 2:

$$P_c = e^{-t/\text{MTBF}} = e^{-2/100} = e^{-0.02} = 0.980$$

If a specific P_c is required for a specified operating time, the required failure rate, or MTBF, can be calculated.

Example 3: A relay is required to have a 0.999 probability of not failing for 10 000 cycles. Find the required failure rate and MCBF.

Solution 3:

$$R = e^{-\lambda t}$$

$$0.999 = e^{-0.001} = e^{-\lambda\left(10^4 \text{ cycles}\right)}$$

Equating exponents gives

$$\lambda\left(10^4 \text{ cycles}\right) = 0.001$$

$$\lambda = \frac{0.001}{10^4} = \frac{1 \text{ failure}}{10^7 \text{ cycles}}$$

The required MCBF is therefore

$$\text{MCBF} = \frac{1}{\lambda} = 10^7 \text{ cycles}$$

Reliability Models

In the following sections we replace $P_c = e^{-\lambda t}$, the reliability of a part, with an R to keep the notation simple.

Calculation of Reliability for Series-Connected Devices

In reliability, devices are considered to be in series if each device is required to operate without failure to obtain system success (ref. 3–5). A system composed of two parts is represented in a reliability diagram, or model, as shown in figure 3–3. If the reliability R for each part is known (probability theorem 2, ch. 2), the probability that the system will not fail is

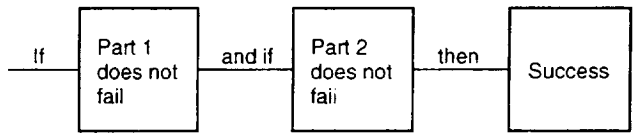


Figure 3–3.—Series model.

$$R_s = R_1 R_2$$

(We assume that the part reliabilities are independent; i.e., the success or failure of one part will not affect the success or failure of another part.) If there are n parts in the system with each one required for system success, the total system reliability is given by

$$R_s = R_1 R_2 R_3 \ldots R_n = \prod_{i=1}^{n} R_i$$

where

R_s probability that system will not fail
R_j reliability of j^{th} part
n total number of parts

The expression

$$R_s = \prod_{j=1}^{n} R_j$$

is often called the product rule.

Example 4: A system has 100 parts, each one required for system success. Find the system reliability R_s if each part has $R = 0.99$.

Solution 4:

$$R_s = \prod_{j=1}^{n} R_j = \prod_{j=1}^{100} R_j = R_1 R_2 R_3 \ldots R_{100}$$

$$= (0.99)(0.99)(0.99)\ldots(0.99) = (0.99)^{100}$$

$$= \left(e^{-0.01}\right)^{100} = e^{-1} = 0.368$$

Therefore, the probability that the system will succeed is about 37 percent.

Example 5: For a typical missile that has 7000 active parts and a reliability requirement of 0.90, each part would have to have a reliability R_p of 0.999985, which is calculated using table A–1:

$$\left(R_p\right)^{7000} = 0.90 = e^{-0.105}$$

Solution 5: Therefore,

$$R_p = \left(e^{-0.105}\right)^{1/7000} = e^{-1.5\times10^{-5}} = e^{-0.000015}$$
$$= 1 - 0.000015 = 0.999985$$

The product rule can also be expressed as

$$R_s = \prod_{j=1}^{n} R_j = R_1 R_2 R_3 \ldots R_n$$
$$= e^{-\lambda_1 t_1} e^{-\lambda_2 t_2} e^{-\lambda_3 t_3} \ldots e^{-\lambda_n t_n}$$
$$= e^{-\left(\lambda_1 t_1 + \lambda_2 t_2 + \lambda_3 t_3 \ldots \lambda_n t_n\right)}$$
$$= \exp\left(-\sum_{j=1}^{n} \lambda_j t_j\right)$$

where

λ_j failure rate of j^{th} part

t_j operating time of j^{th} part

Therefore, if for each series-connected part in a system the failure rate and operating time are known, the system reliability can be calculated by finding $-\sum_{j=1}^{n} \lambda_j t_j$ and raising e to the $-\left(\sum_{j=1}^{n} \lambda_j t_j\right)$ power.

Example 6: Find the system reliability from the model shown in figure 3–4.

Solution 6:

Step 1

$$\sum_{j=1}^{3} \lambda_j t_j = \lambda_1 t_1 + \lambda_2 t_2 + \lambda_3 t_3$$
$$= 10/10^3(10) + 20/10^3(4) + 100/10^3(2)$$
$$= 100/10^3 + 80/10^3 + 200/10^3 = 380/10^3$$

Step 2

$$R_s = \exp\left(-\sum_{j=1}^{3} \lambda_j t_j\right) = e^{-380/10^3} = e^{-0.38} = 0.684$$

If the t_j's are equal (i.e., each part of the device operates for the same length of time), the product rule can further be reduced to

$$R_s = \exp\left(-\sum_{j=1}^{n} \lambda_j\right) t_c$$

where t_c is the common operating time.

Example 7: Find the reliability of the system shown in figure 3–5.

Solution 7:

Step 1

$$\sum_{j=1}^{3} \lambda_j = \lambda_1 + \lambda_2 + \lambda_3 = 7/10^3 + 5/10^3 + 6/10^3 = 18/10^3$$

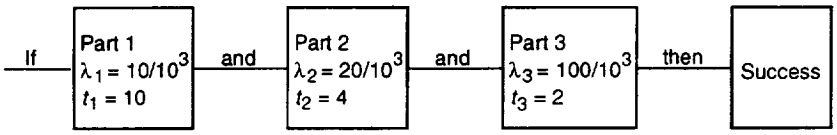


Figure 3–4.—Series model using failure rates and operating times.

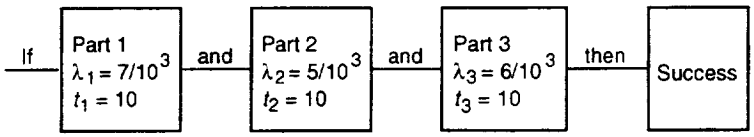


Figure 3–5.—Series model with operating times equal.

Step 2

$$R_s = \exp\left(-\sum_{j=1}^{3}\lambda_j\right)t_c = e^{-18/10^3(10)} = e^{-180/10^3}$$

$$= e^{-0.18} = 0.835$$

Calculation of Reliability for Parallel-Connected Devices (Redundancy)

In reliability, devices are considered to be in parallel if one or more of them can fail without causing system failure but at least one of them must succeed for the system to succeed. First we consider simple redundancy.

Simple redundancy.—If n devices are in parallel so that only one of them must succeed for the system to succeed, the devices are said to be in simple redundancy. The model of a two-part redundancy system presented in figure 3–6 illustrates this concept. In other words, if part 1 fails, the system can still succeed if part 2 does not fail, and vice versa. However, if both parts fail, the system fails.

From probability theorem 3 in chapter 2, we know that the possible combinations of success R and failure Q of two devices are given by

$$R_1R_2 + R_1Q_2 + Q_1R_2 + Q_1Q_2$$

where

R_1R_2 both parts succeed
R_1Q_2 part 1 succeeds and part 2 fails
Q_1R_2 part 1 fails and part 2 succeeds
Q_1Q_2 both parts fail

We also know that the sum of these events equals unity since they are mutually exclusive (i.e., if one event occurs, the others cannot occur). Therefore,

$$R_1R_2 + R_1Q_2 + Q_1R_2 + Q_1Q_2 = 1$$

Because at least one of the parts or devices must succeed in simple redundancy, the probability of this happening is given by

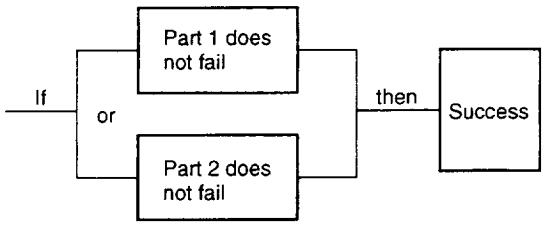


Figure 3–6.—Simple redundancy model.

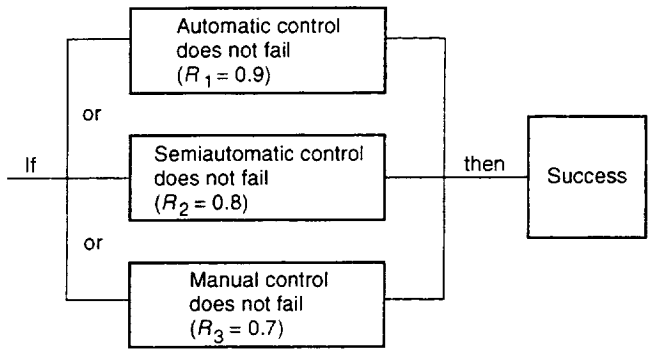


Figure 3–7.—Space capsule guidance model.

$$R_1R_2 + R_1Q_2 + Q_1R_2 = 1 - Q_1Q_2$$

In simple terms, if the only way the redundant system can fail is by all redundant parts failing, the probability of success must be equal to 1 minus the probability that all redundant parts will fail (i.e., $R = 1 - Q$) from probability theorem 1 in chapter 2. This reasoning can be extended to n redundant parts if at least one of the n parts must succeed for the system to succeed.

Example 8: Suppose that a space capsule can be guided three ways: (1) automatically with $R_1 = 0.9$, (2) semiautomatically with $R_2 = 0.8$, (3) manually with $R_3 = 0.7$. The diagram of successful guiding, assuming that the three ways are independent of each other, is shown in figure 3–7. From probability theorem 3 in chapter 2, the possible events are given by

$$R_1R_2R_3 + R_1R_2Q_3 + R_1Q_2R_3 + Q_1R_2R_3 + R_1Q_2Q_3 + Q_1Q_2R_3 + Q_1R_2Q_3 + Q_1Q_2Q_3$$

Because the sum of these probabilities is equal to unity and at least one of the control systems must operate successfully, the probability that guidance will be successful R_{guidance} is

$$\begin{aligned} R_{\text{guidance}} &= R_1R_2R_3 + R_1R_2Q_3 + R_1Q_2R_3 + Q_1R_2R_3 \\ &\quad + R_1Q_2Q_3 + Q_1Q_2R_3 + Q_1R_2Q_3 \\ &= 1 - Q_1Q_2Q_3 = 1 - \left[(1-R_1)(1-R_2)(1-R_3)\right] \\ &= 1 - \left[(1-0.9)(1-0.8)(1-0.7)\right] \\ &= 1 - \left[(0.1)(0.2)(0.3)\right] \\ &= 1 - (0.006) = 0.994 \end{aligned}$$

In general, then, for simple redundancy

$$R_{\text{simple redundant}} = 1 - \prod_{j=1}^{n} Q_j = 1 - (Q_1Q_2Q_3 \ldots Q_n)$$

where

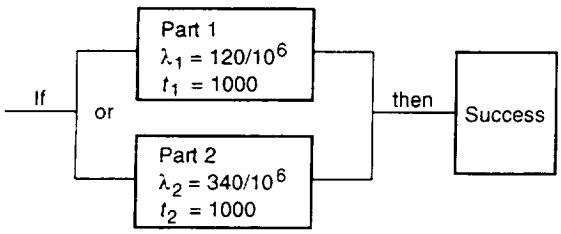


Figure 3–8.—Simple redundancy model using failure rates and operating times.

$\prod_{j=1}^{n} Q_j$ total probability of failure

Q_j total probability of failure of j^{th} redundant part

n total number of redundant parts

Example 9: Find the reliability of the redundant system shown in figure 3–8.

Solution 9:

Step 1—Solve for the reliability of parts 1 and 2:

$$R_1 = e^{-\lambda_1 t_1} = e^{-\left[\left(120/10^6\right)\times 10^3\right]} = e^{-0.120} = 0.887$$

$$R_1 = e^{-\lambda_2 t_2} = e^{-\left[\left(340/10^6\right)\times 10^3\right]} = e^{-0.340} = 0.712$$

Step 2—Solve for the unreliability of each part:

$$Q_1 = 1 - R_1 = 0.113$$
$$Q_2 = 1 - R_2 = 0.288$$

Solve for the reliability of the redundant system:

$$R_{\text{simple redundant}} = 1 - Q_1 Q_2 = 1 - (0.113)(0.288)$$
$$= 1 - 0.033 = 0.967$$

There is a 96.7-percent chance, therefore, that both parts will not fail during the 1000-hr operating time.

Compound redundancy.—Compound redundancy exists when more than one of n redundant parts must succeed for the system to succeed. This can be shown in a model of a three-element redundant system in which at least two of the elements must succeed (fig. 3–9).

From probability theorem 3 in chapter 2, the possible events are

$$R_1R_2R_3 + R_1R_2Q_3 + R_1Q_2R_3 + Q_1R_2R_3 + R_1Q_2Q_3$$
$$+Q_1Q_2R_3 + Q_1R_2Q_3 + Q_1Q_2Q_3$$

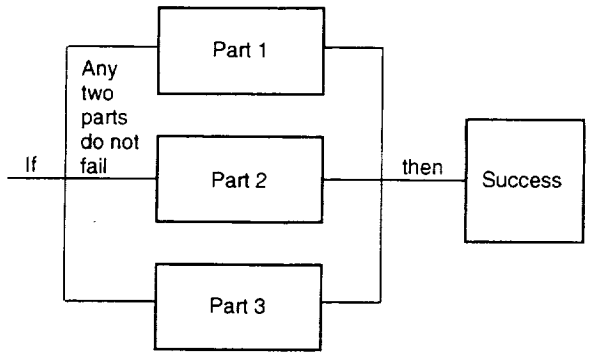


Figure 3–9.—Compound redundancy model.

To simplify the notation, let $R_1 = R_2 = R_3$ and $Q_1 = Q_2 = Q_3$. This reduces the expression to

$$R^3 + R^2Q + R^2Q + R^2Q + RQ^2 + RQ^2 + RQ^2 + Q^3$$

or

$$R^3 + 3R^2Q + 3RQ^2 + Q^3$$

Because the sum of these probabilities equals unity and at least two of the three parts must succeed, the probability for success is given by

$$R_s = R^3 + 3R^2Q = 1 - \left(3RQ^2 + Q^3\right)$$

where $3RQ^2$ represents one part succeeding and two parts failing and Q^3 represents all three parts failing.

Example 10: Assume that there are four identical power supplies in a fire control center and that at least two of them must continue operating for the system to be successful. Let each supply have the same reliability, $R = 0.9$ (which could represent $e^{-\lambda t}$ or R_i or R). Find the probability of system success $R_{\text{simple redundant}}$.

Solution 10: The number of possible events is given by

$$(R+Q)^4 = R^4 + 4R^3Q + 6R^2Q^2 + 4RQ^3 + Q^4$$

The sum of the probabilities of these events equals unity; therefore, the expression for two out of four succeeding is

$$R_s = R^4 + 4R^3Q + 6R^2Q^2 = 1 - \left(4RQ^3 + Q^4\right)$$

Substituting $R = 0.9$ and $Q = 1 - 0.9$ gives

$$R_s = 1 - \left(4RQ^3 + Q^4\right) = 1 - \left[4(0.9)(0.1)^3 + (0.1)^4\right]$$
$$= 1 - \left[(3.6)(0.001) + 0.0001\right] = 1 - (0.0036 + 0.0001)$$
$$= 1 - 0.0037 = 0.996$$

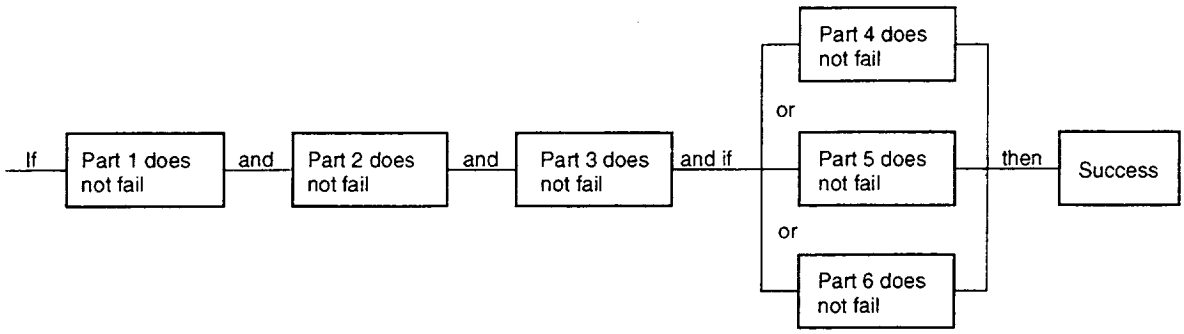


Figure 3–10.—Model of system with series and redundant elements.

Calculation of Reliability for Complete System

To find the reliability for a complete system, begin by developing a model for the system, write the equation for the probability of success from the model, and then use the failure rates and operating times of the system elements to calculate the reliability of the system (refs. 3–6 to 3–8).

Example 11: Consider the system model with series and redundant elements shown in figure 3–10.

Solution 11: The equation can be written directly as

$$R_s = R_1 R_2 R_3 (1 - Q_4 Q_5 Q_6)$$

where $R_1 R_2 R_3$ represents the probability of success of the series parts and $(1 - Q_4 Q_5 Q_6)$ represents the probability of success of the three parts in simple redundancy. If we know that

$$R_1 = 0.99 = e^{-0.01} \qquad R_4 = 0.85$$
$$R_2 = 0.999 = e^{-0.001} \qquad R_5 = 0.89$$
$$R_3 = 0.95 = e^{-0.05} \qquad R_6 = 0.78$$

where R may represent $e^{-\lambda t}$, inherent reliability R_i, or observed product reliability depending on the stage of product development, then the reliability of the system is

$$R_s = e^{-0.01} e^{-0.001} e^{-0.05} [1 - (1 - 0.85)(1 - 0.89)(1 - 0.78)]$$
$$= e^{-0.061} [1 - (0.15)(0.11)(0.22)] = e^{-0.061} (1 - 0.00363)$$
$$= e^{-0.061} e^{-0.0036} = e^{-0.065} = 0.935$$

However, this does not mean that there will be no equipment failures. The system will still succeed even though one or two of the redundant paths have failed.

Example 12: Write the equation for the system shown in figure 3–11.

Solution 12: The equation can be written directly as

$$R_s = R_1 R_2 [1 - (R_3 Q_4 Q_5 + Q_3 R_4 Q_5 + Q_3 Q_4 R_5 + Q_3 Q_4 Q_5)](1 - Q_6 Q_7)$$

where $R_1 R_2$ is the probability that the two parts in series will not fail, $1 - (R_3 Q_4 Q_5 + \ldots + Q_3 Q_4 Q_5)$ is the probability that two out of three of the compound redundant parts will not fail, and $(1 - Q_6 Q_7)$ is the probability that both the simple redundant parts will not fail. If data giving the reliabilities of each part are available, insert this information in the system success equation to find the system reliability.

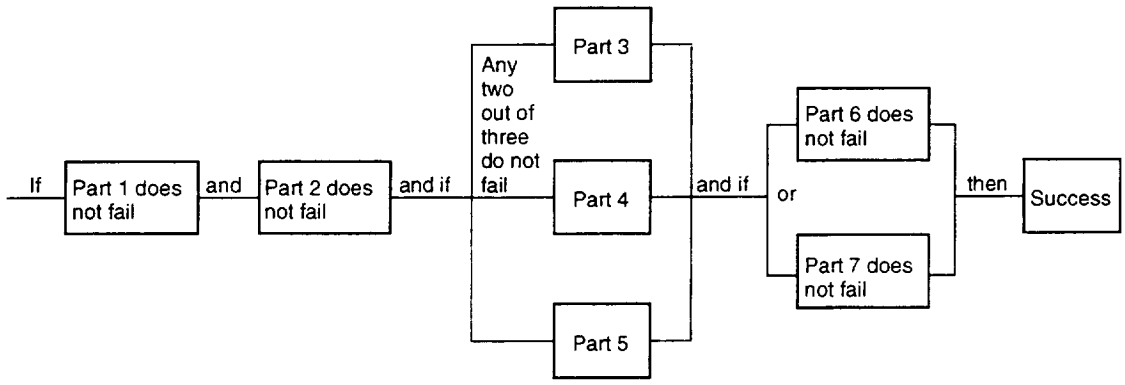


Figure 3–11.—System reliability model using series, simple redundancy, and compound redundancy elements.

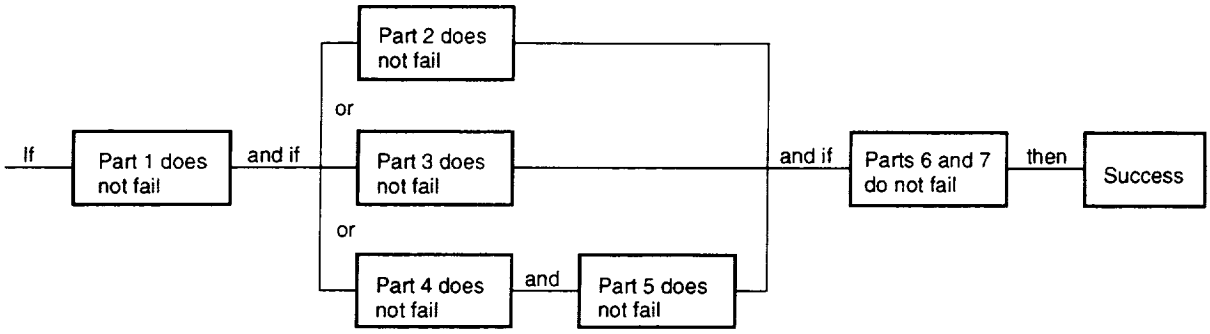


Figure 3–12.—Model with series elements in redundant paths.

Example 13: Write the equation for the system shown in figure 3–12.

Solution 13: The equation can be written directly as

$$R_s = R_1 R_6 R_7 \left\{1 - \left[Q_2 Q_3 (1 - R_4 R_5)\right]\right\}$$

where $R_1 R_6 R_7$ is the reliability of the series parts, $(1 - R_4 R_5)$ is the probability that R_4 or R_5 will fail in the bottom redundant path, and $\{1 - [Q_2 Q_3 (1 - R_4 R_5)]\}$ is the reliability of the three paths in simple redundancy.

Concluding Remarks

Chapter 3 has presented several important concepts that you should have clearly in mind:

(1) The exponential distribution $e^{-\lambda t}$ represents the probability that no catastrophic part failures will occur in a product.

(2) The failure rate λ as used in $e^{-\lambda t}$ is a constant and represents the rate at which random catastrophic failures occur.

(3) Although the cause of failure is known, random failures may still occur.

(4) The mean time between failures (MTBF) is the reciprocal of the failure rate.

(5) In reliability, devices are in series if each one is required to operate successfully for the system to be successful. Devices are parallel or redundant if one or more can fail without causing system failure but at least one of the devices must succeed for the system to succeed.

In addition, you should be able to calculate the following:

(1) The reliability of a device, given failure rate and operating time

(2) The reliability of devices connected in series from the product rule

$$R_s = \prod_{j=1}^{n} R_j$$

(3) The reliability of devices connected in simple redundancy from

$$R_{\text{simple redundant}} = 1 - \prod_{j=1}^{n} Q_j$$

(4) The reliability of n devices connected in compound redundancy by expanding $(R + Q)^n$ and collecting the appropriate terms.

And finally, you should be able to combine the four methods described above to calculate the reliability of a total system.

In 1985, alternative methodologies were introduced in the form of computer reliability analysis programs. One such underlying model uses a Weibull failure rate during the burning, or "infant mortality," period and a constant failure rate during the steady-state period for electronic devices. Initial results indicate that given a 15- to 40-yr system life, the infant mortality period is assumed to last for the first year. Of course, the higher the stress of the environment, the shorter the period of infant mortality. The point is that there are many ways to perform reliability studies, and different methodologies could be equally appropriate or inappropriate. Appendix C describes five distribution functions that can be used for reliability analysis. Table C–1 shows the time-to-failure fit for various systems. The basic criteria relate to the distribution of failures with time.

References

3–1. Failure Distribution Analyses Studies, Vols. I, II, and III. Computer Applications Inc., New York, Aug. 1964. (Avail. NTIS: AD–631525, AD–631526, AD–631527.)

3–2. Hoel, Paul G.: Elementary Statistics. John Wiley & Sons, Inc., 1960.

3–3. Calabro, S.: Reliability Principles and Practices. McGraw-Hill, 1962.

3–4. Reliability Prediction of Electronic Equipment. MIL–HDBK–217E, Jan. 1990.

3–5. Electronic Reliability Design Handbook. MIL–HDBK–338, Vols. I and II, Oct. 1988.

3–6. Bloomquist, C.; and Graham, W.: Analysis of Spacecraft On-Orbit Anomalies and Lifetimes, (PRC R–3579. PRC Systems Sciences Co.; NASA Contract NAS5–27279), NASA CR–170565, 1983.

3–7. Government-Industry Data Exchange Program (GIDEP). Reliability-Maintainability (R–M) Analyzed Data Summaries. Vol. 7, Oct. 1985.

3–8. Kececiouglu, D.: Reliability Engineering Handbook, Vols. 1 and 2. Prentice-Hall, 1991.

Reliability Training[1]

1a. Of 45 launch vehicle flights, 9 were determined to be failures. What is the observed reliability?

A. 0.7 B. 0.8 C. 0.9

1b. What is the observed reliability if the next five flights are successful?

A. 0.72 B. 0.82 C. 0.87

1c. After the five successes of part 1b, how many more successes (without additional failures) are required for a reliability of $R = 0.90$?

A. 20 B. 30 C. 40

2. A three-stage launch vehicle has a reliability for each stage of $R_1 = 0.95$, $R_2 = 0.94$, $R_3 = 0.93$.

a. What is the probability of one successful flight?

A. 0.83 B. 0.85 C. 0.87

b. What is the probability of flight failure for part a?

A. 0.00021 B. 0.15 C. 0.17

c. What is the probability of two successful flights?

A. 0.689 B. 0.723 C. 0.757

3. You are taking a trip in your car and have four good tires and a good spare. By expanding $(R + Q)^5$,

a. How many events (good tires or flats) are available?

A. 16 B. 32 C. 64

b. How many combinations provide four or more good tires?

A. 6 B. 7 C. 16

c. If $R = 0.99$ for each tire and a successful trip means you may have only one flat, what is the probability that you will have a successful trip?

A. 0.980 B. 0.995 C. 0.9990

4. A launch vehicle system is divided into five major subsystems, three of which have already been built and tested. The reliability of each is as follows: $R_1 = 0.95$, $R_2 = 0.95$, $R_3 = 0.98$. The reliability of the overall system must be equal to or greater than 0.85. What will be the minimum acceptable reliability of subsystems 4 and 5 to ensure 85-percent reliability?

A. 0.92 B. 0.95 C. 0.98

[1]Answers are given at the end of this manual.

5a. A launch vehicle test program consists of 20 test firings requiring 90-percent reliability. Five tests have already been completed with one failure. How many additional successes must be recorded to successfully complete the test program?

A. 13 B. 14 C. 15

5b. Based on the probability (four successes in five flights), what is the probability of achieving successful completion of the test program?

A. 0.04 B. 0.167 C. 0.576

6. During individual tests of major launch vehicle subsystems, the reliability of each subsystem was found to be

Subsystem 1 = 0.95
Subsystem 2 = 0.99
Subsystem 3 = 0.89
Subsystem 4 = 0.75

Since all subsystems are required to function properly to achieve success, what increase in reliability of subsystem 4 would be necessary to bring the overall system reliability to 0.80?

A. 15 percent B. 20 percent C. 25 percent

7. Solve for the following unknown values:

a. $\lambda = 750 \times 10^{-6}$ failures/hr; $t = 10$ hr; $R = ?$

A. 0.9925 B. 0.9250 C. 0.9992

b. $\lambda = 8.5$ percent failures/10^3 hr; $t = 3000$ hr; $R = ?$

A. 0.9748 B. 0.7986 C. 0.0781

c. MTBF = 250 failures/hr; $t = 0.5$ hr; $R = ?$

A. 0.9802 B. 0.9980 C. 0.9998

d. $R = 0.999$; $t = 10$ hr; $\lambda = ?$

A. 1000×10^{-9} failures/hr B. 10×10^{-6} failures/hr C. 10 percent failures/10^3 hr

e. MTBF = ?

A. 10^4 failures/hr B. 10^5 failures/hr C. 10^6 failures/hr

8. The a priori MTBF prediction of a printed circuit board was 12.5×10^6 hr. Find the number of expected failures during a 10^8-hr (accelerated) life test of 10 circuit board samples.

A. 12.5 B. 80 C. 125

9a. Write the reliability equation for the battery activation success diagram shown below:

If		And	And	And	Then
Battery activates command (part 1)	Passes umbilical path (part 2)	Initiates EBW 1 (part 3) or EBW 2 (part 4)	Ignites initiator 1 (part 5) or initiator 2 (part 6)	Battery activates (part 7)	Success

A. $R_s = R_1R_2(1 - R_3R_4)(1 - R_5R_6)R_7$ B. $R_s = R_1R_2(1 - Q_3Q_4)(1 - Q_5Q_6)R_7$

9b. If $R = 0.9$ for all series and $R = 0.8$ for all parallel parts, solve for R_s.

A. 0.73 B. 0.26 C. 0.67

10. A launch vehicle subsystem is required to be stored for 10 years (use 9000 hr = 1 year). If the subsystem reliability goal is 0.975.

a. What λ is required with no periodic checkout and repair?

A. 2800×10^{-9} B. 28×10^{-9} C. 280×10^{-9}

b. What λ is required with checkout and repair every 5 years? (Assume 100-percent checkout.)

A. 5600×10^{-9} B. 56×10^{-9} C. 560×10^{-9}

c. What λ is required with checkout and repair every year? (Assume 100-percent checkout.)

A. 2800×10^{-9} B. 28×10^{-9} C. 280×10^{-9}

Chapter 4
Using Failure-Rate Data

Now that you have a working knowledge of the exponential distribution $e^{-\lambda t}$ and have the fundamentals of series and redundant models firmly in mind, the next task is to relate these concepts to your everyday world. To do this, we explore further the meaning of failure rates, examine variables that affect part failure modes and mechanisms, and then use part failure rate data to predict equipment reliability. We introduce a simple technique for allocating failure rates to elements of a system. The concepts discussed in this chapter are tools the designer can use for trading off reliability with other factors such as weight, complexity, and cost. These concepts also provide guidelines for designing reliability into equipment during the concept stage of a program.

Variables Affecting Failure Rates

Part failure rates are affected by (1) acceptance criteria, (2) all environments, (3) application, and (4) storage. To reduce the occurrence of part failures, we observe failure modes, learn what caused the failure (the failure stress), determine why it failed (the failure mechanism), and then take action to eliminate the failure. For example, one of the failure modes observed during a storage test was an "open" connection in a wet tantalum capacitor. The failure mechanism was end seal deterioration, which allowed the electrolyte to leak. One obvious way to avoid this failure mode in a system that must be stored for long periods without maintenance is not to use wet tantalum capacitors. If this is impossible, the best solution would be to redesign the end seals. Further testing would be required to isolate the exact failure stress that produces the failure mechanism. Once isolated, the failure mechanism can often be eliminated through redesign or additional process controls.

Operating Life Test

The tests involved 7575 parts—3930 resistors, 1545 capacitors, 915 diodes, 1080 transistors, and 105 transformers. One-third of the parts were operated at –25 °F, one-third at 77 °F, and one-third at 125 °F. The parts, tested in circuits (printed circuit boards), were derated no more than 40 percent. The ordinate of the curve shows cumulative failures as a function of operating time. For example, at about 240 hours, the first failure was observed and at about 385 hours, the second. Several important observations can be made concerning failure rates and failure modes.

Constant Failure Rate.—Figure 4–1 shows that the failure rate for the first 1600 hr is constant at one failure every 145 hr. This agrees with the constant-λ theory. Bear in mind that constant failure rate is an observation and not a physical law. Depending on the equipment, failure rates may decrease or increase for a period of time.

Random Nature.—Notice that the failures in this constant-failure-rate region are random (in occurrence). For example, two diodes fail, then three transistors, then a silicon switch, then a diode, then a trimpot and a resistor, and so on.

Repetitive Failures.—Figure 4–1 also shows that during the first 1600 hr, only two of these failures involved the same type of device. This is important because in most systems the problems that receive the most attention are the repetitive ones. It should be apparent in this case that the repetitive failures are not the ones that contribute the most to unreliability (failure rate); taking corrective action on the repetitive type of failure would only improve the observed failure rate by 18 percent.

Failure modes.—Table 4–1 shows the observed failure modes (the way the failures were revealed) for the transistor, diode, and resistor failures given in figure 4–1. In table 4–l(a), note that the short failure mode for transistors had an occurrence rate five times that of any other mode. Note also that the eight transistor failures were distributed about evenly in the three environments but that some different failure modes were observed in each environment.

Observe again in table 4–l(b) that the short failure mode for diodes occurred most frequently. The failures were not distributed evenly in each environment, but a different failure mode occurred in each environment.

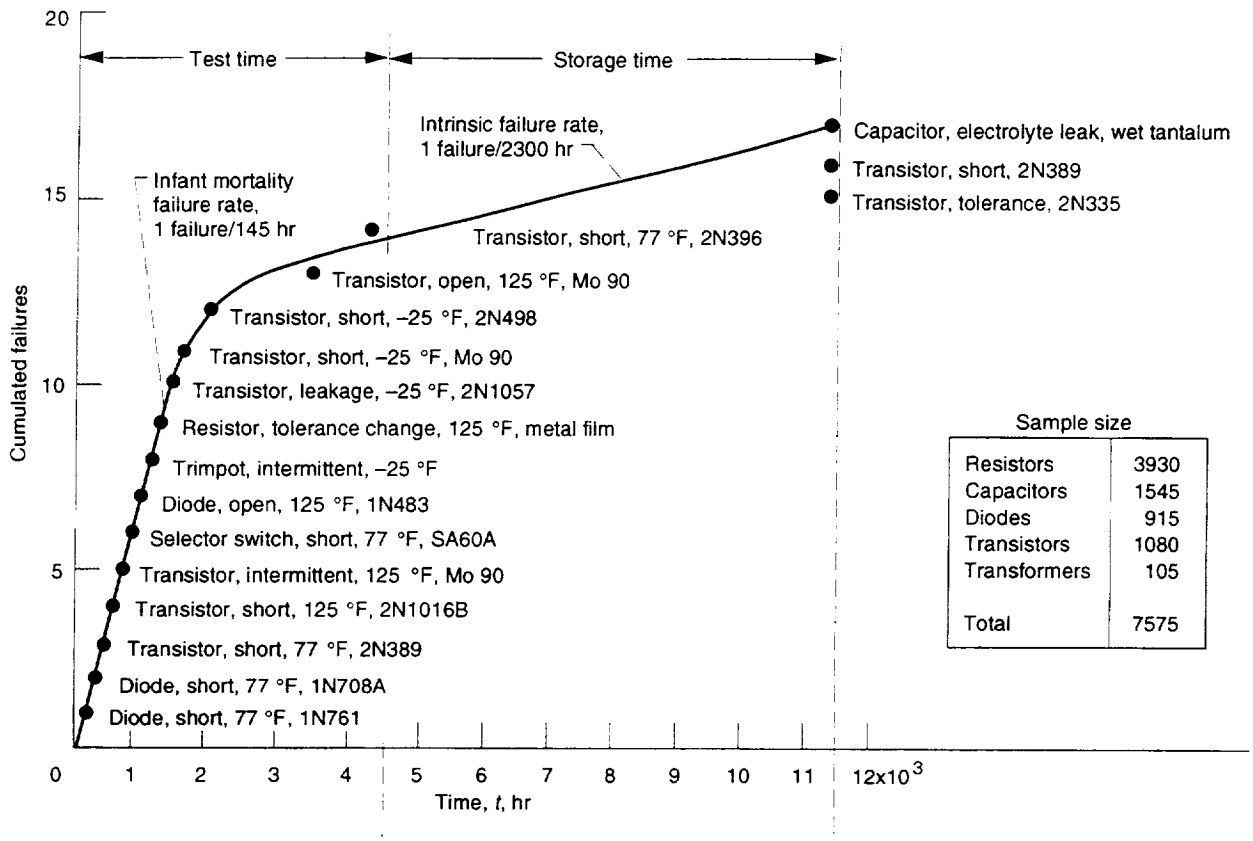


Figure 4–1.—Observed part failures versus test and storage time.

Resistors failed in two modes (table 4–I(c)): one intermittent resistor at low temperatures and one tolerance failure at high temperatures.

Burn-in.—As shown in figure 4–1 after 1600 hr, the failure rate of the 7575 parts dropped by a factor of 7 for the remaining 2900 test hours (3 failures per 2900 hr, failures 12, 13, and 14, as compared with 11 failures per 1600 hr). This is an example of what are commonly called burn-in failures. The first 11 failures represent parts that had some defect not detected by the normal part screening or acceptance tests. Such defects do not reveal themselves until the part has been subjected to operation for some time. As mentioned earlier, eliminating the repetitive failure would only decrease the failure rate in the first 1600 hr by about 18 percent, but if screening tests were sensitive enough to detect all defects, the failure rate would approach the intrinsic failure rate shown in figure 4–1 right from the start.

In summary, some of the observed properties of operating failure rates are as follows:

(1) For complex equipment, the intrinsic failure rate of electronic parts is usually constant in time.

(2) Failures are random, with repetitive failures representing only a small portion of the problems.

(3) Failure modes of parts and equipment vary, depending on the operating environment.

(4) Most parts have a dominant failure mode. For example, the dominant failure mode for semiconductors is shorting.

(5) Rigid part screening and acceptance criteria can substantially reduce operating failure rates by eliminating early failures.

Storage Test

After the operating test, the parts were put in storage for approximately 7000 hr (10 months) and then were retested to determine the effect of storage on parts. As shown in figure 4–1, three failures (14, 15, and 16) were observed at the end of the storage period. Note that the average failure rate observed in storage (one failure per 2300 hr) is close to the same rate observed in the previous 2900 hr of operation. Thus, it can be concluded that storage does produce part failures and that the storage failure rate may be as high as the operating rate. Industry is conducting a great deal of research on this problem because storage failure rates become a significant factor in the reliability of unmanned systems and affect considerably the maintenance policy of manned systems.

Summary of Variables Affecting Failure Rates

Part failure rates are thus affected by

TABLE 4–1.—FAILURE MODES

(a) Transistors

Observed part failure mode	Temperature, °F			Total failures	Observed failure rate, failures/hr
	–25	77	125		
Open	----------	----------	MD–90	1	$0.206/10^6$
Short	MD–90	2N389	2N1016B	5	$1.03/10^6$
	2N498	2N396	----------		
Intermittent	----------	----------	MD–90	1	$.206/10^6$
Leakage	2N1057	----------	----------	1	$.206/10^6$
Totals	3	2	3	8	$1.65/10^6$

(b) Diodes

Open	----------	----------	1N483	1	$0.24/10^4$
Short	----------	1N761	----------	3	$.73/10^4$
		1N708A			
		SA60A			
Totals	0	3	1	4	$0.97/10^4$

(c) Resistors

Intermittent	Trimpot	----------	----------	1	$0.06/10^6$
Tolerance	----------	----------	Metal film	1	$.06/10^6$
Totals	1	0	1	2	$0.12/10^6$

TABLE 4–3.—STRESS RATIOS THAT MEET ALLOCATION REQUIREMENT

Part temperature, °C	Stress ratio, W					
	0.1	0.2	0.3	0.4	0.5	0.6
	Failure rate of derated part per 10^6 hr, λ_p					
30					0.23	0.22
40				0.24		
50			0.24			
60		0.25				
70	0.25					

TABLE 4.-2.—FAILURE RATE CALCULATION

(a) Tactical fire control station logic gate

Component	Stress ratio, W	Number used, N	Failure rate of derated part at 40 °C λ_p, failures/10^6 hr	Application factor for vehicle, ground mounted, K_A	Total failure rate, $\lambda_T = N\lambda_p K_A$, failures/$10^6$ hr
Resistor, composition (2000 Ω)	0.5	1	0.0035	10	0.035
Resistor, composition (180 000 Ω)	.5	1	.0035	↓	.035
Resistor, composition (22 000 Ω)	.6	1	.0038		.038
Resistor, composition (6500 Ω)	.5	2	.0035		.070
Transistor, germanium (PNP type)	<1 W; 0.4 normalized junction temperature	1	1.3	8	10.400
Diode, 1N31A	.3	1	3.5	5	17.500
				Total, $\lambda_c = \Sigma\lambda_T = 29.68$	

(b) Proposed logic gate

Resistor, film (1300 Ω)	0.8	1	0.19	0.3	0.057
Resistor, film (3320 Ω)	.2	↓	.14	.3	.042
Resistor, film (46 600 Ω)	.2		.14	.3	.042
Transistor, silicon (NPN type)	<1 W; 0.15 normalized junction temperature		.165	8	1.320
Diode, 1N31A	.2	5	3.0	5	75.000
				Total, $\lambda_c = \Sigma\lambda_T = 76.461$	

(1) Acceptance criteria
(2) All environments
(3) Application
(4) Age or storage

To find ways of reducing the occurrence of part failures, we observe failure modes, learn what caused the failure (the failure stress), determine why it failed (the failure mechanism), and then take action to eliminate the failure. For example, one of the failure modes observed during the storage test was an "open" in a wet tantalum capacitor. The failure mechanism was deterioration of the end seals, which allowed the electrolyte to leak. One obvious way to avoid this failure mode in a system that must be stored for long periods without maintenance is not to use wet tantalum capacitors. If this is impossible, the next best thing would be to redesign the end seals. This would no doubt require further testing to isolate the exact failure stress that produces the failure mechanism. Once isolated, the failure mechanism can often be eliminated through redesign or additional process controls.

One of the best known methods of representing part failures is the use of failure rate data. Figure 4–2 (from ref. 4–1) shows a typical time-versus-failure-rate curve for flight hardware. This is the well-known "bathtub curve," which over the years has become widely accepted by the reliability community and has proven to be particularly appropriate for electronic equipment and systems. It displays the sum of three failure rate quantities: quality (QFR), stress (SFR), and wearout (WFR).

Zone 1, the infant mortality period, is characterized by an initially high failure rate (QFR). This is normally the result of poor design, use of substandard components, or lack of adequate controls in the manufacturing process. When these mistakes are not caught by quality control operations, an early failure is likely to result. Early failures can be eliminated by a "burn-in" period during which time the equipment is operated at stress levels closely approximating the intended actual operating conditions. The equipment is then released for actual use only when it has successfully passed through the burn-in period. For most well-described complex equipment, a 100-hr failure-free burn-in is usually adequate to cull out a large proportion of the infant mortality failures caused by stresses on the parts.

Zone II, the useful life period, is characterized by an essentially constant failure rate (SFR). This is the period dominated by chance failures, defined as those failures that result from strictly random or chance causes. They cannot be eliminated by either lengthy burn-in periods or good preventive maintenance practices.

Equipment is designed to operate under certain conditions and to have certain strength levels. When these strength levels are exceeded because of random unforeseen or unknown events, a chance failure will occur. Although reliability theory and practice are concerned with all three types of failure, the primary concern is with chance failures since they occur during the useful life of the equipment. Figure 4–2 is somewhat

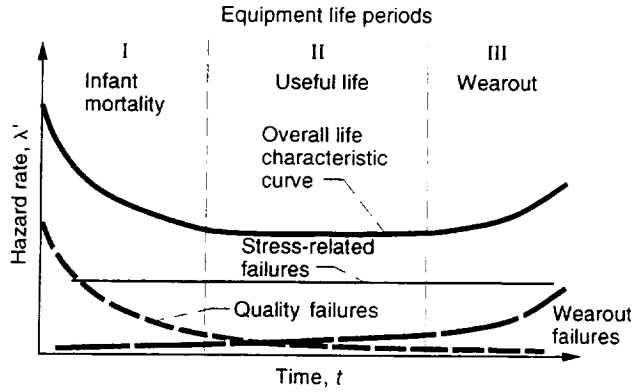


Figure 4–2.—Hazard rate versus equipment life periods.

deceiving because zone II is usually much longer than zone I or III. The time when a chance failure will occur cannot be predicted, but the likelihood or probability that one will occur during a given period of time within the useful life can be determined by analyzing the equipment design. If the probability of a chance failure is too great, either design changes must be introduced or the operating environment made less severe.

The SFR period is the basis for the application of most reliability engineering design methods. Because it is constant, the exponential distribution of time to failure is applicable and is the basis for the design and prediction procedures spelled out in documents such as MIL–HDBK–217E (ref. 4–2).

The simplicity of the approach (utilizing the exponential distribution, as previously indicated) makes it extremely attractive. Fortunately, it is widely applicable for complex equipment and systems. If complex equipment consists of many components, each having a different mean life and variance that are randomly distributed, then the system malfunction rate becomes essentially constant as failed parts are replaced. Thus, even though the failures might be wearout failures, the mixed population causes them to occur at random intervals with a constant failure rate and exponential behavior. This has been verified for much equipment from electronic systems to rocket motors.

Zone III, the wearout period, is characterized by an increasing failure rate (WFR) resulting from equipment deterioration due to age or use. For example, mechanical components, such as transmission bearings, will eventually wear out and fail regardless of how well they are made. Early failures can be postponed and the useful life extended by good design and maintenance practices. The only way to prevent failure due to wearout is to replace or repair the deteriorating component before it fails.

Because modern electronic equipment is almost completely composed of semiconductor devices that really have no short-term wearout mechanism, except for perhaps electromigration, one might question whether predominantly electronic equipment will even reach zone III of the bathtub curve.

Different statistical distributions might be used to characterize each zone. Hazard rate has been defined for five different failure distribution functions Depending on which distribution

fits the hazard rate data best, a failure distribution function can be selected. The infant mortality period for the typical hazard rate in figure 4–2 might be represented by the Weibull distribution, the useful life period by the exponential distribution, and the wearout period by the log normal distribution.

Part Failure Rate Data

It is common in the field of reliability to represent part integrity or reliability in terms of failure rate or mean time between failures (MTBF). In general, part failure rates are presented as a function of temperature and electrical stress as shown in figure 4–3. The family of curves on the graph represents different applied electrical stresses in terms of a stress ratio or derating factor. For example, if a part is to operate at temperature A and is derated 20 percent (stress ratio, 0.8), that part will have a failure rate of $\lambda = 0.8$ as shown. If the part is derated 70 percent (stress ratio, 0.3), it will have a failure rate of $\lambda = 0.3$, etc. Failure rate is usually given in failures per 10^6 hr although as indicated in chapter 3, other dimensions are used depending on who publishes the data.

The current authoritative failure rate data published by the Department of Defense are in MIL–HDBK–217E (ref. 4–2). The MIL–HDBK–217 series is a direct result of the 1952 AGREE effort mentioned in chapter 1. The publications listed in table 1–1 and in references 4–3 to 4–5 are also offshoots of this effort to meet the need for authoritative, statistically based part failure rates. Because new data on both existing and new state-of-the-art parts are constantly being generated and analyzed, failure rate handbooks do change. Therefore, be sure to use the latest version available. Even the latest version of the data used for compiling the handbook may not represent the parts you are using. The best procedure is to use your own failure rate data with modern computer-aided software to simulate your designs.

As emphasized in chapter 3, failure rates are statistical, and there is no such thing as an absolute failure rate. Consider the simple definition of failure rate:

$$\lambda = \frac{\text{Number of observed failures}}{\text{Total operating time}}$$

Obviously, if today we observe two failures in 100 hr and tomorrow we accumulate no more failures, the new failure rate is two failures in 124 hr. Then, if a failure occurs in the next 1-hr period, the failure rate is three failures in 125 hr. Therefore, we can never know what the true failure rate is, but we can determine representative failure rates or best estimates from many hours of observed operating time. This type of failure rate data is presented in the MIL–HDBK–217 series.

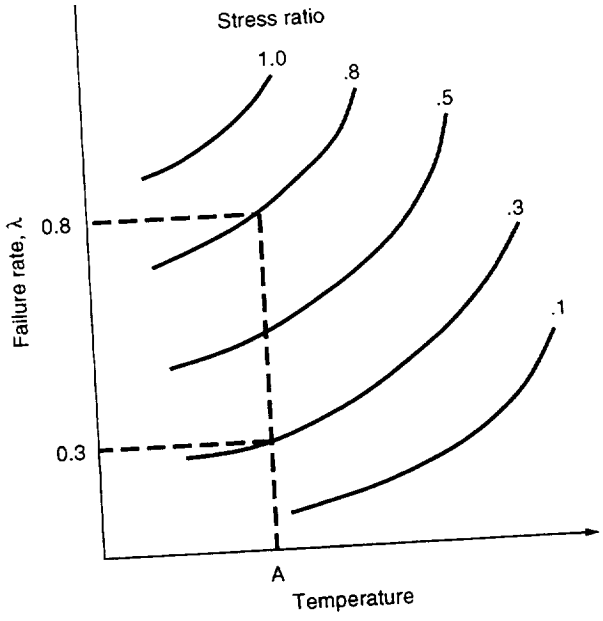


Figure 4–3.—Failure rate versus electrical stress ratio and temperature.

Improving System Reliability Through Part Derating

The best way to explain how to derate a component is to give an example. Consider two 20-V wet-slug tantalum capacitors, both to be operated at a component temperature of 165 °F. One is to be operated at 20 V and the other at 12 V. First, find the stress ratio or operating-to-rated ratio for both applications:

$$\text{Stress ratio} = \frac{\text{Operating voltage}}{\text{Rated voltage}}$$

Hence, one capacitor has a stress ratio of 1.0,

$$\text{Stress ratio} = \frac{20\ \text{V}}{20\ \text{V}} = 1.0$$

and the other, a stress ratio of 0.6,

$$\text{Stress ratio} = \frac{12\ \text{V}}{20\ \text{V}} = 0.6$$

(A stress ratio of 0.6 is the same as "derating" the component 40 percent.) To find the failure rate λ for each capacitor, go to the MIL–HDBK–217E (ref. 4–2) table for MIL–C–3965 glass-sealed, wet-slug capacitors. Move horizontally across the 165 °F line to the vertical 0.6 and 1.0 stress ratio columns and read directly:

Importance of Learning From Each Failure

When a product fails, a valuable piece of information about it has been generated because we have the opportunity to learn how to improve the product if we take the right actions:

Failures can be classified as

(1) Catastrophic (a shorted transistor or an open wire-wound resistor)
(2) Degradation (change in transistor gain or resistor value)
(3) Physical wearout (brush wear in an electric motor)

These three failure categories can be subclassified further:

(1) Statistically independent (a shorted capacitor in a radio-frequency amplifier being unrelated to a low-emission cathode in a picture tube)
(2) Cascade (the shorted capacitor in the radio-frequency amplifier causing excessive current to flow in its transistor and burning the collector beam lead open)
(3) Common mode (a circuit board used for primary control of a process and a backup circuit board both burned out by an over-voltage condition in a power supply that feeds the two of them)

On the basis of the following categories, much can be learned from each failure that occurs during flight acceptance testing for a mission: good failure reporting, conducting failure analyses, maintaining a concurrence system, and taking corrective action. Failure analysis determines what caused the part to fail. Corrective action ensures that the cause is dealt with. Concurrence informs management of actions being taken to avoid another failure. These data enable all personnel to compare the part ratings with the use stresses and to verify that the part is being used with a known margin.

Failure Reporting, Analysis, Corrective Action, and Concurrence

Many different methods can be used to record reliability data for any given project. The Department of Defense has standardized a method on DD form 787–1. A simple form that tells the whole story on one sheet of paper is NASA–C–8192 (fig. 4-6). The method that you use to record reliability data will have to fit your needs. Keep your form simple and easy to fill out, and get approval from management.

Case Study—Achieving Launch Vehicle Reliability

Design Challenge

The launch vehicle studied requires the highest acceleration and velocity and the shortest reaction time of any developed. As such, the design challenges were formidable; typical in-flight environments include random vibration of 61 g's rms up to 3 kHz, mechanical shock at 25 000 g's peak (between 5 and 10 kHz), linear acceleration well in excess of 100 g's, acoustics of 150 dB, and aerodynamic heating up to 6200 °F. The development philosophy was that a vehicle be launched from a tactical silo with the initial design. Although many changes occurred during the 13-year development, the first flight test vehicle was not greatly different from the 70 now deployed.

Subsystem Description

The vehicle is launched from an underground silo, which also serves as a storage container during the multiyear design life. Adjacent to the silo and integral to it is a small compartment housing the ground support equipment. This equipment is used to conduct periodic tests of the vehicle electronics, to prepare the vehicle for launch, and to launch the vehicle. It also maintains the silo environment at 80±10 °F and 50 percent or less relative humidity.

The vehicle is predominantly in a power-off storage mode when deployed in its silo. A periodic test of flight electronics is conducted automatically every 4 weeks. In a multiyear design life, the flight electronics accumulate about 11 min of operating time and 43 830 hr of storage time. The ratio of storage time to operating time is nearly 240 000:1.

Approach to Achieving Reliability Goals

Reliability mathematical models were developed early in the research and development program. From these models it was apparent that the following parameters were the most important in achieving the reliability goals:

(1) Electronic storage failure rate during a multiyear design life (i.e., storage failures)
(2) Percent testability of missile electronics (i.e., MIL–STD–471A, ref. 4–6)
(3) Periodic test interval for missile electronics
(4) Severity of in-flight environments (acceleration, shock, vibration, and aerodynamic heating)

National Aeronautics and Space Administration
Glenn Research Center

Page _____ of _____

PROBLEM REPORT # _____ (Hardware __ Software __)

A. Project Name _____ Procedure No. _____ Date Identified _____

Assy/CSCI Name _____ ID No. _____ Location _____

Type: 1. Eng/Qual ___ Process: 1. Inspect ___ 3. Design ___ 5. Test ___ -Type: _____
2. Flight ___ 2. Assemble ___ 4. Code ___ _____
3. GSE ___

B. Background Info. & Descriptions: *(use continuation sheets as needed)*

Initiator _____ Date _____

C. Analysis/Root Cause/Effect on System *(use continuation sheets as needed)*

Is damage assessment required?_____Yes (Is work sheet attached?_____Yes)

Defect(s) info. (Name, ID, Lot code, Supplier, affected routines/sub-routines/programs, etc.)

Defect Code:_____

Problem Type: ___ Nonconformance ___ Failure Analyst _____ Date _____

D. Disposition: Rework/Rewrite ___ Repair/Patch ___ Use as is ___ Return ___ Scrap ___ Request Waiver ___

E. Corrective Action: *(use continuation sheets as needed)*

Initiated: Eng Chg Order ___, Software Chg Req ___, Waiver Req ___, Request/Order # _____

Project Eng _____ OMS&A _____ Reviewed on: ____/____/____

F. Corrective Action Follow-up: ____/____/____ By *(name & title)*: _____

G. Project Office Approval Signature(s) & Date	OMS&A Approval Signature(s) & Date

NASA-C-8192 (Rev. 4-97) Page 1 of 2 Distribution: Project Mgr. (orig.), OMS&A, Hardware File (Ref. PAI# 440)

(a)

Figure 4.6.—Failure report and analysis forms. (a) Problem report. (b) Damage assessment worksheet. (c) Defect codes.

INSTRUCTIONS (Please print/write legibly)

Problem Report # — Unique number assigned by OMS&A PRACA Administrator.
(Hardware__ Software__) — Analyst select 1 of 2 categories.

Section A — To be completed by person who discovered the problem
Project Name — Name or Acronym of project.
Procedure No. — Title and/or No. of procedure/instructions used to carry out required task.
Date Identified — Date when nonconformance is found or failure occurred.
Assy/CSCI Name — Name of specific pkg., assy., sub-assy. or software pkg. with problem.
ID No. — Part No., Serial No. if there are multiple parts of same design, or SCM# (SW Config. Mgnt. #).
Location — Location where problem is identified, e.g. GRC, KSC, EMI Lab, Machine Shop, etc.
Type — Choose 1 of 3 choices "Engineering/Qualification, Flight or Ground Support Equip."
Process — Choose 1 of 5 choices "Inspection, Assembly, Design, Code or Test".
Test Type — Applied for test processes only, eg. Burn-in, Vib., Thermal Cycle, Integration, Acceptance, etc.

Section B — To be completed by person who discovered the problem
Background Info & Descriptions — How much operating time/cycles did the package have when the problem occurred? Record what was actually measured (actual data), and what it should have been (specifications); and which computer or micro. was running the software?
Initiator — Name of person who initiate report. **Date** — Report date.

Section C — To be completed by responsible Project Engineer/Analyst
Analysis/Root Cause/Effect on System — Brief summary of analysis, describe root cause(s), and effect on system if root cause(s) is not eliminated.
Defective Part(s) Info. — Record defective part(s) name, identification (P/N & S/N), model, lot code, supplier/manufacturer.
Problem Type — Choose 1 of 2 choices "Nonconformance or Failure".
Analyst — Name of analyst. **Date** — Analysis complete date.

Section D — Responsible Project Engineer(s) will choose 1 of 6 disposition choices.
Rework/Rewrite — Correct hardware/software to conform to requirements (dwgs., specs., procedures, etc.).
Repair/Patch — Modify hardware or patch software programs to usable condition.
Use as is — Accept hardware/software as is, without any modifications; or "Work around" – Software remains as is, but further action is required on operator or other systems.
Return — Return to supplier for corrective action (rework, repair, replace, analysis, etc.).
Scrap — Isolate defective material for details analysis, or discard un-usable material.
Request Waiver — Initiate a Waiver Request for authorization to vary from specified requirements.

Section E — Joint effort of Project Engr., OMS&A Rep. & Specialist(s) as needed
Corrective Action — Record specific actions required to eliminate problem(s) and prevent recurrence. Identify extent of software regression testing, and affected routines/programs, including any ECO# (Eng Chg Order), SCQ# (Software Chg Request), and Waiver Request# initiated.
Project Eng. — Responsible project engineer's signature.
OMS&A — Cognizant OMS&A representative's signature.
Reviewed on — Date when Corrective Action plan is reviewed, or Problem Review Board meets.

Section F — To be completed by OMS&A Representative
Corrective Action Follow-up — Date when corrective action is verified. Assure approved waiver is attached if one has been requested. This will be the official "Problem Closure Date".
Verified By — Name of OMS&A Rep. who completed the follow-up.

Section G — Approval Signature Requirements
Problem identified during assembly/inspection — Required sign-off by Project Eng. & OMS&A Rep.
Problem identified during test — Required signatures of Project Engineer, OMS&A Rep., Project Assurance Manager, and Project Manager.

**** Training on PRACA System is available through Assurance Management Office****

NASA-C-8192 (Rev 4-97) Page 2 of 2
(a)

Figure 4.6.—(a) Concluded.

Glenn Research Center

DATE _____

SHEET ____ OF ____

SYSTEM _____

COMPILED BY _____

REFERENCE DRAWING _____

APPROVED BY _____

ITEM/FUNCTIONAL IDENTIFICATION	FUNCTION	FAILURE CAUSES	DAMAGE MODE	DAMAGE EFFECTS			REMARKS
				LOCAL EFFECTS	NEXT HIGHER LEVEL	END EFFECTS	

DAMAGE ASSESSMENT WORKSHEET

(b)

PAI#440
ATTCH.#3.2.6

Figure 4-6.—(b) Damage assessment worksheet.

ATTACHMENT 3.2.7

DEFECT CODES

INITIAL DEFECT (ONLY) CODES		FINAL DEFECT CODES	
TEST FAILURE	CODE	CONTAMINATION	CODE
Component Select (Separate Test)	101	Fluid	251
Combined (POT/COAT)	103	Biological	252
POST (POT/COAT)	105	Corrosion	254
Performance/Functional	106	Particulate	255
Shock	107	Foreign Object	256
Thermal Cycle	109	Contaminated	1006
EMI/EMC	111		
Burn-In	113	ELECTRICAL	
Pre (POT/COAT)	115		
Vibration	117	Incorrect Mounting	261
Thermal Vacuum	119	Connector Damaged	262
X-Ray Examination Reject	120	Incorrect Lead Bend	263
Launch Site Test (Ground Equipment)	122	Unqualified Part	264
Acoustics	123	Short Lead	265
Continuity/Ground	125	Damaged Component	266
Launch Site Test (Airborne Equipment)	126	Long Lead	267
Engine Leaks	127	Burnt/Discolored	268
Leak Test	128	Lead/Wire Damaged	269
Model Survey	131	Wire Size Incorrect	270
Structural Load	132	Birdcaged	271
Thermal Balance	133	Crimping Incorrect	272
Pressurization	134	Insulation Damaged	273
Proof Pressure	135	Missing Part	274
Appendage Deployment	136	Polarity Incorrect	275
Phasing Test	137	Dirty Relay Contacts	276
Alignment Test	138	**Routing Incorrect	277
Weight and CG	139	**Miswired	278
		**Other	279
SUSPECT		**Wrong Part	280
NOTE: Temporary code must be changed before final closeout.		Incorrect Reference Designators	2028
Suspect	750	MECHANICAL	
Suspect as a result of DC&R activity	760		
		Incorrect Part	281
		Binding Stuck or Jammed	282
		Dissimilar Metals	283
		Excess Bonding	284
		Holes Incorrect	285
		Lack of Proper Lubrication	286
		Insufficient Bonding	288
		Interference	289
		Bent Contacts/Pins	290
		Misaligned	291
		Missing Part	292
		Improper Assembly	293
		Safety Wire Items	294
		Weight	295
		Torque Values Incorrect	296
		Part Damaged	298
		Does Not Engage or Lock Correctly	299
		Incorrect Dimensions	2001

(c)

Figure 4–6.—(c) Defect codes.

FINAL DEFECT CODES (Continued)

MECHANICAL (continued)	CODE
Location	2002
Missing or Extra	2003
Insert	2004
Rework/Repair Damages	2025
Detail Out of Tolerance	6001
Layout	6002
Bend Radius/Angle	6003
Made in Reverse	6004
Undersize Machine/Grind	6005
Incorrect Loft Lines Used	6006
DAMAGE	
Packaging/Handling	301
Launch	303
During Fabrication	305
During Usage	306
During Transportation	307
During Test	308
Damage	1009
Damaged PWB	2046
DIMENSIONAL	
Inside Dimension Distorted	401
Incorrect Length	402
Inside Dimension Undersize	403
Incomplete-Missing	404
Outside Dimension Distorted	405
Mislocated Feature	406
Outside Dimension Oversize	407
Surface Finish	408
Thickness Oversize	409
Outside Dimension Undersize	410
Thickness Undersize	411
Incorrect Width	412
Inside Dimension Oversize	413
Inside Diameter Undersize	416
Inside Diameter Oversize	417
Outside Diameter Undersize	418
Outside Diameter Oversize	419
Flatness	420
Straightness	421
Roundness	422
Cylindricity	423
Perpendicularity	424
Angularity	425
Parallelism	426
Profile	427
Runout-Total Runout	428
True Position	429

DIMENSIONAL (continued)	CODE
Burrs-Sharp Edges	431
Threads	432
Angle	433
Depth	434
DOCUMENTATION	
Other Documentation	450
Test Reports/Certs in Error/Not Complete	452
Test Reports/Certs Not Received	453
Missing/Lost MARS	455
MARS in Error	456
Missing/Lost Process Plan	457
Incorrect Entry Process Plan	458
Process Plan Not To Latest DCN	459
Q Codes (Other than Test Reports/Certs)	470
PLASTICS	
Improper Cure/Mix	475
Delamination	476
Discontinuities (Holes/Blisters/Voids)	477
Fiber Content	478
Flexural	479
Lap Shear	480
Exposed Circuitry	482
Incorrect Coating	484
Incorrect Bonding	485
FINISH	
Adhesion	501
Blistered/Flaking	502
Color	503
Cracked/Crazed	504
Incorrect	505
Pitted/Porous	506
No Samples	507
Rough/Irregular	508
Thickness	509
Scratched	510
IDENTIFICATION	
Incomplete	551
Incorrect	552
Smeared/Illegible	554
Missing	556
MATERIALS PROPERTIES	
Chemical	611
Metallurgical	612
Improper Mix/Cure	613

(c)

Figure 4–6.—(c) Continued.

FINAL DEFECT CODES (continued)

MATERIAL PROPERTIES (continued)	CODE
Heat Treat Material Response	615
Mechanical	617
Voids/Inclusions	618
Crack/Fracture	619
Voids/Porosity/Inclusions/Cracks	6013
Certification	6014
Incorrect Material	6015
Incorrect Dimensions	6016
Chemical Composition	6017
Moisture Content	6018
Pot Life	6019
Tensile Strength	6020
Yield	6021
Hardness	6022
Cure Hardness	6023
Peel Strength	6024
SOLDER	
Cold Joint	702
Hole in Solder	703
Fractured Joint	704
Pitted/Porous	705
Insufficient	706
Excess Flux	707
Excess Solder	708
Solder/Ball Splash	709
Dewetted Joint	710
Lifted Pads	712
Measling	713
Insulation in Solder	721
Potential Short	722
Bridging	723
Improper Tinning	724
Manual Soldering Discrepancy	2047
Machine Soldering Discrepancy	2048
Contaminated Joint	2049
Corrosion/Oxidation	2050
NO DEFECT	
NOTE: This code to be used for MARS closures where no discrepancies were identified.	755

MISCELLANEOUS	CODE
Test Error	860
Process/Steps Missed	861
No Evidence of Source Inspection	862
Procurement Error	863
Destructive Physical Analysis (DPA) Reject	864
Particle Impact Noise Detection (PIND) Reject	865
Defective Tool/Test Equipment	870
Incorrect Assembling	872
Integrity Seal Missing or Broken	875
Intermittent Operations	877
Launch Usage	879
Leakage	881
Out of Calibration	883
Shipped Short	885
Burst/Ruptured	886
Failed Due to Associated Equipment	887
Expanded (Normal Life)	888
Time/Operational, Temperature Sensitive, Expirations	895
Procedure Not Followed	1006
Proof Test	2026
Missing Operation	2027
Contamination	6025
All Trailer Problems	6026
Documentation/Certification Problems	6027
History Jacket Problems	6028
Directed Rejection Item	6029
WELDS	
Cracks	902
Porosity	903
Lack of Fusion	904
Burn Through	905
Lack of Penetration	906
Laps	907
Mismatch/Suck-In	910
Location	912
Build Up	921
Craters	922
Discoloration	923
Fill-Up	924
Length	925
Preparation	926
Profile	927
Undercut	928
Oxidation	2063
Metal Expulsion	2064

(c)

Figure 4–6.—(c) Continued.

FINAL DEFECT CODES (continued)

ELECTRONIC/COMPUTERS	CODE
Faulty Program or Disk	931
Unable to Load Program	932
Nonprogrammed Halt	933
Illegal Operation or Address	934
Computer Memory Error/Defect	935
Input/Output Pulse Distortion	936
Low Power Output	937
Frequency Out of Band, Unstable or Incorrect	938
Commercial Part Failure	941
Communication/Transmission Line Disturbance	943
Externally Induced Transient	945
COMPONENT LEAD WELDING	
(EMF only)	
Excessive Embedment	950
Cracks	951
Voids	952
Excessive Expulsion	953
Open/Missed Welds	954
Damaged Ribbon/Lead	955
Dimensions Incorrect	956
Sleeving Missing	957
Insufficient Heat/Cold Weld	958
Misrouted	959
Insufficient Fillet	960
Ribbon/Lead Misalignment	961
Ribbon/Lead Length Incorrect	962
ASSEMBLY/INSTALLATIONS	
Parts Mismatched	2019
Fastener Wrong or Damaged	2020
Damaged or Missing Seals	2021
Missing/Improperly Installed	2022
Parts Missing/Wrong/Damaged	2023
Improper Configuration	2024
RESISTANCE WELDING	
Resistance Weld Defects	2067

BONDING/COMPOSITIES/POTTING	CODE
Separation/Delamination	2013
Improper Cure	2014
Incorrect Lay-Up/Set-Up	2015
Test Specimen Failure Missing	2016
Voids/Blisters/Bridging/Pits	2017
Damage	2018
Mission Operation	2051
Damaged	2052
CONNECTORS-COMPONENTS/EEE	
Exceeds PDA	2041
Outside of SPC Boundaries	2042
X-ray to Applicable MIL Spec	2043
Improper Testing	2044
Noisy Output	2045
TOOLING FUNCTION	
Incomplete Hardware	6007
Burrs	6006
Inadequate Structure	6009
Discrepant Drill Bushing	6010
Improper Insert/Bushing	6012
FUSION WELDING	
Fusion Weld Defects	2066
TUBE/HOSE	
Damaged Flares/Lip Seals	2005
Incorrect Contours/Bends	2006
Wrong or Binding B-Nuts Sleeves	2007
Dimensional	2008
Expended	2009
Damaged Braid	2010
Cracks	2011
CHEMICAL/PLATING/LUBE/PAINT	
Contamination	2012

(c)

Figure 4–6.—(c) Concluded.

Launch and Flight Reliability

The flight test program demonstrated the launch and flight reliability of the vehicle. The ultimate flight program success ratio of 91 percent exceeded the overall availability-reliability goal by a comfortable margin.

Field Failure Problem

Twenty-six guidance sections failed the platform caging test portion of the launch station periodic tests (LSPT's). These failures resulted in a major alarm powerdown. An investigation was conducted.

Description of launch station periodic tests.—The system test requirements at the site include a requirement for station periodic tests upon completion of cell or vehicle installation and every 28 days thereafter. LSPT's check the overall system performance to evaluate the readiness of a cell. During an LSPT, the software initiates a test of the vehicle and ground equipment, data processing system, and radar interfaces. Any nonconformance during an LSPT is logged by the data processor and printed out, and the time from initiation of LSPT to failure is recorded. During an LSPT, the platform spin motor is spun up and held at speed for approximately 10 sec. After this, the system is returned to normal.

An LSPT consists of two phases:

(1) Spinup, a power-up phase to spin the gyros, align the platform, verify platform null, and check airborne power supply operation
(2) A detailed test of airborne electronics in the radio-frequency test phase

Initial failure occurrence.—Cell 3 on remote farm 1 (R1C3) experienced an LSPT failure (a major alarm powerdown) 5.936 sec after "prep order," the command to ready the vehicle for launch. The failure did not repeat during four subsequent LSPT's. RIC3 had previously passed three scheduled LSPT's before failure. A total of four cells on remote farms 1 and 2 had experienced similar failures. Two of the failures occurred at 5.360 sec (an inverter test to determine if ac power is available). Two occurred at 5.936 sec (caging test to determine if the platform is nulled to the reference position; see fig. 4–7).

Replacement of failed guidance and control sections (G&C) 28, 102, and 86 led to successful LSPT's. G&C 99, which failed only once during in-cell testing, was left on line. G&C's 28, 102, and 86 were returned to Martin Marietta, Orlando, for analysis of the pre failed condition.

Failure verifica nd troubleshooting.—The test plan that was generate mitted testing the failed G&C's in a horizontal marriage test and a G&C test to maximize the probability of duplicating the field failures. Test results confirmed site failures for both the caging null and the inverter null during a horizontal marriage test on G&C 102, a G&C level test

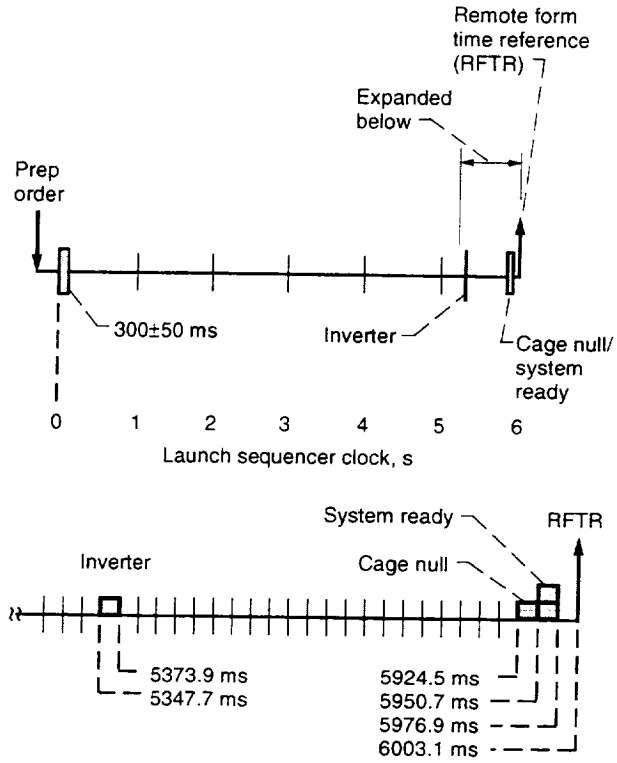


Figure 4–7.—System spinup tests. (Gate times are within ± 50 ms of that shown because of data processor tolerances.)

on G&C's 28 and 86, and an autopilot level test on G&C 102. G&C 102 failed caging null four times and inverter null once at horizontal marriage. An evaluation of the inverter null failure revealed that a high caging amplifier output caused the launch sequencer level detector to become offset during inverter monitoring, resulting in the major alarm even though the autopilot inverter voltage was normal. Launch sequencer offset may or may not occur with an uncaged platform depending on the amplitude of the caging amplifier output when the inverter voltage is monitored. Therefore, both the inverter null and the caging null LSPT failures at site were attributed to failure of the platform to cage.

An autopilot acceptance test tool was modified to permit monitoring of the platform spin motor voltage (800 Hz, 8 V, 3 ϕ) and the spin motor rotation detector (SMRD). During a spinup test on autopilot 69 (G&C 102), recordings indicated sustained caging oscillation. The SMRD showed no evidence of spin motor operation even though all autopilot voltages were correct, including the spin motor excitation voltage at the platform terminals. Further verification was obtained by listening for characteristic motor noises with a stethoscope.

G&C 86 failed the G&C level test because of caging null and inverter null alarms. Then, 3.5 sec into the third run, the caging loop stopped oscillating, but the platform did not cage in time to pass the test. The next run met all G&C test requirements.

It appeared obvious that the spin motor started spinning in the middle of the run.

G&C 28 failed one run of the G&C level test; however, it met all requirements in the autopilot level test. This means that the spin successfully met its acceptance test procedure requirements. A hesitation was noted during two of the seven spinup tests conducted. Platform 127 was heated to normal on the gyro test set. Its resistances were checked and found to meet specification requirements. No attempt was made to start platform 127's spin motor at platform level. Both units were hand-carried to the subcontractor for failure analysis. The subcontractor was familiar with the construction of the platform and had the facilities to disassemble the platform without disturbing the apparently intermittent failure condition.

Verification test conclusions.—Verification tests isolated the site LSPT failures to a failure of the platform spin motor to spin up, thereby causing major alarms at the inverter null or caging null gate. During testing, three of the first four failed platforms caged upon repeated application of voltage. Once the platform caged, the platform, autopilot, and G&C met all system test requirements. On the basis of these results, it was decided to repeat LSPT's up to 10 times after a site failure before removing the G&C. If the LSPT's were successful, the G&C would be left on line.

Measurements at platform level indicated the problem was internal to the platform and that all resistances and the platform temperature were correct. Subcontractor representatives reviewed the test results and concurred that the problem was internal to the platform.

Mechanical Tests

The spin motor breakaway torque was measured with a gram gage on platform 127 and was found to be normal (750 dyne cm). Dynamometer tests were performed on both platforms. The dynamometer is an instrument that measures rotation torque by slowly rotating the rotor of the spin motor while recording the stator rotational torque. The dynamometer is used during initial builds to establish the spin motor bearing preload (torque). The spin motor generates approximately 4000 dyne cm of starting torque with normal excitation voltage; 800 dyne cm of this torque is used to overcome the inertia and frictional torque of the motor.

Platform 140 was tested on the dynamometer and produced the torque peaks of 3400 and 3100 dyne cm shown in figure 4–8. The torque peaks were three revolutions apart. This is four times the normal running torque level for a new spin motor and about four times the torque level for this spin motor for the rest of its run. The torque increase lasted for about one-half a revolution and repeated within three revolutions. The spin motor bearings were cleaned and reassembled. Two large torque spikes of approximately 3000 dyne cm were observed on the first revolution. A 2200-dyne cm torque hump, one revolution in duration, was centered at the beginning of the second revolution. From these results, it was concluded that something in the spin motor bearing was causing an abnormal frictional load there. This result isolated the problem to the spin motor bearing area and eliminated the motor electrical characteristics as a contributor.

Runup and Rundown Tests

A series of tests were performed on spin motors 96 and 140 to determine the effect of motor running time on spin motor start and running torque. Figure 4–9 shows the change in rundown time with a change in motor run time.

Summary of Case Study

Field problem cause.—The 26 LSPT failures at the site were caused by the failure of the G&C platform spin motors to spin up within 6 sec after the command to ready the vehicle for launch. It was determined that the spin motors did not start with the normal application of voltage. A polymer film had formed on the bearing surfaces during testing at 175 °F and caused the balls to stick to the outer race. This film was identified as one from the alkyl phenol and alkyl benzene families, and its source was determined to be uncured resins from the bearing retainer.

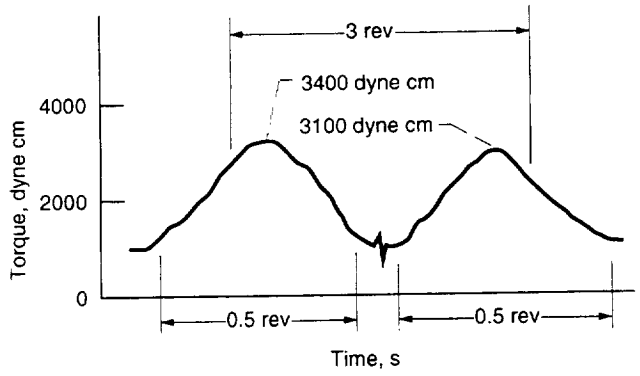


Figure 4–8.—Platform dynamometer torque test.

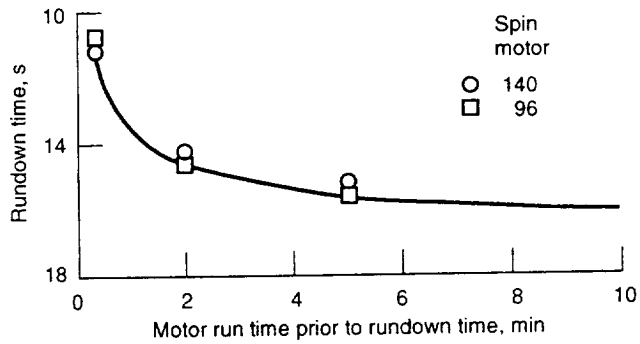


Figure 4–9.—Rundown time versus motor run time.

Polymer film.—A film approximately 900 Å thick had formed on the metal surfaces of the bearings of failed spin motors. The amount of material generated was ~10^{-7} g/ball. To put this number in proper perspective, 2×10^{-4} g of oil is put on the bearing race during initial build, and 2×10^{-3} g of oil is impregnated in the bearing retainer.

Alkyl phenol/alkyl benzene is a generic identification of a family of organic compounds. Further analysis identifies the major compounds in the family as phenol and methyl phenol (alkyl phenols) and toluene, xylene, and benzene (alkyl benzenes). A phenolic polymeric film would have the gummy, adhesive, and insolubility properties detected in the analysis. There is little doubt that the gummy film detected was a phenol-based material.

Source of phenol.—Phenols are used in three areas of the spin motor. A phenolic adhesive bonds the stator laminations together and bonds the hysteresis ring to the rotor. The bonding processes adequately cure the phenol to the point where uncured phenols would not be present. Also, the stator laminations are coated with epoxy after bonding. The remaining source is the paper phenolic retainer, which serves as a spacer and a lubrication source for the spin motor bearings. Mass spectral analysis of the retainers yielded spectra essentially identical to the spectrum of the coating on the failed bearings. The conclusion of this analysis is that the source of the phenolic is uncured phenolic resins or resin compounds in the retainer.

Retainer processing.—The retainer material is manufactured to military specifications by a vendor and is screened to tighter vehicle requirements for specific gravity. There is no specific requirement concerning uncured resins in the retainer material. The vendor estimated an upper limit of 1 percent of uncured resin in the retainer raw material. One percent would provide 3×10^{-5} g of uncured resins, more than sufficient to cause the spin motor problem.

The finished retainer material is cleaned by an extraction process with benzene or hexane. This process does not remove a significant amount of uncured resins. Therefore, if uncured resins survive the vendor processing, they will remain in the uncured state in the installed retainers.

Mechanism of film formation.—It is theorized that the uncured resins are transferred from the retainer to the bearing surfaces through the natural lubricating process of the retainer. Running the spin motors generates centrifugal forces that sling the excess oil off the rotating surfaces, leaving a thin film of oil. The force of gravity during subsequent storage of the motor causes the already thin film to become thinner on the top surfaces and thicker on the lower surfaces. This redistribution process involves only the oil and leaves more viscous contaminants in place. Subsequent running of the motor will cause replacement of oil on the oil-free surfaces. The source of the replacement oil is the retainer capillaries. This replacement process will cause the oil to bring any uncured phenolics to the surface of the retainer. The metal surfaces will then become lubricated with oil containing a small percentage of uncured resins. Subsequent storage cycles and running will continue this redistribution process, steadily increasing the phenolic concentration. Exposure to a temperature of 175 °F and extended operational maintenance gradually cure these phenolics in two stages. Initially, a highly viscous gummy residue is formed; finally, a hard, insoluble polymer film is formed on the metal surfaces. The film forms a bond between the balls and the races. The coating builds up to the point where the spin motor torque cannot overcome the bond at the initial power application.

Extent of problem.—An analysis of failed and unfailed field units proved that not all platforms are susceptible to this failure. Obviously, a high percentage are susceptible, since 26 failures have been experienced. It is likely that many unfailed platforms contain some small percentage of uncured resins.

The significantly higher failure rate in the units with higher serial numbers points to a process (or common) failure mode. All evidence points to lot-to-lot variations in the amount of uncured resins present in the retainer raw material. Traceability from retainer lot to individual platform spin motor was not possible in this case, but such records should be available. The 26 units that have failed and the failure rate at the 14-day interval bound the total platform failure rate. The number of spares available is adequate to meet system life and reliability requirements.

Site reliability.—The site system reliability goal allows approximately two G&C failures per month for any cause. Analysis of test data indicates the goal can be achieved at either a 7-day test interval (0.8 failure/month) or a 14-day test interval (1.5 failures/month). It cannot be achieved at a 21-day interval (7.7 failures/month) or a 28-day interval (8.6 failures/month). Even though at least 74 percent of the site failures were restarted, a limited number of spare G&C's are available.

Tests at the site revealed that most failed spin motors can be restarted within 10 power applications and once started will perform properly. The site procedure was revised to leave any failed G&C's that restart within 10 attempts on line. Platforms that did not start within 10 attempts were returned to the contractor and were restarted by repetitive application of overvoltage or reverse voltage up to the motor saturation limit. These data support the conclusion that the failure mode was the formation of a film bond on the race and that increasing the inverter output voltage to the motor saturation limit would not eliminate the problem.

Current site operating procedures provide a 14-day LSPT interval with a 10-min run time. This enables the G&C failure rate to meet system reliability goals. The vehicle site is currently being deactivated. If reactivation should be required, the repair of all defective or support platforms should be included as part of that effort.

Concluding Remarks

Now that you have completed chapter 4, several concepts should be clear.

(1) The failure rate of complex equipment is usually considered to be a constant.
(2) Most failures are random, with repetitive failures representing a small portion of unreliability.
(3) The rate at which failures occur depends upon
    (a) The acceptance criteria, which determine how effectively potential failures are detected
    (b) All applied stresses, including electrical, mechanical, and environmental. (As these stresses increase, the failure rate usually increases.)
(4) Published failure rate data represent the potential failures expected of a part. The rate at which these failures are observed depends on the applied electrical stresses (the stress ratio) and the mechanical stresses (the K_A factor).
(5) In general, failure rate predictions are best applied on a relative basis.
(6) Failure rate data can be used to provide reliability criteria to be traded off with other performance parameters or physical configurations.
(7) The reliability of a device can be increased only if the device's failure mechanisms and their activation causes are understood.

In addition, you should be able to use failure rate data to predict the failure rate expected of a design, and consequently, to calculate the first term, P_c, of inherent reliability. Finally, you should be able to allocate failure rate requirements to parts after having been given a reliability goal for a system or the elements of a system.

References

4–1. Electronic Reliability Design Handbook. MIL–HDBK–338, Oct. 1988.
4–2. Reliability Prediction of Electronic Equipment. MIL–HDBK–217E, Jan. 1990.
4–3. Taylor, J.R.: Handbook of Piece Part Failure Rates. Martin Marietta Corp., June 1970. (Avail. NTIS, AD–B007168L.)
4–4. Bloomquist, C.; and Graham, W.: Analysis of Spacecraft On-Orbit Anomalies and Lifetime. (PRC R–3579, PRC Systems Sciences Co.; NASA Contract NAS5–27279), NASA CR–170565, 1983.
4–5. Government-Industry Data Exchange Program (GIDEP), Reliability Maintainability (R-M) Analyzed Data Summaries, vol. 7, Oct. 1985.
4–6. Maintainability Demonstration. MIL–STD–471A, Jan. 10, 1975.
4–7. Reliability Modeling and Prediction. MIL–STD–756B, Aug. 1982.
4–8. Lloyd, D.K.; and Lipow, M.: Reliability: Management, Methods, and Mathematics. Prentice-Hall, 1962.
4–9. Landers, R.R.: Reliability and Product Assurance. Prentice-Hall, 1963.
4–10. Anstead, R.I.; and Goldberg, E.: Failure Analysis of Electronic Parts Laboratory Methods. NASA SP–6508, 1975.
4–11. Devaney, J.R.; Hill, G.L; and Seippel, R.G.: Failure Analysis Mechanisms, Techniques and Photo Atlas. Failure Recognition and Training Service Inc., Monrovia, CA, 1985.
4–12. Smith, G., et al.: How to Avoid Metallic Growth Problems on Electronic Hardware. IPC-TR-476, Institute of Printed Circuits, Sept. 1977.

Reliability Training[1]

1a. Using the failure rate data in table 4–4, calculate the flight failure rate for a launch vehicle electronic subsystem consisting of the following parts (assume $K_A = 1000$):

Component	Number of parts, N
Resistor, G657109/10	5
Resistor, variable, 11176416	1
Capacitor, G657113	3
Diode, G6557092	3
Transistor, 11176056	4
Integrated circuit, analog, 11177686	1

A. 195 failures per 10^9 hr B. 195 000 failures per 10^9 hr C. 195 000 failures per 10^6 hr

1b. Assume the flight failure rate for this circuit is 500 000 failures per 10^9 hr. Calculate the reliability of the circuit for a 0.01-hr flight.

A. 0.9995 B. 0.99995 C. 0.999995

2. The a posteriori flight failure rate of a launch vehicle is 440 000 failures per 10^9 hr.

a. If the storage failure rate is 0.3 of the operating rate, how long can the vehicle be stored with a 90.4 percent probability of no failures?

A. 30 days B. 40 days C. 50 days

b. After 1450 hr (2 months) in storage the vehicle is removed and checked out electronically. If the vehicle passes its electronic checkout and the checkout equipment can detect only 80 percent of the possible failures, what is the probability that the vehicle is good? (Ignore test time.)

A. 0.962 B. 0.858 C. 0.946

3. A subassembly in a piece of ground support equipment has a reliability requirement of 0.995. Preliminary estimates suggest that the subassembly will contain 300 parts and will operate for 200 hr. What is the average part failure rate required to meet the reliability goal?

A. 25×10^{-6} B. $16\,667\times10^{-9}$ C. 83×10^{-9}

4. A piece of ground support equipment has a reliability goal of 0.9936. It contains four subassemblies of approximately equal risk.

a. What is the allocated reliability goal of each of the four subassemblies?

A. 0.99984 B. 0.9984 C. 0.9884

b. Allocating further into subassembly 1, assume the goal is 0.998. Solve for the average part failure rate given the following:

Estimated parts count: 100
Estimated operating time: 10 hr

A. $20\,000\times10^{-9}$ B. 2000×10^{-9} C. 200×10^{-9}

[1]Answers are given at the end of this manual.

TABLE 4–4.—SELECTED LISTING—APPROVED ELECTRONIC FAILURE RATES FOR LAUNCH VEHICLE APPLICATION[a]

Part number	Part	Operating mode[b]	Nonoperating mode[b]
		Failure rate, failures/10^9 hr	
Integrated circuits			
11177680/81/82/83/84/85	Digital	10	3
11177686	Analog	30	10
Transistors			
6557155	Double switch	10	3
6557318/19	Medium-power switch	20	↓
6557046	PNP type transistor	↓	↓
11176911	Medium-power switch	↓	↓
11176056	High-speed switch	↓	↓
11177685	Field-effect transistor	↓	↓
6310038	2N5201	10	↓
6557072	2N918 (unmatched)	50	5
Diodes			
6557061	Rectifier and logic (5 V)	20	3
6557092	Rectifier and logic (30 V)	5	↓
6557123	Rectifier and logic (50 V)	↓	↓
6557125	Rectifier and logic (600 V)	↓	↓
11176912	Rectifier and logic (400 V)	↓	↓
Resistors			
6557018	2.5-W wirewound	2	1
6557015	1/8-W wirewound	3	2
6557016/17	1- and 2-W wirewound	2	5
6557030	1/10-W fixed film	1	5
6557031	6-W wirewound	5	5
6557109/10	1/4-W fixed composition	1	2
6557329	1/8-W fixed film	1	3
11176416	1-W variable metal film	50	10.3
Capacitors			
G657020/21/22	Fixed glass	0.1	0.1
G657113/173	Fixed ceramic	5	1
G657114	Fixed ceramic	10	1
G657119/120	Solid tantalum	2	1
G657202	Precision, fixed ceramic	50	3
Relays			
11176326/453	DPDT armature	100	20
Transformers (RF)			
11301034/35/43/49		10	5
11301064		1	5
RF coil			
G657140/41		3	2
G657178/81		10	2
RF filter			
G657189		50	5

[a]Current failure rate data are available from refs. 4–1 and 4–4.
[b]Applies to all slash numbers of parts shown (worst case shown).

Chapter 5
Applying Probability Density Functions

The inherent reliability of equipment is defined in chapter 3 as

$$R_i = e^{-\lambda t} P_t P_w$$

where

R_i	probability of no failures
$e^{-\lambda t}$	probability of no catastrophic part failures
P_t	probability of no tolerance failures
$2P_w$	probability of no wearout failures

Before discussing the P_t and P_w terms in the next chapter, it is necessary to understand probability density functions and cumulative probability functions. These concepts form another part of probability theory not discussed in chapter 2. First, in this chapter, the theory of density and cumulative functions is discussed in general; then the normal, or Gaussian, distribution is discussed in detail. This normal distribution is used extensively later in the manual.

Probability Density Functions

If a chance variable x can take on values only within some interval, say between a and b, the probability density function $p(x)$ of that variable has the property that (ref. 5–1)

$$\int_a^b p(x)\,dx = 1$$

In other words, the area under the curve $p(x)$ is equal to unity. This is shown in figure 5–1.

In the language of probability, the probability of x being within the interval (a,b) is given by

$$P(a \le x \le b) = \int_a^b p(x)\,dx = 1$$

In other words the probability that x lies between a and b is 1. This should be clear, since x can take only values between a and b.

In a similar fashion, we can find the probability of x being within any other interval, say between c and d, from

$$P(c \le x \le d) = \int_c^d p(x)\,dx$$

which is shown in figure 5–2.

Example 1: Suppose we were to perform an experiment in which we measured the height of oak trees in a 1-acre woods. The result, if our measuring accuracy is ±5 ft, might look like the histogram shown in figure 5–3.

The value at the top of each histogram cell (or bar) indicates the number of trees observed to have a height within the boundaries of that cell. For example, 19 trees had a height between 0 and 10 feet, 17 trees had a height between 10 and 20 feet, and so on. The figure shows that 100 trees were observed.

Now let us calculate values for the ordinate of the histogram so that the area under the histogram equals unity. Then, we will establish a probability density function for the tree heights. Since we observed 100 trees, it should be apparent that if the calculated ordinate of a cell times the width of the cell (the cell area) yields the percentage of 100 trees in that cell, the sum of the percentage in all cells will have to equal 100 percent. Of, if the percentages are expressed as decimal fractions, their sum will equal 1, which will be the total area under the histogram. Therefore,

$$\text{Ordinate of cell} = \frac{\text{Percent of trees in cell}}{\text{Width of cell}}$$

For the cell 0 to 10 feet, which has 19 percent of the trees in it,

$$\text{Ordinate of cell} = \frac{19}{100} \times \frac{1}{10} = 0.019$$

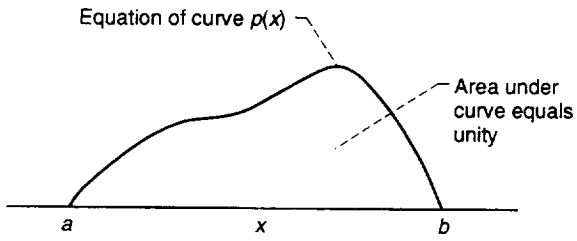


Figure 5–1.—Probability density function curve.

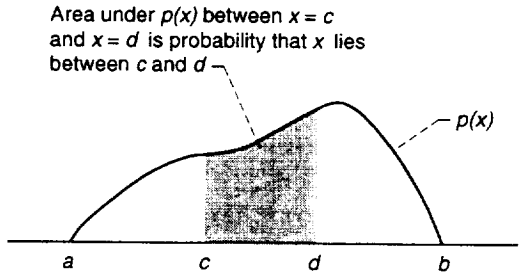


Figure 5–2.—Application of probability density function.

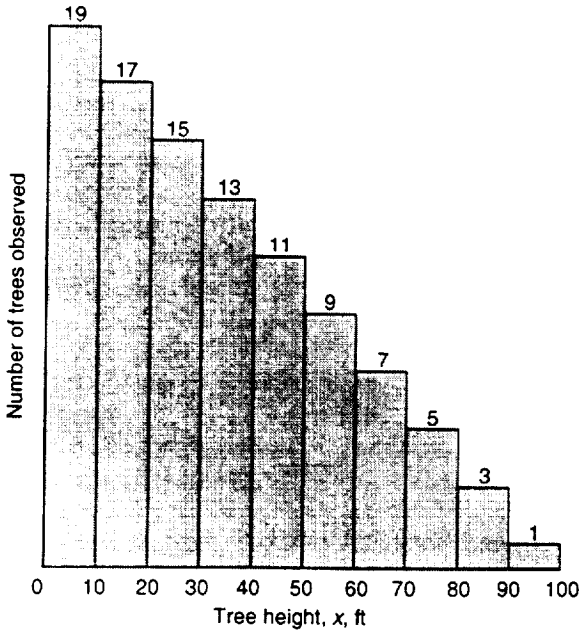


Figure 5–3.—Height of trees observed in 1-acre woods.

As a check, we can see that

Ordinate of cell = 0.019 × Cell width (10) = 0.19, or 19 percent

In a similar fashion, the ordinates for the other cells can be calculated and are shown in table 5–1 and figure 5–4.

The next step (fig. 5–4) is to draw a line through the midpoint of the cells. The equation for this line is called the probability density function $p(x)$ and has the form

$$p(x) = -0.0002x + 0.02$$

The area under the curve is (ref. 5–2)

$$\text{Area} = \int_0^{100} p(x)\,dx \int_0^{100} (-0.0002x + 0.02)\,dx$$

$$= -\frac{x^2}{10^4} + 0.02x\Big|_0^{100} = -\frac{(100)^2}{10^4} + 0.02(100)$$

$$= -\frac{10^4}{10^4} + 2 = -1 + 2 = 1$$

This agrees with our requirement that the area under a probability density function equal unity.

TABLE 5–1.—CALCULATION OF CELL ORDINATES FOR TREE DATA

Cell	Ordinate	Area, cell width times cell ordinate
0–10	$\frac{19}{100 \times 10} = 0.019$	0.19
10–20	$\frac{17}{10^3} = 0.017$	.17
20–30	$\frac{15}{10^3} = 0.015$	.15
30–40	$\frac{13}{10^3} = 0.013$	.13
40–50	$\frac{11}{10^3} = 0.011$	.11
50–60	$\frac{9}{10^3} = 0.009$	.09
60–70	$\frac{7}{10^3} = 0.007$	.07
70–80	$\frac{5}{10^3} = 0.005$	.05
80–90	$\frac{3}{10^3} = 0.003$	.03
90–100	$\frac{1}{10^3} = 0.001$	.01
	Total area	1.00

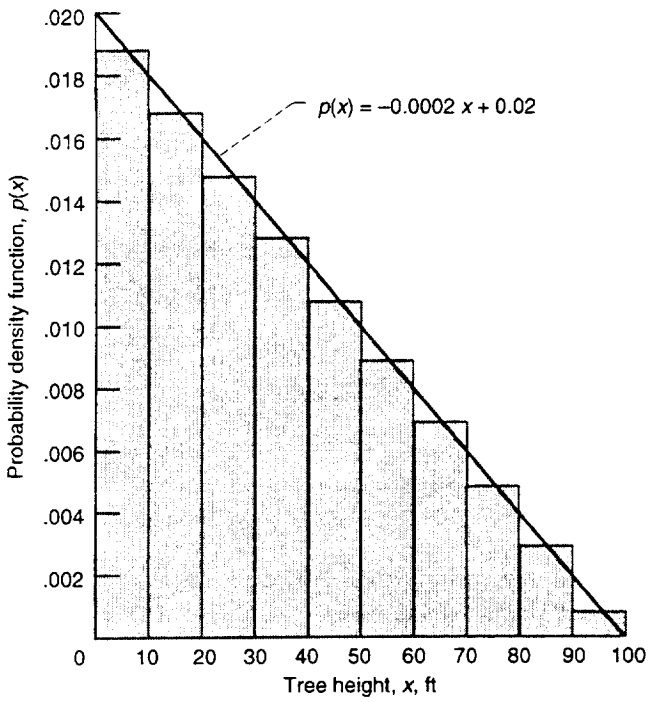


Figure 5–4.—Probability density function for tree heights.

Application of Density Functions

Now let us see how we can apply the density function to the tree data. To find the percentage of trees between 60 and 80 feet high, solve for

$$P(60 \le x \le 80) = \int_{60}^{80} p(x)\,\mathrm{d}x = \int_{60}^{80} (-0.0002x + 0.02)\,\mathrm{d}x$$

$$= -\frac{x^2}{10^4} + 0.02x\Big|_{60}^{80} = -\frac{1}{10^4}\left(80^2 - 60^2\right) + 0.02(80 - 60)$$

$$= -\frac{1}{10^4}(2800) + 0.4 = -0.28 + 0.4$$

$$= 0.12, \text{ or } 12 \text{ percent}$$

Figure 5–3 shows that this answer is correct, since 12/100 trees were observed to have a height between 60 and 80 feet.

Another way to look at this example is that there is only a 12-percent chance that a tree picked at random from the 1-acre area would have a height between 60 and 80 feet. In a similar fashion, we can calculate the probability that a tree would have any range of heights within the boundary of 0 to 100 feet.

In the tree example, we were able to measure the trees in a particular part of the woods and to obtain a height density function for those trees. But what do we do if we are interested in a different area of woods and for some reason we are not able to go out and measure the trees? We would probably assume that the acre we measured was representative of all other acres

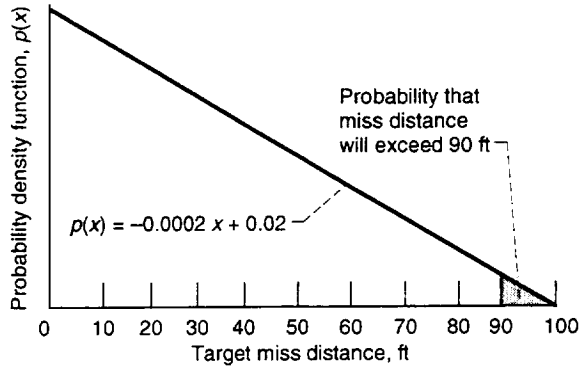


Figure 5–5.—Probability density function for missile target miss distance.

in the same woods. If we accept this assumption, we could then use our experience (the established density function) to predict the distribution of tree heights in an unmeasured acre. And this is exactly what is done in industry.

As you can see, if we know what the density functions are for such things as failure rates, operating temperatures, and missile accuracy, it is easy to determine the probability of meeting a failure rate requirement for equipment (such as a missile) specified to operate in some temperature range with a required accuracy.

Example 2: Suppose that a missile has a maximum target miss distance requirement of 90 feet and that after several hundred firings, the probability density function for miss distance is

$$p(x) = -0.0002x + 0.02 \quad \text{where } 0 \le x \le 100$$

which is the same as the $p(x)$ for the tree example and is shown in figure 5–5.

To predict the probability that the next missile fired will miss the target by more than 90 feet, solve for

$$P(90 \le x \le 100) = \int_{90}^{100} (-0.0002x + 0.02)\,\mathrm{d}x$$

$$= -\frac{x^2}{10^4} + 0.02x\Big|_{90}^{100}$$

$$= -\frac{1}{10^4}\left(100^2 - 90^2\right) + 0.02(100 - 90)$$

$$= -\frac{1900}{10^4} + 0.02(10)$$

$$= -0.19 + 0.2 = 0.01, \text{ or } 1 \text{ percent}$$

In other words, there is a 99-percent chance that the missile will hit within 90 feet of the target and a 1-percent chance that it will not. This is shown as the shaded area under the density function in figure 5–5.

Cumulative Probability Distribution

Another practical tool in probability calculation is the cumulative probability distribution $F(x)$ from reference 5–3. An $F(x)$ curve for the tree example in the preceding section is shown in figure 5–6. The curve represents the cumulative area under the probability density function $p(x)$. The ordinates of the curve were calculated as shown in table 5–2.

The cumulative curve can be used to solve the same problems that the density curve was used to solve.

Example 3: Referring again to example 1, suppose that we want to know the probability that a particular tree selected at random from the woods will have a height between 30 and 50 feet.

Solution 3A: Using the density function for tree height,

$$P(30 \le x \le 50) = \int_{30}^{50} (-0.0002x + 0.02)\,dx$$

$$= -\frac{x^2}{10^4} + 0.02x \Big|_{30}^{50}$$

$$= -\frac{1600}{10^4} + 0.40$$

$$= -0.16 + 0.40 = 0.24, \text{ or } 24 \text{ percent}$$

Solution 3B: Using the cumulative curve shown in figure 5–5,

$$P(30 \le x \le 50) = F(50) - F(30) = 0.75 - 0.51$$

$$= 0.24, \text{ or } 24 \text{ percent}$$

which agrees with solution 3A.

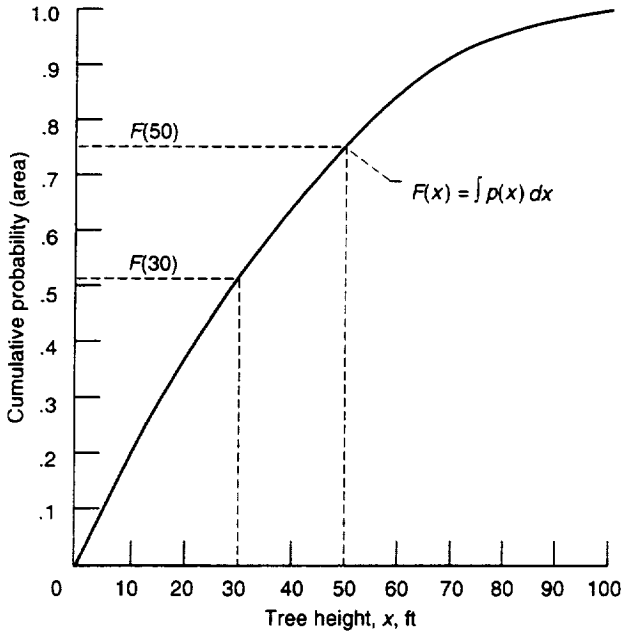


Figure 5–6.—Cumulative probability function for tree heights.

TABLE 5–2.—ORDINATES FOR CUMULATIVE DISTRIBUTION OF TREE DATA

Tree height, ft	Area under $p(x)$ curve	Ordinate of $p(x)$ curve (cumulative area)
0–10	0.19	0.19
10–20	.17	.36
20–30	.15	.51
30–40	.13	.64
40–50	.11	.75
50–60	.09	.84
60–70	.07	.91
70–80	.05	.96
80–90	.03	.99
90–100	.01	1.00

Note that in working out solution 3A, the next-to-last step $(0.75 - 0.51)$ is the same as the next-to-last step of solution 3B. The reason for this is that the equation of the cumulative probability function $F(x)$ is found from

$$F(x) = \int p(x)\,dx$$

and

$$\int_a^b p(x)\,dx = F(b) - F(a)$$

For the tree example

$$F(x) = \int (-0.0002x + 0.02)\,dx = -\frac{x^2}{10^4} + 0.02x$$

Consequently, we can find the probability of a variable x being within some interval by using the cumulative function $F(x)$ even though the cumulative graph is not available.

Example 4: What is the probability that a tree selected at random will have a height less than 20 feet?

Solution 4:

$$P(0 \le x \le 20) = \int_0^{20} p(x)\,dx = F(20) - F(0)$$

$$= -\frac{x^2}{10^4} + 0.02x \Big|_0^{20}$$

$$= \left[-\frac{20^2}{10^4} + 0.02(20)\right] - 0$$

$$= -0.04 + 0.4 = 0.36, \text{ or } 36 \text{ percent}$$

which agrees with a graphical solution.

Some general rules for the use of the cumulative function $F(x)$ are

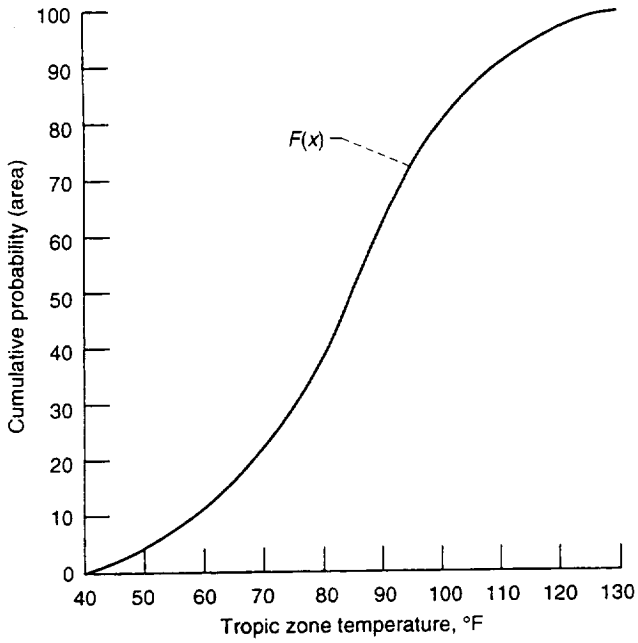


Figure 5–7.—Cumulative distribution of tropic zone temperatures.

$$(1)\ P(x \le a) = F(a)$$

$$(2)\ P(x \ge a) = 1 - F(a)$$

$$(3)\ P(a \le x \le b) = F(b) - F(a)$$

Example 5: Suppose that we would like to know the probability of equipment seeing tropic zone temperatures above 120 °F during operation because at or above 120 °F, we have to add a costly air-conditioning system to cool the equipment. If we could obtain the temperature data, we might find that the cumulative distribution for tropic zone temperatures would be that shown in figure 5–7.

Solution 5: From the curve, the probability of observing a temperature at or above 120 °F is given by

$$P(\text{temp} \ge 120\,^\circ\text{F}) = 1 - F(120\,^\circ\text{F}) = 1 - 0.97$$
$$= 0.3, \text{ or } 3 \text{ percent}$$

With only a 3-percent chance of temperatures above 120 °F, we probably would decide against air conditioning (all other parameters, such as failure rate, being equal).

Normal Distribution

One of the most frequently used density functions in reliability engineering is the normal, or Gaussian, distribution. A more descriptive term, however, is the normal curve of error because it represents the distribution of errors observed from repeated

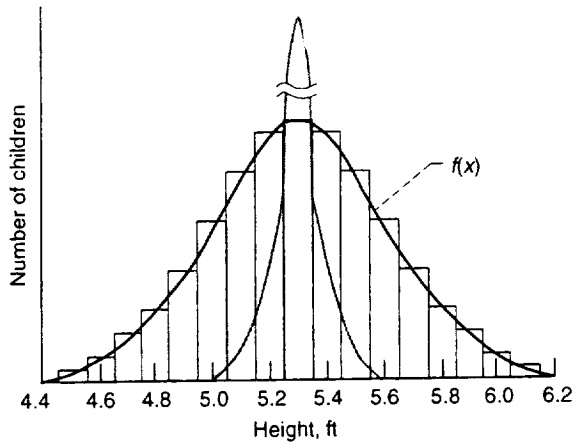


Figure 5–8.—Histogram and density function for heights of children.

measurements of an object or some physical phenomenon (ref. 5–4).

Example 6: Assume that we need to measure the heights of eighth-grade children. A histogram of the children's heights would resemble the curve in figure 5–8. If, as in our tree example, we calculate an ordinate for the histogram so that the area under the histogram equals unity and then connect the midpoints of each cell, we obtain a smooth curve as shown in figure 5–8. This curve represents the density function for the heights of the children. Such a curve (sometimes called a bell curve) is the shape of the normal distribution. We say that the children's heights are distributed normally.

Normal Density Function

The equation for the density function $p(x)$ of the normal distribution is

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\bar{x})^2/2\sigma^2}$$

This curve is shown in figure 5–9. The function $p(x)$ has two parameters. The first is the mean $\bar{x}$ calculated from

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \text{ where}$$

where

n total number of measurements or observations
x_i value of i^{th} measurement

The mean, therefore, is the arithmetic average of the measurements. From example 6, we would add all the heights observed and then divide by the number of children measured to obtain a mean or average height. The mean of all the children's heights from the data in figure 5–8 is 5.3 ft.

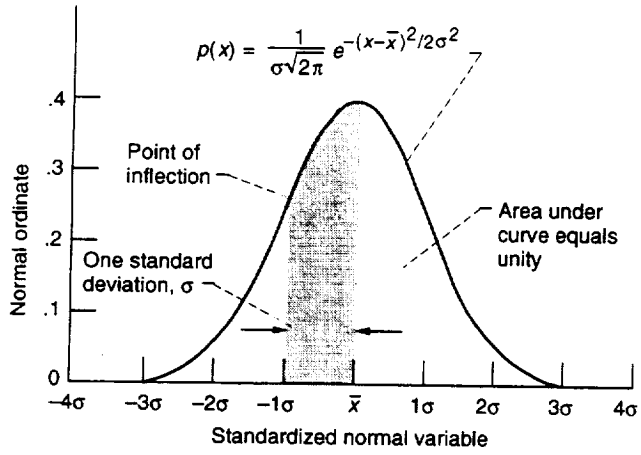


Figure 5–9.—Normal density function.

The second parameter of $p(x)$ is the standard deviation σ calculated from

$$\sigma = \left[\sum_{i=1}^{n} \frac{(x_i - \bar{x})^2}{n-1} \right]^{1/2}$$

where

$\bar{x}$ mean of measurements

x_i value of i^{th} measurement

n total number of measurements

Note that $n - 1$ is used in the equation to give an unbiased sampling distribution. In the general definition of σ, n instead of $n - 1$ would be used.

The standard deviation is the square root of the variance, which is denoted by σ^2. The magnitude of the variance, as well as the standard deviation, indicates how far all the measurements deviate from the mean. The standard deviation of the children's height data, for example, is approximately 0.3 ft. If the range of heights observed had been from 5 to 5.6 ft, the standard deviation would have been approximately 0.1 ft; with this standard deviation, the distribution would look squeezed together, as shown by the dashed curve in figure 5–8. However, the area under the dashed curve would still equal the area under the solid curve.

Properties of Normal Distribution

The normal density function is a continuous distribution from $-\infty$ to ∞. It is symmetrical about the mean and has an area equal to unity as required for probability density functions. For the normal distribution, the standard deviation is the distance on the abscissa from the mean $\bar{x}$ to the intercept on the abscissa of a line drawn perpendicular to the abscissa through the point of inflection on the curve. This is shown in figure 5–9. It is also shown that equal increments of the standard deviation can be laid out to the left (−) and the right (+) of the mean $\bar{x}$.

TABLE 5–3.—AREAS BETWEEN $-z$ AND z

z	Area under curve	Probability $-z \le x \le z$
1	0.683	$P(-1\sigma \le x \le 1\sigma)$
2	.9545	$P(-2\sigma \le x \le 2\sigma)$
3	.9973	$P(-3\sigma \le x \le 3\sigma)$
4	.999937	$P(-4\sigma \le x \le 4\sigma)$
5	.999999426	$P(-5\sigma \le x \le 5\sigma)$
6	.99999999803	$P(-6\sigma \le x \le 6\sigma)$
7	.999999999992	$P(-7\sigma \le x \le 7\sigma)$

As you will recall, in determining probabilities from a density function, we need to calculate the area under the curve $p(x)$. When using the normal density function, it is common practice to relate areas to the standard deviation. In general, for the area under the curve between the values of z and $-z$, standard deviations can be found from

$$p(-z \le x \le z) = \text{Area} = \int_{-z}^{z} \frac{1}{\sigma\sqrt{2\pi}} e^{-1/2\left(z^2\right)} dz$$

The areas for various values of z are shown in table 5–3. This table shows that the area under the normal curve between 1σ and -1σ is 0.683, or 68.3 percent; the area under the normal curve between 2σ and -2σ is 0.9545, or 95.45 percent, and so forth.

Example 7: The term "3σ limit" refers to the area under the normal curve between 3σ and -3σ, which is 0.9973, or 99.73 percent, as shown in table 5–3. Therefore, if a power supply output is defined as 28±3 V and the ±3 V represents a 3σ limit, 99.73 percent of all such power supplies will have an output between 25 and 31 V. The percentage of supplies having an output greater than 31 V and less than 25 V will be $1 - 0.9973 = 0.0027$, or 0.27 percent, as shown in figure 5–10.

Up to now we have been working with areas under the normal density function between integers of σ, that is, 1, 2, 3, and so on. In practice, however, we are usually interested in the area between decimal fractions of σ, those being 1.1, 2.3, et cetera. We have also been using z to represent the number of standard deviations that a particular limit value is from the mean. For instance, in the power supply example, 25 V was given as being three standard deviations from the mean of 28 V. It is better when working in decimal fractions of σ to let $z = (x - \bar{x}) / \sigma$ where $x - \bar{x}$ is the distance from the mean $\bar{x}$ to the limit value and σ is the standard deviation. Going back to the supply example, our lower limit was 25 V, which was 3 V from the mean of 28 V, and the standard deviation was 1 V; therefore, $z = (25 - 28)/1 = -3$.

TABLE 5-4.—AREAS IN TWO TAILS OF NORMAL CURVE AT SELECTED VALUES OF z
[From reference 5-1.]

z	0	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
0	1.0000	0.9920	0.9840	0.9761	0.9681	0.9601	0.9522	0.9442	0.9362	0.9283
.1	.9203	.9124	.9045	.8966	.8887	.8808	.8729	.8650	.8572	.8493
.2	.8415	.8337	.8259	.8181	.8103	.8026	.7949	.7872	.7795	.7718
.3	.7642	.7566	.7490	.7414	.7339	.7263	.7188	.7114	.7039	.6965
.4	.6892	.6818	.6745	.6672	.6599	.6527	.6455	.6384	.6312	.6241
.5	.6171	.6101	.6031	.5961	.5892	.5823	.5755	.5687	.5619	.5552
.6	.5485	.5419	.5353	.5287	.5222	.5157	.5093	.5029	.4965	.4902
.7	.4839	.4777	.4715	.4654	.4593	.4533	.4473	.4413	.4354	.4295
.8	.4237	.4179	.4122	.4065	.4009	.3953	.3898	.3843	.3789	.3735
.9	.3681	.3628	.3576	.3524	.3472	.3421	.3371	.3320	.3271	.3222
1.0	.3173	.3125	.3077	.3030	.2983	.2937	.2891	.2846	.2801	.2757
1.1	.2713	.2670	.2627	.2585	.2543	.2501	.2460	.2420	.2380	.2340
1.2	.2301	.2263	.2225	.2187	.2150	.2113	.2077	.2041	.2005	.1971
1.3	.1936	.1902	.1868	.1835	.1802	.1770	.1738	.1707	.1676	.1645
1.4	.1615	.1585	.1556	.1527	.1499	.1471	.1443	.1416	.1389	.1362
1.5	.1336	.1310	.1285	.1260	.1236	.1211	.1188	.1164	.1141	.1118
1.6	.1096	.1074	.1052	.1031	.1010	.0989	.0969	.0949	.0930	.0910
1.7	.0891	.0873	.0854	.0836	.0819	.0801	.0784	.0767	.0751	.0735
1.8	.0719	.0703	.0688	.0672	.0658	.0643	.0629	.0615	.0601	.0588
1.9	.0574	.0561	.0549	.0536	.0524	.0512	.0500	.0488	.0477	.0466
2.0	.0455	.0444	.0434	.0424	.0414	.0404	.0394	.0385	.0375	.0366
2.1	.0357	.0349	.0340	.0332	.0324	.0316	.0308	.0300	.0293	.0285
2.2	.0278	.0271	.0264	.0257	.0251	.0244	.0233	.0232	.0226	.0220
2.3	.0214	.0209	.0203	.0198	.0193	.0188	.0183	.0178	.0173	.0168
2.4	.0164	.0160	.0155	.0151	.0147	.0143	.0139	.0135	.0131	.0128
2.5	.0124	.0121	.0117	.0114	.0111	.0108	.0105	.0102	.00988	.00960
2.6	.00932	.00905	.00879	.00854	.00829	.00805	.00781	.00759	.00736	.00715
2.7	.00693	.00673	.00653	.00633	.00614	.00596	.00578	.00561	.00544	.00527
2.8	.00511	.00495	.00480	.00465	.00451	.00437	.00424	.00410	.00398	.00385
2.9	.00373	.00361	.00350	.00339	.00328	.00318	.00308	.00298	.00288	.00279

z	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
3	0.00270	0.00194	0.00137	$0.0^{3}967$	$0.0^{3}674$	$0.0^{3}465$	$0.0^{3}318$	$0.0^{3}216$	$0.0^{3}145$	$0.0^{3}962$
4	$.0^{4}633$	$.0^{4}413$	$.0^{4}267$	$.0^{4}171$	$.0^{4}108$	$.0^{5}680$	$.0^{5}422$	$.0^{5}260$	$.0^{6}159$	$.0^{6}958$
5	$.0^{6}573$	$.0^{6}340$	$.0^{6}199$	$.0^{6}116$	$.0^{7}666$	$.0^{7}380$	$.0^{7}214$	$.0^{7}120$	$.0^{8}663$	$.0^{8}364$
6	$.0^{8}197$	$.0^{8}106$	$.0^{9}565$	$.0^{9}298$	$.0^{9}155$	$.0^{10}803$	$.0^{10}411$	$.0^{10}208$	$.0^{10}105$	$.0^{11}520$

Symmetrical Two-Limit Problems

In this discussion the term "symmetrical two-limit problems" refers to the area under the density function at equal values of z from both sides of the mean. The power supply example was this type, since we were concerned with the area between -3σ and 3σ from the mean $\bar{x}$. To work these problems when z is a decimal fraction, we use tables of areas in the two tails of the normal curve.

Table 5–4 shows tabulated areas in two tails of the normal curve for selected values of z from the mean $\bar{x}$. For example, when $z = 3.0$, the table shows that 0.00270 of the total area lies in the two tails of the curve below -3σ and above 3σ. Because the curve is symmetrical, 0.00135 of the area will lie to the left of -3σ and 0.00135 to the right of 3σ. Note that this agrees with figure 5–10 for the power supply example.

Example 8 (using table 5–4): Suppose that a circuit design requites that the gain β of a transistor be no less than 30 and

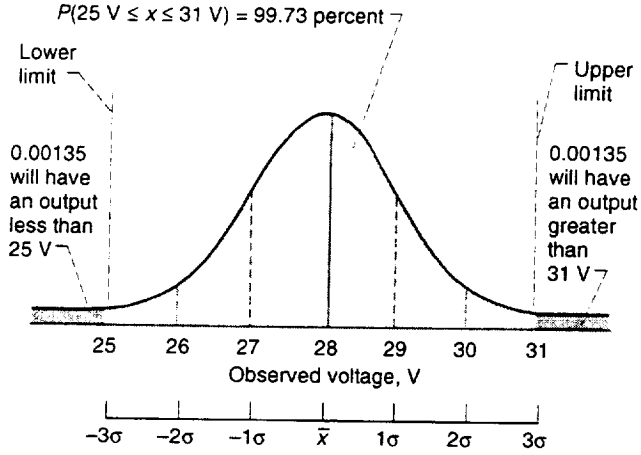


Figure 5–10.—Probability density functions for power supply outputs.

no greater than 180. The mean $\bar{x}$ of the β density function of a particular transistor is 105 with a standard deviation of 32. What percentage of the transistors will have a β within the required limits?

Solution 8:

Step 1—Solve for z.

$$x - \bar{x} = 105 - 30 = 180 - 105 = 75$$

Since σ is given as 32,

$$z = \frac{75}{32} = 2.34$$

Step 2—From table 5–4, the area in the two tails when $z = 2.34$ is 0.0193. Therefore, two tail tables 0.00965 of the transistors will have a β below 30 and 0.00965 will have a β above 180.

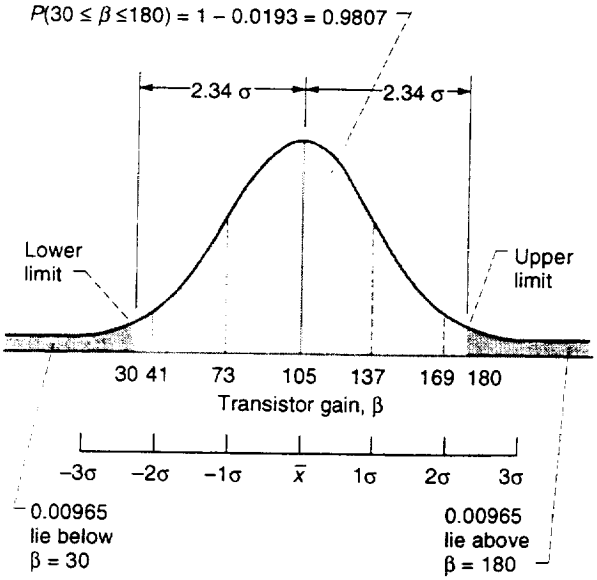


Figure 5–11.—Transistor gain.

Step 3—Now find $P(30 \le \beta \le 180)$. Since 0.0193 of the transistors will have a β below 30 or above 180, then $1 - 0.0193$ must give the percentage that will lie between 30 and 180. This is $1 - 0.0193 = 0.9807$, or 98.07 percent, as shown in figure 5–11. If we were to buy 100 000 of these transistors, we would expect 98 070 of them to have a β between 30 and 180. The remaining 1930 would not meet our β requirements.

One-Limit Problems

In many applications, engineers are interested only in one-sided limits, an upper or lower limit, rather than a two-sided upper and lower limit. In these cases, they are interested in the area under one tail of the density function as shown in figure 5–12. Tabulated values of the area in one tail of the normal density function at selected values of z are given in table 5–5.

Example 9: Suppose an exploding bridgewire (EBW) power supply is required to produce an output voltage of at least 1500 V. At this output voltage or greater, all the bridgewire detonators will explode. If the mean output of all such supplies is known to be 1575 V and the standard deviation is 46 V, what is the probability that an output of 1500 V or greater will be observed?

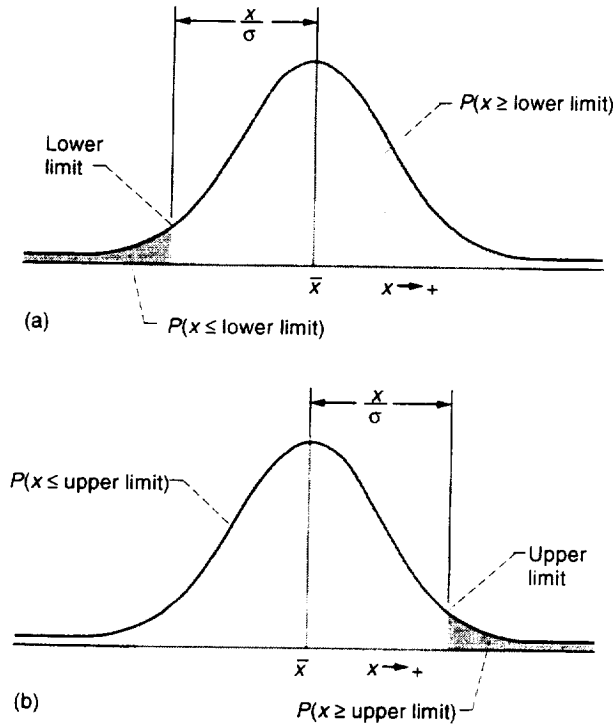
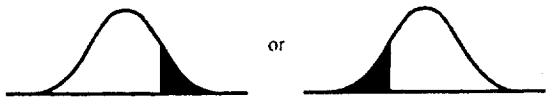


Figure 5–12.—Example of one-limit problems. (a) Lower limit. (b) Upper limit.

TABLE 5-5.—AREAS IN ONE TAIL OF NORMAL CURVE AT SELECTED VALUES OF z
[From reference 5-1.]



z	0	0.01	0.02	0.03	0.04	0.05	0.06	.0.07	0.08	0.09
0	0.5000	0.4960	0.4920	0.4880	0.4840	0.4801	0.4761	0.4721	0.4681	0.4641
.1	.4602	.4562	.4522	.4483	.4443	.4404	.4364	.4325	.4286	.4247
.2	.4207	.4168	.4129	.4090	.4052	.4013	.3974	.3936	.3897	.3859
.3	.3821	.3783	.3745	.3707	.3669	.3632	.3594	.3557	.3520	.3483
.4	.3446	.3409	.3372	.3336	.3300	.3264	.3228	.3192	.3156	.3121
.5	.3085	.3050	.3015	.2981	.2946	.2912	.2877	.2843	.2810	.2776
.6	.2743	.2709	.2676	.2643	.2611	.2578	.2546	.2514	.2483	.2451
.7	.2420	.2389	.2358	.2327	.2296	.2266	.2236	.2206	.2177	.2148
.8	.2119	.2090	.2061	.2033	.2005	.1977	.1949	.1922	.1894	.1867
.9	.1841	.1814	.1788	.1762	.1736	.1711	.1685	.1660	.1635	.1611
1.0	.1587	.1562	.1539	.1515	.1492	.1469	.1446	.1423	.1401	.1379
1.1	.1357	.1335	.1314	.1292	.1271	.1251	.1230	.1210	.1190	.1170
1.2	.1151	.1131	.1112	.1093	.1075	.1056	.1038	.1020	.1003	.0985
1.3	.0968	.0951	.0934	.0918	.0901	.0885	.0869	.0853	.0838	.0823
1.4	.0808	.0793	.0778	.0764	.0749	.0735	.0721	.0708	.0694	.0681
1.5	.0668	.0655	.0643	.0630	.0618	.0606	.0594	.0582	.0571	.0559
1.6	.0548	.0537	.0526	.0516	.0505	.0495	.0485	.0475	.0465	.0455
1.7	.0446	.0436	.0427	.0418	.0409	.0401	.0392	.0384	.0375	.0367
1.8	.0359	.0351	.0344	.0336	.0329	.0322	.0314	.0307	.0301	.0294
1.9	.0287	.0281	.0274	.0268	.0262	.0256	.0250	.0244	.0239	.0233
2.0	.0228	.0222	.0217	.0212	.0207	.0202	.0197	.0192	.0188	.0183
2.1	.0179	.0174	.0170	.0166	.0162	.0158	.0154	.0150	.0146	.0143
2.2	.0139	.0136	.0132	.0129	.0125	.0122	.0119	.0016	.0113	.0110
2.3	.0107	.0104	.0102	.00990	.00964	.00939	.00914	.00889	.00866	.00842
2.4	.00820	.00798	.0076	.00755	.00734	.00714	.00695	.00676	.00657	.00639
2.5	.00621	.00604	.00587	.00570	.00554	.00539	.00523	.00508	.00494	.00480
2.6	00466	.00453	.00440	.00427	.00415	.00402	.00391	.00379	.00368	.00357
2.7	.00347	.00336	.00326	.00317	.00307	.00298	.00289	.00280	.007272	.00264
2.8	.00256	.00248	.00240	.00233	.00226	.00219	.00212	.00205	.00199	.00193
2.9	.00187	.00181	.00175	.00169	.00164	.00159	.00154	.00149	.00144	.00139

z	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
3	0.00135	$0.0^{3}968$	$0.0^{3}687$	$0.0^{3}483$	$0.0^{3}337$	$0.0^{3}233$	$0.0^{3}159$	$0.0^{3}108$	$0.0^{4}723$	$0.0^{4}481$
4	$.0^{4}317$	$.0^{4}207$	$.0^{4}133$	$.0^{5}854$	$.0^{5}541$	$.0^{5}6340$	$.0^{5}211$	$.0^{5}130$	$.0^{6}793$	$.0^{6}479$
5	$.0^{6}287$	$.0^{6}170$	$.0^{7}996$	$.0^{7}579$	$.0^{7}333$	$.0^{7}190$	$.0^{7}107$	$.0^{8}599$	$.0^{8}332$	$.0^{8}182$
6	$.0^{9}987$	$.0^{9}530$	$.0^{9}282$	$.0^{9}149$	$.0^{10}777$	$.0^{10}402$	$.0^{10}206$	$.0^{10}104$	$.0^{11}523$	$.0^{11}260$

Solution 9:
Step 1—Calculate z.

$$z = \frac{\text{Mean limit}}{\sigma} = \frac{1575 - 1500}{46} = \frac{75}{46} = 1.63$$

Step 2—Find the area in one tail of the normal curve at z from the mean. From table 5–5 the tail area at $z = 1.63$ from the mean is given as 0.0516. Therefore, there is a 0.0516 probability that an observed output will be below 1500 V.

Step 3—Find the probability that the output will be 1500 V or greater. Since from step 2 $P(x \leq 1500) = 0.0516$,

$$P(x > 1500) = 1 - P(x \leq 1500) = 1 - 0.0516$$
$$= 0.9484, \text{ or } 94.84 \text{ percent}$$

We can therefore expect to obtain a 1500-V output voltage level 94.84 percent of the time. Or to express it another way, 94.84 percent of the supplies will produce an output above the

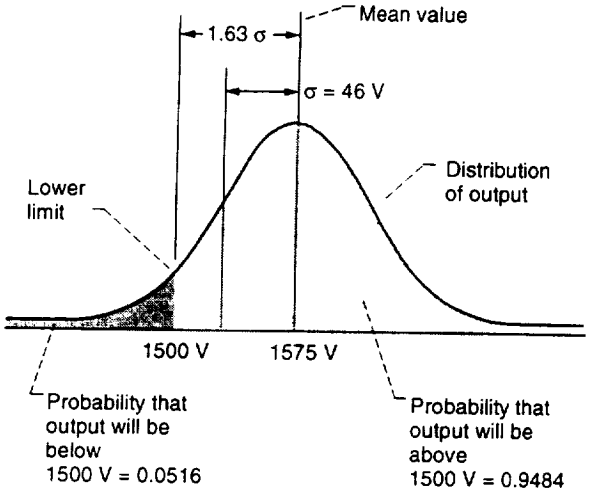


Figure 5–13.—Exploding bridgewire power supply output.

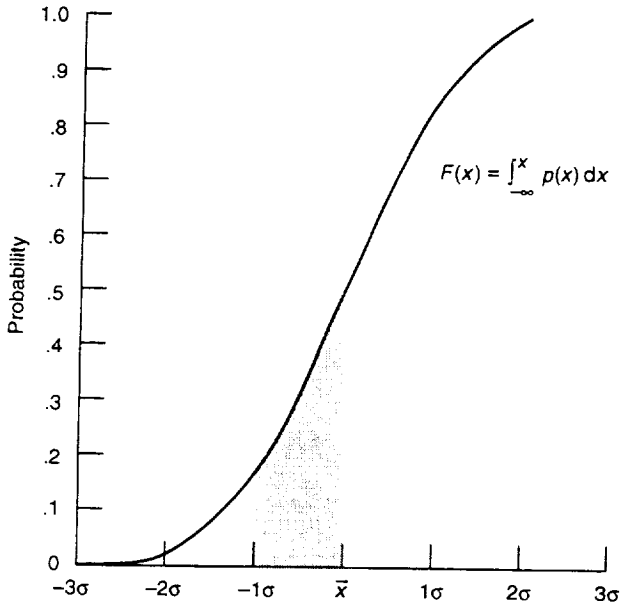


Figure 5–14.—Cumulative normal curve.

minimum requirement of 1500 V. This result is shown in figure 5–13. Associated with the probability density function $p(x)$ of the normal distribution is a cumulative probability distribution denoted by $F(x)$. As shown in the integral formulas of chapter 2, the relation between the two is given by

$$F(x) = \int p(x)\,dx$$

So, for the normal distribution

$$F(x) = \frac{1}{\sigma\sqrt{2\pi}} \int e^{-1/2[(x-\bar{x})/\sigma]^2}\,dx$$

or in z notation

$$F(z) = \frac{1}{\sqrt{2\pi}} \int e^{-(1/2)z^2}\,dz$$

A graph of $F(x)$ is shown in figure 5–14. Recall that in discussing cumulative functions earlier, $F(x)$ was called the cumulative area under the density curve. Looking at figure 5–14, then, you can see that

(1) $F(\bar{x}) = 0.5$, or that 50 percent of the area under the normal distribution is between $-\infty$ and the mean $\bar{x}$, or that there is a 50-percent probability that a variable x lies in the interval $(-\infty, \bar{x})$

(2) $1 - F(\bar{x}) = 0.5$, or that 50 percent of the area under the normal distribution is between the mean $\bar{x}$ and ∞; or that there is a 50-percent probability that a variable x lies in the interval $(\bar{x}, \infty)$

(3) The area between -1σ and $\bar{x}$ is

$$P(-1\sigma \le x \le \bar{x}) = F(\bar{x}) - F(-1\sigma)$$
$$= 0.5 - 0.16 = 0.34$$

or that there is a 0.34-probability that a variable x will lie between the mean $\bar{x}$ and -1σ.

For more accurate work, the cumulative areas for selected values of z have been tabulated and are shown in tables 5–6 and 5–7. Table 5–6 shows the cumulative areas for values of z from $-\infty$ to 0, which are illustrated in figure 5–15. Table 5–6 shows that

(1) At $z = 0$ (i.e., when the distance from the limit to $\bar{x}$ is 0), the cumulative area from $-\infty$ to $\bar{x}$ is 0.5000, or 50 percent

(2) At $z = -1.0$, the cumulative area from $-\infty$ to -1σ is 0.1587, or 15.87 percent

(3) At $z = -2.0$, the cumulative area from $-\infty$ to -2σ is 0.02275, or 2.275 percent

Table 5–7 shows the cumulative areas for values of z from 0 to ∞, which is illustrated in figure 5–16.

In both tables the value of z is the same as $F(x)$. It therefore follows that

(1) The probability of the variable x lying between $-\infty$ and $\bar{x}$ is

$$P(-\infty \le x \le \bar{x}) = F(\bar{x}) - F(-\infty)$$
$$= F(z = 0) - F(z = -\infty)$$
$$= 0.5 - 0 = 0.5, \text{ or } 50 \text{ percent}$$

(2) The probability of the variable x lying between -2.1σ and 3.2σ is

TABLE 5-6.—CUMULATIVE NORMAL DISTRIBUTION FROM $z = -\infty$ to 0

[From reference 5-2.]

z	0	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
−0	0.5000	0.4960	0.4920	0.4880	0.4840	0.4801	0.4761	0.4721	0.4681	0.4641
− .1	.4602	.4562	.4522	.4483	.4443	.4404	.4364	.4325	.4286	.4247
− .2	.4207	.4168	.4129	.4090	.4052	.4013	.3974	.3936	.3897	.3859
− .3	.3821	.3783	.3745	.3707	.3669	.3632	.3594	.3557	.3520	.3483
− .4	.3446	.3409	.3372	.3336	.3300	.3264	.3228	.3192	.3156	.3121
− .5	.3085	.3050	.3015	.2981	.2946	.2912	.2877	.2843	.2810	.2776
− .6	.2743	.2709	.2676	.2643	.2611	.2578	.2546	.2514	.2483	.2451
− .7	.2420	.2389	.2358	.2327	.2297	.2266	.2236	.2206	.2177	.2148
− .8	.2119	.2090	.2061	.2033	.2005	.1977	.1949	.1922	.1894	.1867
− .9	.1841	.1814	.1788	.1762	.1736	.1711	.1685	.1660	.1635	.1611
−1.0	.1587	.1562	.1539	.1515	.1492	.1469	.1446	.1423	.1401	.1379
−1.1	.1357	.1335	.1314	.1292	.1271	.1251	.1230	.1210	.1190	.1170
−1.2	.1151	.1131	.1112	.1093	.1075	.1056	.1038	.1020	.1003	.09853
−1.3	.09680	.09510	.09342	.09176	.09012	.08851	.08691	.08534	.08379	.08226
−1.4	.08076	.07927	.07780	.07636	.07493	.07353	.07215	.07078	.06944	.06811
−1.5	.06681	.06552	.06426	.06301	.06178	.06057	.05938	.05821	.05705	.05592
−1.6	.05480	.05370	.05262	.05155	.05050	.04947	.04846	.04746	.04648	.04551
−1.7	.04457	.04363	.04272	.04182	.04093	.04006	.03920	.03864	.03754	.03673
−1.8	.03593	.03515	.03438	.03362	.03288	.03216	.03144	.03074	.03005	.02938
−1.9	.02872	.02807	.02743	.02680	.02619	.02559	.02500	.02442	.02385	.02330
−2.0	.02275	.02222	.02169	.02118	.02068	.02018	.01970	.01923	.01876	.01831
−2.1	.01786	.01743	.01700	.01659	.01618	.01578	.01539	.01500	.01463	.01426
−2.2	.01390	.01355	.01321	.01287	.01255	.01222	.01191	.01160	.01130	.01101
−2.3	.01072	.01044	.01017	$.0^{2}9903$	$.0^{2}9642$	$.0^{2}9387$	$.0^{2}9137$	$.0^{2}8894$	$.0^{2}8656$	$.0^{2}8424$
−2.4	$.0^{2}8198$	$.0^{2}7976$	$.0^{2}7760$	$.0^{2}7549$	$.0^{2}7344$	$.0^{2}7143$	$.0^{2}6947$	$.0^{2}6756$	$.0^{2}6569$	$.0^{2}6387$
−2.5	$.0^{2}6210$	$.0^{2}6037$	$.0^{2}5868$	$.0^{2}5703$	$.0^{2}5543$	$.0^{2}5386$	$.0^{2}5234$	$.0^{2}5085$	$.0^{2}4940$	$.0^{2}4799$
−2.6	$.0^{2}4661$	$.0^{2}4527$	$.0^{2}4396$	$.0^{2}4269$	$.0^{2}4145$	$.0^{2}4025$	$.0^{2}3907$	$.0^{2}3793$	$.0^{2}3681$	$.0^{2}3573$
−2.7	$.0^{2}3467$	$.0^{2}3364$	$.0^{2}3264$	$.0^{2}3167$	$.0^{2}3072$	$.0^{2}2980$	$.0^{2}2890$	$.0^{2}2803$	$.0^{2}2718$	$.0^{2}2635$
−2.8	$.0^{2}2555$	$.0^{2}2477$	$.0^{2}2401$	$.0^{2}2327$	$.0^{2}2256$	$.0^{2}2186$	$.0^{2}2118$	$.0^{2}2052$	$.0^{2}1988$	$.0^{2}1926$
−2.9	$.0^{2}1866$	$.0^{2}1807$	$.0^{2}1750$	$.0^{2}1695$	$.0^{2}1641$	$.0^{2}1589$	$.0^{2}1538$	$.0^{2}1489$	$.0^{2}1441$	$.0^{2}1395$
−3.0	$.0^{2}1350$	$.0^{2}1306$	$.0^{2}1264$	$.0^{2}1223$	$.0^{2}1183$	$.0^{2}1144$	$.0^{2}1107$	$.0^{2}1070$	$.0^{2}1035$	$.0^{2}1001$
−3.1	$.0^{3}9676$	$.0^{3}9354$	$.0^{3}9043$	$.0^{3}8740$	$.0^{3}8447$	$.0^{3}8164$	$.0^{3}7888$	$.0^{3}7622$	$.0^{3}7364$	$.0^{3}7114$
−3.2	$.0^{3}6871$	$.0^{3}6637$	$.0^{3}6410$	$.0^{3}6190$	$.0^{3}5976$	$.0^{3}5770$	$.0^{3}5571$	$.0^{3}5377$	$.0^{3}5190$	$.0^{3}5009$
−3.3	$.0^{3}4834$	$.0^{3}4665$	$.0^{3}4501$	$.0^{3}4342$	$.0^{3}4189$	$.0^{3}4041$	$.0^{3}3897$	$.0^{3}3758$	$.0^{3}3624$	$.0^{3}3495$
−3.4	$.0^{3}3369$	$.0^{3}3248$	$.0^{3}3131$	$.0^{3}3018$	$.0^{3}2909$	$.0^{3}2803$	$.0^{3}2701$	$.0^{3}2602$	$.0^{3}2507$	$.0^{3}2415$
−3.5	$.0^{3}2326$	$.0^{3}2241$	$.0^{3}2158$	$.0^{3}2078$	$.0^{3}2001$	$.0^{3}1926$	$.0^{3}1854$	$.0^{3}1785$	$.0^{3}1718$	$.0^{3}1653$
−3.6	$.0^{3}1591$	$.0^{3}1531$	$.0^{3}1473$	$.0^{3}1417$	$.0^{3}1363$	$.0^{3}1311$	$.0^{3}1261$	$.0^{3}1213$	$.0^{3}1166$	$.0^{3}1121$
−3.7	$.0^{3}1078$	$.0^{3}1036$	$.0^{4}9961$	$.0^{4}9574$	$.0^{4}9201$	$.0^{4}8842$	$.0^{4}8496$	$.0^{4}8162$	$.0^{4}7841$	$.0^{4}7532$
−3.8	$.0^{4}7235$	$.0^{4}6948$	$.0^{4}6673$	$.0^{4}6407$	$.0^{4}6152$	$.0^{4}5906$	$.0^{4}5569$	$.0^{4}5442$	$.0^{4}5223$	$.0^{4}5012$
−3.9	$.0^{4}4810$	$.0^{4}4615$	$.0^{4}4427$	$.0^{4}4247$	$.0^{4}4074$	$.0^{4}3908$	$.0^{4}3747$	$.0^{4}3594$	$.0^{4}3446$	$.0^{4}3304$
−4.0	$.0^{4}3167$	$.0^{4}3036$	$.0^{4}2910$	$.0^{4}2789$	$.0^{4}2673$	$.0^{4}2561$	$.0^{4}2454$	$.0^{4}2351$	$.0^{4}2252$	$.0^{4}2157$
−4.1	$.0^{4}2066$	$.0^{4}1978$	$.0^{4}1894$	$.0^{4}1814$	$.0^{4}1737$	$.0^{4}1662$	$.0^{4}1591$	$.0^{4}1523$	$.0^{4}1458$	$.0^{4}1395$
−4.2	$.0^{4}1335$	$.0^{4}1277$	$.0^{4}1222$	$.0^{4}1168$	$.0^{4}1118$	$.0^{4}1069$	$.0^{4}1022$	$.0^{5}9774$	$.0^{5}9345$	$.0^{5}8934$
−4.3	$.0^{5}8540$	$.0^{5}8163$	$.0^{5}7801$	$.0^{5}7455$	$.0^{5}7124$	$.0^{5}6807$	$.0^{5}6503$	$.0^{5}6212$	$.0^{5}5934$	$.0^{5}5668$
−4.4	$.0^{5}5413$	$.0^{5}5169$	$.0^{5}4935$	$.0^{5}4712$	$.0^{5}4498$	$.0^{5}4294$	$.0^{5}4098$	$.0^{5}3911$	$.0^{5}3732$	$.0^{5}3561$
−4.5	$.0^{5}3398$	$.0^{5}3241$	$.0^{5}3092$	$.0^{5}2949$	$.0^{5}2813$	$.0^{5}2682$	$.0^{5}2558$	$.0^{5}2439$	$.0^{5}2325$	$.0^{5}2216$
−4.6	$.0^{5}2112$	$.0^{5}2013$	$.0^{5}1919$	$.0^{5}1828$	$.0^{5}1742$	$.0^{5}1660$	$.0^{5}1581$	$.0^{5}1506$	$.0^{5}1434$	$.0^{5}1366$
−4.7	$.0^{5}1301$	$.0^{5}1239$	$.0^{5}1179$	$.0^{5}1123$	$.0^{5}1069$	$.0^{5}1017$	$.0^{6}9680$	$.0^{6}9211$	$.0^{6}8765$	$.0^{6}8339$
−4.8	$.0^{6}7933$	$.0^{6}7547$	$.0^{6}7178$	$.0^{6}6827$	$.0^{6}6492$	$.0^{6}6173$	$.0^{6}5869$	$.0^{6}5580$	$.0^{6}5304$	$.0^{6}5042$
−4.9	$.0^{6}4792$	$.0^{6}4554$	$.0^{6}4327$	$.0^{6}4111$	$.0^{6}3906$	$.0^{6}3711$	$.0^{6}3525$	$.0^{6}3348$	$.0^{6}3179$	$.0^{6}3019$
$-\infty$	0	0	0	0	0	0	0	0	0	0

TABLE 5-7.—CUMULATIVE NORMAL DISTRIBUTION FROM $z = 0$ to ∞

[From reference 5-2.]

z	0	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
0	0.5000	0.5040	0.5080	0.5120	0.5160	0.5199	0.5239	0.5279	0.5319	0.5359
.1	.5398	.5438	.5478	.5517	.5557	.5596	.5836	.5675	.5714	.5753
.2	.5793	.5832	.5871	.5910	.5948	.5987	.6026	.6064	.6103	.6141
.3	.6179	.6217	.6255	.6293	.6331	.6368	.6406	.6443	.6480	.6517
.4	.6554	.6591	.6628	.6664	.6700	.6736	.6772	.6808	.6844	.6879
.5	.6915	.6950	.6985	.7019	.7054	.7088	.7123	.7157	.7190	.7224
.6	.7257	.7291	.7324	.7357	.7389	.7422	.7454	.7486	.7517	.7549
.7	.7580	.7611	.7642	.7673	.7703	.7734	.7764	.7794	.7823	.7852
.8	.7881	.7910	.7939	.7967	.7995	.8023	.8051	.8078	.8106	.8133
.9	.8159	.8186	.8212	.8238	.8264	.8289	.8315	.8340	.8365	.8389
1.0	.8413	.8438	.8461	.8485	.8508	.8531	.8554	.8577	.8599	.8621
1.1	.8643	.8665	.8686	.8708	.8729	.8749	.8770	.8790	.8810	.8830
1.2	.8849	.8869	.8888	.8907	.8925	.8944	.8962	.8980	.8997	.90147
1.3	.90320	.90490	.90658	.90824	.90988	.91149	.91309	.91466	.91621	.91774
1.4	.91924	.92073	.92220	.92364	.92507	.92647	.92785	.92922	.93056	.93189
1.5	.93319	.93448	.93574	.93699	.93822	.93943	.94062	.94179	.94295	.94408
1.6	.94520	.94630	.94738	.94845	.94950	.95053	.95154	.95254	.95352	.95449
1.7	.95543	.95637	.95728	.95818	.95907	.95994	.96080	.96164	.96246	.96327
1.8	.96407	.96485	.96562	.96638	.96712	.96784	.96856	.96926	.96995	.97062
1.9	.97128	.97193	.97257	.97320	.97381	.97441	.97500	.97558	.97615	.97670
2.0	.97725	.97778	.97831	.97882	.97932	.97982	.98030	.98077	.98124	.98169
2.1	.98214	.98257	.98300	.98341	.98382	.98422	.98461	.98500	.98537	.98574
2.2	.98610	.98645	.98679	.98713	.98745	.98778	.98809	.98840	.98870	.98899
2.3	.98928	.98956	.98983	.$9^2$0097	.$9^2$0358	.$9^2$0613	.$9^2$0863	.$9^2$1106	.$9^2$1344	.$9^2$1567
2.4	.$9^2$1802	.$9^2$2024	.92240	.$9^2$2451	.$9^2$2656	.$9^2$2857	.$9^2$3053	.$9^2$3244	.$9^2$3431	.$9^2$3613
2.5	.$9^2$3790	.$9^2$3963	.$9^2$4132	.$9^2$4297	.$9^2$4457	.$9^2$4614	.$9^2$4766	.$9^2$4915	.$9^2$5060	.$9^2$5201
2.6	.$9^2$5339	.$9^2$5473	.$9^2$5604	.$9^2$5731	.$9^2$5855	.$9^2$5975	.$9^2$6093	.$9^2$6207	.$9^2$6319	.$9^2$6427
2.7	.$9^2$6533	.$9^2$6636	.$9^2$6736	.$9^2$6833	.$9^2$6928	.$9^2$7020	.$9^2$7110	.$9^2$7197	.$9^2$7282	.$9^2$7365
2.8	.$9^2$7445	.$9^2$7523	.$9^2$7599	.$9^2$7673	.$9^2$7744	.$9^2$7814	.$9^2$7882	.$9^2$7948	.$9^2$8012	.$9^2$8074
2.9	.$9^2$8134	.$9^2$8193	.$9^2$8250	.$9^2$8305	.$9^2$8359	.$9^2$8411	.$9^2$8462	.$9^2$8511	.$9^2$8559	.$9^2$8605
3.0	.$9^2$8650	.$9^2$8694	.$9^2$8736	.$9^2$8777	.$9^2$8817	.$9^2$8856	.$9^2$8893	.$9^2$8930	.$9^2$8965	.$9^2$8999
3.1	.$9^2$0324	.$9^2$0646	.$9^3$0957	.$9^3$1260	.$9^3$1553	.$9^3$1863	.$9^3$2112	.$9^3$2378	.$9^3$2636	.$9^3$2886
3.2	.$9^3$3129	.$9^3$3363	.$9^3$3590	.$9^3$3810	.$9^3$4024	.$9^3$4230	.$9^3$4429	.$9^3$4623	.$9^3$4810	.$9^3$4991
3.3	.$9^3$5166	.$9^3$5355	.$9^3$5499	.$9^3$5658	.$9^3$5811	.$9^3$5959	.$9^3$6103	.$9^3$6242	.$9^3$6376	.$9^3$6505
3.4	.$9^3$6631	.$9^3$6752	.$9^3$6869	.$9^3$6982	.$9^3$7091	.$9^3$7197	.$9^3$7299	.$9^3$7398	.$9^3$7493	.$9^3$7585
3.5	.$9^3$7674	.$9^3$7759	.$9^3$7842	.$9^3$7922	.$9^3$7999	.$9^3$8074	.$9^3$8146	.$9^3$8215	.$9^3$8282	.$9^3$8347
3.6	.$9^3$8409	.$9^3$8469	.$9^3$8527	.$9^3$8583	.$9^3$8637	.$9^3$8689	.$9^3$8739	.$9^3$8787	.$9^3$8834	.$9^3$8879
3.7	.$9^3$8922	.$9^3$8964	.$9^4$0039	.$9^4$0426	.$9^4$0799	.$9^4$1158	.$9^4$1504	.$9^4$1838	.$9^4$2159	.$9^4$2468
3.8	.$9^4$2765	.$9^4$3052	.$9^4$3327	.$9^4$3593	.$9^4$3848	.$9^4$4094	.$9^4$4331	.$9^4$4558	.$9^4$4777	.$9^4$4988
3.9	.$9^4$5190	.$9^4$5385	.$9^4$5573	.$9^4$5753	.$9^4$5926	.$9^4$6092	.$9^4$6253	.$9^4$6406	.$9^4$6554	.$9^4$6696
4.0	.$9^4$6833	.$9^4$6964	.$9^4$7090	.$9^4$7211	.$9^4$7327	.$9^4$7439	.$9^4$7546	.$9^4$7649	.$9^4$7748	.$9^4$7843
4.1	.$9^4$7934	.$9^4$8022	.$9^4$8106	.$9^4$8186	.$9^4$8263	.$9^4$8338	.$9^4$8409	.$9^4$8477	.$9^4$8542	.$9^4$8605
4.2	.$9^4$8665	.$9^4$8723	.$9^4$8778	.$9^4$8832	.$9^4$8882	.$9^4$8931	.$9^4$8978	.$9^5$40226	.$9^5$0655	.$9^5$1066
4.3	.$9^5$1460	.$9^5$1837	.$9^5$2199	.$9^5$2545	.$9^5$2876	.$9^5$3193	.$9^5$3497	.$9^5$3788	.$9^5$4066	.$9^5$4332
4.4	.$9^5$4587	.$9^5$4831	.$9^5$5065	.$9^5$5288	.$9^5$5502	.$9^5$5706	.$9^5$5902	.$9^5$6089	.$9^5$6268	.$9^5$6439
4.5	.$9^5$6602	.$9^5$6759	.$9^5$6908	.$9^5$7051	.$9^5$7187	.$9^5$7318	.$9^5$7442	.$9^5$7561	.$9^5$7675	.$9^5$7784
4.6	.$9^5$7888	.$9^5$7987	.$9^5$8081	.$9^5$8172	.$9^5$8258	.95834	.$9^5$8419	.$9^5$8494	.$9^5$8566	.$9^5$8634
4.7	.$9^5$8699	.$9^5$8761	.$9^5$8821	.$9^5$8877	.$9^5$8931	.$9^5$8983	.$9^6$0320	.$9^6$0789	.$9^6$1235	.$9^6$1661
4.8	.$9^6$2067	.$9^6$2463	.$9^6$2822	.$9^6$3173	.$9^6$3508	.$9^6$3827	.$9^6$4131	.$9^6$4420	.$9^6$4696	.$9^6$4958
4.9	.$9^6$5208	.$9^6$5446	.$9^6$5673	.$9^6$5889	.$9^6$6094	.$9^6$6289	.$9^6$6475	.$9^6$6652	.$9^6$6821	.$9^6$6981
∞	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

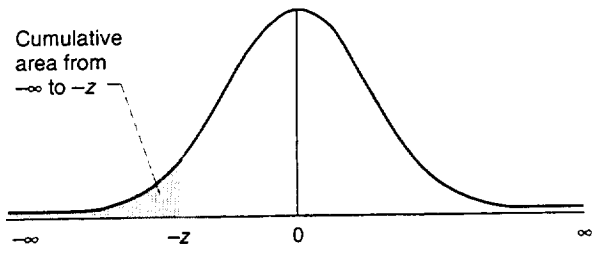


Figure 5–15.—Cumulative areas for values of z from −∞ to 0.

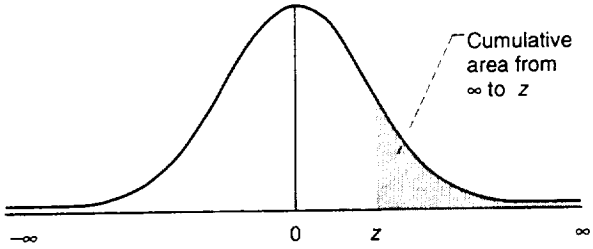


Figure 5–16.—Cumulative areas for values of z from 0 to ∞.

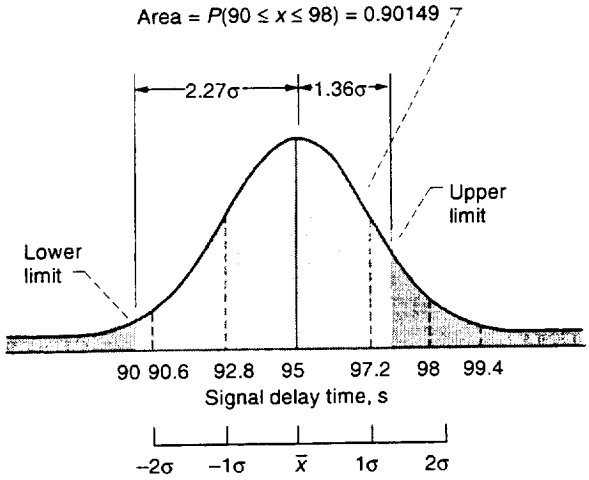


Figure 5–17.—Signal delay time.

$$P(-2.1\sigma \le x \le 3.2\sigma) = F(3.2) - F(-2.1)$$
$$= F(z = 3.2) - F(z = -2.1)$$
$$= 0.9993129 - 0.01786$$
$$= 0.9814529, \text{ or } 98 \text{ percent}$$

Nonsymmetrical Two-Limit Problems

The cumulative function is useful for solving nonsymmetrical two-limit problems, which are in practice the most frequently encountered.

Example 10: Suppose that a time-delay relay is required to delay the transmission of a signal at least 90 sec but no more than 98 sec. If the mean "time out" of the specific type of relay is 95 sec and the standard deviation is 2.2 sec, what is the probability that the signal will be delayed within the specified times?

Solution 10:

Step 1—Find $F(98 \text{ sec})$. Since the mean is given as 95 sec and the standard deviation as 2.2 sec,

$$z = \frac{\text{Limit} - \text{Mean}}{\sigma} = \frac{98 - 95}{2.2} = \frac{3}{2.2} = 1.36$$

From table 5–7,

$$F(98 \text{ sec}) = F(z) = F(1.36) = 0.91309$$

Step 2—Find $F(90 \text{ sec})$. Since the mean is 95 sec and the standard deviation is 2.2 sec,

$$z = \frac{90 - 95}{2.2} = \frac{-5}{2.2} = -2.27$$

From table 5–6,

$$F(90 \text{ sec}) = F(z) = F(-2.27) = 0.01160$$

Step 3—Find $P(90 \le x \le 98)$. From steps 1 and 2,

$$P(90 \le x \le 98) = F(98) - F(90) = 0.91309 - 0.01160$$
$$= 0.90149, \text{ or } 90 \text{ percent}$$

There exists, therefore, a 90-percent probability that the signal will be delayed no less than 90 sec and no more than 98 sec, as shown in figure 5–17.

Application of Normal Distribution to Test Analyses and Reliability Predictions

This section gives two examples of how the normal distribution techniques may be applied to the analysis of test data of certain devices and how the results of the analysis may be used to estimate or predict the outcome of actual tests (ref. 5–5). Many similar examples are given in chapter 6.

Example 11: For this two-limit problem, assume that a door hinge has a pin pull-force requirement of 12 ± 4.64 lb. Assume further that we have received 116 door hinges and have actually measured the pin pull-force required for 16 of them as part of an acceptance test. The results of the test are shown in table 5–8 and in the histogram of figure 5–18. We now want to apply normal distribution theory and then estimate what percentage of the remaining 100 door hinges will meet the pin pull-force requirement.

Solution 11:

Step 1—Solve for the mean of the test data $\bar{x}$. We have already seen that

TABLE 5–8.—RESULTS OF DOOR HINGE ACCEPTANCE TEST

Pull-force required, lb	Number of occurrences
8	1
10	3
12	7
14	4
16	1
Total	16

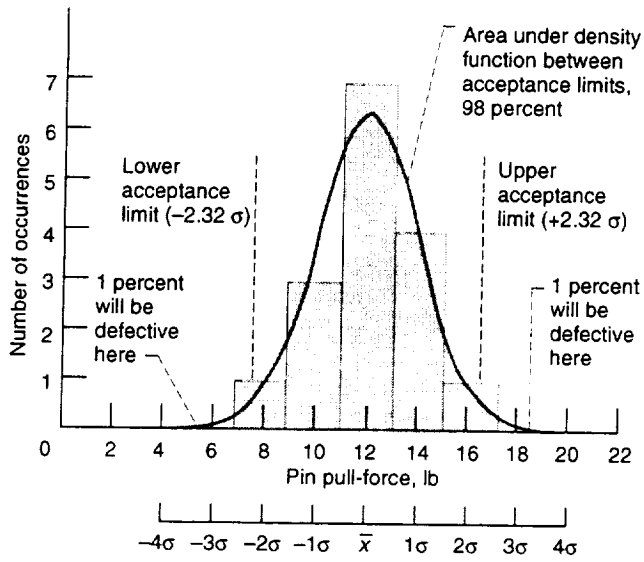


Figure 5–18.—Door hinge test results.

$$\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n}$$

where

x_i value of i^{th} measurement

n total number of measurements

Let x = pound forces so that

$x_1 = 8$ $x_9 = 12$

$x_2 = 10$ $x_{10} = 12$

$x_3 = 10$ $x_{11} = 12$

$x_4 = 10$ $x_{12} = 14$

$x_5 = 12$ $x_{13} = 14$

$x_6 = 12$ $x_{14} = 14$

$x_7 = 12$ $x_{15} = 14$

$x_8 = 12$ $x_{16} = 16$

and let n = 16 (number of occurrences). The mean $\bar{x}$ is therefore

$$\bar{x} = \frac{\sum_{i=1}^{16} x_i}{n} = \frac{8 + 3(10) + 7(12) + 4(14) + 16}{16}$$

$$= 12 \text{ lb (rounded to two places)}$$

Step 2—Solve for the standard deviation σ. We have also seen that

$$\sigma = \left[\frac{\sum_{i=1}^{n} (x_{i-\bar{x}})^2}{n-1} \right]^{1/2}$$

where

$\bar{x}$ observed mean

x value of i^{th} measurement

n total number of measurements

Solve for $\sum_{i=1}^{n} (x_{i-\bar{x}})^2$:

$$\sum_{i=1}^{n} (x_{i-\bar{x}})^2 = \sum_{i=1}^{16} (x_i - 12)^2$$

$$= (8-12)^2 + 3(10-12)^2 + 7(12-12)^2 + 4(14-12)^2 + (16-12)^2$$

$$= (-4)^2 + 3(-2)^2 + 7(0)^2 + 4(2)^2 + (4)^2$$

$$= 16 + 12 + 0 + 16 + 16 = 60$$

Then solve for $\frac{\sum_{i=1}^{16}(x_i-12)^2}{n-1}$:

$$\frac{\sum_{i=1}^{16}(x_i-12)^2}{n-1}=\frac{60}{16-1}=\frac{60}{15}=4$$

Finally solve for σ:

$$\sigma=\left[\frac{\sum_{i=1}^{16}(x_i-12)^2}{n-1}\right]^{1/2}=\sqrt{4}=2\text{ lb}$$

Step 3—With a mean of $\bar{x}$ = 12 lb and a standard deviation of σ = 2 lb, figure 5–18 shows that

(1) The lower pull-force limit of 7.36 lb is z = (7.36 – 12)/2 = –2.32 standard deviations from the mean.

(2) The upper limit of 16.64 lb is z = (16.64 – 12)/2 = 2.32 standard deviations from the mean.

Consequently, the percentage of door hinges that should fall within the 12 ± 4.64-lb tolerance is given by

$$\begin{aligned}P(-2.32\sigma \le x \le 2.32\sigma) &= F(2.32)-F(-2.32)\\ &= 0.98983-0.01017\\ &\text{(from tables 5–6 and 5–7)}\\ &= 0.97966\text{, or 98 percent}\end{aligned}$$

This says that 98 percent of the door hinges should fall within the 12 ± 4.64-lb tolerance and that 2 percent should be outside the required tolerance. However, none of the 16 samples was outside the tolerance. So where are the 2 percent that the analysis says are defective? The answer is that the 2 percent of defective door hinges are in the 100 not tested.

We can make this statement by assuming that if we had tested all 100 door hinges, we would have expected to observe the same mean ($\bar{x}$ = 12 lb) and standard deviation (σ = 2 lb) that we did with the 16 samples. Note that this assumption is subject to confidence limits discussed in chapter 6. If we accept this assumption, we would expect to find 2 of the 100 door hinges

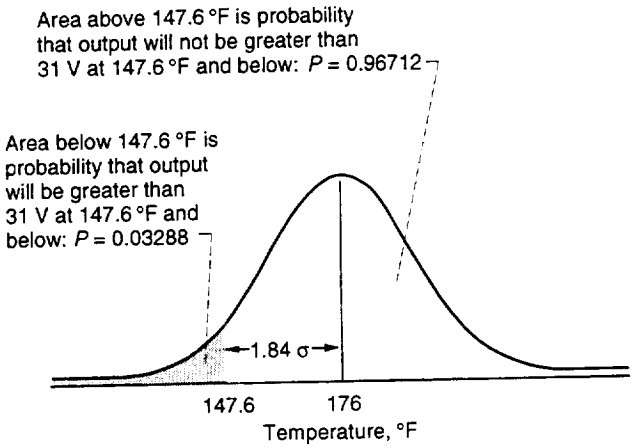


Figure 5–19.—Failure distribution of power supplies.

defective: one would have a pull force less than 7.36 lb (the lower limit) and the other, a pull force greater than 16.64 lb (the upper limit). This is also shown in figure 5–18.

However, considering the 16 door hinges to be actually representative of all such door hinges, we could predict that only 98 percent of such door hinges produced would meet the acceptance criteria of a 12 ± 4.64-lb pin pull force.

Example 12: In this one-limit problem, 10 power supplies are selected out of a lot of 110 and tested at increasing temperatures until all exceed a maximum permissible output of 31 V. The failure temperatures in degrees centigrade of the 10 supplies are observed to be

$x_1 = 57$	$x_6 = 60$
$x_2 = 65$	$x_7 = 75$
$x_3 = 53$	$x_8 = 82$
$x_4 = 62$	$x_9 = 71$
$x_5 = 66$	$x_{10} = 69$

Find the probability that the remaining 100 supplies will have an output greater than 31 V at 50 °C and below.

Solution 12:

Step 1—Solve for the mean $\bar{x}$:

$$\bar{x}=\frac{\sum_{i=1}^{10}x_i}{10}=\frac{57+65+53+62+66+60+75+82+71+69}{10}$$

$$=\frac{660}{10}=66\text{ °C}$$

Step 2—Solve for the standard deviation σ. First,

$$\sum_{i=1}^{10}(x_i-66)^2=(57-66)^2+(65+66)^2+(53-66)^2$$
$$+(62-66)^2+(66-66)^2+(60-66)^2$$
$$+(75-66)^2+(82-66)^2+(71-66)^2+(69-66)^2$$
$$=81+1+169+16+0+36+81+256+25+9$$
$$=674$$

Then

$$\sigma=\left[\frac{\sum_{i=1}^{10}(x_i-66)^2}{n-1}\right]^{1/2}=\left(\frac{674}{9}\right)^{1/2}$$
$$=8.7\ ^{\circ}\text{C (rounded to two places)}$$

Step 3—Solve for z = (Limit – Mean)/σ. With an observed mean $\bar{x}$ = 66 and a standard deviation σ = 8.7, the 50 °C limit is z = (50 – 66)/8.7 = – 16/8.7 = – 1.84 observation locations in standard deviations from the mean.

Step 4—Look at table 5–6 and find the cumulative area from $-\infty$ to σ = – 1.84. This is given as 0.03288. Therefore, there is a 3.288-percent probability that the remaining 100 supplies will have an output greater than 31 V at 50 °C and below. This is shown in figure 5–19.

Effects of Tolerance on a Product

Because tolerances must be anticipated in all manufacturing processes, some important questions to ask about the effects of tolerance on a product are

(1) How is the reliability affected?
(2) How can tolerances be analyzed and what methods are available?
(3) How are tolerance failures affected?

Electrical circuits are often affected by part tolerances (circuit gains can shift up or down, and transfer function poles or zeros can shift into the right-hand s-plane, causing oscillations). Mechanical components may not fit together or may be so loose that excessive vibration causes failure (refs. 5–6 to 5–8).

Notes on Tolerance Accumulation: A How-To-Do-It Guide

General.—The notation used in calculating tolerance is

T	tolerance
σ_V	standard deviation
V	dependent variable subject to tolerance accumulation
x	independent, measurable parameter
1,2,3,n	subscript notation for parameters
i	generalized subscript (i.e., i = 1,2,3,. . .,n for x_i)

Tolerance is usually $\pm 3\sigma$. When in doubt, find out. Note that when T is expressed in percent, always convert to engineering units before proceeding. The mean or average is $\bar{V}=f(\bar{x}_1,\bar{x}_2,\bar{x}_3,\ldots,\bar{x}_n)$. The coefficient of variation is $C_v=(\sigma/V)\times 100$ = percent.

Worst-case method.—The worst-case method is as follows:

$$V=f\left[(\bar{x}_1+T_1),(\bar{x}_2+T_2),(\bar{x}_3+T_3),\ldots,(\bar{x}_n+T_n)\right]$$
$$-V=f\left[(\bar{x}_1-T_1),(\bar{x}_2-T_2),(\bar{x}_3-T_3),\ldots,(\bar{x}_n-T_n)\right]$$

Actually,

$$\pm V=f\left[(\bar{x}_1\pm T_1),(\bar{x}_2\pm T_2),(\bar{x}_3\pm T_3),\ldots,(\bar{x}_n\pm T_n)\right]$$

where the plus or minus sign is selected for maximum V and then selected to give minimum V. If these $\pm V$ worst-case limits are acceptable, go no farther. If not, try the root-sum-square method.

Root-sum-square method.—The root-sum-square method is valid only if the $f(x$'s) are algebraically additive (i.e., when V is a linear function of the x's):

$$\pm V=\bar{V}\pm 3\sigma_v$$

where

$$\sigma_v^2=\sigma_1^2+\sigma_2^2+\sigma_3^2+\ldots+\sigma_n^2$$

and

$$\sigma_i=\frac{T_i}{3}\quad \text{if } T_i=\pm 3\sigma$$

Stated another way

$$\mp V = \bar{V} \pm \left[\sum_{i=1}^{n}\left(\frac{T_i}{3}\right)^2\right]^{1/2}$$

If these $\pm V$ root-sum-square limits are acceptable, go no farther. If they are not acceptable or the $f(x$'s) involve products or quotients, try the perturbation or partial derivative methods.

Perturbation method.—The perturbation method is as follows:

$$\mp V = \bar{V} \pm 3\sigma_v$$

where

$$\sigma_v^2 = \left(\bar{V}_{\Delta x_1} - \bar{V}\right)^2 + \left(\bar{V}_{\Delta x_2} - \bar{V}\right)^2 + \ldots + \left(\bar{V}_{\Delta x_n} - \bar{V}\right)^2$$

and where

$$\bar{V}_{\Delta x_1} = f\left[(\bar{x}_1 \pm \sigma_1),(\bar{x}_2 \pm \sigma_2),(\bar{x}_3 \pm \sigma_3),\ldots,(\bar{x}_n \pm \sigma_n)\right]$$

The $\pm V$ limits are valid if $C_v = \left(\sigma_v / \bar{V}\right) \times 100 \leq 10$ percent.

Partial derivative method.—The partial derivative method is as follows:

$$\mp V = \bar{V} \pm 3\sigma_v$$

where

$$\sigma_v^2 = \left(\frac{\partial V}{\partial x_1}\right)\sigma_{x_1}^2 + \left(\frac{\partial V}{\partial x_2}\right)^2 \sigma_{x_2}^2 + \ldots + \left(\frac{\partial V}{\partial x_n}\right)^2 \sigma_{x_n}^2$$

The $\pm V$ limits are valid if $C_v = \left(\sigma_v / \bar{V}\right) \times 100 \leq 10$ percent.

Thus, four methods are available for estimating the effects of tolerance on a product. The worst-case method can be used on any problem. In those cases where the $\pm V$ worst-case limits are not acceptable, other methods can be tried. The root-sum-square method is usually valid if the functions are algebraically additive. The perturbation or partial derivative methods are valid only if the coefficient of variation is less than or equal to 10 percent.

Estimating Effects of Tolerance

The following examples illustrate how these tolerance equations can be used. Consider a stacked tolerance problem where the dependent variable is a linear function—three variables added to give $\bar{V}$:

$$\bar{V} = f(\bar{x}_1, \bar{x}_2, \bar{x}_3)$$
$$\bar{V} = \bar{x}_1 + \bar{x}_2 + \bar{x}_3$$
$$T = 3\sigma$$

where

$$\bar{x}_1 = 1 \pm 0.1 \text{ mil}$$
$$\bar{x}_2 = 2 \pm 0.1 \text{ mil}$$
$$\bar{x}_3 = 3 \pm 0.1 \text{ mil}$$

Now, find $\bar{V}$ and the expected range of V:

$$\bar{V} = 1 + 2 + 3 = 6 \text{ mils}$$

Using the worst-case method, with positive tolerance

$$\bar{V}_+ = (1 + 0.1) + (2 + 0.1) + (3 + 0.1) = 6.3_+$$

and with negative tolerance

$$\bar{V}_- = (1 - 0.1) + (2 - 0.1) + (3 - 0.1) = 5.7_-$$

or

$$\bar{V}_\pm = 6 \pm 0.3 \text{ mil}$$

In the worst-case method, the tolerance on $\bar{V}$ (i.e., 0.3 mil) is worse than the $3\sigma_v$ tolerance. Tolerance can and often does cause fit problems and circuit problems. Therefore, in some cases we need to know what tolerance is acceptable.

Using the root-sum-square method,

$$\bar{V} = 6 \text{ mils}$$

and

$$\sigma_1 = \frac{0.1}{3} = 0.033 = \sigma_2 = \sigma_3$$

$$\sigma_v = \left(\sigma_1^2 + \sigma_2^2 + \sigma_3^2\right)^{1/2} = \left(3\sigma_1^2\right)^{1/2}$$

$$\left[3(0.033)^2\right]^{1/2} = 0.0572$$

$$3\sigma_v = 0.172$$

so that

$$\bar{V}_\pm = 6 \pm 0.172 \text{ mil}$$

In the root-sum-square method, the T value of 0.172 is the 3σ tolerance on V.

As a second example, consider a volume problem that has three variables in multiplication. Find $\overline{V}$ and the expected range of V:

$$\overline{V} = \overline{L}\,\overline{W}\overline{H} = 10 \text{ ft} \times 5 \text{ ft} \times 2 \text{ ft} = 100 \text{ ft}^3$$

First, convert percent tolerances to engineering units:

$$\overline{L} = 10 \text{ ft} \pm 10 \text{ percent} = 10 \text{ ft} \pm 10 \text{ ft} \times 0.1 = 10 \text{ ft} \pm 1 \text{ ft}$$

$$\overline{W} = 5 \text{ ft} \pm 10 \text{ percent} = 5 \text{ ft} \pm 5 \text{ ft} \times 0.1 = 5 \text{ ft} \pm 0.5 \text{ ft}$$

$$\overline{H} = 2 \text{ ft} \pm 5 \text{ percent} = 2 \text{ ft} \pm 2 \text{ ft} \times 0.05 = 2 \text{ ft} \pm 0.1 \text{ ft}$$

$$T = \pm 3\sigma$$

Using the worst-case method,

$$V_{\pm} = (10 \pm 1) \times (5 \pm 0.5) \times (2 \pm 0.1) = 11 \times 5.5 \times 2.1$$
$$\text{or } 9 \times 4.5 \times 1.9 = 127 \text{ or } 77$$

The root-sum-square method cannot be used because these variables are not algebraically additive. Using the perturbation method,

$$V = \overline{V} \pm 3\sigma_V$$

where

$$\sigma_V = \left[\left(\overline{V}_{\Delta L} - \overline{V}\right)^2 + \left(\overline{V}_{\Delta W} - \overline{V}\right)^2 + \left(\overline{V}_{\Delta H} - \overline{V}\right)^2\right]^{1/2}$$

$$= \left\{\left[\left(\overline{L} + \frac{T_L}{3}\right)\overline{W}\overline{H} - \overline{V}\right]^2 + \left[\left(\overline{W} + \frac{T_W}{3}\right)\overline{L}\overline{H} - \overline{V}\right]^2 + \left[\left(\overline{H} + \frac{T_H}{3}\right)\overline{L}\overline{W} - \overline{V}\right]^2\right\}^{1/2}$$

$$\sigma_L = \frac{T_L}{3} = \frac{1}{3} = 0.33 \text{ ft}$$

$$\sigma_W = \frac{T_W}{3} = \frac{5}{3} = 0.17 \text{ ft}$$

$$\sigma_H = \frac{T_H}{3} = \frac{0.1}{3} = 0.03 \text{ ft}$$

$$\sigma_V = \left\{\left[(10 + 0.33)(5)(2) - 100\right]^2 + \left[(5 + 0.17)(10)(2) - 100\right]^2 + \left[(2 + 0.03)(10)(5) - 100\right]^2\right\}^{1/2}$$

$$= \left[(100.3 - 100)^2 + (103.4 - 100)^2 + (101.5 - 100)^2\right]$$

$$= (10.89 + 11.56 + 2.25)^{1/2} = \sqrt{25} = 5$$

$$V = \overline{V} \pm 3\sigma_V = 100 \pm 15 \text{ ft}^3$$

Checking the validity gives

$$C_V = \frac{\sigma_V}{V} = \frac{5}{100} \times 10^2 = 5 \text{ percent}$$

which is less than 10 percent. This solution is a better estimate of the effects of tolerance on volume. Note also that various values can now be estimated for different types of problems regarding this volume because it has been represented as a normal distribution function.

Using the partial derivative method, again

$$V_{\pm} = \overline{V} \pm 3\sigma_V$$

where

$$\sigma_V = \left[\left(\frac{\partial V}{\partial x_1}\right)^2 \sigma_{x_1}^2 + \ldots + \left(\frac{\partial V}{\partial x_n}\right)^2 \sigma_{x_n}^2\right]^{1/2}$$

$$V = LWH, \quad \frac{\partial V}{\partial L} = WH, \quad \frac{\partial V}{\partial W} = LH, \quad \frac{\partial V}{\partial H} = LW$$

$$\sigma_L = 0.33 \text{ ft}, \quad \sigma_W = 0.17 \text{ ft}, \quad \sigma_H = 0.03 \text{ ft}$$

$$\sigma_V = \left[(WH)_L^2 \sigma_L^2 + (LH)_W^2 \sigma_W^2 + (LW)_H^2 \sigma_H^2\right]^{1/2}$$

$$= \left[(5 \times 2)^2 (0.33)^2 + (10 \times 2)^2 (0.17)^2 + (10 \times 5)^2 (0.03)^2\right]^{1/2}$$

$$= (10.9 + 11.6 + 2.25)^{1/2} = \sqrt{25} = 5$$

$$V = 100 \pm 15 \text{ ft}^3$$

This method is more work and gives the same results as the perturbation method. Because the $C_V = 5$ percent, which is less than 10 percent, the method would be suitable to use.

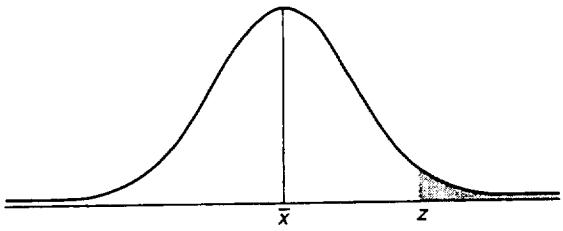
Concluding Remarks

Now that you have completed chapter 5, you should have a clear understanding of the following concepts:

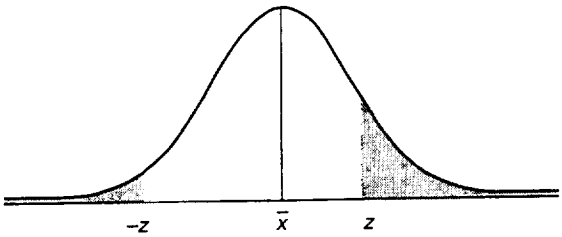
(1) A probability density function $p(x)$ for a random variable describes the probability that the variable will take on a certain range of values.

(2) The area under the density function is equal to unity, which means that the probability is 1 that the variable will be within the interval described by the density function. For example, the normal distribution describes the interval from $-\infty$ to ∞.

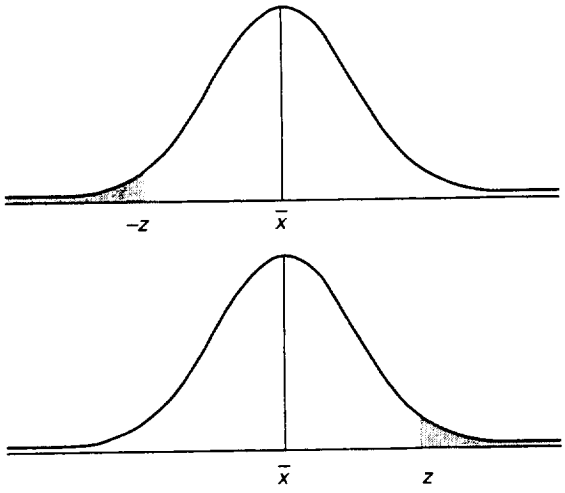
(3) Associated with each probability density function is a cumulative probability distribution $F(x)$ that represents the cumulative sum of the areas under the density function.



(a) Symmetrical two-limit problems, which are concerned with the probability of a variable taking on values within equal distances from both sides of the mean.



(b) Nonsymmetrical two-limit problems, which are similar to (a) but within unequal distances from both sides of the mean of the density function.



(c) One-limit problems, which are concerned with the probability of a variable taking on values above or below some limit represented by some distance from the mean of the density function.

(4) The normal distribution (also called the bell curve, the Gaussian distribution, and the normal curve of error) is a probability density function. Using the normal distribution, you should be able to solve the following types of problems:

(5) You should be able to take data measurements of a certain device and calculate the mean of the data given by

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

and the standard deviation of the data given by

$$\sigma = \left[\sum_{i=1}^{n} \frac{(x_i - \bar{x})^2}{n-1}\right]^{1/2}$$

and

$$z = \frac{x_i - \bar{x}}{\sigma}$$

Using the data mean and standard deviation, you should then be able to estimate the probability of failures occurring when more of the same devices are tested or operated.

(6) The worst-case method can be used on any problem:
- (a) Limits will be defined.
- (b) No estimates can be made from the population distribution.

(7) The root-sum-square method only applies to algebraic variables that are additive.

(8) The perturbation or partial derivative methods are only valid if the coefficient of variation is 10 percent or less.

References

5–1. Croxton, F.E.: Tables of Areas in Two Tails and in One Tail of the Normal Curve. Prentice-Hall Inc., 1949.

5–2. Hald, A.: Tables of the Cumulative Normal Distribution. John Wiley & Sons, Inc., 1949.

5–3. Failure Distribution Analyses Study. Vols. 1, 2, and 3, Computer Applications Inc., New York, 1964. (Avail. NTIS, AD–631525, AD–631526, AD–631527.)

5–4. Hoel, P.G: Elementary Statistics. John Wiley & Sons, 1960.

5–5. Berretonni, J.N: Practical Applications of the Weibull Distribution, Industrial Quality Control, vol. 21, no. 2, Aug. 1964, pp. 71–79.

5–6. Reliability Prediction of Electronic Equipment, MIL–HDBK–217E, Jan. 1990.

5–7. Electronic Reliability Design Handbook, MIL–HDBK–338, vols. 1 and 2, Oct. 1988.

5–8. Reliability Modeling and Prediction, MIL–STD–756B, Aug. 1982.

Reliability Training[1]

1. A unit is required to operate at 100 °F. If tests show the mean strength of the data for the unit is 123 °F and the standard deviation is 9 °F, what is the probability that the unit will operate successfully; that is, $P(x \geq 100\ °F)$?

   A. 0.5234 B. 0.2523 C. 0.9946 D. 0.9995

2. A pressure vessel (including a factor of safety) has an upper operating limit of 8000 psi. Burst tests show a mean strength of 9850 psi and a standard deviation of 440 psi. What is the probability of pressure vessel failure; that is, $P(x \leq 8000\ \text{psi})$?[2]

   A. 0.0^4267 B. 0.0^4133 C. 0.0^4317

3. A memory drum is required to reach sink speed and stabilize in 15.5 sec at 125 °F. Five drums are tested with these stabilizing time results: 13.2, 12.3, 14.8, 10.3, and 12.9 sec.

   a. What is the mean stabilizing time?

   A. 13.1 B. 10.7 C. 12.7

   b. What is the standard deviation?

   A. 1.63 B. 1.45 C. 1.32

   c. What is the estimated percentage of drums out of specification; that is, $P(x > 15.5\ \text{sec})$?

   A. 6.7 B. 8.5 C. 4.3

4. A pyrotechnic gyro has an uncaging time requirement of 142 ± 20 msec. Six gyros were tested resulting in these uncaging times: 123, 153, 140, 129, 132, and 146 msec.

   a. What is the mean uncaging time?

   A. 133.2 msec B. 135.2 msec C 137.2 msec

   b. What is the standard deviation?

   A. 10.2 B. 11.2 C. 11.9

   c. What is the estimated percentage of gyros within specification; that is, $P(122 \leq x \leq 162\ \text{msec})$?

   A. 89.8 B. 96.8 C. 82.6

5. A hydraulic pressure line was designed to the following stresses:

   (a) Maximum operating pressure (actual), 1500 psi
   (b) Design pressure (10-percent safety factor), 1650 psi

   Tests of the pressure line indicated a mean failure pressure of 1725 psi and a standard deviation of 45 psi.

   a. What is the reliability of the line when the design pressure limits are considered?

   A. 0.10 B. 0.90 C. 0.95

[1]Answers are given at the end of this manual.

[2]The superscripted numbers in the answers are shorthand for 2.67×10^{-6}.

b. What is the reliability of the line when the maximum operating pressure is considered?

A. 0.99 B. 0.90 C. 0.80

6. A communications network requires a 1300-msec watchdog delay after initiation. A sample of 10 delays was tested from a rack of 100 delays. The time delays of the circuits are as shown:

Circuit number	Delay, msec
1	1250
2	1400
3	1700
4	1435
5	1100
6	1565
7	1485
8	1385
9	1350
10	1400

a. What is the average (mean) delay time?

A. 1386 msec B. 1400 msec C. 1407 msec

b. What is the standard deviation?

A. 52.7 B. 87.1 C. 163.4

c. On the basis of this sample, what percentage of the 100 circuits will meet specifications (1300-msec or greater delay)?

A. 75 B. 80 C. 90

7. A circuit contains four elements in series. Their equivalent resistance values are

Element	Nominal resistance, R, ohm	Tolerance,[a] T, percent
A	100	±10
B	20	±1
C	10	±5
D	10	±5

[a]Where $\pm T = \pm 3\sigma$.

a. What is the nominal or mean total resistance $\overline{R}_T$?

A. 120 Ω B. 140 Ω C. 160 Ω

b. What are the worst-case R values (upper number, maximum; lower number, minimum)?

A. 131.6 Ω
118.7 Ω

B. 176.3 Ω
146.2 Ω

C. 151.2 Ω
128.8 Ω

c. Using the root-sum-square method, what is the probability that $\overline{R}_T \geq 135\ \Omega$?

A. 0.905 B. 0.962 C. 0.933

d. Using the perturbation method, what is the probability that $\overline{R}_T \geq 135\ \Omega$?

A. 0.905 B. 0.962 C. 0.933

8. Given power (watts) = I^2R, where $I = 0.5\ A$, $T_I = \pm 5$ percent, $R = 100\ \Omega$, and $T_R = \pm 10$ percent. (Note: $\pm T = \pm 3\sigma$.)

a. What is the nominal or mean power output $\overline{P}$?

A. 25 W B. 20 W C. 30 W

b. What are the worse-case $\overline{P}$ values (upper number, maximum; lower number, minimum)?

A.	B.	C.
26.6 W	35.2 W	30.3 W
18.2 W	22.6 W	20.3 W

c. Using the perturbation method, what is the probability that $(23.5 \leq \overline{P} \leq 26.5)$?

A. 0.94 B. 0.80 C. 0.86

d. What is the C_V (in percent) for the perturbation method used in question 8c?

A. 12 B. 8 C. 4.6

e. Is the root-sum-square method valid for solving the probability problem 8c?

A. Yes B. No

f. Using the partial derivative method, what is the probability that $23.5 \leq \overline{P} \leq 26.5$?

A. 0.942 B. 0.803 C. 0.857

Chapter 6
Testing for Reliability

In chapters 3 and 4, we discussed the methods used to predict the probability that random catastrophic part failures would occur in given products and systems. These analytical techniques are well established (ref. 6–1). Yet, we should keep in mind that they are practical only when adequate experimental data are available in the form of part failure rates. In other words, their validity is predicated on great amounts of empirical information.

Such is not the case when we undertake similar analyses to determine the influence of tolerance and wearout failures on the reliability of a product. An understanding of these failure modes depends on experimental data in the form of probability density functions such as those discussed in chapter 5. In general, such data are unavailable on items at the part or system level; this kind of information must be developed empirically through reliability test methods.

Chapter 6 reviews and expands the terms used in the reliability expression given in chapter 2 and then shows how the terms can be demonstrated or assessed through the application of attribute test, test-to-failure, and life test methods (ref. 6–2).

Demonstrating Reliability

Recall from chapter 2 that one way to define product reliability is as the probability that one or more failure modes will not be manifested (ref. 6–3). This can be written as

$$R = P_c P_t P_w \left(K_q K_m K_r K_\ell K_u\right)$$

where

P_c probability that catastrophic part failures will not occur
P_t probability that out-of-tolerance failures will not occur
P_w probability that wearout failures will not occur
K_q probability that quality test methods and acceptance criteria will not degrade inherent reliability
K_m probability that manufacturing processes, fabrication, and assembly techniques will not degrade inherent reliability
K_r probability that reliability engineering activities will not degrade inherent reliability
K_ℓ probability that logistics activities will not degrade inherent reliability
K_u probability that user or customer will not degrade inherent reliability

The term $P_c P_t P_w$ denotes inherent reliability R_i; $(K_q K_m K_r K_\ell K_u)$ are factors that affect the probability of the three modes of failure occurring during hardware manufacture and use rather than occurring from unreliable hardware design.

First, we illustrate how the empirical value of these terms affects product reliability. Then, we discuss the particular test methods used to develop these values. Assume that a device was designed with a reliability requirement of 0.996. This means that only 4 out of 1000 such devices can fail. The device contains 1000 parts, it has a function to perform within a tolerance of $X \pm 2$ percent, and it must operate for a mission cycle of 1000 hours at 50 °C.

P_c Illustrated

If we know the number and types of parts in the device plus the applied stresses and part failure rates used in the exponential distribution, $e^{-t\left(\sum \lambda\right)}$ we can estimate the probability that no catastrophic part failure will occur during the mission cycle. Assuming, for example, that our estimate is $P_c = 0.999$ (i.e., one device in 1000 will incur a catastrophic part failure during the mission cycle), the product reliability of the device becomes

$$R = P_c P_t P_w (K\text{ - factors}) = e^{-t\left(\sum \lambda\right)} P_t P_w (K\text{ - factors})$$
$$= 0.999 P_t P_w (K\text{ - factors})$$

P_t Illustrated

Suppose that we now test one of the devices at 50 °C. If the functional output is greater than the specified tolerance of $X \pm 2$ percent, the reliability of that particular device is zero. It is zero because P_t is zero (i.e., $R = (0.999)(0)P_w(K\text{-factors}) = 0$). We can say, however, that the device will continue to operate in an out-of-tolerance condition with a probability of no catastrophic failures equal to 0.999 just as we predicted. To understand this, recall that part failure rates reflect only the electrical, mechanical, and environmental stresses applied to the individual parts. For this reason, a prediction on the basis of such data will neglect to indicate that (1) the parts have been connected to obtain a specified function, (2) a tolerance analysis of the function has been performed, or (3) the parts are packaged correctly. In other words, P_c represents only how well the individual parts will operate, not how well the combined parts will perform.

If nine more of the devices are tested at 50 °C with all the output functions remaining within the $X \pm 2$ percent tolerance, P_t becomes 9/10 = 0.9 and the reliability of the device $R = (0.999)(0.9)P_w(K\text{-factors})$. Because the reliability requirement of the device is 0.996, it should be clear that P_t must be greater than 0.996. Let us assume then that 1000 devices are tested at 147 °F with only one tolerance failure, which produces an observed $P_t = 999/1000 = 0.999$. The reliability of the device is now

$$R = (0.999)(0.999)P_w(K\text{ - factors}) = 0.998P_w(K\text{ - factors})$$

Note that, because operating time is accumulated during original functional testing, it is possible for random catastrophic part failures to occur. Remember, however, that this type of failure is represented by P_c and not P_t.

P_w Illustrated

Now let us take another operating device and see whether wearout failures will occur within the 1000-hour mission cycle. If, as run time is accumulated, a faulty function output or catastrophic failure is caused by a *wear mechanism,* the reliability of the device again becomes zero. It is zero because P_w is zero as shown in the equation

$$R = (0.999)(0.999)(0)(K\text{ - factors}) = 0$$

Note the emphasis on the words "wear mechanism." Because it is possible to experience random catastrophic part failures and even out-of-tolerance conditions during a test for wearout, it is absolutely necessary to perform physics-of-failure analyses. This is essential to ascertain if the failures are caused by true physical wear before including them in the P_w assessment.

So far, the first two terms, P_c and P_t, combine to yield a probability of (0.999)(0.999) = 0.998. As a result, the remaining terms, P_w (K-factors), must be no less than 0.998 if the 0.996 device requirement is to be satisfied. Therefore, we assume that we have demonstrated a P_w of 0.999, which reduces the device reliability to

$$R = P_cP_tP_w(K\text{ - factors}) = (0.999)(0.999)(0.999)(K\text{ - factors})$$
$$= 0.997(K\text{ - factors})$$

K-Factors Illustrated

Since testing obviously must be conducted on real hardware, the K-factors as well as the P terms of reliability are present in every test sample. Establishing values for the K-factors requires that all failures observed during a test be subjected to physics-of-failure analyses to identify specific failure mechanisms. Actually, the action taken to prevent the recurrence of an observed failure mechanism determines the factor that caused the failure. A failure that can be prevented by additional screening tests as part of the quality acceptance criteria is charged to the K_q factor; one that requires additional control over some manufacturing process is charged to the K_m factor, and so on. Failures that require changes in documentation, design, and tolerance would be charged to the P_c, P_t, or P_w terms as applicable.

The least important aspect of testing is the ability to charge an organization or function with responsibility for a failure. More important is the need to prevent observed failures from recurring. This requires that corrective action be made a recognized part of each reliability test program.

Getting back to the illustration, we assume that one failure out of 1000 devices was caused by one of the K-factors even though it could have been observed during a P_c, P_t, or P_w failure evaluation. This reduces the reliability of the device to

$$R = P_cP_tP_w(K\text{ - factors}) = (0.999)(0.999)(0.999)(0.999)$$
$$= 0.996$$

which indicates that the device met its requirement.

Test Objectives and Methods

The purpose of the preceding illustration was to provide a better understanding of (1) how the P terms and the K-factors relate to physical hardware and (2) the techniques for demonstrating the terms through testing. Table 6–1 shows the suggested test methods. We say "suggested" because any of the test methods can be used if certain conditions are met (ref. 6–4). These conditions are pointed out as each method is discussed. Table 6–1 indicates the most efficient methods by assigning

TABLE 6–1.—TEST METHOD PRIORITIES FOR DEMONSTRATING RELIABILITY

Reliability term	Suggested test method		
	Attribute tests	Tests to failure	Life tests
P_c	2	3	1
P_t	3	1	2
P_w	3	2	1
K-factors	3	1	2

priority numbers from 1 to 3 (with 1 being the most efficient and 3 the least).

Test Objectives

At least 1000 test samples (attribute tests) are required to demonstrate a reliability requirement of 0.999. Because of cost and time, this approach is impractical. Furthermore, the total production of a product often may not even approach 1000 items. Because we usually cannot test the total production of a product (called product population), we must demonstrate reliability on a few samples. Thus, the main objective of a reliability test is to test an available device so that the data will allow a statistical conclusion to be reached about the reliability of similar devices that will not or cannot be tested. That is, the main objective of a reliability test is not only to evaluate the specific items tested but also to provide a sound basis for predicting the reliability of similar items that will not be tested and that often have not yet been manufactured.

As stated, to know how reliable a product is, one must know how many ways it can fail and the types and magnitudes of the stresses that produce such failures. This premise leads to a secondary objective of a reliability test: to produce failures in the product so that the types and magnitudes of the stresses causing such failures can be identified. Reliability tests that result in no failures provide some measure of reliability but little information about the population failure mechanisms of like devices. (The exceptions to this are not dealt with at this time.)

In subsequent sections, we discuss statistical confidence attribute test, test-to-failure, and life test methods, explain how well these methods meet the two test objectives, show how the test results can be statistically analyzed, and introduce the subject and use of confidence limits.

Attribute Test Methods

Qualification, preflight certification, and design verification tests are categorized as attribute tests (refs. 6–5 and 6–6). They are usually go/no-go and demonstrate that a device is good or bad without showing how good or how bad. In a typical test, two samples are subjected to a selected level of environmental stress, usually the maximum anticipated operational limit. If both samples pass, the device is considered qualified, preflight certified, or verified for use in the particular environment involved (refs. 6–7 and 6–8). Occasionally, such tests are called tests to success because the true objective is to have the device pass the test.

An attribute test is usually not a satisfactory method of testing for reliability because it can only identify gross design and manufacturing problems. It can be used for reliability testing only when a sufficient number of samples are tested to establish an acceptable level of statistical confidence.

Statistical Confidence

The statistical confidence level is the probability that the corresponding confidence interval covers the true (but unknown) value of a population parameter. Such a confidence interval is often used as a measure of uncertainty about estimates of population parameters. In other words, rather than express statistical estimates as point estimates, it is much more meaningful to express them as a range (or interval), with an associated probability (or confidence) that the true value lies within such an interval.

It should be noted however, that statistical confidence intervals can be difficult to evaluate (see also refs. 6–4 and 6–9). For simple distributions in reliability, intervals and levels are calculated in a straightforward manner. For more complicated or multiparameter distributions, especially where parameter estimates are not statistically independent, such intervals and levels can be very difficult to calculate.

To illustrate further the limitations of attribute test methods, we apply statistics to the test results. Figure A–4(a) in appendix A shows on the ordinate the number of events (successes) necessary to demonstrate a reliability value (abscissa) for various confidence levels (family of curves) when no failures are observed. Figures A–4(b) to (f) provide the same information when one to five failures are observed.

From the results of two devices tested with no failures, figure A–4(a) shows that we can state with 50-percent confidence that the population reliability of such devices is no less than 71 percent. Fifty-percent confidence means that there is a 50-percent chance that we are wrong and that the reliability of similar untested devices will actually be less than 71 percent. Similarly, we can also state from the same figure that we are 60 percent confident that the reliability of all such devices is 63 percent. But either way, the probability of success is less than encouraging.

To gain a better understanding of figure A–4 and the theory behind it, let us stop for a moment and see how confidence levels are calculated. Recall from chapter 2 that the combination of events that might result from a test of two devices was given by

$$R^2 + 2RQ + Q^2 = 1$$

where

R^2 probability that both devices will pass
$2RQ$ probability that one device will pass and one will fail
Q^2 probability that both devices will fail

In the power supply example, we observed the first event R^2 because both supplies passed the test. If we assume a 50-percent probability that both will pass, we can set $R^2 = 0.50$ and solve for the reliability of the device as follows:

$$R^2 = 0.50$$

$$R = \sqrt{0.50} = 0.71$$

We then can say with 50-percent confidence that the population reliability of the device is no less than 0.71. By assuming a 50-percent chance, we are willing to accept a 50-percent risk of being wrong, hence the term "50 percent confident." If we want only to take a 40-percent risk of being wrong, we can again solve for R from

$$R^2 = 0.40$$

$$R = \sqrt{0.40} = 0.63$$

In this case, we can be 60 percent confident that the population reliability of the devices is no less than 0.63.

Selection of the confidence level is a customer's or engineer's choice and depends on the amount of risk he is willing to take on being wrong about the reliability of the device. The customer usually specifies the risk he is willing to take in conjunction with the system reliability requirement. As higher confidence levels (lower risk) are chosen, the lower the reliability estimate will be. For example, if we want to make a 90-percent confidence (10-percent risk) statement based on the results of the test to success of two devices, we simply solve

$$R^2 = (1 - \text{Confidence level}) = 1 - 0.90 = 0.10$$

so that

$$R = \sqrt{0.10} = 0.316$$

Table 6–2 illustrates how the reliability lower bound changes with various confidence levels. The curves in figure A–4 are developed in a similar manner. In figure A–4(b), which is used when one failure is observed, for 10 samples tested with one observed failure, the statistically predicted or demonstrated reliability at 90-percent confidence is 0.66. This answer is found by solving

TABLE 6–2.—RELIABILITY AND CONFIDENCE LEVEL FOR TWO-SAMPLE ATTRIBUTE TEST WITH NO FAILURES

Confidence level, percent	Reliability, R	Risk, percent
10	0.95	90
50	.71	50
60	.63	40
70	.55	30
80	.45	20
90	.32	10
99	.10	1

$$R^{10} + 10R^9Q = 1 - 0.90$$

$$R = 0.663$$

which agrees with the figure to two places.

Application.—The discussion thus far has underscored the shortcomings of attribute tests when sample sizes are small. Tests involving only two or three samples may reveal gross errors in hardware design or manufacturing processes, but when relied upon for anything more, the conclusions become risky (refs. 6–7 and 6–8).

Attribute tests can be useful in testing for reliability when a sufficient sample size is used. For example, 10 samples tested without failure statistically demonstrate a population reliability of 0.79 at 90-percent confidence; 100 tests without failure demonstrate a population reliability of 0.976 at 90-percent confidence. To understand better the application of attribute tests and the use of figure A–4, consider the following examples:

Example 1: During the flight testing of 50 missiles, five failures are observed. What confidence do we have that the missile is 80 percent reliable?

Solution 1: From figure A–4(f) the answer is read directly to be a 95-percent confidence level. The a posteriori reliability of these 50 missiles, or that derived from the observed facts, is still 45/50 = 90 percent. Thus, future flights will be at least 80 percent reliable with a 5-percent risk of being wrong.

Example 2: An explosive switch has a reliability requirement of 0.98. How many switches must be fired without a failure to demonstrate this reliability at 80-percent confidence?

Solution 2: From figure A–4(a), the answer is read directly as 80 switches.

Example 3: A test report states that the reliability of a device was estimated to be 0.992 at 95-percent confidence based on a test of 1000 samples. How many failures were observed?

Solution 3: In figure A–4(d), the 95-percent confidence curve crosses the 1000-event line at R = 0.992. Therefore, three failures were observed.

In these examples, the population reliability estimates may represent any of the P terms or the K-factors in the expression for product reliability, depending on the definition of failure used to judge the test results. For a device that is judged only on its capability to remain within certain tolerances, the reliability would be the P_t term. Had catastrophic failures been included, we would have demonstrated the P_c and P_t terms. In general, attribute tests include all failure modes as part of the failure definition and, consequently, the associated reliability is product reliability with both the P terms and the K-factors included.

Attribute test/safety margin slide rule.—A special-purpose slide rule that was developed to facilitate determining attribute test/safety margin confidence levels will be available in class for these exercises. (See the back of this manual for the slide rule and the instructions to assemble it.)

Examples 4 (confidence level for attribute test): Attribute tests are tests to success. The objective is for a selected number of samples, called tests on the slide rule, to operate successfully at some predetermined stress level. Some tests, however, may fail. This slide rule handles combinations of up to 1000 tests and up to 500 failures. The answer is a direct population reliability reading of the untested population at a selected confidence level. Six confidence levels from 50 to 90 percent are available. (The statistical basis for this rule is the χ^2 approximation of binomial distribution.)

Example 4a: Fifteen items are tested with one failure observed. What is the population reliability at 70-percent confidence level?

Solution 4a: Set one failure on the movable slide above the 70-percent confidence level index. Read from TOTAL NUMBER OF TESTS the tests for a population reliability of 0.85 at 70-percent confidence level. By setting one failure at successive levels of confidence this example gives these population reliabilities: 0.710 at 95-percent confidence level, 0.758 at 90 percent, 0.815 at 80 percent, 0.873 at 60 percent, and 0.895 at 50 percent.

Example 4b: A population reliability of 0.9 at 95-percent confidence level is desired. How many tests are required to demonstrate this condition?

Solution 4b: Set zero failures at the 95-percent confidence level index. From TOTAL NUMBER OF TESTS read 29 tests directly above 0.90 population reliability. Therefore, 29 tests without failure will demonstrate this combination. If, however, one failure occurs, set one failure at 95 percent. Then 46 others must pass the test successfully. Progressively more observed failures such as 10 (set of 10 at 95 percent) require 170 successes (160 + 10).

Examples 5 (confidence level for safety margins): Safety margin S_M indicates the number of standard deviations σ_M between some preselected reliability boundary R_b and the mean of the measured sample failure distribution. Thus, $S_M = (\bar{X}_M - R_b) \div \sigma_M$, where $\bar{X}_M$ and σ_M are the measured mean and standard deviation of the samples under test. The larger the sample size, the more nearly the measured S_M approaches the safety margin of the untested population S_D. This rule equates S_M for six levels of confidence for sample sizes N between 5 and 80. (Statistical basis for this rule: noncentral t distribution.)

Example 5a: Ten items are tested to failure with an observed or measured S_M of 5.8. What is the lower expected safety margin of the untested population at 90-percent confidence?

Solution 5a: Set 5.8 on the movable slide at the top window for the S_M value. Under $N = 10$ on the 90-percent window, read $S_D \geq 3.9$. Without moving the slide, for successive levels of confidence, 4.45 at 80 percent, 4.85 at 70 percent, 5.21 at 60 percent, and 5.57 at 50 percent.

Example 5b: *Six* samples are available for test. What S_M is required to demonstrate a population safety margin of 4.0 or greater at 90-percent confidence level?

Solution 5b: Using the 90-percent window, set $S_D = 4.0$ opposite $N = 6$. At S_M read 7.1. Therefore, test results of 7.1 or greater will demonstrate $S_D \geq 4.0$ at a 90-percent confidence level. If 25 samples are available for test, set $S_D = 4.0$ opposite $N = 25$ on the 90-percent window. An S_M of only 5.0 or greater would demonstrate 4.0 or greater safety margin at 90-percent confidence.

Sneak circuits.—During attribute testing, the flight hardware may sometimes not work properly because of a sneak circuit. A sneak circuit is defined for both hardware and software as follows (ref. 6–10):

(1) Hardware: a latent condition inherent to the system design and independent of component failure that inhibits a desired function or initiates an undesired function (path, timing, indication, label)

(2) Software: an unplanned event with no apparent cause-and-effect relationship that is not dependent on hardware failure and is not detected during a simulated system test (path, timing, indication, label)

Each sneak circuit problem should be analyzed, a cause determined, and corrective action implemented and verified. References 6–10 to 6–12 give a number of examples of how this can be done:

(1) Reluctant Redstone—making complex circuitry simple
(2) F–4 example
(3) Trim motor example
(4) Software example

A few minutes spent with one of these references should solve any sneak circuit problem.

Attribute test summary.—In summary, four concepts should be kept in mind:

(1) An attribute test, when conducted with only a few samples, is not a satisfactory method of testing for reliability, but it can identify gross design and manufacturing problems.

(2) An attribute test is an adequate method of testing for reliability only when sufficient samples are tested to establish an acceptable level of statistical confidence.

(3) Some situations dictate attribute tests or no tests at all (e.g., limited availability or the high cost of samples, limited time for testing, test levels that exceed the limits of test equipment, and the need to use the test samples after testing).

(4) Confidence, a statistical term that depends on supporting statistical data, reflects the amount of risk we are willing to take when stating the reliability of a product.

Test-To-Failure Methods

The purpose of the test-to-failure method is to develop a failure distribution for a product under one or more types of stress. Here, testing continues until the unit under test ceases to function within specified limits. Alternatively, test to failure may be accomplished by increasing electrical load or mechanical load until a failure is induced. The results are used to calculate the probability of the failure of the device for each load. In this case, the failures are usually tolerance or physical wearout. The test-to-failure method is also valuable because we can determine the "spread" or standard deviation of the loads that cause failure (or the spread of the times to failure, etc.). This spread has a significant effect on the overall reliability.

In this discussion of test-to-failure methods, the term safety factor S_F is included because it is often confused with safety margin S_M. Safety factor is widely used in industry to describe the assurance against failure that is built into structural products. Safety factor S_F can be defined as

$$S_F = \frac{\overline{X}_{\text{avg}_S}}{R_b}$$

where

$\overline{X}_{\text{avg}_s}$ mean strength of material

R_b reliability boundary, the maximum anticipated operating stress level the component receives

We choose to define "safety margin" S_M by taking into account the standard deviation or the spread of the data; hence, S_M is the number of standard deviations of the strength distribution that lie between the reliability boundary R_b and the mean strength $\overline{X}_{\text{avg}_s}$:

$$S_M = \frac{\left(\left|R_b - \overline{X}_{\text{avg}_s}\right|\right)}{\sigma_s}$$

where σ_s is the standard deviation of the strength distribution.

Using S_F presents little risk when we deal with materials with clearly defined, repeatable, and "tight" strength distributions, such as sheet and structural steel or aluminum. However, when we deal with plastics, fiberglass, and other metal substitutes or processes with wide variations in strength or repeatability, using S_M provides a clearer picture of what is happening. In most cases, we must know the safety margin to understand how useful the safety factor is.

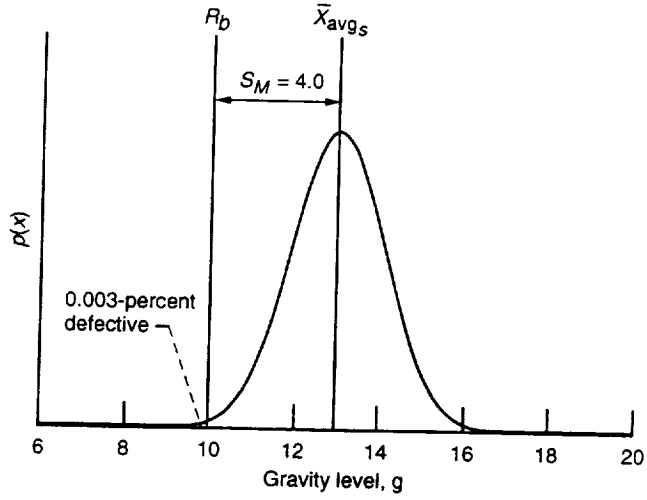


Figure 6–1.—Test-to-failure method applied to metallic structure. Mean strength of material, $\overline{X}_{\text{avg}_S}$, 13; reliability boundary, R_b, 10; standard deviation, s_S, 0.75; safety factor, S_F, 13/10 or 1.3; safety margin, S_M, (|10–13|)/0.75 or 4.0; probability of defect, 0.00003 or 0.003 percent.

Consider the example of the design of a support structure to hold cargo in a launch vehicle. The component strength is expressed and represented by its ability to withstand a particular g force. Structural members (consisting of various materials) are tested with a mechanical load until failure occurs.

We may have materials with clearly defined, repeatable, and tight strength distributions, such as sheet and structural steel or aluminum. Here, using S_F presents little risk (see fig. 6–1 for metallic structure where a normal (Gaussian) distribution is assumed). Alternatively, we may have plastics, fiberglass, and other metal substitutes or processes with wide variations in strength or repeatability and using S_M provides a clearer picture of a potential problem (see fig. 6–2 for a metal substitute, a composite).

To use and benefit from this concept we need to

(1) Know the material strengths and distributions
(2) Identify the reliability boundary R_b for the loading of the material
(3) Know the safety margin to understand the usefulness of the safety factor

Using safety margins in this way in the design process has a major benefit because they provide a clearer picture of what is happening in the real world by taking strength distributions into account. Also, the difference in the probability of defects (calculated by solving for the area under the normal distribution curve to the left of R_b) is better reflected in the difference in the strength margins.

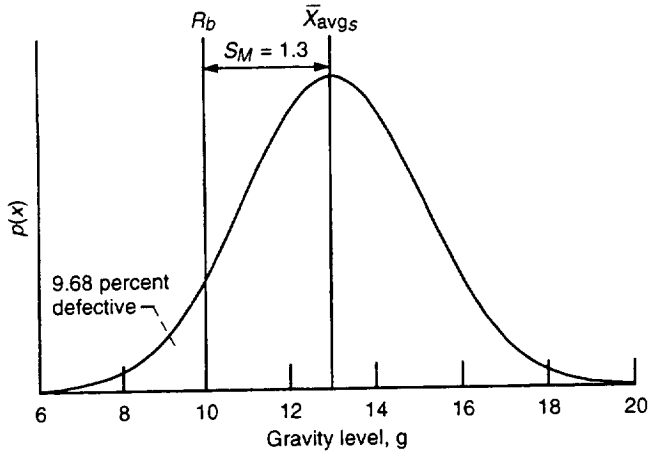


Figure 6–2.—Test-to-failure method applied to metal substitute (composite). Mean strength of material, $\bar{X}_{avg_S}$, 13; reliability boundary, R_b, 10; standard deviation, σ_S, 2.308; safety factor, S_F, 13/10 or 1.3; safety margin, S_M, (|10–13|)/2.308 or 1.3; probability of defect, 0.0968 or 9.68 percent.

In summary, test-to-failure methods can be used to develop a strength distribution that provides a good estimate of tolerance and physical wearout problems without the need for the large samples required for attribute tests (note that extrapolation outside the range of data should be avoided). The results of a test-to-failure exposure of a device can be used to predict the reliability of similar devices that cannot or will not be tested. Testing to failure also provides the means for evaluating the failure modes and mechanisms of devices so that improvements can be made. It was also shown that a safety factor is much more useful if the associated safety margin is known.

Test procedure and sample size.—Devices that are not automatically destroyed upon being operated are normally not expended or destroyed during a functional test. Electronic equipment usually falls into this category. For such equipment, a minimum sample size of five is necessary, each sample being subjected to increasing stress levels until failure occurs or the limits of the testing facility are reached. In the latter case, no safety margin calculation is possible because no failures are observed. Here, we must rely on intuition when deciding the acceptability of the device.

Test-to-failure procedure and sample size requirements for one-shot devices are different because a one-shot device is normally expended or destroyed during a functional test. Ordinance items such as squib switches fall into this category. For such devices, at least 20 samples should be tested, but 30 to 70 would be more desirable. At least 12 failures should be observed during a test. In a typical one-shot test, of which there are many variations, a sample is tested at the reliability boundary and, if it passes, a new sample is tested at predetermined stress increments until a failure occurs. Then, the next sample is tested at one stress increment below the last failure. If this sample passes, the stress is increased one increment for the next sample. This process, depicted in figure 6–3, continues until at least 12 failures have been observed.

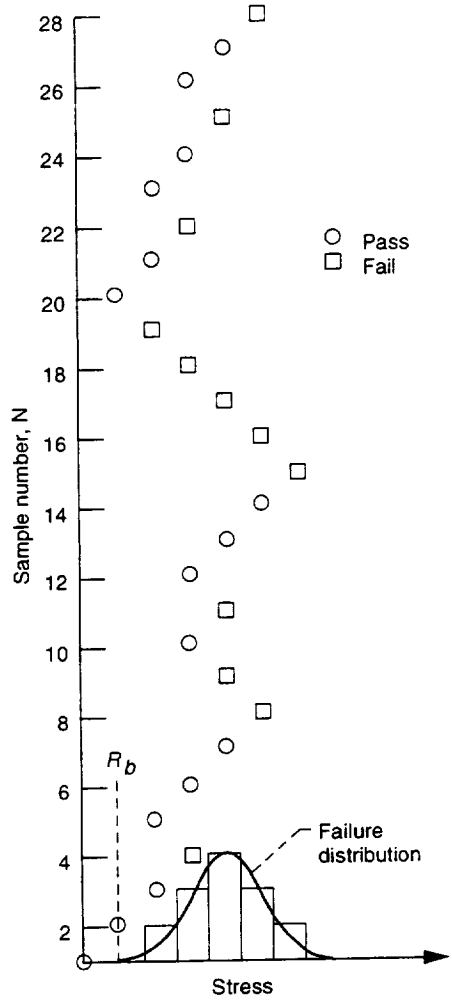


Figure 6–3.—Example of one-shot test-to-failure procedure.

Safety margins for single failure modes.—For devices that exhibit a single failure mode during a test-to-failure exposure, the safety margin and the reliability are calculated by the technique just discussed in the definition of safety margin. The following examples further illustrate the method and show the practical results.

Example 6: A test was conducted on a vendor's 0.25- and 0.50-W film resistors to evaluate their ability to operate reliably at their rated power levels. Thirty samples of each type were tested by increasing the power dissipation until the resistance change exceeded 5 percent. The results are shown in figure 6–4, from which the following points are noteworthy:

(1) The mean strength of the 0.25-W resistor was less than half the mean strength of the 0.50-W resistor: $\bar{x}_{0.25} = 1.19$ W compared with $\bar{x}_{0.50} = 2.6$ W. This was to be expected since the 0.50-W resistor was larger, had more volume, and could dissipate more energy.

(2) The standard deviation of the 0.25-W resistor was almost the same as that for the 0.50-W resistor: $\sigma_{0.25} = 0.272$ W; $\sigma_{0.50} = 0.332$ W. This was also expected because both resistors were made by the same manufacturer and were subjected to the same process controls and quality acceptance criteria.

(3) The 0.50-W resistor, because of its higher mean strength, had a safety margin of 6.32 with reference to its rated power dissipation of 0.50 W. According to table 5–5, this means that only 0.0^9149 resistors would exceed a 5-percent resistance change when applied at 0.50 W. The 0.25-W resistor, because of its lower mean strength, had a safety margin of only 3.45 with reference to its rated power of 0.25 W. According to table 5–5 again, this means that 0.0^3337 resistors would exceed a 5-percent resistance change when applied at 0.25 W. Derating the 0.25 W to 0.125 W increased the safety margin to 3.92 and decreased the expected number of failures to 0.0^4481, an improvement factor of 7.5. This, of course, is the reason for aerating components, as discussed in chapter 4. Although we have indicated that a safety margin of 6.32 has statistical meaning, in practice a population safety margin of 5 or higher indicates that the applicable failure mode will not occur unless, of course, the strength distribution deviates greatly from a normal distribution.

Example 7: A fiberglass material to be used for a flame shield was required to have a flexural strength of 15 000 psi. The results of testing 59 samples to failure are presented in figure 6–5. The strength distribution of the material was calculated to have a mean of 19 900 psi and a standard deviation of 4200 psi. The safety margin was then calculated as

$$S_M = \frac{15\,000 - 19\,000}{4200} = 1.17$$

Because, from table 5–7, $S_M = \bar{x}_s / \sigma_s = 1.17$ indicates that 87.9 percent of the samples will fail at reliability boundaries above 15 000 psi, we can see that 12.1 percent will fail at boundaries below 15 000 psi. This analysis is optimistic in that

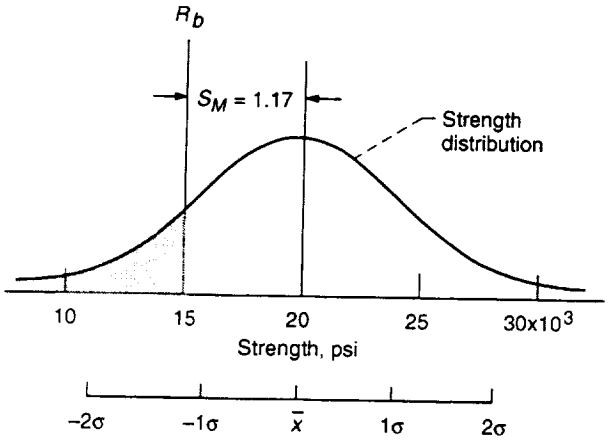


Figure 6–5.—Strength distribution in fiberglass material. $\bar{X}_S = 19\,000$ psi; $\sigma_S = 4200$ psi.

11/59 = 18.7 percent actually did fail below 15 000 psi. The test also shows that the reliability of the flame shield could be improved either by selecting another type of material to obtain a higher mean strength or by changing the fabrication processes to reduce the large strength deviation.

Example 8: Samples of transistors from two vendors were tested to failure under high temperatures. Failure was defined as any out-of-tolerance parameter. The results shown in figure 6–6 indicate that vendor B's materials, design, and process control were far superior to vendor A's as revealed by the large differences in mean strength and standard deviation. With an S_M of 1.41, 7.9 percent of vendor A's transistors would fail at the 74 °C reliability boundary; with an S_M of 8.27, vendor B's transistors would not be expected to fail at all. It is unlikely that an attribute test would have identified the better transistor.

Example 9: Squib switch samples were tested to failure under vibration in accordance with the procedure for testing one-shot items. The results are shown in figure 6–7, where the mean and standard deviations of the failure distribution have been calculated from the failure points observed. As shown, $\bar{x}_s = 14$ g's and $\sigma_s = 1.04$ g's to produce a safety margin of 3.84 with reference to the reliability boundary of 10 g's.

The preceding examples have shown how the P_t product reliability term can be effectively demonstrated through test-to-failure methods. This has been the case because each example except the squib switch involved a tolerance problem. The examples also show that the K_m factor plays an important role in product reliability and that control over K-factors can ensure a significant increase in reliability.

Multiple failure modes.—Most products perform more than one function and have more than one critical parameter for each function. In addition, most products are made up of many types of materials and parts and require many fabrication processes during manufacture. It follows then that a product can exhibit a variety of failure modes during testing.

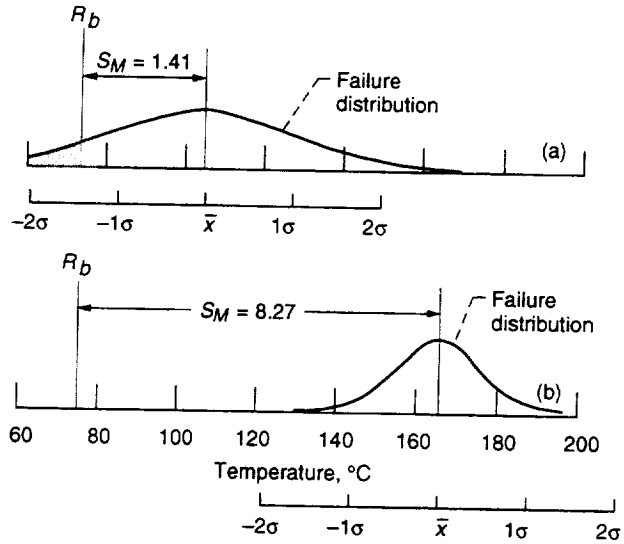


Figure 6–6.—Test-to-failure results for two transistors. (a) Vendor A. $\bar{X}_S = 105$ °C; $\sigma_S = 22$ °C. (b) Vendor B. $\bar{X}_S = 165$ °C; $\sigma_S = 11$ °C.

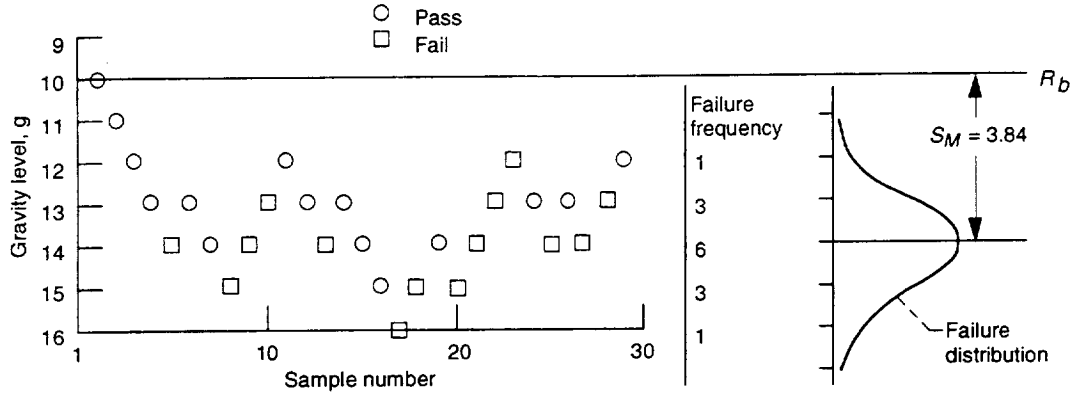


Figure 6–7.—Vibration test-to-failure results of one-shot device (squib switch). $\bar{X}_S$ = 14 g's; σ_S = 1.04 g's.

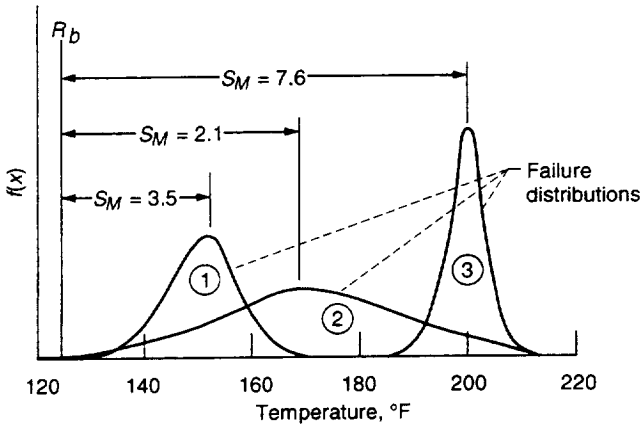


Figure 6–8.—Test-to-failure results when multiple failure modes are observed.

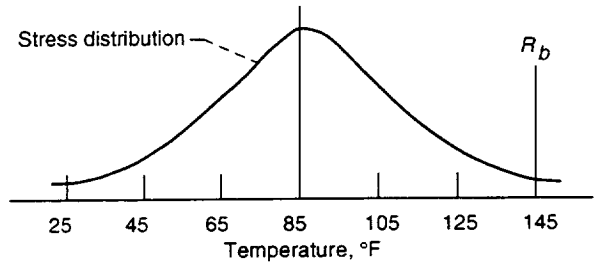


Figure 6–9.—Stress distribution for operating temperature. $\bar{X}_S$ = 85 °F; σ_S = 20 °F.

In the conduct of a test to failure, each failure mode detected must be evaluated individually; that is, a failure distribution must be developed for each failure mode and safety margins must be calculated for each individual failure distribution. Moreover, as mentioned before, at least five samples or failure points are needed to describe each failure mode distribution.

To see this more clearly, consider the test results shown in figure 6–8. Here, each of the three failure modes observed is described in terms of its own failure distribution and resulting safety margin with reference to the same reliability boundary. If these failure modes are independent and each represents an out-of-tolerance P_t condition, the P_t of the test device is given by

$$P_{t,\text{total}} = P_{t,1}(S_M = 3.5)P_{t,2}(S_M = 2.1)P_{t,3}(S_M = 7.6)$$
$$= (0.9998)(0.9821)(1.00) = 0.9819$$

This also shows that the independent evaluation of each failure mode identifies the priorities necessary to improve the product. For example, the elimination of failure mode 2, either by increasing $P_{t,2}$ to 1 or by eliminating the mode altogether increases $P_{t,\text{total}}$ from 0.9819 to 0.9998.

When stress distribution is known.—When safety margins are calculated with reference to a single point or a fixed reliability boundary, the resulting reliability estimate is conservative because it is assumed that the equipment will always be operated at the reliability boundary. As an illustration, figure 6–9 shows the stress distribution for the operating temperature of a device and the maximum anticipated operating limit (145 °F), which is given in the device specifications and would normally be considered the reliability boundary.

Figure 6–10 shows the strength distribution of the device for high temperatures and also that a safety margin for the device, when referenced to the 145 °F reliability boundary, is 1.54, or a reliability of 93.8 percent. We know, however, that the 145 °F limit is the 3σ limit of the stress distribution and will occur only 0.135 percent of the time. The question is, How does this affect the estimated reliability of the device in the temperature environment?

If we select random values from the stress and strength distribution and subtract the stress value from the strength value, a positive result indicates a success—the strength exceeds the stress. A negative result indicates a failure—the stress exceeds the strength. With this knowledge, we can calculate a *difference distribution* and through the application of the safety margin technique, solve for the probability of the strength being greater than the stress (i.e., success). This difference distribution is also distributed normally and has the following parameters:

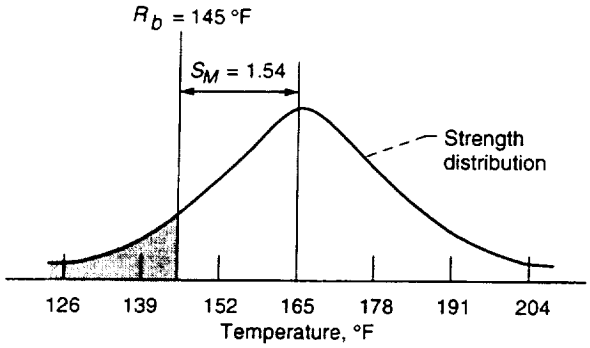


Figure 6–10.—Strength distribution for operating temperature. $\bar{X}_S$ = 165 °F; σ_S = 13 °F.

$$\bar{x}_{\text{difference}} = \bar{x}_s - \bar{x}_{\text{stress}}$$

$$\sigma_{\text{difference}} = \left(\sigma_s^2 - \sigma_{\text{stress}}^2\right)^{1/2}$$

From the strength and stress distribution parameters given in the preceding example (figs. 6–9 and 6–10),

$$\bar{x}_{\text{difference}} = 165 - 85 = 80\ °\text{F}$$

$$\sigma_{\text{difference}} = \left(20^2 + 13^2\right)^{1/2} = 24\ °\text{F}$$

This distribution is shown in figure 6–11.

Because positive numbers represent success events, we are interested in the area under the difference distribution that includes only positive numbers. This can be calculated by using zero as the reliability boundary and solving for the safety margin from

$$S_M = \frac{0 - \bar{x}_s}{\sigma_s} = \frac{0 - 80}{24} = 3.33$$

This 3.33 safety margin gives a reliability of 0.9996 when the stress distribution is considered. Comparing this result with the estimated reliability of 0.938 when the reliability boundary point estimate of 145 °F was used shows the significance of knowing the stress distribution when estimating reliability values.

Confidence levels.—As discussed before, the main objective in developing a failure distribution for a device by test-to-failure methods is to predict how well a population of like devices will perform. Of course, such failure distributions, along with the resulting safety margins and reliability estimates, are subject to error. Errors result from sample size limitations in much the same way as the demonstrated reliability varies with sample size in attribute testing. Specifically, the mean and the standard

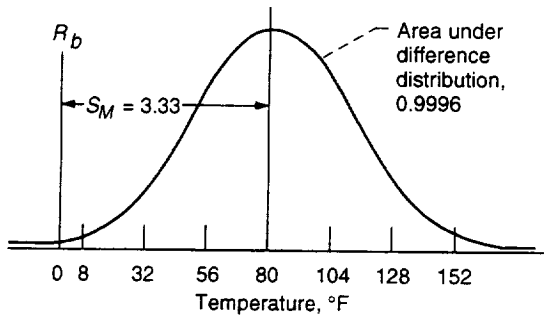


Figure 6–11.—Strength and stress difference distribution. $\bar{X}_S$ = 80 °F; σ_S = 24 °F.

TABLE 6–3.—CONFIDENCE LEVEL TABLES FOR VARIOUS SAMPLE SIZES

Confidence level, percent	Sample size			
	5 to 12	13 to 20	21 to 29	30 to 100
	Confidence level tables			
99	A–3(a)	A–3(b)	A–3(c)	A–3(d)
95	A–4(a)	A–4(b)	A–4(c)	A–4(d)
90	A–5(a)	A–5(b)	A–5(c)	A–5(d)

deviations of the strength distribution must be adjusted to reflect the sample size used in their calculation. For this purpose, tables A–3 to A–5 in appendix A have been developed by using the noncentral t distribution. Table 6–3 shows the applicable appendix A tables for selected confidence levels and sample sizes, and the examples that follow illustrate their use.

Example 10: Upon being tested to failure at high temperatures, 10 devices were found to have a failure distribution of $\bar{x}_s$ = 112.7 °C and σ_s = 16 °C. The reliability boundary was 50 °C. Find the safety margin and reliability demonstrated at 90-percent confidence.

Solution 10:

Step 1—Solve first for the observed safety margin.

$$S_M = \frac{R_b - \bar{x}_s}{\sigma_s} = \frac{50 - 112.7}{16} = 3.92$$

From table 5–7, the observed reliability is 0.99996.

Step 2—Now in appendix A refer to table A–5(a), which deals with 90-percent confidence limits for safety margins, and follow across to column N = 10, the number of samples. The values under the N headings in all the tables listed in table 6–3 represent the observed safety margins for sample sizes as calculated from raw test data. The S_M column lists corresponding population safety margins for the observed safety margins shown under the N headings. Finally, corresponding population reliability estimates are shown under the P_x headings, which may represent P_t or P_w as applicable.

Step 3—Proceed down the $N = 10$ column to 3.923, the observed safety margin derived in step 1.

Step 4—Having located $S_M = 3.923$ with 10 samples, follow horizontally to the left to find the demonstrated population safety margin in the S_M column. This is 2.6.

Step 5—With a population S_M of 2.6, follow the same line to the right to find the population reliability estimate under the P_x heading. This value is 0.9953. Recall that the observed safety margin was 3.923 and the observed reliability, 0.99996.

Example 11: Twelve gyroscopes were tested to failure by using time as a stress to develop a wearout distribution. The wearout distribution was found to have an $\bar{x}_s$ of 5000 hours and a σ_s of 840 hours. Find the P_w demonstrated at 95-percent confidence with a reliability boundary of 1000 hours.

Solution 11:

Step 1—The sample safety margin is

$$S_M = \frac{1000 - 5000}{840} = 4.76$$

Step 2—The population safety margin at 95-percent confidence with a 12-sample safety margin of 4.76 is read directly from table A–4(a) to be 3.0.

Step 3—For a population S_M of 3.0, the corresponding P_w under the P_x column is 0.9986. Therefore, 99.86 percent of the gyroscopes will not wear out before 1000 hours have been accumulated.

Safety factor.—This section is included in the discussion of test-to-failure methods because the term "safety factor" is often confused with safety margin. It is used widely in industry to describe the assurance against failure that is built into structural products. There are many definitions of safety factor S_F, with the most common being the ratio of mean strength to reliability boundary:

$$S_F = \frac{\bar{x}_s}{R_b}$$

When dealing with materials with clearly defined, repeatable, and "tight" strength distributions, such as sheet and structural steel or aluminum, using S_F presents little risk. However, when dealing with plastics, fiberglass, and other metal substitutes or processes with wide variations in strength or repeatability, using S_M provides a clearer picture of what is happening (fig. 6–12). In most cases, we must know the safety margin to understand how accurate the safety factor may be.

Test-to-failure summary.—In summary, you should understand the following concepts about test-to-failure applications:

(1) Developing a strength distribution through test-to-failure methods provides a good estimate of the P_t and P_w product

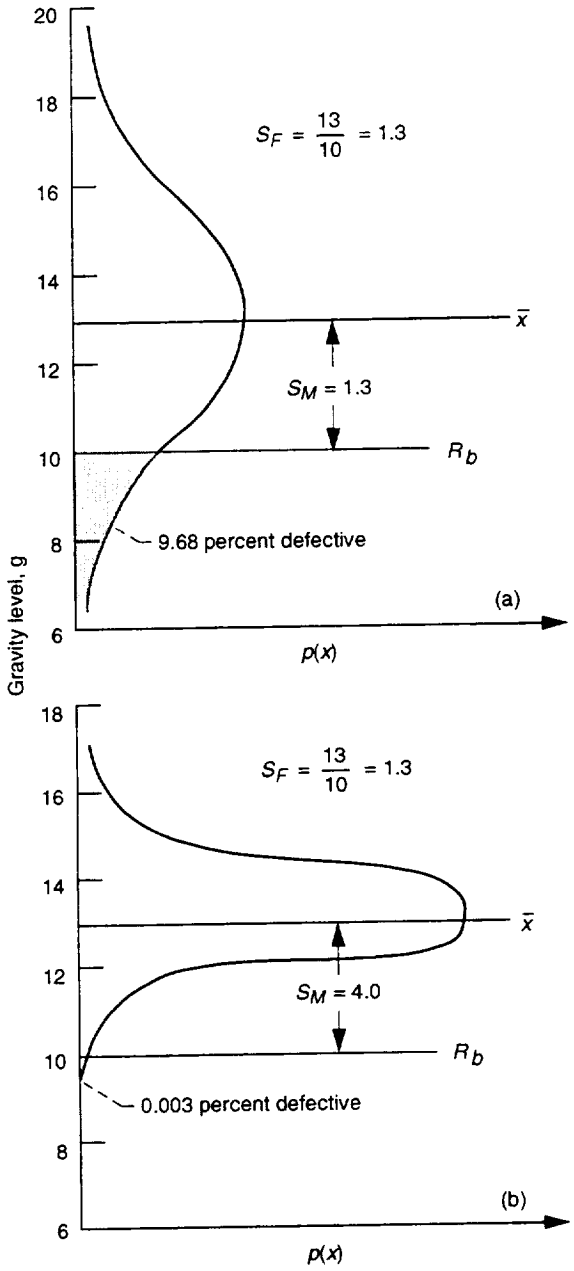


Figure 6–12.—Two structures with identical safety factors ($S_F = 13/10 = 1.3$) but with different safety margins. (a) Structure A. (b) Structure B.

reliability terms without the need for the large samples required for attribute tests.

(2) The results of a test-to-failure exposure of a device can be used to predict the reliability of similar devices that cannot or will not be tested.

(3) Testing to failure provides a means of evaluating the failure modes and mechanisms of devices for improvement purposes.

(4) Testing to failure allows confidence levels to be applied to the safety margins and to the resulting population reliability estimates.

(5) To know how accurate a safety factor may be, we must also know the associated safety margin.

Life Test Methods

Life tests are conducted to illustrate how the failure rate of a typical system or complex subsystem varies during its operating life. Such data provide valuable guidelines for controlling product reliability. They help to establish burn-in requirements, to predict spare part requirements, and to understand the need for or lack of need for a system maintenance program. Such data are obtained through laboratory life tests or from the normal operation of a fielded system.

Life tests are performed to evaluate product failure-rate characteristics. If failures include all causes of system failure, the failure rate of the system is the only true factor available for evaluating the system's performance. Life tests at the parts level often require large sample sizes if realistic failure-rate characteristics are to be identified and laboratory life tests are to simulate the major factors that influence failure rates in a device during field operations. Furthermore, the use of running averages in the analysis of life data will identify burn-in and wearout regions if such exist. Failure rates are statistics and therefore are subject to confidence levels when used in making predictions (see refs. 6–13 to 6–17).

Figure 6–13 illustrates what might be called a failure surface for a typical product. It shows system failure rate versus operating time and environmental stress. These three parameters describe a surface such that, given an environmental stress and an operating time, the failure rate is a point on the surface.

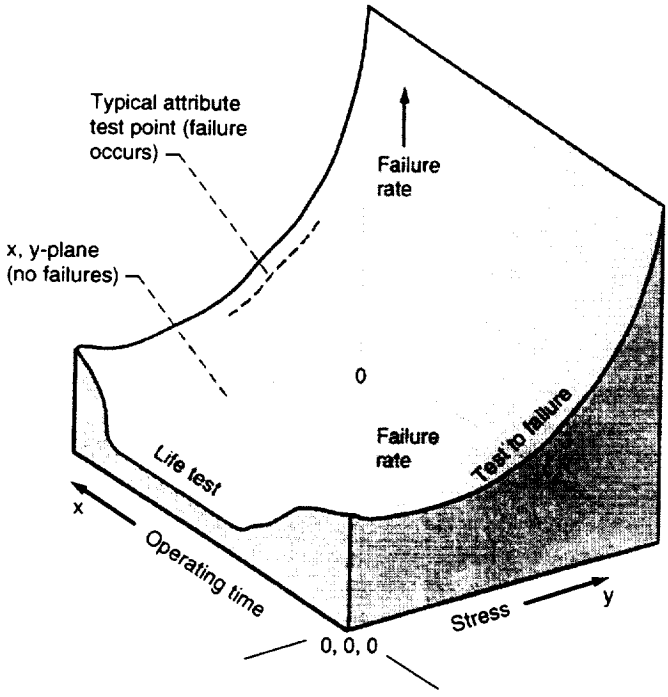


Figure 6–13.—Product failure surface.

Test-to-failure methods generate lines on the surface parallel to the stress axis; life tests generate lines on the surface parallel to the time axis. Therefore, these tests provide a good description of the failure surface and, consequently, the reliability of a product.

Attribute tests result only in a point on the surface if failures occur and a point somewhere on the x,y-plane if failures do not occur. For this reason, attribute testing is one of the least desirable methods for ascertaining reliability. Of course, in the case of missile flights or other events that produce go/no-go results, an attribute analysis is the only way to determine product reliability.

Application.—Although life test data are derived basically for use in evaluating the failure characteristics of a product, byproducts of the evaluation may serve many other purposes. Four of the most frequent are

(1) To serve as acceptance criteria for new hardware. For example, a product may be subjected to a life test before it is accepted for delivery to demonstrate that its failure rate is below some predetermined value. Examples of such applications are burn-in or debugging tests and group B life tests conducted on electronic parts. Some manufacturers of communications satellites subject all electronic parts to a 1200-hour burn-in test and use only the ones that survive.

(2) To identify product improvement methods. Here, life tests serve a dual purpose by providing hardware at essentially no cost for physics-of-failure analyses. In turn, these analyses identify failure mechanisms and the action needed to reduce effectively a product's failure rate. In the past 10 years, this has resulted in significant part failure-rate reductions. In fact, the failure rates of some components have been reduced so far that accelerated life tests (life tests at elevated stress levels) and test-to-failure techniques must be employed to attain reliability improvements in a reasonable timeframe.

(3) To establish preventive maintenance policies. Products with known or suspected wear mechanisms are life tested to determine when the wearout process will begin to cause undesirable failure-rate trends. Once the wearout region is established for a product, system failures can be reduced by implementing a suitable preventive maintenance plan or overhaul program. This is effectively illustrated in figure 6–14, which shows the failure-rate trend in a commercial jet aircraft subsystem. Here, the upward trend after 4000 hours of opera-

Figure 6–14.—Failure-rate characteristics of commercial jet electronic subsystem.

tion was revealed to be caused by a servomechanism that required lubrication. By establishing a periodic lubrication schedule for the mechanism, further failures were eliminated. Note that this subsystem also exhibited burn-in and intrinsic-failure-rate regions.

(4) To assess reliability. Here, tests are performed or life data are collected from fielded systems to establish whether contractual reliability requirements are actually being met. In cases of noncompliance and when the field failures are analyzed, one of the preceding methods is employed to improve the product, or else a design change is implemented. The effectiveness of the corrective action is then evaluated from additional life data. Because life-test-observed failure rates include catastrophic, tolerance, wearout, and K-factor failures, life tests usually demonstrate product reliability.

Test procedure and sample size.—Conducting a life test is fairly straightforward. It involves only the accumulation of equipment operating time. Precautions must be taken, however, when the test is conducted in a laboratory. Operating conditions must include all the factors that affect failure rates when the device is operated tactically. Major factors are environment, power-on and power-off times, power cycling rates, preventive maintenance, operator tasks, and field tolerance limits. Ignoring any of these factors may lead to an unrealistic failure-rate estimate.

When accelerated life tests are conducted for screening purposes, stress levels no greater than the inherent strength of the product must be chosen. The inherent strength limit can be evaluated through test-to-failure methods before the life tests are conducted.

Experience with nonaccelerated life tests of military standard electronic parts for periods as long as 5000 hours indicates that an average of one to two failures per 1000 parts can be expected. For this reason, life tests will not provide good reliability estimates at the part level except when quantities on the order of 1000 or more parts are available. On the other hand, life tests are efficient at the system level with only one sample as long as the system is fairly complex (includes several thousand parts).

Life tests intended to reveal the wearout characteristics of a device may involve as few as five samples, although from 20 to 30 are more desirable if a good estimate of the wearout distribution is to be obtained.

Analyzing life test data.—Recall from chapter 3 that an empirical definition of mean time between failures (MTBF) was given as

$$\text{MTBF} = \frac{\text{Total test hours}}{\text{Total observed failures}}$$

Remember also that because this expression neglects to show when the failures occur, it assumes an intrinsic failure rate and therefore an intrinsic mean time between failures, or MTBF. The assumption of an intrinsic failure rate may not be valid in some cases, but life test results have traditionally been reported this way.

To see this illustrated, consider the results of a 4000-hour life test of a complex (47 000 parts) electronic system as shown in figure 6–15. This graph plots cumulatively in terms of the times the 47 failures are observed so that the slopes of the lines represent the failure rate. The solid line shows the system failure rate that resulted from assuming an intrinsic failure rate, which was

$$\lambda = \frac{\text{Total failures}}{\text{Total operation time}} = \frac{47}{4000} = 1 \text{ failure/ } 86 \text{ hours}$$

From the plotted test data, it is obvious that this intrinsic failure rate was not a good estimate of what really happened. The plotted data indicate that there were two intrinsic-failure-rate portions: one from 0 to 1000 hours and the other from 1000 to 4000 hours. In the 0- to 1000-hour region, the actual failure rate was

$$\lambda = \frac{35}{1000} = 1 \text{ failure/ } 29 \text{ hours}$$

or about 3 times higher than the total average failure rate of 1/86 hours; in the 1000- to 4000-hour region, the actual failure rate was

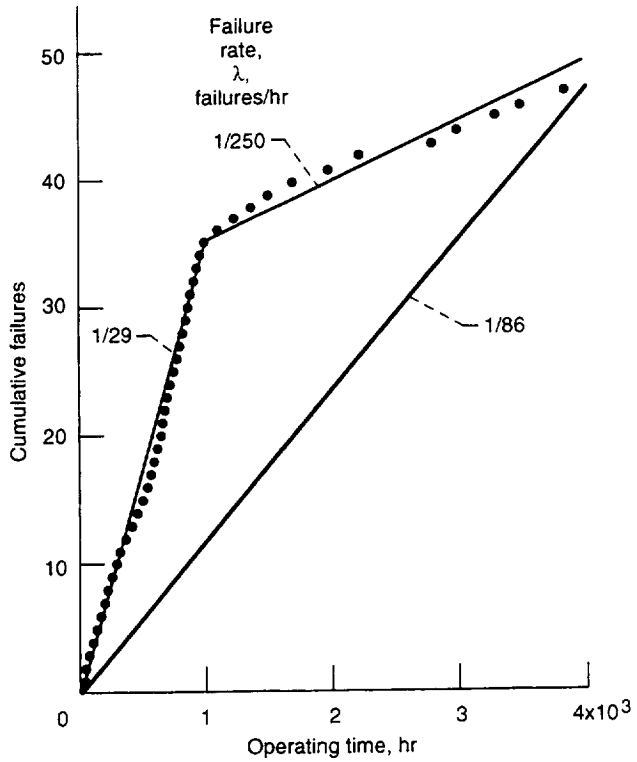


Figure 6–15.—Results of complex electronic system life test.

$$\lambda = \frac{12}{3000} = 1 \text{ failure/ } 250 \text{ hours}$$

or about 2.9 times lower than the average.

This illustration establishes the desirability of knowing when failures occur, not just the number of failures. The results of analyzing data by regions can be used to evaluate burn-in and spare parts requirements. The burn-in region was identified to be from 0 to 1000 hours because after this time the failure rate decreased by a factor of 8.6.

This result also has a significant effect on logistics. For example, if we assume that the system will accumulate 1000 hours per year, we can expect during the first year to replace 35 parts:

$$\left(\frac{1 \text{ failure}}{29 \text{ hours}} \times 1000 \text{ hours}\right)$$

whereas during the next and subsequent years we can expect to make only four replacements:

$$\left(\frac{1 \text{ failure}}{250 \text{ hours}} \times 1000 \text{ hours}\right)$$

Using the average failure rate of 1 failure/86 hours, we would have to plan, however, for 28 replacements every year. Obviously, the cost impact of detailed analysis can be substantial.

Running averages.—When system failure rates are irregular or when there is a need to evaluate the effect of different operating conditions on a system, running average analyses are useful. This can best be illustrated through the example presented in figure 6–16. A 300-hour running average in 50-hour exposures is shown for a complex system during an engineering evaluation test. (Running averages are constructed by finding the failure rate for the first 300 hours of operation, then dropping the first 50 hours and picking up the 300- to 350-hour interval and calculating the new 300-hour regional failure rate, and then repeating the process by dropping the second 50 hours of data and adding the next 50 hours for the total test period.) From the resultant curve, you can readily see (1) the effects of the debugging test, (2) the increase in failure rate during the high-temperature test and the decrease after that test, (3) another increase during low-temperature exposure and the subsequent decrease, (4) a slight increase caused by vibration, and (5) a continuously decreasing rate as the test progressed. The curve indicates that the system is the most sensitive to high temperature and that because the failure rate continued to decrease after high-temperature exposure, exposure to high temperatures is an effective way to screen defective parts from the system. Because the failure rate continued to decrease after the tests were completed, neither low temperature nor vibration caused permanent damage to the system.

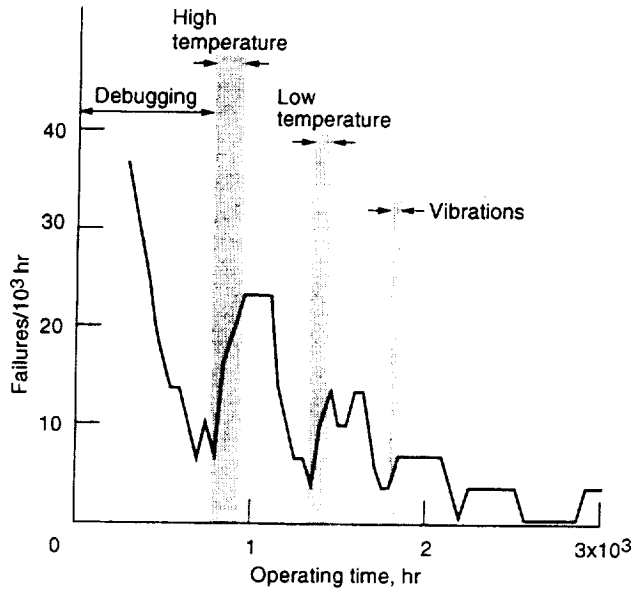


Figure 6–16.—Running average failure-rate analysis of life test data (300-hr running average in 50-hr increments).

At the end of the 3000-hour period, the failure rate was 3.3 failures per 1000 hours. This reflected a tenfold decrease from the initial failure rate during debugging, typical of the results observed for many complex systems. An example of a running average failure-rate analysis that identifies a system wearout region is shown in figure 6–17. The increasing failure rate after 3000 hours was caused by relay failures (during approximately 10 000 cycles of operation). This type of information can be used to establish a relay replacement requirement as part of a system preventive maintenance plan.

Confidence levels.—As discussed in chapter 4, failure rates are statistical. Consequently, they are subject to confidence levels just as attribute and test-to-failure results are influenced by such factors. Confidence levels for intrinsic failure rates are calculated by using table A–2 in appendix A.

To use this table, first calculate the total test hours accumulated from

$$t = \sum_{i=1}^{n} N_i T_i$$

where

N_i i^{th} unit tested
t_i test time of N_i
n total units tested

Then find under the number of failures observed during the test the tolerance factor for the desired confidence level. The lower limit for the MTBF at the selected confidence level is then found from

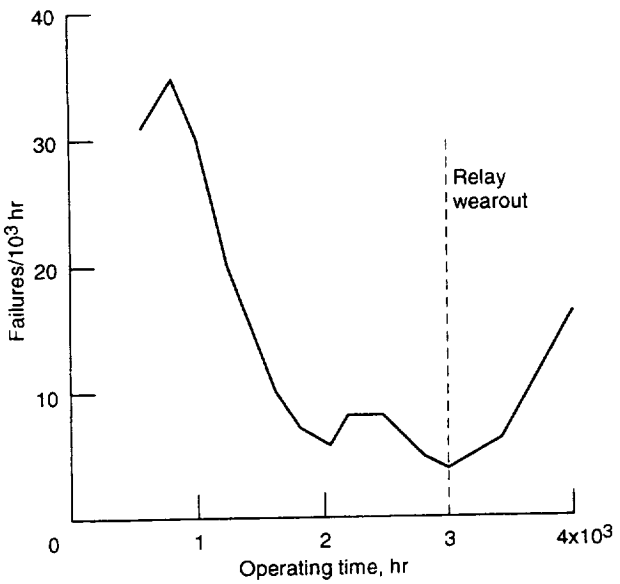


Figure 6–17.—Running average failure-rate analysis of life test data identifying wearout region (600-hr running average in 200-hr increments).

$$\text{MTBF} = \frac{t}{\text{Tolerance factor}}$$

and the upper limit for failure rate from

$$\lambda = \frac{\text{Tolerance factor}}{t}$$

Example 13: A system was life tested for 3000 hours, during which six failures were observed. What is the demonstrated 80-percent-confidence MTBF?

Solution 13:

Step 1—Solve for the total test hours.

$$t = \sum_{i=1}^{n} N_i T_i = 1 \times 3000 = 3000$$

Step 2—From table A–2 find the tolerance factor for six failures at 80-percent confidence to be 9.0.

Step 3—Solve for the demonstrated MTBF.

$$\text{MTBF} = \frac{t}{\text{Tolerance factor}} = \frac{3000}{9} = 333 \text{ hours}$$

in contrast to the observed MTBF of 3000/6 = 500 hours.

Example 14: Had four of the six failures in example 13 been observed in the first 1000 hours, what would be the demonstrated MTBF at 80-percent confidence in the region from 1000 to 3000 hours?

Solution 14:

Step 1—The total test time is given as t = 2000 hours.

Step 2—From table A–2 find the tolerance factor for two failures at 80-percent confidence to be 4.3.

Step 3—Find the demonstrated MTBF at 80-percent confidence after 1000 to 3000 hours.

$$\text{MTBF} = \frac{2000}{4.3} = 465 \text{ hours}$$

Example 15: It is desired to demonstrate an 80-hour MTBF on a computer at 90-percent confidence. How much test time is required on one sample if no failures occur?

Solution 15:

Step 1—From table A–2 find the tolerance factor for no failures at 90-percent confidence to be 2.3.

Step 2—Because the desired 90-percent-confidence MTBF is given as 80 hours and the tolerance factor is known, calculate the total test time required from

$$t = (\text{MTBF})(\text{Tolerance factor}) = (80)(2.3) = 184 \text{ hours}$$

to prove that 184 hours with no failures demonstrates an 80-hour MTBF at 90-percent confidence.

A good discussion of fixed time and sequential tests is given in MIL–STD–781D (ref. 6–3).

Life test summary.—In summary, the following concepts are reiterated:

(1) Life tests are performed to evaluate product failure-rate characteristics.

(2) If "failures" include all causes of system failure, the failure rate of the system is the only true factor available for evaluating the system's performance.

(3) Life tests at the part level require large sample sizes if realistic failure-rate characteristics are to be identified.

(4) Laboratory life tests must simulate the major factors that influence failure rates in a device during field operations.

(5) The use of running averages in the analysis of life data will identify burn-in and wearout regions if such exist.

(6) Failure rates are statistics and therefore are subject to confidence levels when used in making predictions.

Conclusion

When a product fails, whether during a test or from service, a valuable piece of information about it has been generated. We have the opportunity to learn how to improve the product if we take the right actions.

Much can be learned from each failure by using good failure reporting, analysis, and a concurrence system and by taking corrective action. Failure analysis determines what caused the part to fail. Corrective action ensures that the cause is dealt with.

With respect to testing, experimentation and evaluation to determine failure modes and effects greatly benefit reliability analysis. They do so by giving precise answers to the questions of why and how a product or component fails. Testing helps to reduce high development risks associated with a completely new design, to analyze high-risk portions of the design, and to confirm analytical models.

Attribute tests, although not the most satisfactory method of testing, can still identify gross design and manufacturing problems. Test-to-failure methods can be used to develop a strength distribution that gives a good estimate of tolerance and physical wearout problems without the need for large samples required in attribute tests. Life tests are performed to evaluate product failure-rate characteristics but the tests must be carefullydesigned.

All these test methods can be used to establish system level reliability and when conducted properly and in a timely fashion, can give valuable information about product behavior and overall reliability.

References

6–1. Reliability Program for Systems and Equipment Development and Production. MIL–STD–785B, July 1986.

6–2. Bazovsky, I.: Reliability Theory and Practice. Prentice-Hall, 1963.

6–3. Reliability Testing for Engineering Development, Qualification, and Production. MIL–STD–781D, Oct. 1986.

6–4. Reliability Test Methods, Plans, and Environments for Engineering Development, Qualification, and Production. MIL–HDBK–781, July 1987.

6–5. Laube, R.B.: Methods to Assess the Success of Test Programs. J. Environ. Sci., vol. 26, no. 2, Mar.–Apr. 1983, pp. 54–58.

6–6. Test Requirements for Space Vehicles. MIL–STD–1540B, Oct. 1982.

6–7. Haugen, E.B.: Probabilistic Approaches to Design. John Wiley & Sons, 1968.

6–8. Kececioglu, D.; McKinley, J.W.; and Saroni, M.J.: A Probabilistic Method of Designing Specified Reliabilities Into Mechanical Components With Time Dependent Stress and Strength Distributions. NASA CR–72836, 1967.

6–9. Laubach, C.H.: Environmental Acceptance Testing. NASA SP–T–0023, 1975.

6–10. Sneak Circuit Analysis. Boeing Safety Seminar, Boeing Systems Division, Aug. 1985.

6–11. Sneak Circuit Analysis. Naval Avionics Center, R&M–STD–R00205, May 1986.

6–12. Sneak Circuit Application Guidelines. Rome Air Development Center, RADC–TR–82–179, June 1989, (Avail. NTIS AD–A118479.)

6–13. Electronic Reliability Design Handbook. MIL–HDBK–338, vols. 1 and 2, Oct. 1988.

6–14. Reliability Modeling and Prediction. MIL–STD–756B, Aug. 1982.

6–15. Reliability Prediction of Electronic Equipment. MIL–HDBK–217F, Notice A, Jan. 1992.

6–16. Kregszig, E.: Introductory Mathematical Statistics, Principles and Methods, John Wiley & Sons, Inc., 1970.

6–17. Robert, N.H.: Mathematical Methods in Reliability Engineering. McGraw Hill, 1964.

Reliability Training[1]

1. Seven hydraulic power supplies were tested in a combined high-temperature and vibration test. Outputs of six of the seven units tested were within limits.

   a. What is the observed reliability R of the seven units tested?

      A. 0.825 B. 0.857 C. 0.913

   b. What is the predicted population reliability R at 80-percent confidence?

      A. 0.50 B. 0.75 C. 0.625

   c. How many tests (with one failure already experienced) are needed to demonstrate $R = 0.88$ at 80-percent confidence?

      A. 24 B. 15 C. 30

2. A vibration test was conducted on 20 autopilot sensing circuits with these results: Mean $\bar{x}_s = 7.8$ g's; standard deviation $\sigma_s = 1.2$ g's; reliability boundary $R_b = 6$ g's.

   a. What is the observed safety margin S_M?

      A. 2.0 B. 1.0 C. 1.5

   b. What is the observed reliability R?

      A. 0.900 B. 0.935 C. 0.962

   c. What is the predicted population safety margin S_M at 80-percent confidence?

      A. 1.19 B. 2.19 C. 3.19

   d. What is the predicted population reliability R at 80-percent confidence?

      A. 0.75 B. 0.95 C. 0.88

   e. How could the autopilot be made more reliable?

      A. Add brackets, thicker mounting materials, stiffer construction.

      B. Control material tolerances more tightly; inspect torque values and weld assemblies.

      C. Use vibration isolators.

      D. All of the above.

3. Twenty-five low-pressure hydraulic line samples were tested to destruction. These lines are rated to carry 30 psia (R_b); $\bar{x}_s = 31.5$ psia; $\sigma_s = 0.75$ psia.

   a. What is the observed S_M of these test items?

      A. 1.0 B. 2.0 C. 3.0

[1] Answers are given at the end of this manual. Please assemble and use the slide rule at the back of this manual to do this problem set.

b. What is the predicted population safety margin S_M at 90-percent confidence?

A. 0.95 B. 1.25 C. 1.51

c. The design requirement calls for an $S_M \geq 4.0$ at 90-percent confidence. After discussing the problem with the designer, it was learned that the 30-psia rating included a 2.5-psia "pad." Using the corrected R_b of 27.5 psia, now what are the S_M and S_D at 90-percent confidence?

i. S_M (observed) = ?

A. 4.22 B. 5.33 C. 6.44

ii. S_D (predicted) = ?

A. 4.28 B. 3.75 C. 4.80

Chapter 7
Software Reliability

Software reliability management is highly dependent on how the relationship between quality and reliability is perceived. For the purposes of this manual, quality is closely related to the process, and reliability is closely related to the product. Thus, both span the life cycle.

Before we can stratify software reliability, the progress of hardware reliability should be briefly reviewed. Over the past 25 years, the industry has observed (1) the initial assignment of "wizard status" to hardware reliability for theory, modeling, and analysis, (2) the growth of the field, and (3) the final establishment of hardware reliability as a science. One of the major problems was aligning reliability predictions and field performance. Once that was accomplished, the wizard status was removed from hardware reliability. The emphasis in hardware reliability from now to the year 2000, as discussed in chapter 1, will be on system failure modes and effects.

Software reliability has reached classification as a science for many reasons. The difficulty in assessing software reliability is analogous to the problem of assessing the reliability of a new hardware device with unknown reliability characteristics. The existence of 30 to 50 different software reliability models indicates the organization in this area. As discussed in chapter 1, hardware reliability started at a few companies and later was the focus of the AGREE reports. The field then logically progressed through different models in sequence over the years. Along the same lines, numerous people and companies have simultaneously entered the software reliability field in their major areas: namely, cost, complexity, and reliability. The difference is that at least 100 times as many people are now studying software reliability as initially studied hardware reliability. The existence of so many models and their purports tends to mask the fact that several of these models have shown excellent correlations between software performance predictions and actual software field performance; for instance, the Musa model as applied to communications systems and the Xerox model as applied to office copiers. There are also reasons for not accepting software reliability as a science, and they are briefly discussed here.

One impediment to the establishment of software reliability as a science is the tendency toward programming development philosophies such as (1) "do it right the first time" (a reliability model is not needed), or (2) "quality is a programmer's development tool," or (3) "quality is the same as reliability and is measured by the number of defects in a program and not by its reliability." All these philosophies tend to eliminate probabilistic measures because the managers consider a programmer to be a software factory whose quality output is controllable, adjustable, or both. In actuality, hardware design can be controlled for reliability characteristics better than software design can. Design philosophy experiments that failed to enhance hardware reliability are again being formulated for software design. (Some of the material in this chapter is reprinted with permission from ref. 7–1.) Quality and reliability are not the same. Quality is characteristic and reliability is probabilistic. Our approach draws the line between quality and reliability because quality is concerned with the development process and reliability is concerned with the operating product. Many models have been developed and a number of the measurement models show great promise. Predictive models have been far less successful partly because a data base (such as MIL–HDBK–217E (ref. 7–2) for hardware) is not yet available for software. Software reliability often has to use other methods; it must be concerned with the process of software product development.

Models

The development of techniques for measuring software reliability has been motivated mainly by project managers who not only need ways of estimating the manpower required to develop a software system with a given level of performance but also need techniques for determining when this level of performance has been reached. Most software reliability models presented to date are still far from satisfying these two needs.

Most models assume that the software failure rate will be proportional to the number of implementation and design errors

in the system without taking into account that different kinds of errors may contribute differently to the total failure rate. Eliminating one significant design error may double the mean time to failure, whereas eliminating 10 minor implementation errors (bugs) may have no noticeable effect. Even assuming that the failure rate is proportional to the number of bugs and design errors in the system, no model considers that the failure rate will then be related to the system workload. For example, doubling the workload without changing the distribution of input data to the system may double the failure rate.

Software reliability models can be grouped into four categories: time domain, data domain, axiomatic, and other.

Time Domain Models

Models formulated in the time domain attempt to relate software reliability (characterized, for instance, by a mean-time-to-failure (MTTF) figure under typical workload conditions) to the number of bugs present in the software at a given time during its development. Typical of this approach are the models presented by Shooman (ref. 7–3), Musa (ref. 7–4), and Jelinsky and Moranda (ref. 7–5). Removing implementation errors should increase MTTF, and correlating bug removal history with the time evolution of the MTTF value may allow the prediction of reliability when a given MTTF will be reached. The main disadvantages of time domain models are that bug correction can generate more bugs and that software unreliability can be due not only to implementation errors but also to design (specification) errors, characterization, and simulation during testing of the typical workload.

The Shooman model (ref. 7–3) attempts to estimate the software reliability—that is, the probability that no software failure will occur during an operating time interval $(0,t)$—from an estimate of the number of errors per machine-language instruction present in a software system after T months of debugging. The model assumes that at system integration there are E_i errors present in the system and that the system is operated continuously by an exerciser that emulates its real use. The hazard function after T months of debugging is assumed to be proportional to the remaining errors in the system. The reliability of the software system is then assumed to be

$$R(t) = e^{-CE(r,T)}$$

where $E(r,T)$ is the remaining number of errors in the system after T months of debugging and C is a proportionality constant. The model provides equations for estimating C and $E(r,T)$ from the results of the exerciser and the number of errors corrected.

The Jelinsky-Moranda model (ref. 7–5) is a special case of the Shooman model. The additional assumption made is that each error discovered is immediately removed, decreasing the remaining number of errors by one. Assuming that the amount of debugging time between error occurrences has an exponential distribution, the density function of the time of discovery of the i^{th} error, measured from the time of discovery of the $(i-1)^{th}$ error is

$$p(t_i) = \lambda(i)\, e^{-\lambda(i)t_i}$$

where $\lambda(i) = f(N-i+1)$ and N is the number of errors originally present. The model gives the maximum likelihood estimates for N and f.

The Jelinsky-Moranda model has been extended by Wolverton and Schick (ref. 7–6). They assume that the error rate is proportional not only to the number of errors but also to the time spent in debugging, so that the chance of discovery increases as time goes on. Thayer, Lipow, and Nelson (ref. 7–7) give another extension in which more than one error can be detected in a time interval, with no correction being made after the end of this interval. New maximum likelihood estimators of N and f are also given.

All the models presented so far attempt to predict the reliability of a software system after a period of testing and debugging. In a good example of an application of this type of model, Miyamoto (ref. 7–8) describes the development of an on-line, real-time system for which a requirement is that the mean time between software errors (MTBSE) has to be longer than 30 days. The system will operate on a day-by-day basis, 13 hours a day. (It will be loaded every morning and reset every evening.) The requirement is formulated so that the value of the reliability function $R(t)$ for t = 13 hours has to be greater than $e^{(-13/MTBSE)} = 0.9672$. Miyamoto also gives the MTBSE variations in time as a function of the debugging time. The MTBSE remained low for most of the debugging period, jumping to an acceptable level only at the end. The correlation coefficient between the remaining number of errors in the program and the failure rate was 0.77, but the scatter plot shown is disappointing and suggests that the correlation coefficient between the failure rate and any other system variable could have given the same value. In the same paper, Miyamoto describes in detail how the system was tested.

None of the above models takes into account that in the process of fixing a bug, new errors may be introduced in the system. The final number given is usually the mean time between software errors, but only Miyamoto points out that this number is valid only for a specific set of workload conditions.

Other models for studying the improvement in reliability of a software item during its development phase exist, such as Littlewood (ref. 7–9), in which the execution of a program is simulated with continuous-time Markov switching among smaller programs. This model also demonstrates that under certain conditions in the software system structure, the failure process will be asymptotically Poisson. Trivedi and Shooman (ref. 7–10) give another Markov model, in which the most probable number of errors that will have been corrected at any time t is based on preliminary modeling of the error occurrence and repair rates. The model also predicts the system's availabil-

ity and reliability at time t. Schneidewind (ref. 7–11) describes a model which assumes that the failure process is described by a nonhomogeneous Poisson process. The rate of error detection in a time interval is assumed to be proportional to the number of errors present during that interval. This leads to a Poisson distribution with a decreasing hazard rate.

Data Domain Models

Another approach to software reliability modeling is studying the data domain. The first model of this kind is described by Nelson (ref. 7–12). In principle, if sets of all input data upon which a computer program can operate are identified, the reliability of the program can be estimated by running the program for a subset of input data. Thayer, Lipow, and Nelson (ref. 7–7) describe data domain techniques in more detail. Schick and Wolverton (ref. 7–13) compare the time domain and data domain models. However, different applications will tend to use different subsets of all possible input data, yielding different reliability values for the same software system. This fact is formally taken into account by Cheung (ref. 7–14), in which software reliability is estimated from a Markov model whose transition probabilities depend on a user profile. Cheung and Ramamoorthy (ref. 7–15) give techniques for evaluating the transition probabilities for a given profile.

In the Nelson model (ref. 7–12) a computer program is described as a computable function F defined on the set $E = (E_i, i = 1, \ldots, N)$, where E includes all possible combinations of input data. Each E_i is a sample of data needed to make a run of the program. Execution of a program produces, for a given value of E_i, the function value $F(E_i)$.

In the presence of bugs or design errors, a program actually implements F'. Let E_e be the set of input data such that $F'(E_e)$ produces an execution failure (execution terminates prematurely, or fails to terminate, or the results produced are not acceptable). If N_e is the quantity of E_i leading to failure F_e,

$$p = \frac{N_e}{N}$$

is the probability that a run of the program will result in an execution failure. Nelson defines the reliability R as the probability of no failures or

$$R = 1 - p = 1 - \frac{N_e}{N}$$

In addition, this model is further refined to account for the fact that the inputs to a program are not selected from E with equal a priori probability but are selected according to some operational requirement. This requirement may be characterized by a probability distribution $(P_i, i = 1, \ldots, N)$, P_i being the probability that the selected input is E_i. If we define the auxiliary variables Y_i to be 0 if a run with E_i is successful, and 1 otherwise,

$$p = \sum_{i=1}^{N} P_i Y_i$$

where p is again the probability that a run of the program will result in an execution failure.

A mathematical definition of the reliability of a computer program is given as the probability of no execution failures after n runs:

$$R(n) = R^n = (1 - p)^n$$

The model elaborates on how to choose input data values at random for E according to the probability distribution P_i to obtain an unbiased estimator of $R(n)$. In addition, if the execution time for each E_i is also known, the reliability function can be expressed in terms of the more conventional probability of no failure in a time interval $(0, t)$.

Chapter 6 in Thayer, Lipow, and Nelson (ref. 7–7) extends the previous models to take into account how the testing of input data sets should be partitioned. Also discussed are the uncertainty in predicting reliability values, the effect of removing software errors, and the effect of program structure.

Axiomatic Models

The third category includes models in which software reliability (as well as software quality in general) is postulated to obey certain universal laws (Ferdinand and Sutherla, ref. 7–16; Fitzsimmons and Love, ref. 7–17). Although such models have generated great interest, their general validity has never been proven and, at most, they only give an estimate of the number of bugs present in a program.

The best-known axiomatic model is the so-called software science theory developed by Halstead (see ref. 7–18). Halstead used an approach similar to thermodynamics to provide quantitative measures of program level, language level, algorithm purity, program clarity, effect of modularization, programming effort, and programming time. In particular, the estimated number of bugs in a program is given by the expression

$$B = K\left(\frac{V}{E_0}\right)$$

where

K proportionality constant
E_0 mean number of mental discriminations between errors made by programmer

TABLE 7–1.—CORRELATION OF EXPERIENCE TO SOFTWARE BUG PREDICTION BY AXIOMATIC MODELS

Reference	Correlation coefficient between predicted and real number of bugs
Funami and Halstead (ref. 7–19)	0.98, 0.83, 0.92
Cornell and Halstead (ref. 7–20)	0.99
Fitzsimmons and Love (ref. 7–17):	
System A	0.81
System B	.75
System C	.75
Overall	.76

V volume of algorithm implementation, $N \log_2(n)$

where

N program length
n size of vocabulary defined by language used

More specifically,

$$N = N_1 + N_2$$

$$n = n_1 + n_2$$

where

N_1 total number of occurrences of operators in a program
N_2 total number of occurrences of operands in a program
n_1 number of distinct operators appearing in a program
n_2 number of distinct operands appearing in a program

and E_0 has been empirically estimated to be approximately 3000.

Many publications have either supported or contradicted the results proposed by the software science theory, including a special issue of the IEEE Transactions on Software Engineering (ref. 7–18). Though unconventional, the measures proposed by the software science theory are easy to compute, and in any case it is an alternative for estimating the number of bugs in a software system. Table 7–1 shows a correlation coefficient between the real number of bugs found in a software project and the number predicted by the software science theory for several experiments. There are significant correlations with error occurrences in the programs, although the data reported by Fitzsimmons and Love (ref. 7–17) (obtained from three General Electric software development projects totaling 166 280 statements) show weaker correlation than the original values reported by Halstead.

Other Models

The model presented by Costis, Landrault, and Laprie (ref. 7–21) is based on the fact that for well-debugged programs, a software error results from conditions on both the input data set and the logical paths encountered. We can then consider these events random and independent of the past behavior of the system (i.e., with constant failure rate). Also, because of their rarity, design errors or bugs may have the same effect as transient hardware faults.

The model is built on the following assumptions:

(1) The system initially possesses N design errors or bugs that can be totally corrected by N interventions of the maintenance team.

(2) The software failure rate is constant for a given number of system design errors.

(3) The system starts and continues operation until a fault is detected; it then passes to a repair state. If the fault is due to a hardware transient, the system is put into operation again after a period of time for which the probability density function is assumed to be known. If the fault is due to a software failure, maintenance takes place, during which the error may be removed, more errors may be introduced, or no modifications may be made to the software.

The model computes the availability of the system as a function of time by using semi-Markovian theory. That is, the system will make state transitions according to the transition probabilities matrix, and the time spent in each state is a random variable whose probability density function is either assumed to be known or is measurable. The main result presented by Costis, Landrault, and Laprie (ref. 7–21) is how the availability of the system improves (when all the design errors have been removed) as the design errors are being removed under some restrictive conditions. They show that the minimum availability depends only on the software failure rate at system integration and not on the order of occurrence of the different types of design errors. The presence of different types of design errors only extends the time necessary to approach the asymptotic availability.

The mathematics of the model is complex, requiring numerical computation of inverse Laplace transforms for the transition probabilities matrix, and it is not clear that the parameters needed to simulate a real system accurately can be easily measured from a real system.

Finally, some attempts have been made to model fault-tolerant software through module duplication (Hecht, ref. 7–22) and warnings about how not to measure software reliability (Littlewood, ref. 7–23).

None of the preceding models characterizes system behavior accurately enough to give the user a guaranteed level of performance under general workload conditions. They estimate the number of bugs present in a program but do not provide any

accurate method of characterizing and measuring operational system unreliability due to software. There is a large gap between the variables that can be easily measured in a running system and the number of bugs in its software. Instead, a cost-effective analysis should allow precise evaluation of software unreliability from variables easily measurable in an operational system, without knowing the details of how the software has been written.

Trends and Conclusions

With software reliability being questioned as a science, programming process control appears to be the popular answer to both software reliability and software quality. Measurements of the programming process are supposed to ensure the generation of an "error-free" programming product, if such an achievement is possible. Further, quality and productivity measurements combined with select leading process indicators are supposed to fulfill the control requirements for developing quality software. This so-called answer is similar to a philosophy that failed in attempts to develop hardware reliability control. Reliability should be used to predict field performance. Especially with real-time communications and information management systems, the field performance requirements vastly overshadow the field defect level requirements. How can we change the present popular trend (toward programming process control) to one that includes a probabilistic reliability approach? The answer is not a simple one; these models must be finely balanced so that a clear separation of reliability and quality can be achieved.

The trends for reliability tasks in the large-scale integrated circuit (LSI) and very large-scale integrated circuit (VLSI) hardware areas are in the failure modes and effects analysis and the control of failures. The same emphasis can be placed on software (programming bugs or software errors). Once this is done, reliability models can reflect system performance due to hardware and software "defects" because their frequency of occurrence and the effects of their presence in the operation will be known. This philosophy focuses on the complete elimination of critical defects and the specified tolerance level of minor defects. Normally, minor defects are easier to find and more numerous than the most critical defects and therefore dominate a defect-removal-oriented model.

We conclude that the proper method for developing quality programming products combines quality, reliability, and a selective measurements program. In addition, a redirection of the programming development process to be based in the future on the criticality of defects, their number, and their budgeting at the various programming life-cycle phases is the dominant requirement. A reliability growth model will monitor and control the progress of defect removal for the design phases and prove a direct correlation to actual system field performance. With such an approach, a system can be placed in operation at a customer site at a preselected performance level as predicted by the growth model.

Software

For several reasons, we have discussed software models before describing software. The reader should not be biased or led to a specific type of software. Few papers on software reliability make a distinction between product software, embedded software, applications software, and support software. In addition, the models do not distinguish between vendor-acquired software and in-house software and combinations of these.

Categories of Software

According to Electronic Design Magazine, the United States supports at least 50 000 software houses, each grossing approximately $500 000 per year. It is projected that software sales in the United States will surpass hardware sales and reach the $60 billion range. International competition will eventually yield error-free software.

In-house and vendor-acquired software can be categorized as follows:

(1) Product
(2) Embedded
(3) Applications
(4) Support

Product software.—This categorization is from the viewpoint of the software specialist. Communications digital switching systems software is included as "product software" along with the software for data packet switching systems, text systems, etc.

Embedded software.—This category comprises programming systems embedded in physical products to control their operational characteristics. Examples of products are radar controllers, boiler controls, avionics, and voice recognition systems.

Applications software.—This category is usually developed to service a company's internal operations. The accounting area of this category covers payroll systems, personnel systems, etc. The business area includes reservations systems (car, motel), delivery route control, manufacturing systems, and on-line agent systems.

Support software.—This category consists of the software tools needed to develop, test, and qualify other software products or to aid in engineering design and development. The category includes compilers, assemblers, test executives, error seeders, and development support systems.

Vendor-acquired software.—This software can be absorbed by the previous four categories and is only presented here for clarification. It includes FORTRAN compilers, COBOL compilers, assemblers, the UNIX operating system, the ORACLE data base system, and application packages.

Processing Environments

Software can usually be developed in three ways: (1) interactive, (2) batch, and (3) remote job entry. In the operational environment, these expand to include real time. Real-time development can be characteristic of both product software and embedded software. However, because product software and embedded software differ greatly in their requirements and in their development productivity and quality methodologies, they should not be combined (e.g., avionics has size, weight, and reliability requirements resulting in dense software of a type that a communications switching system does not have).

Severity of Software Defects

We must categorize and weigh the effects of failures. The following four-level defect severity classification is presented in terms of typical software product areas:

(1) System unusable (generic: frequent system crashes)
  (a) Management information system (MIS) software defects: inability to generate accounts payable or to access data base; improper billing
  (b) Computer-aided design (CAD), manufacturing (CAM), and engineering (CAE) defects: inability to use systems; CAD produces incorrect designs
  (c) Telephone switching defects: frequent service outages; loss of emergency communications service
  (d) Data communications defects: loss of one or more signaling channels; unrecoverable errors in transmission; erratic service
  (e) Military system defects: success of mission jeopardized; inability to exercise fire control systems; loss of electronic countermeasure capabilities
  (f) Space system defects: success of space mission jeopardized; risk of ground support team or flight crew life; loss of critical telemetry information
  (g) Process control defects: waste of labor hours, raw materials, or manufactured items; loss of control resulting in contamination or severe air and water pollution
(2) Major restrictions (generic: loss of some functions)
  (a) MIS software defects: loss of some ticket reservation centers or loss of certain features such as credit card verification
  (b) CAD/CAM/CAE defects: loss of some features in computer-aided design such as the update function; significant operational restrictions in CAM or CAE areas; faults produced for which there is no workaround
  (c) Telephone switching defects: loss of full traffic capability; loss of billing
  (d) Data communications defects: occasional loss of consumer data; inability to operate in degraded mode with loss of equipment
  (e) Military system defects: significant operational restrictions; loss of intermediate fast frequency function in detection systems; loss of one or more antijamming features
  (f) Space system defects: occasional loss of telemetry data and communications; significant operational or control restrictions
  (g) Process control defects: process cannot consistently handle exceptions; inability to complete all process control functions
(3) Minor restrictions (generic: loss of features; inability to effectively modify program)
  (a) MIS software defects: mishandling of records; system occasionally cannot handle exceptions
  (b) CAD/CAM/CAE defects: occasional errors produced in design system; faults produced for which there are workarounds
  (c) Telephone switching defects: loss of some support feature, such as call forwarding or conferencing
  (d) Data communications defects: occasional inability to keep up with data rate or requests; occasional minor loss of data transmitted or received
  (e) Military system defects: loss of some operational modes such as tracking history, monitor or slave model of operation, multiple option selection
  (f) Space system defects: occasional loss of update information or frame; occasional loss of subframe synchronization or dropouts of some noncritical measurements
  (g) Process control defects: problems that require a workaround to be implemented; minor reductions in rate or throughput; manual intervention at some points in the process
(4) No restrictions (generic: cosmetic; misleading documentation; inefficient machine/person interface)

Software Bugs Compared With Software Defects

Software bugs are not necessarily software defects: the term "defect" implies that removal or repair is necessary, and the term "bug" implies removal, some degree of correction, or a certain level of toleration. A recent example of bug toleration from the telecommunications industry is contained in reference 7–24:

> It is not technically or economically feasible to detect and fix all software problems in a system as large as No. 4 Electronic Switching System (ESS). Consequently, a strong emphasis has been placed on making it sufficiently tolerant of software errors to provide successful operation and fault recovery in an environment containing software problems.

Various opinions exist in the industry about what constitutes a software failure. Definitions range from a software failure being classed as any software-caused processor restart or memory reload to a complete outage. One argument against assigning an MTBF to software-caused processor restarts or memory reloads is that if the system recovers in the proper manner by itself, there has not been a software failure, only a software fault or the manifestation of a software bug. From a systems reliability viewpoint, if the system recovers within a reasonable time, the event is not to be classed as a software failure.

Hardware and Software Failures

Microprocessor-based products have more refined definitions. Four types of failure may be considered: (1) hardware catastrophic, (2) hardware transient, (3) software catastrophic, and (4) software transient. In general, the catastrophic failures require a physical or remote hardware replacement, a manual or remote unit restart, or a software program patch. The transient failure categories can result in either restarts or reloads for the microprocessor-based systems, subsystems, or individual units and may or may not require further correction. A recent reliability analysis of such a system assigned ratios to these categories. Hardware transient faults were assumed to occur at 10 times the hardware catastrophic rate, and software transient faults were assumed to occur at 100 to 500 times the software catastrophic rate.

The time of day is of great concern in reliability modeling and analysis. Although hardware catastrophic failures occur at any time of the day, they often manifest themselves during busier system processing times. On the other hand, hardware transient failures generally occur during the busy hours as do software transient failures. The availability of restart times is also critical and in the example presented in reference 7–25, the system downtime is presented as a function of the MTBF of the software and the reboot time. When a system's predicted reliability is close to the specified reliability, such a sensitivity analysis must be performed.

Reference 7–26 presents a comprehensive summary of developed models and methods that encompass software lifecycle costs, productivity, reliability and error analysis, complexity, and the data parameters associated with these models and methods. The various models and methods are compared in reference 7–26 on a common basis, and the results are presented in matrix form.

Manifestations of Software Bugs

Many theories, models, and methods are available for quantifying software reliability. Nathan (ref. 7–27) stated, "It is contrary to the definition of reliability to apply reliability analysis to a system that never really works. This means that the software which still has bugs in it really has never worked in the true sense of reliability in the hardware sense." This statement agrees with reference 7–24, which says that large, complex software programs used in the communications industry are usually operating with some software bugs. Thus, a reliability analysis of such software is different from a reliability analysis of established hardware. Software reliability is not alone in the need for establishing qualitative and quantitative models. Reference 7–28 discusses the "bathtub curve" and the effect of recent data on electronic equipment failure rate, and reference 7–30 discusses the effects of deferred maintenance and nonconstant software and hardware fault rates.

In the early 1980's work was done on a combined hardware/software reliability model. Reference 7–30 states, "The use of steady-state availability as a reliability/maintainability measure is shown to be misleading for systems exhibiting both hardware and software faults." The authors develop a theory for combining well-known hardware and software models in a Markov process and consider the topic of software bugs and errors based on their experience in the telecommunications field. To synthesize the manifestations of software bugs, we must note some of the hardware trends for these systems:

(1) Hardware transient failures increase as integrated circuits become denser.

(2) Hardware transient failures tend to remain constant or increase slightly with time after the infant mortality phase.

(3) Hardware (integrated circuit) catastrophic failures decrease with time after the infant mortality phase.

These trends affect the operational software of communications systems. If the transient failures increase, the error analysis and system security software are called into action more often. This increases the risk of misprocessing a given transaction in the communications system. A decrease in the catastrophic failure rate of integrated circuits can be significant, as described in reference 7–13, which predicts an order-of-magnitude decrease in the failure rate of 4K memory devices between the first year and the twentieth year. We also tend to over-simplify the actual situations. Even with five vendors of these 4K devices, the manufacturing quality control person may have to set up different screens to eliminate the defective devices from different vendors. Thus, the system software will see many different transient memory problems and combinations of them in operation.

Central control technology has prevailed in communications systems for 25 years. The industry has used many of its old modeling tools and applied them directly to distributed control structures. Most modeling research was performed on large duplex processors. With an evolution through forms of multiple duplex processors and load-sharing processors and onto the present forms of distributed processing architectures, the

TABLE 7–2.—CRITICALITY INDEX

Bug manifestation rate	Defect removal rate	Level of criticality	Failure type	Failure characteristic
4 per day	1 per month	5	Transient	Errors come and go
3 per day	1 per week	4	Transient	Errors are repeated
2 per week	1 per month	3	Transient or catastrophic	Service is affected
1 per month	2 per year	2	Transient or catastrophic	System is partially down
1 per two years	1 per year	1	Catastrophic	System stops

modeling tools need to be verified. With fully distributed control systems the software reliability model must be conceptually matched to the software design to achieve valid predictions of reliability.

The following trends can be formulated for software transient failures:

(1) Software transient failures decrease as the system architecture approaches a fully distributed control structure.

(2) Software transient failures increase as the processing window decreases (i.e., less time allowed per function, fast timing mode entry, removal of error checking, removal of system ready checks, etc.)

A fully distributed control structure can be configured to operate as its own error filter. In a hierarchy of processing levels, each level acts as a barrier to the level below and prevents errors or transient faults from propagating through the system. Central control structures cannot usually prevent this type of error propagation.

If the interleaving of transaction processes in a software program is reduced, such as with a fully distributed control architecture, the transaction processes are less likely to fail. This is especially true with nonconsistent user interaction as experienced in communications systems. Another opinion on software transient failures is that the faster a software program runs, the more likely it is to cause errors (such as encountered in central control architectures). Some general statements can be formulated:

(1) In large communications systems, software transient failures tend to remain constant, and software catastrophic failures tend to decrease with time.

(2) In small communications systems, software transient failures decrease with time.

(3) As the size of the software program increases, software transient failures decrease and hardware failures increase.

A "missing link" needs further discussion. Several methods can be used to quantify the occurrence of software bugs. However, manifestations in the system's operations are detrimental to the reliability analysis because each manifestation could cause a failure event. The key is to categorize levels of criticality for bug manifestations and estimate their probability of occurrence and their respective distributions. The importance of this increases with the distribution of the hardware and software. Software reliability is often controlled by establishing a software reliability design process. Reference 7–24 presents techniques for such a design process control. The final measure is the system test, which includes the evaluation of priority problems and the performance of the system while under stress as defined by audits, interrupts, reinitialization, and other measurable parameters. The missing link in quantifying software bug manifestations needs to be found before we can obtain an accurate software reliability model for measuring tradeoffs in the design process on a predicted performance basis. If a software reliability modeling tool could additionally combine the effects of hardware, software, and operator faults, it would be a powerful tool for making design tradeoff decisions. Table 7–2 is an example of the missing link and presents a five-level criticality index for defects. Previously, we discussed a four-level defect severity classification with level four not causing errors. These examples indicate the flexibility of such an approach to criticality classification.

Software reliability measurement and its applications are discussed in reference 7–31 for two of the leading software reliability models, Musa's execution time model and Littlewood's Bayesian model. Software reliability measurement has made substantial progress and continues to progress as additional projects collect data. The major hurdle in establishing a software reliability measurement tool for use during the requirement stage is under way.

Comparing references 7–32 and 7–31 yields an insight into the different methods of achieving software reliability. The method described in reference 7–32 concentrates on the design process meeting a present level of reliability or performance at the various project design stages. When the system meets its final software reliability acceptance criteria, the process is complete. Reference 7–31 describes a model that provides the design process with a continuous software reliability growth prediction. The Musa model can compare simultaneous software developments and can be used extensively in making design process decisions. An excellent text on software

reliability based on extensive data gathering was published in 1987 (ref. 7–33).

We can choose a decreasing, constant, or increasing software bug removal rate for systems software. Although each has its application to special situations and systems, a decreasing software bug removal rate will generally be encountered. Systems software also has advantages in that certain software defects can be temporarily patched and the permanent patch postponed to a more appropriate date. Thus, this type of defect manifestation is treated in general as one that does not affect service, but it should be included in the overall software quality assessment. The missing link concerns software bug manifestations. As described in reference 7–34, until the traditional separation of hardware and software systems is overcome in the design of large systems, it will be impossible to achieve a satisfactory performance benchmark. This indicates that software performance modeling has not yet focused on the specific causes of software unreliability.

References

7-1. Siewiorek, D.P.; and Swarz, R.S.: The Theory and Practice of Reliable System Design. Digital Press, Bedford, MA, 1982, pp. 206–211.

7-2. Reliability Prediction of Electronic Equipment. MIL–HDBK–217E, Jan. 1990.

7-3. Shooman, M.L.: The Equivalence of Reliability Diagrams and Fault-Free Analysis. IEEE Trans. Reliab., vol. R–19, no. 2, May 1970, pp. 74–75.

7-4. Musa, J.D.: A Theory of Software Reliability and Its Applications. IEEE Trans. Software Eng., vol. SE–1, no. 3, Sept. 1975, pp. 312–327.

7-5. Jelinsky, Z.; and Moranda, P.B.: Applications of a Probability Based Method to a Code Reading Experiment. Record 1973: IEEE Symposium on Computer Software Reliability, IEEE, New York, 1973, pp. 78–82.

7-6. Wolverton, R.W.; and Schick, G.J.: Assessment of Software Reliability. TWE–SS–73–04, Los Angeles, CA, 1974.

7-7. Thayer, T.A.; Lipow, M.; and Nelson, E.C.: Software Reliability: A Study of a Large Project Reality. North Holland, 1978.

7-8. Miyamoto, I.: Software Reliability in Online Real Time Environment. International Conference on Reliable Software, IEEE/Automation Industries, Inc., Silver Spring, MD, 1975, pp. 518–527.

7-9. Littlewood, B.: A Reliability Model for Markov Structured Software. International Conference on Reliable Software, IEEE/Automation Industries, Inc., Silver Spring, MD, 1975, pp. 204–207.

7-10. Trivedi, A.K.; and Shooman, M.L.: A Many-State Markov Model for the Estimation and Prediction of Computer Software Performance. International Conference on Reliable Software, IEEE/Automation Industries, Inc., Silver Spring, MD, 1975, pp. 208–220.

7-11. Schneidewind, N.F.: Analysis of Error Processes in Computer Software. International Conference on Reliable Software, IEEE/Automation Industries, Inc., Silver Spring, MD, 1975, pp. 337–346.

7-12. Nelson, E.C.: A Statistical Basis for Software Reliability Assessment. TRW, 1973.

7-13. Schick, G.J.; and Wolverton, R.W.: An Analysis of Computing Software Reliability Models. IEEE Trans. Software Eng., vol. SE–4, no. 2, Mar. 1978, pp. 104–120.

7-14. Cheung, R.C.: A User-Oriented Software Reliability Model. IEEE Trans. Software Eng., vol. SE–6, no. 6, Mar. 1970, pp. 118–125.

7-15. Cheung, R.C.; and Ramamoorthy, C.V.: Optimum Measurement of Program Path Frequency and Its Applications. International Federation of Automatic Control: 6th World Congress, Instrument Society of America, 1975, vol. 4, paper 34–3.

7-16. Ferdinand, A.E.; and Sutherla, T.W.: A Theory of Systems Complexity. Int. J. Gen. Syst., vol. 1, no. 1, 1974, pp. 19–33.

7-17. Fitzsimmons, A.; and Love, T.: Review and Evaluation of Software Science. Comput. Surv., vol. 10, no. 1, Mar. 1978, pp. 3–18.

7-18. Commemorative Issue in Honor of Dr. Maurice H. Halstead. IEEE Trans. Software Eng., vol. SE–5, no. 2, Mar. 1979.

7-19. Funami, Y.; and Halstead, M.H.: A Software Physics Analysis of Akiyama's Debugging Data. Purdue University, CSD TR–144, 1975.

7-20. Cornell, L.; and Halstead, M.H.: Predicting the Number of Bugs Expected in a Program Module. Purdue University, CSD TR–202, 1976.

7-21. Costis, A.; Landrault, C.; and Laprie, J.C.: Reliability and Availability Models for Maintained Systems Featuring Hardware Failures and Design Faults. IEEE Trans. Comput., vol. C–27, June 1978, pp. 548–560.

7-22. Hecht, H.: Fault-Tolerant Software for Real-Time Applications. Comput. Surv., vol. 8, no. 4, Dec. 1976, pp. 391–407.

7-23. Littlewood, B.: How To Measure Software Reliability and How Not To. IEEE Trans. Software Eng., vol. SE–5, no. 2, June 1979, pp. 103–110.

7-24. Davis, E.A.; and Giloth, P.K.: Performance Objectives and Service Experience. Bell Syst. Tech. J., vol. 60, no. 6, July-Aug. 1981, pp. 1203–1224.

7-25. Aveyard, R.L.; and Man, F.T.: A Study on the Reliability of the Circuit Maintenance System–IB. Bell Syst. Tech. J., vol. 59, no. 8, Oct. 1980, pp. 1317–1332.

7-26. Software Engineering Research Review: Quantitative Software Models. Report No. SPR–1. Data and Analysis Center for Software (DACS), Griffiss AFB, NY, 1979.

7-27. Nathan, I.: A Deterministic Model To Predict 'Error-Free' Status of Complex Software Development. Workshop on Quantitative Software Models. IEEE, New York, 1979.

7-28. Wong, K.L.: Unified Field (Failure) Theory-Demise of the Bathtub Curve. Annual Reliability and Maintainability Symposium, IEEE, New York, 1981, pp. 402–407.

7-29. Malec, H.A.: Maintenance Techniques in Distributed Communications Switching Systems. IEEE Trans. Reliab., vol. R–30, no. 3, Aug. 1981, pp. 253–257.

7-30. Angus, J. E.; and James, L. E.: Combined Hardware/Software Reliability Models. Annual Reliability and Maintainability Symposium, IEEE, New York, 1982, pp. 176–181.

7-31. Musa, J.D.: The Measurement and Management of Software Reliability. IEEE Proceedings, vol. 68, no. 9, Sept. 1980, pp. 1131–1143.

7-32. Giloth, P.K.; and Witsken, J.R.: No. 4 ESS-Design and Performance of Reliable Switching Software. International Switching Symposium (ISS '81–CIC), IEEE, 1981, pp. 33AI/1–9.

7-33. Musa, J.D.; Iannino, A.; and Okamoto, K.: Software Reliability. McGraw-Hill, 1987.

7-34. Malec, H.A.: Transcribing Communications Performance Standards Into Design Requirements. ITT Adv. Technol. Center Tech. Bul., vol. 2, no. 1, Aug. 1981.

Reliability Training[1]

1. In-house and vendor-acquired software can be classified into what four categories?

   A. Product, embedded, applications, and error-free software

   B. Useful, embedded, applications, and harmful software

   C. Product, embedded, applications, and support software

2. Name the four categories of software reliability models.

   A. Time domain, data axiom, corollary, and many

   B. Time domain, data domain, axiomatic, and other

   C. Time axiom, data domain, frequency domain, and corollary

3. Can the bug manifestation rate be

   A. Equal to the defect removal rate?
   B. Greater than the defect removal rate?
   C. Less than the defect removal rate?
   D. All of the above?

4. What are the various software processing environments?

   A. Interactive, batch, remote job entry, and real time

   B. Hyperactive, batch, close job entry, and compressed time

   C. Interactive, batch, real job entry, and remote time

5. Name the four levels of severity for software defect categorizations.

   A. Generic system, functional, category restrictions, and working

   B. System unusable, major restrictions, minor restrictions, and no restrictions

   C. System unusable, system crashes, loss of features, and minor bugs

6. An online, real-time system has a mean time between software errors of 15 days. The system operates 8 hours per day. What is the value of the reliability function? Use the Miyamoto model.

   A. 0.962

   B. 0.999

   C. 0.978

7. Is it always necessary to remove every bug from certain software products?

   A. Yes

   B. No

   C. Don't know

8. Name the four types of hardware and software failure.

   A. Hardware part, hardware board, software module, software plan

   B. Hardware plan, hardware build, software cycle, software type cycle

   C. Hardware catastrophic, hardware transient, software catastrophic, software transient

[1]Answers are given at the end of this manual.

Reference Document for Inspection: "Big Bird's" House Concept

What is desired: bird house

For whom (client, customer, user): "Big Bird" (the tall yellow bird on "Sesame Street")

Why: Why not, even Oscar the Grouch has a house.

"Big Bird's" General Concept

"Big Bird" needs a house (and he's willing to pay for it) and he wants it big enough for him to live in (he's over 6 feet tall). He wants to be able to enter and leave the house comfortably, to be able to lock out the big bad wolves (even those dressed as granny), the materials used to be strong enough to support his weight (he's not particularly svelte), and to be weather proofed enough to keep him dry and warm in stormy weather, as defined by the post office (rain, sleet, hail, snow, wind).

Class Meeting Exercise: Requirements Inspection

Statement of Problem:	"Big Bird" has no house.	**Life Cycle Stage**
Done	Step 1: Build a house.	Concept
	Step 2: State the kind of house desired.	Requirements System Subsystem
Done		
To be inspected		
	Step 3: Make drawings of desired house.	Design
	Step 4: Build house.	Development
	Step 5: Walk through house (open doors and windows).	Test
	Step 6: Pay for house.	Delivery
	Step 7: Live in house.	Operation and maintenance

Note: At any step, perform analysis and SQUAWK if changes are needed.

Reference Document for Inspection System Requirements

"Big Bird's" House Systems Requirements

Excuse Me, Are Those Requirements?

Well, yes, after a bit of questioning and head scratching, the following system requirements were defined:

1. The house shall accommodate "Big Bird" and his belongings.
2. The house shall provide easy access to "Big Bird."
3. The building materials shall be strong enough to support "Big Bird" (who is, ahem, rather rotund).
4. The building materials shall deny entrance to big bad wolves (straw definitely being out of favor).
5. The house shall have security measures to prevent easy access to any nefarious beings intending "fowl" play.
6. The building materials shall be weather proof and found in nature.
7. The building materials shall be low cost (even birds have budgets).
8. The house shall be one room.
9. The house shall have one door.
10. The house shall have a floor.
11. The house shall have a roof.
12. The house shall have one window.
13. The house shall rest on level ground beneath his tree.
14. There will be no electricity, plumbing, heating, or air conditioning (client has feathers, candles, a bird bath, and ice cream).
15. Client will bring his own bed (BHOB).
16. The cost of the house shall not exceed 80 bird bucks.

"Big Bird's" Requirements Checklist

Clarity

1. Are requirements specified in an implementation-free way so as not to obscure the original requirements?
2. Are implementation, method, and technique requirements kept separate from functional requirements?
3. Are the requirements clear and unambiguous (i.e., are there aspects of the requirements that you do not understand; can they be misinterpreted)?

Completeness

1. Are requirements stated as completely as possible? Have all incomplete requirements been captured as TBD's?
2. Has a feasibility analysis been performed and documented?
3. Is the impact of not achieving the requirements documented?
4. Have trade studies been performed and documented?
5. Have the security issues of hardware, software, operations personnel, and procedures been addressed?
6. Has the impact of the project on users, other systems, and the environment been assessed?
7. Are the required functions, external interfaces, and performance specifications prioritized by need date? Are they prioritized by their significance to the system?

Compliance

1. Does this document follow the project's system documentation standards?
2. Does it follow JPL's standards?
3. Does the appropriate standard prevail in the event of inconsistencies?

Consistency

1. Are the requirements stated consistently without contradicting themselves or the requirements of related systems?
2. Is the terminology consistent with the user and/or sponsor's terminology?

Correctness

1. Are the goals of the system defined?

Data Usage

1. Are "don't care" condition values truly "don't care?" ("Don't care" values identify cases when the value of a condition or flag is irrelevant, even though the value may be important for other cases.)
2. Are "don't care" condition values explicitly stated? (Correct identification of "don't care" values may improve a design's portability.)

Functionality

1. Are all functions dearly and unambiguously described?
2. Are all described functions necessary and together sufficient to meet mission and system objectives?

Interfaces

1. Are all external interfaces clearly defined?
2. Are all internal interfaces clearly defined?
3. Are all interfaces necessary, together sufficient, and consistent with each other?

Maintainability

1. Have the requirements for system maintainability been specified in a measurable, verifiable manner?
2. Are requirements written to be as weakly coupled as possible so that rippling effects from changes are minimized?

Performance

1. Are all required performance specifications and the amount of performance degradation that can be tolerated explicitly stated (e.g., consider timing, throughput, memory size, accuracy, and precision)?
2. For each performance requirement defined,
   a. Do rough estimates indicate that they can be met?
   b. Is the impact of failure to meet the requirement defined?

Reliability

1. Are clearly defined, measurable, and verifiable reliability requirements specified?
2. Are there error detection, reporting, and recovery requirements?
3. Are undesired events (e.g., single-event upset, data loss or scrambling, operator error) considered and their required responses specified?
4. Have assumptions about the intended sequence of functions been stated? Are these sequences required?
5. Do these requirements adequately address the survivability after a software or hardware fault of the system from the point of view of hardware, software, operations personnel, and procedures?

Testability

1. Can the system be tested, demonstrated, inspected, or analyzed to show that it satisfies requirements?
2. Are requirements stated precisely to facilitate specification of system test success criteria and requirements?

Traceability

1. Are all functions, structures, and constraints traced to mission/system objectives?
2. Is each requirement stated in a manner that it can be uniquely referenced in subordinate documents?

"Big Bird's" Formal Inspection Subsystem Requirements
'Subsystem Requirements' Written for Big Bird's Approval

1. The house shall be made of wood.	Acceptable ☐	or	Major Missing Type ☐ Minor Wrong Origin ☐ Open issue Extra Defect classification
2. The house shall be nailed together.	Acceptable ☐	or	Major Missing Type ☐ Minor Wrong Origin ☐ Open issue Extra Defect classification
3. The house size shall be 4 cubits by 4 cubits by 3 cubits. (If cubits were good enough for Noah, they are good enough for us.)	Acceptable ☐	or	Major Missing Type ☐ Minor Wrong Origin ☐ Open issue Extra Defect classification
4. The door shall be made of balsa wood.	Acceptable ☐	or	Major Missing Type ☐ Minor Wrong Origin ☐ Open issue Extra Defect classification
5. The door opening shall be 4 inches by 8 feet.	Acceptable ☐	or	Major Missing Type ☐ Minor Wrong Origin ☐ Open issue Extra Defect classification
6. The door shall have a lock and key.	Acceptable ☐	or	Major Missing Type ☐ Minor Wrong Origin ☐ Open issue Extra Defect classification
7. The door shall have a door knob and hinges.	Acceptable ☐	or	Major Missing Type ☐ Minor Wrong Origin ☐ Open issue Extra Defect classification
8. The door shall be on the same wall as the window.	Acceptable ☐	or	Major Missing Type ☐ Minor Wrong Origin ☐ Open issue Extra Defect classification
9. The door shall be 12 meters from the mah jong set, shall be glued with silly putty to the wall, and shall play the Hallelujah Chorus when the doorbell is rung by the wolves wanting to eat Big Bird.	Acceptable ☐	or	Major Missing Type ☐ Minor Wrong Origin ☐ Open issue Extra Defect classification
10. The floor shall be carpeted with a silk comforter (cost of 100 bird bucks; client has cold feet).	Acceptable ☐	or	Major Missing Type ☐ Minor Wrong Origin ☐ Open issue Extra Defect classification
11. The roof shall be shingled.	Acceptable ☐	or	Major Missing Type ☐ Minor Wrong Origin ☐ Open issue Extra Defect classification
12. The shingles shall be taffy.	Acceptable ☐	or	Major Missing Type ☐ Minor Wrong Origin ☐ Open issue Extra Defect classification
13. The house shall be painted blue.	Acceptable ☐	or	Major Missing Type ☐ Minor Wrong Origin ☐ Open issue Extra Defect classification
14. The window shall be 3 by 3 feet.	Acceptable ☐	or	Major Missing Type ☐ Minor Wrong Origin ☐ Open issue Extra Defect classification
15. The window shall have interior locking wood shutters (wolf proofing).	Acceptable ☐	or	Major Missing Type ☐ Minor Wrong Origin ☐ Open issue Extra Defect classification
16. The window shall have a screen.	Acceptable ☐	or	Major Missing Type ☐ Minor Wrong Origin ☐ Open issue Extra Defect classification
17. The screen shall be made of oriental tissue paper.	Acceptable ☐	or	Major Missing Type ☐ Minor Wrong Origin ☐ Open issue Extra Defect classification

"Big Bird's" Formal Inspection Subsystem Requirements 'Subsystem Requirements' Written for Big Bird's Approval (Concluded)

18. The window shall open, close, and lock.	Acceptable ☐	or	Major Missing Type ☐ Minor Wrong Origin ☐ Open issue Extra Defect classification
19. The window shall have double thermal glass panes.	Acceptable ☐	or	Major Missing Type ☐ Minor Wrong Origin ☐ Open issue Extra Defect classification
20. The window shall be placed next to the flumajubit and the whupinsnapper.	Acceptable ☐	or	Major Missing Type ☐ Minor Wrong Origin ☐ Open issue Extra Defect classification
21. The house shall be insulated (see 10 above).	Acceptable ☐	or	Major Missing Type ☐ Minor Wrong Origin ☐ Open issue Extra Defect classification
22. The insulation shall be cough lozenges (Smith Brothers, cherry flavor).	Acceptable ☐	or	Major Missing Type ☐ Minor Wrong Origin ☐ Open issue Extra Defect classification
23. The house shall have one bed (cost of 100 bird bucks).	Acceptable ☐	or	Major Missing Type ☐ Minor Wrong Origin ☐ Open issue Extra Defect classification
24. The cost of the house shall be 300 bird bucks.	Acceptable ☐	or	Major Missing Type ☐ Minor Wrong Origin ☐ Open issue Extra Defect classification

Chapter 8
Software Design Improvements

Part I—Software Benefits and Limitations

Introduction

Computer hardware and associated software have been used for many years to process accounting information, to analyze test data, and to perform engineering analysis. Now, computers and software control everything from automobiles to washing machines and the number and type of applications are growing at an exponential rate. The size of individual programs has had similar growth. Furthermore, software and hardware are used to monitor and/or control potentially dangerous products and safety-critical systems. These uses include everything from airplanes and braking systems to medical devices and nuclear plants.

The benefits to systems of using software are reduction in weight, better optimization, autonomous action taken in emergencies, more features and, hence, flexibility for users of computer-based products, increased capabilities, better design analysis, and identification of the causes of problems.

What is the benefit of weight reduction? Using a computer system to control aircraft and spacecraft has tremendous weight and cost advantages over relying upon conventional electromechanical systems and personnel (who could be better used elsewhere).

Some of the questions software designers ask are, How can this hardware and software be made more reliable? How can software quality be improved? What methodology needs to be provided on large and small software products to improve the design? How can software be verified?

Software reliability.—Software reliability includes the probability that the program (in terms of the computer and its software) being executed will not deliver erroneous output. People have come to trust computer-generated results (assuming that the input data are correct); however, we are now beginning to encounter problems. Recently a manufacturer reported that its motherboards, which employed a particular IDE (integrated drive electronics) controller, "when using certain operating systems have the potential for data corruption that could manifest itself as a misspelled word in a document, incorrect values or account balances in accounting software, ... or even corruption of an entire partition or drive." The potential for data errors due to software embedded in certain Pentium computer chips has also been discovered (ref. 8–1).

Importance of reliability.—The tremendous growth in the use of software to control systems has also drawn attention to the importance of reliability. Critical life-support systems and flight controls on military and civilian aircraft use software. For example, mechanical interlocks, which prevent unsafe conditions from occurring (such as disabling power when an instrument cover is removed), are being replaced with software-controlled interlocks.

The size of the software also continues to grow, making it more costly to find and fix errors. From a few lines of code 20 years ago to 500 000 source lines of code (SLOC) for only the flight software of the space shuttle (ref. 8–2) and 1.588 million SLOC for the F–22 fighter (ref. 8–3). The application of software in the automotive industry has increased from an 8-bit processor that controlled engine applications to a powerful personal computer that added more built-in diagnostics and systems controls. Also, because of its complexity, only 1 percent of major software projects are finished on time and budget and 25 percent are never finished at all (ref. 8–4).

Some problems have become apparent. There occasionally exists a lack of discipline in generating software; people treat software controls very lightly and often have not attempted to

predict the reliability and safety implications of their software. Hence, there are many potential and unrecognized pitfalls in the application of software that are only now being realized. Many serious incidents in safety-critical applications may have been related to software and the complex control interfaces that often accompany software controlled systems. One example occurred when "in 1983 a United Airlines Boeing 767 went into a 4-minute powerless glide after the pilot was compelled to shut down both engines." This was due to a computerized engine-control system (in an attempt to optimize fuel efficiency) that ordered the engines to run at a speed where ice buildup and overheating occurred (ref. 8–5).

A China Airlines A300–600R Airbus crashed in part because of cockpit confusion. "Essentially, the crew had to choose between allowing the aircraft to be governed by its automatic pilot or to fly it manually. Instead, they took a halfway measure, probably because they failed to realize that their trimmable horizontal stabilizer (THS) had moved to a maximum noseup deflection as an automatic response to a go-around command. It was defeating their effort to bring the aircraft's nose down with elevator control ... (ref. 8–6)."

Because of these problems, we need to ask the following questions: What computer system errors can occur? What are the risks to the system from software? Why do accidents involving software happen—from both the systems engineering and the software engineering viewpoint? What are some software reliability or (safety) axioms that can be applied to software development? How can we be aware of the real risks and dangers from the application of software to a control and sensor problem?

Software quality.—How can the design of software be improved? Part II of this chapter, Software Quality and the Design and Inspection Process, will answer these questions. It will also discuss the following topics: useful software quality metrics, tools to improve software quality, software specifications, assessing the quality and reliability of software, specifications to improve software safety, tools that affect software reliability and quality, factors that affect tradeoffs and costing when software quality is evaluated.

Software safety.—Software development is now a key factor affecting system safety because of the often catastrophic effects of software errors. Therefore, a system can only be safe if its software cannot cause the hardware to create an unsafe condition. Software safety is the effective integration of software design, development, testing, operation, and maintenance into the system development process. A safety-critical computer software component (SCCSC) is one whose errors can result in a potential hazard, loss of predictability, or loss of system control. System functions are safety-critical when the software operations that, if not performed, performed out-of-sequence, or performed incorrectly can result in improper control functions that could directly or indirectly cause or allow a hazardous condition to exist. How can this software be improved?

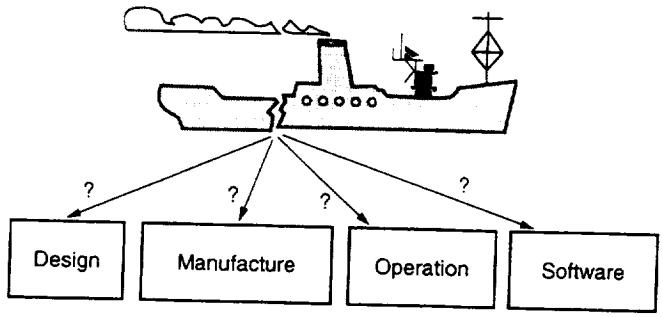


Figure 8–1.—Failure origins.

Overview: How Do Failures Arise?

Generally, we can say that all failures come from the design or manufacturing process or from the operation of the equipment (the computer), its associated software, and the system it controls (fig. 1). Software is becoming a critical source of failures because they often occur in unexpected ways. Through a long history of the design process and particularly in the design of mechanisms or structures, the type and severity of failures have become well known. Hardware failures can often be predicted, inspections can be set up to look for potential failures, and the manufacturing process can be changed to make a mechanical system more reliable.

Although a small anomaly or error in the design or operation of a mechanical system often produces a predictable and corresponding failure, software is different. An incorrect bit, a corrupted line of code, or an error in logic can have disastrous consequences. Testing a mechanical system (though not perfect) can be set up to validate all "known" events; on the other hand, software with only a few thousand SLOC may contain hundreds of decision options with millions of potential outcomes that cannot all be tested for or even predicted. Also, historically the design and behavior of mechanical systems have been well known, so expanding the performance envelope of the design led to a new system that was similar to the old one. The behavior of the new mechanical system was predictable. This does not hold true for software because minor changes in a program can lead to major changes in output.

Error types.—The types and sources of errors that can occur in a computer-controlled system are presented in figure 2 and are described next:

- Hardware failure in the computer: common to all electrical devices
- Hardware logic errors (in program logic controllers (PLC's)): mistakes in design or manufacture
- Coding errors: mistakenly written into program or program became corrupted
- Requirements errors: missing, incomplete, ambiguous, or contradictory specifications

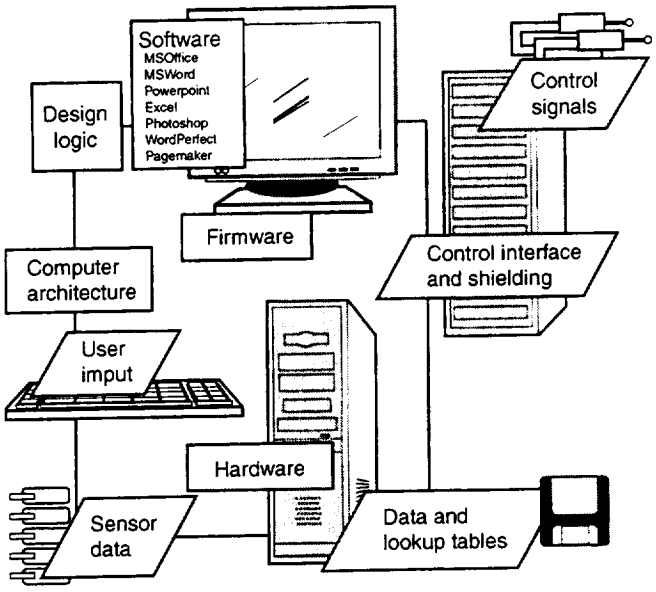


Figure 8–2.—Types of errors.

- Unintended outcome or state: logic errors in the program code for a given set of inputs
- Corrupted data: partially failed sensors or errors in internal lookup tables
- User interface problems: several sources (e.g., multiple points to turn off computer control of a system or keyboard buffers are too small)
- Faulty software tools (e.g., finite-element structural analysis code generation programs): errors in logic and outputs
- Execution problems
  - Variations in computer architecture from platform to plat form causing software verified on one platform to behave differently on another platform
  - Faulty or difficult-to-use interfaces between computers or between computers and sensors

Hardware and software failure differences.—In comparison with the methods used to verify the reliability of system hardware components, those used for software prediction, inspection, testing, and reliability verification differ greatly. The reason for the differences is the nonphysical, abstract nature of software, the failures of which are almost always information design oversights or programming mistakes and are not caused by environmental stresses or cumulative damage. Furthermore, the design rules for mechanical systems are usually well known, a vast amount of historical data on similar systems being available along with mathematical models of wear, fatigue, electrical stress, and so forth to make life predictions. Each software system is often unique. Even with some code reuse, complexity makes reapplication difficult. Some features of software and hardware reliability are compared in table 8–1.

Types of Software

Software types are classified on the bases of timing and control, run methodology, and run environment.

Timing and control.—Software risks and their impact on systems and data can be evaluated based on how the software interacts with the system, how humans interact with the system and the software, and whether this activity is carried on in real time. Factors to evaluate are whether (1) the software controls a system or just provides information, (2) real-time human interference and evaluation of output are allowed, (3) the software output time is critical or nontime critical, and (4) the data supplied by informational software are critical or noncritical. These factors are summarized in table 8–2. The reader should also consult MIL–STD–882C, System Safety Program Requirements (ref. 8–7), for types of software based on levels of control and hazard criticality.

Run methodology.—Another classification of software is based on run methodology and includes these types:

TABLE 8–1.—HARDWARE AND SOFTWARE FAILURE DIFFERENCES

Category	Hardware	Software
Reliability prediction tools	Many mathematical models exist for predicting wear, fatigue life, and electronic component life.	Reliability predictions are nearly impossible due to the nonrandom distribution of errors.
Causes of failures	Wearout, misuse, inadequate design, manufacture or maintenance or incorrect use can contribute to failures	Poor design affects software (the computer system on which the software resides can also fail).
Redundancy	Hardware reliability is usually improved with redundancy.	Software reliability (except possibly for multiple voting systems) is not improved with redundancy.
Hard or soft failures	Soft failures (some degradation in service before complete failure) often occur due to wear, chemical action, electrical degradation, etc.	Usually no soft failures occur (however, there may be some recovery routines that can take the system to a safe state, etc.)
Maintenance	Usually testing and maintenance improve hardware and increase reliability	Software reprogramming may introduce new and unpredictable failure modes into the system. Reliability may be decreased. Any change to the code should require complete retesting of the software, but this is usually not done.
Reliability prediction methodology	Design theory, a history of previous systems and load predictions all allow excellent reliability prediction	Software reliability is a function of the development process.

TABLE 8–2.—CLASSIFICATION OF SOFTWARE BASED ON LEVEL OF HAZARD AND CONTROL

Software control	Information	Human/other control	Real time	Examples
Autonomous control exercised over hazardous systems.	Some information may be available but insufficient for real-time interference.	May be possible but not desirable; often no other independent safety systems	Yes	Space shuttle main engine and solid rocket booster ignition sequence
Semiautonomous control exercised over hazardous systems.	Real-time information is available to allow human/other system interaction and control.	Possible and desirable under some circumstances; other independent safety systems or ability to disengage	Yes	Aircraft terrain-following system, medication dispensing device, nuclear power plant safety systems, automatic go around mode in aircraft (override)
Mix of computer and human control over hazardous systems.	Real-time information is available to allow human interaction and control. Human control of some functions.	Yes, required for some subsystems of operation; other independent safety systems	Yes	Aircraft fly-by-wire systems of unstable aircraft (example BN–2) where computer translates pilots control requests into feasible flight surface modifications
No, but generates information requiring immediate human action.	Complete real-time information presented to allow human control over hazardous systems.	Human interaction required to properly control the system; other independent safety systems	Yes	Aircraft collision avoidance systems, nuclear power plant instrumentation, hospital patient vital signs
No, but human action based on information.	Information not presented in real time. Software does provide critical information.	Human actions and decisions directly influenced by information; other checks	No	Statistical process control information of machine tools, historical medical information summaries
No, but human action based on information.	Information not presented in real time. Software does not provide critical information.	Human actions and decisions directly influenced by the information	No	Financial and economic data

Interactive: a program that is continuously running and interacting with the operator

Batch: a single run or process of a program (often acting on data, such as a finite-element analysis) from which a single output will occur

Remote job entry: a software environment in which programs are submitted or started by others from remote locations who usually seek a single output

Environment.—Software may be classified according to the environment in which it operates:

Embedded: a computer code written to control a product; usually resides on a processor that is part of the product; has typical applications as boiler controllers, washing machine and automobile computer controls

Applications: program that analyzes data; often runs as a batch job on a computer with limited input from the user once the job is submitted; operates in payroll systems, finite-analysis programs, material requirements planning (MRP) systems (to update sections)

Support: software tools that may be considered another class of programs; used to develop, test, and qualify other software products or to aid in engineering design and development; has typical applications as compilers, assemblers, computer-aided-software engineering tools (CASE)

Types of Computer System Errors

The following examples are problems that have been observed with the application of software to control processes and systems.

Space probe.—Clementine 1, which successfully mapped the Moon's surface, was to have a close encounter with a near-Earth asteroid. A hardware or software malfunction on the spacecraft "resulted in a sequencing mode that triggered an opening of valves for four of the spacecraft's 12 attitude control thrusters, allowing all the hydrazine propellant to be used up (ref. 8–8)."

Chemical plant.—Programmers did not fully understand the way a chemical plant operated. The specifications stated that if an alarm occurred, all process control settings were to be frozen. The resulting computer system software released a catalyst into a reactor and began to increase cooling water flow to it. While the flow was increasing, the system received an oil sump, oil low alarm, and froze the flow of cooling water at too slow a rate. The result was that "the reactor overheated and the pressure release valve vented a quantity of noxious fumes into the atmosphere (ref. 8–4)."

Space Shuttle.—An aborted mission nearly occurred during the first flight of Endeavor to rendezvous and repair an Intelsat satellite. The software routine used to calculate rendezvous firings "failed to converge to a solution due to a mismatch

between the precision of the state-vector tables, which describes the position and velocity of the Shuttle (ref. 8–2)."

Airliner.—A laptop computer used by a passenger on a Boeing 747–400 flying over the Pacific caused the airliner's navigation system to behave erratically. When the computer was brought to the flight deck and turned on, "the navigation displays went crazy (ref. 8–9)."

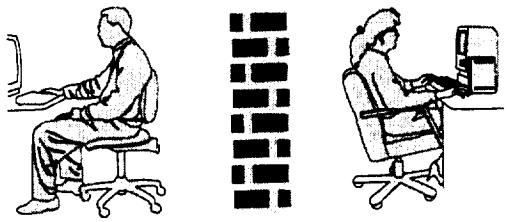


Figure 8–3.—Sources of error based on organizational problems.

Sources of Errors

Investigating the sources of problems should take precedence over finding the errors in the software logic. Anytime an analog and/or an electromechanical control system is replaced by a computer system, many unique problems can occur.

Organizational problems.—Determining the causes of errors and eliminating them requires an analysis of the procedures, organizational arrangements, and methodology that cause problems with software. Figure 3 gives an overview of the following organizational problems:

(1) Communication between the software programmer and the systems or design engineer: The designer does not know the software and the programmer does not know the system with all its potential failure modes (they do not have domain-specific knowledge). Programmers frequently fail to understand the potential for problems if certain actions do not occur in a logical sequence. For example, "start heater and add fluids to boiler" may be "logical" programming sequences, but what if the computer has a fault after the heater is started, before enough fluid is added to the boiler? Similarly, design and safety engineers frequently lack knowledge about specific software, the way it will control the system, and the potential for software problems. They treat the computer and its software as a black box with no regard for the consequences if the unit fails. Consequently, in the past, system safety engineers ignored software or looked at it superficially when analyzing systems.

(2) Documentation standards for software, testing, and verification: Many problems are caused by the practices of not documenting the software analysis and the procedures for inspection, testing, and last-minute fixes without retesting and reverification. Design and verification tools may not exist. Formal procedures for software inspection may not exist or the procedures may be in place but may be essentially ignored by the software development group. For example, a potential flight problem was noticed on one experiment scheduled to fly in space to evaluate the effects of microgravity. To correct it, the software was changed during a preflight checkout on a holiday, but the change was not verified. During the mission, the heaters on a device developed only 25 percent of the needed power because the simple software change caused the loss of some mission data.

(3) Standardization of software structure: In many organizations, not requiring adherence to software standards contributes to many system failures. Trying to be elegant in writing software, using complex techniques, and neglecting internal comments and written documentation can seriously affect the quality of software and decrease its reuse.

(4) Configuration control management over software changes: During software development and maintenance, unauthorized or undocumented changes made by a programmer to fix a possible mistake may cause many problems down the line. Toward the end of a project, pressure to complete the job encourages code changes without proper review or documentation.

(5) Silver bullets: Over reliance on silver bullets to solve a company's software problems results in real issues being overlooked. One of the most difficult problems to deal with is unrealistic hope that an advance in software development technology, a new code-generating tool, or object-oriented super code will make software generation problems disappear. This reliance also manifests itself when state-of-the-art techniques are exclusively relied upon in lieu of using good documentation, formal requirements, and continuous interface between software, design, and safety personnel.

(6) Personnel: A greater attempt to keep good programming talent should be made because a turnover results in a loss of corporate knowledge, reduces the reuse of code, and causes problems with software maintenance.

(7) Software reuse: When existing software could be reused, many software programs are started from scratch (again with little control over how the code is to be written). Note that a careful reuse of codes has saved time and manpower.

Design and requirements problems.—Poor analysis and flowdown of requirements specifications for an individual project can cause errors, delays, and cost overruns:

(1) Requirements: Poorly defined requirements for a specific software project can cause a cost overrun and increase the probability that code logic errors will be introduced. When real-time systems are developed for new applications or applications outside the normal areas of the software engineer's expertise, additional requirements are needed to implement the basic system. Frequently discovered while the software development process is well underway, these requirements are often inconsistent, incomplete, incomprehensible, contradictory, and ambiguous.

(2) Additional features: Adding new features to the software is also major problem and is based on the perception that long after programming has started, requirements for new features can be added with little negative effect. However, such additions to performance requirements adversely affect system software because it must be changed for each new requirement. Because each change risks increasing errors in the function or logic, design engineers must always ask, Have the requirements been analyzed as a complete set?

(3) Anticipating problems: More attention must be given to protecting the software-controlled system from off-nominal environments and to anticipating what states the system can reach through an unexpected series of events. Too often, the emphasis is on fulfilling performance requirements without carefully analyzing what can go wrong with the system.

(4) Software and/or hardware interaction: Problems result from a lack of understanding of how the program will actually run once a system is operational. Software may not be able to process all the sensor data during a clock cycle, or it may be unable to deal with changes in physical conditions and processors.

(5) Isolating processes: Adding too many unnecessary software processes on a computer controlling a safety-critical system can reduce assurance that critical processes will be handled properly (safety critical refers to systems whose failures can cause a loss of life, the mission, or the system).

Other problem areas.—In addition to the associated hardware, sensors, and interfaces that can also increase the risk of errors, other problems concern incorrect data, the reliability of the system, and the production, distribution, and maintenance of software.

(1) Reliability: The reliability and survivability of the computer hardware, sensors, and power supplies are often not adequately planned for. The central processing unit (CPU), memory, or disk drives of a computer can fail, the system can lose power, excess heat or voltage spikes can cause unanticipated errors in performance and output, or the system can completely shut down.

(2) System and/or sensor interfaces: The interfaces between sensors and other mechanical devices can fail, resulting in damage to cables and the failure of power supplies to sensors or servocontrollers. Often the anticipation of these events and effective solutions are not handled adequately.

(3) Radio frequency noise: The effect of radio frequency (RF) noise is often unanticipated. It can cause a computer processor, its memory, and input/output devices to operate improperly, or it can cause errors or erroneous readings from sensors, poorly shielded cables, connectors, and interface boards (e.g., fiber optic to digital conversion).

(4) Manufacture and maintenance: Improper handling of the manufacture, reproduction, and distribution of software results in compilation errors and improper revisions of code being distributed. Integration problems can occur while assembling the code, linking program modules together, and transferring files. Poor control over maintenance upgrades of software and firmware also causes errors from improperly loading programs, using the wrong batch files, and patching to the wrong revision of software.

TABLE 8-3.—SOURCES OF ERRORS BY PERCENT

Logic	21.9
Input/output	14.74
Data handling	14.49
Computation	8.34
Preset data base	7.83
Documentation	6.25
User interface	7.70
Routine-to-routine interface	5.62

A Rome Laboratories study classified errors by percentage of occurrence (table 8–3), which reveals the importance of interface design and documentation (ref. 8–10).

Tools to Improve Software System Reliability and Safety

For each of the aforementioned problem-causing agents, the following tools minimize risk and may even eliminate the problem.

Organizational improvement.—Various tools and techniques properly applied and supported at all organizational levels can greatly improve software reliability and safety.

(1) Communication: Improve communication between designers, software engineers, and safety engineers through concurrent engineering, safety review teams, and joint training. Concurrent engineering with regular meetings between design and software engineers to review specifications and requirements will improve communications. Continuous discussions with the end users will help them to understand the background of the various system performance requirements. Joint training and cross training will encourage them to develop informal relationships and communication. Software safety review committees consisting of design, software, and safety personnel who continually meet to review software specifications and implementation will assure that safety-critical software performs properly and that specifications be carefully written, not just in "legal" terms but with clear descriptions of how the system should work.

(2) Documentation: Improve software documentation standards, testing, and verification procedures. Encourage the application of standards for all software projects, including general requirements for all system development projects, the industry or military standards to be followed, and the documents to be generated for a specific product. These documents

may include a software version description document (see part II for more details) and plans for software management, assurance, configuration management, requirements specifications, and testing.

(3) Standardization: Set and enforce software structure standards to delineate what is and what is not allowed. The programmer should not design a "clever" program that cannot be readily understood or debugged. Enforce safe subsets of programming language, coding standards, and style guides.

(4) Configuration management: Implement consistent controls over software changes and the change approval process by using software development products that include software configuration management and code generation tools. Computer-aided-software-engineering (CASE) tools and other configuration management techniques can automatically compare software revisions with previous copies and limit unapproved changes. Other programming tools provide mission simulation and module interface documentation.

(5) Silver bullets: The introduction of major changes in the procedures for generating software must be scrupulously reviewed and their impact on the software personnel, maintenance, and standardization evaluated carefully. Major disruptions to personnel can result from any major change in the way a product is designed and developed; therefore, careful and complete training of personnel, a free flow of information about the new system, assurances as to the support of existing programmers, and the gradual introduction of the new methods (e.g., starting on one small project) are required. Projects already underway and those scheduled to begin may or may not benefit from the changes.

(6) Personnel: Provide incentives to keep good programming talent and maintain the corporate knowledge base. The programmers should have a mix of programming skills and experience and the ability to transmit practical programming knowledge to new programmers who only have classroom training with little or no insight into real-world problems. Keeping senior programmers or senior managers who can review software and participate in independent verification and validation (IV&V) of software across missions or products is also beneficial, as is retaining workers who know the software systems that support software maintenance and new applications of the code. Provide training in the proper methodologies.

Software should be modularized to facilitate changes and maintenance. The modules should have low coupling (the number of links between modules is minimized) and have high cohesion (the level of self-containment).

Use a "clean room approach" to develop software. This approach implies a highly structured programming environment with tight control of the specifications for the software and system and support and adherence to the software analysis specifications.

(7) Software reuse: Encourage the reuse of software with strict controls imposed over software structure and procedures for code reuse. Software modules and/or software reuse also improve reliability because of the benefits derived from faults removed in prior usage. Modularized software with well-documented and verifiable inputs and outputs also enhances maintainability. Lewis Research Center's launch vehicle programs are reused for each mission with only minor modifications, and excellent reliability results have been achieved.

Design and requirements improvements.—The hardware and the software must be integrated to work together. This integration includes the entire system with input sensors and signal conditioners, analog-to-digital (A/D) boards, the computer hardware and software itself, and the output devices (control actuators). Basic design methodology can improve software as well; thus, the following approaches support this concept:

(1) Requirements: Spend sufficient time defining and understanding requirements. The system, software, and safety engineers should work with the end user to develop requirements, to express the requirements in mutually understandable language, and to design requirements that are testable and verifiable.

(2) Additional features: Limit changes in requirements once the software design process begins. Question whether an additional feature is really necessary or if, instead, functionality should be reduced to achieve safety and basic performance goals. A large number of ancillary noncritical devices and special graphical user interfaces may not be necessary and may only complicate and slow the system.

Avoid developing a false sense of security by putting software in its proper place of importance. Erroneously, many people think that a computer controlling a system can never fail and will believe computer-controlled readouts rather than rely on their own good senses.

(3) Anticipating problems: Fully analyze the ways the software-controlled system can fail and the undesirable states the system can attain. Then, implement procedures and methods to ensure that these undesirable states and failure modes cannot be attained and that they are not attainable through some unusual (though not impossible) combinations of software states, environment, and/or input data. Such steps will ensure the system's invulnerability to these failures.

Use error detection, correction, and recovery software development to achieve fault tolerance. Examples of common errors include inconsistent data in data bases, process deadlock, starvation and premature termination, runtime failures due to out-of-range values, attempts to divide by zero, and lack of storage for dynamically allocated objects. Although software does not degrade, it is virtually impossible to prove the correctness of large, complex, real-time systems. The selective use of logic engines can be effective in reducing uncertainty about a system's performance.

Use software that can detect and properly handle runtime errors and software controls that assume the worst and prepare

for it, such as undesirable states the computer can attain and the ways each of these states can be prevented. Make a careful analysis of responses to failed or suspect sensors.

Software capable of real-time diagnosis of its own hardware and sensors is very useful. Memory can be protected with parity, error-correcting code, and read-only circuitry in memory. Messages received should be checked for accuracy, and routes can be automatically changed when errors are detected. Predefined system exceptions and user-defined fault exceptions should be designed into application software. Predefined exceptions can be raised by runtime systems so the software should also have built-in or operating system recovery procedures. Information for recovery includes processor identification, process name, data reference, memory location, error type, and time of detection.

(4) Software and/or hardware interaction: Computer timing problems and buffer overload problems must be eliminated. If all alarms and sensors cannot be read in one clock cycle of the CPU, errors may occur or alarms may be missed. Overloaded buffers can result in CPU lockup.

Load balancing should be a part of the operating system software routines because failures are often caused by overloading one or more processors in the system. A few examples of overloading are caused by an increase in message traffic or the inability of a processor to perform within time constraints. In these cases, a potential tool to support complex systems is dynamic traffic time sharing in which message streams are distributed among identical processors with a traffic coordinator keeping track of the relative load among processors.

(5) Isolating processes: Systems for safety-critical applications need to be separate from everything else. System specifications often require gathering data from hundreds of sensors and performing all sorts of noncritical tasks. Segregating these noncritical tasks in a separate computer system will often improve chances that safety-critical functions will be not be disrupted by defects in noncritical resources. Safety-critical modules should be "firewalled," and proven hardware and technology should be used for critical systems. "Flight-proven" older computer systems and software that do the job should be chosen over newer computers whose standards are rapidly evolving where critical applications are involved.

Analog interlocks on safety-critical systems should be replaced with software interlocks only with the greatest of care. A thorough, well-documented analysis of what would happen with a computer failure and with a system failure that the interlock protects should also be made. An example of the problem of replacing mechanical interlocks with software interlocks involves a radiation therapy machine. An early model of the therapy machine had a hardware interlock to prevent radiation overdoses. When the interlock was removed on a later model and replaced with software logic, several people were killed from a radiation overdose. The problem was caused by the operator interface, poorly documented data input procedures, and inadequate safety procedures. The earlier model never experienced the problem because the program did not control the interlock (ref. 8–11).

In many cases, safety-critical systems can have an analog process (or a stand-alone computer) capable of taking over if the primary computer fails. If a computer control fails on a process plant, an analog backup system (which is presumably controlled by the computer) could keep the process running (though at less than optimum conditions). Alternatively, control actuators could go to a safe position if a failure occurred. Usually, the process must be allowed to proceed to some nominal conditions (e.g., partial cooling water or partial product inflow into a process) before shutting down.

Monitor the health of the backup systems and the output of software control commands independently of the main control computer. A separate computer should be performing health checks on the main computer and on safety-critical sensor outputs.

Conduct special tests to verify the performance of safety-critical software. This testing should verify that the software responds correctly and safely to single and multiple failures or alarms; that it properly handles operator input or sensor errors (e.g., data from a failed sensor); that it does not perform any unintended routines; that it detects failures and takes action with respect to entry into and execution of safety-critical software components; and that it is able to receive alarms or other inhibit commands.

Formal methods can use abstract models and specification languages to develop correct requirements. Logic engines can be used to prove the correctness of the requirements.

For many years, the Lewis Research Center's launch vehicle program verified the software for each mission by running the complete program in the mission simulation lab. All the mission constants and components were checked and verified. Lewis never lost a vehicle because of software problems.

Other improvements.—The hardware/software system must also be integrated with input sensors and signal conditioners (e.g., analog-to-digital boards) and the output devices (e.g., servocontrolled actuators). Because the reliability of all this hardware is also an issue, some basic approaches to total system performance follow:

(1) Reliability: The reliability and survivability of the electronic components associated with the software control system can be improved by properly protecting components from vibration, excess heat and voltage, and current spikes. Properly maintained grounding and shielding also must be assured with maintenance training and documentation. Robust sensors, actuators, and interfaces also contribute to a more reliable system. Sensor failure can cause the wrong data to be processed. Even the fraying of cables has been linked with possible uncontrolled changes in aircraft flight surface actuation. The

reliability of computer-controlled output devices (servo-actuators, valves, relays) must also be verified. Because output devices may be subject to noise problems, error recovery and restart procedures should be included in software and properly tested.

Passive controls should be designed so that failures cause the system to go to a safe state. If input commands or sensor readings are suspect, the system should go to a safe condition, which is accomplished by an analog backup or an autonomous software module that should be in a separate backup system.

Multiple voting systems (multiple computers running the same task in parallel with independently written programs) might help to improve reliability. Although this concept is beneficial in theory, some studies suggest that common software logic faults arise from common requirements. Furthermore, maintenance and configuration management of this type of system is greatly complicated by having different active versions of code (ref. 8–12). Multiple computers with software written for the same functional output but developed independently is one way to handle the critical problem of software taking the operator to a condition that was never intended. Systems should sense the occurrence of anomalies and alert the operator. Health monitoring of the controlled system and the computer itself, including frequent self-checks, should be part of the program.

Redundant systems need to have separate power sources and locations (to avoid common mode failures). Use uninterrupted power supplies for critical software systems. Have battery backup for as long as needed to switch to manual operation. Avoid a common power supply that can send a surge to all devices at once or can shut off all devices at once.

A distributed system can also be used to improve reliability. The system can sense problems in one processor and transfer its work to another processor or system. Hardware components degrade with time and represent the most important factor in ensuring reliability of real-time systems. However, note that the complexities of a distributed system can cause new problems that possibly reduce reliability. For example, the synchronization and precision of numerical values between programs and communications procedures can cause errors. More resources are also consumed for coding and testing and programs become larger (with more chance for error).

(2) System and/or sensor interfaces: The computer and sensor interfaces must be thoroughly tested to prevent mechanical failures, intermittent contacts, connector problems, and noise. Again, provisions for data out of acceptable ranges must be made.

(3) Radio frequency noise: Radio frequency (RF) noise problems can be avoided. Input and output data should be validated before use. The software should check for data outside valid ranges and take appropriate action such as setting off an alarm or shutting down the system. Proper maintenance procedures and training in the removal and replacement of grounding and shielding should be developed. The interaction of and possible need for separate analog and digital grounds should also be investigated. Thorough system testing in all anticipated environments should be performed.

(4) Manufacturing and maintenance: The duplication, loading, and maintenance of software must be planned and controlled. Procedures must be developed to assure that the proper code is loaded on each processor model. All new compilations of code must be verified. Buggy compilers can introduce defects. Subtle changes from one revision of an operating system to another can cause a difference in response to the same code. Procedures and requirements for maintenance upgrades must also be developed. The updated software should be adequately tested and verified (to the same level and extent and to the same requirements as the operating software) for accuracy (performance), reliability, and maintainability. New software should be modularized and uploaded as individual modules when maintenance is being performed. Also, whenever possible, issue firmware changes as fully populated and tested circuit cards (not as individual chips).

Software Development Tools

Several methods can be used to analyze and verify software.

Fault tree analysis.—This can identify critical faults and potential faults or problems. Then, all the conditions that can lead to these faults are considered and diagrammed.

Petri net analysis.—This provides a way to model systems graphically. A Petri net has a set of symbols that show inputs, outputs, and states with nodes that are either "places" (represented by circles) or "transitions" (represented by vertical lines). When all the places with connections to a transition are marked, the net is "fired" by removing marks from each input place and adding a mark to each place pointed to by the transition (the output places) (ref. 8–4).

Hazard analysis.—This uses formal methods to identify hazards and evaluate software systems (ref. 8–11).

Formal logic analyzers.—These are logic engines that can verify specifications. Some source analyzers can reveal logic problems in code and branching problems.

Pseudocodes.—These are used for program design and verification. They are similar to programming languages but are not compiled. They have the flow and naming notation of programming language but have a readable style that allows someone to better understand program logic (ref. 8–4).

State transition diagrams (STD's).—These are graphs that show the possible states of the system as nodes and the possible changes that may take as lines. They can highlight poor architecture or unnecessarily complex computer code (ref. 8–4).

Software failure mode effects analysis (FMEA).—These analyze what can go wrong with the software and with the system itself. The FMEA should analyze whether the system is

fault tolerant with respect to hardware failures and make certain that the system specifications are complete. The actual failure of the computer hardware usually results in a hard failure and the effects are easily identified. However, the effects of failures handled by software may not be so clear. For example, how does the software handle the loss of one piece of sensor data or a recovery from a fault?

Software Safety Axioms and Suggestions

These axioms should be read and reread and the principles behind them thoroughly understood.

(1) Persons who design software should not write the code and those who write the code should not do the testing.

(2) Accidents are caused by incomplete or wrong assumptions about the system or process being controlled. Actual coding errors are less frequent perpetrators of accidents.

(3) Unhandled controlled system states and environmental conditions are a big cause of "software malfunctions."

(4) The lack of up-to-date professional standards in software engineering and/or the lack of the use of these standards is a root cause of many problems.

(5) Changes to the original system specifications should be limited.

(6) It is impossible to build a complex software system to behave exactly as it should under all conditions.

(7) Software safety, quality, and reliability are designed in, not tested in.

(8) Upstream approaches to software safety are most effective.

(9) Software alone is neither safe nor unsafe.

(10) Many software bugs are timing problems that are difficult to test for.

(11) Software often fails because it goes somewhere that the programmer does not think it can get to.

(12) Software systems do not work well until they have been used.

(13) Mathematical functions implemented by software are not continuous functions but have an arbitrary number of discontinuities.

(14) Engineers believe one can design "black box tests" on software systems without the knowledge of what is inside the box.

(15) Safety-critical systems should be kept as small and as simple as possible; any functions that are not safety critical should be moved to other modules.

(16) A software control system should be treated as a single-point failure (in the past the software was often ignored).

(17) What must not happen should be decided at the outset and then one should make sure that the program cannot get there.

(18) The system should be fault tolerant and able to recover from faults and instruction jumps.

(19) Independent verification and validation (IV&V) of software should be used.

Conclusions

Software is now used in many safety-critical applications and each system has the potential to be a single-point failure or to be zero fault tolerant; that is, a single failure will cause the system to fail or if a computer is controlling a hazardous function, a single failure can cause a hazardous condition to exist.

Potential problems with software are not well understood. Computers controlling a system (the computer hardware, the software, the sensors, and output devices that direct the flow of energy) are not a black box that can be ignored in a safety, reliability, or risk evaluation. However, if handled and applied properly, software and hardware may be used to control a system and thus can be a valuable design option.

The software development process can be improved by good communication, documentation, standardization, and configuration management. Other major factors in proper software development are correct and understandable requirements. Factors that help to improve confidence in the system are anticipating problems, properly handling errors, and improving hardware reliability. Methods to validate and improve software quality (and safety) are discussed in part II.

Part II—Software Quality and the Design and Inspection Process

Software Development Specifications

Improving software with standards and controls must include the following:

- Robust design: making software fault tolerant
- Process controls: standardizing the software development process
- Design standards: standardizing the software specifications
- Inspection: standardizing the software requirements inspection process
- Code inspection: standardizing the software code inspection process

Precise and easily readable documentation and specifications are necessary for a successful software project. Ideally, formal methods and specifications language should be used and once written, must be understood and adhered to. To accomplish this process requires team participation in document and specification generation and also real support of the specifications, documentation, and the verification of software conformance and validation by upper management and the team.

Some of these documents and related practices should include

(1) A formal software management plan that includes the software development cycle, the configuration management plan, approval authority, and group charter and responsibilities. (This plan would specify what other documentation is required, how interfaces are to be controlled, and what the quality assurance and verification requirements are.)
(2) A formal software design specification that includes architecture specifications and hardware interfaces
(3) A software development plan that describes development activities, facilities, personnel, activity flow, and the development tools for software generation
(4) A plan for formal inspection of software that includes
(a) a software quality assurance plan to integrate hardware and software safety, quality, and reliability
(b) a software verification test specification
(c) a software fault tolerance and failure modes and effects analysis specification
(5) A software safety program plan that includes a software safety handbook and reliability practices specifications
(6) A formal plan for maintenance and operation
(7) Configuration management and documentation plans that specify recording all changes to software and the reasons for the changes. (Records should include design changes that require software modifications or any change in the functional capabilities, performance specifications, or allocation of software to components or interfaces.)
(8) Interface control documentation that specifies linking hardware and software, vendor-supplied software, and internally generated software
(9) Failure review boards to review bugs, the bug removal process, and the overall effect of bugs on the system
(10) Lessons learned to be used to document problems and the solutions to eliminate repetition of errors
(11) Test plans that will, to the greatest extent possible, validate the software system

Once these documents are developed and the procedures set up, they must be implemented, enforced, and maintained. A software system safety working team (multidisciplined) can assist software engineering and continually monitor adherence to the documentation. They also have to engender respect for the need to follow the specifications, not mandate them and walk away. Therefore, the team and software engineering management must educate programmers in the understanding and use of specifications (ref. 8–13).

Specifications and Programming Standards

Structured programming with a well-defined design approach and extensive commenting benefits the software design process. Standardizing formats, nomenclature, language, compilers, and platforms for the software contributes to project success as well. Besides many excellent internal company standards for software development, a number of documents exist to help in the standardization and to gauge the maturity of software development. Some of these documents are

(1) The Software Engineering Institute (SEI) Capability Maturity Model (CMM) is a method for assessing the software engineering capabilities of development organizations. It evaluates the level of process control and methodology in developing software and is designed to rank the "maturity" of the company and its ability to undertake major software development projects.
(2) ISO 9000–3 Software Guidelines, Part 3, Guidelines for the application of ISO 9001 to the development, supply, and maintenance of software is intended to provide suggested controls and methods.
(3) IEEE Software Engineering Standards Collections include 22 standards (1993 edition) covering terminology, quality assurance plans, configuration management, test documentation, requirements specifications, maintenance, metrics, and other subjects.
(4) NASA-developed software standards include NSS 1740.13, INTERIM, June 1994, NASA Software Safety Standards that expands on the requirements of NASA Management Instruction (NMI) 2410.10, NASA Software Management Assurance and Engineering Policy. These documents contain a detailed reference document list.
(5) DOD Standards include MIL–STD–882C, System Safety Program Requirements (ref. 8–7), DOD–STD–2167A, Defense System Software Development (MIL–STD–498) (ref. 8–14), software development (e.g., ref. 8–15), and Documentation, and numerous other standards and guidelines (for reference only).

NASA Software Inspection Activities

We now want to focus on one area of the software documentation, testing, inspection, and qualification process: the software inspection activity. This inspection process includes (1) metrics, (2) software inspection training, and (3) formal software inspection. Inspection activities include

- Implementation of requirements
- Review of pseudocode
- Review of mechanics
- Review of data structure
- "Walkthrough" of code
- Verification and validation
- Independent verification and validation

The objectives of formal inspection include (1) removing defects as early as possible in the development process, (2) having a structured, well-defined review process for finding and fixing defects, (3) generating metrics and checklists used to

improve quality, (4) following total quality management (TQM) techniques such as working together as a team, and (5) taking responsibility for a work product shared by the author's peers.

To achieve these objectives, specifications must be reviewable, formally analyzable, and usable by the designers and the assurance and safety engineers. Furthermore, the specifications must support completeness and robustness checks and they must support the generation of mission test data.

Formal design requirements and inspections.—The objective of inspection is to remove defects at the earliest possible point in the product development life cycle. The product can be a document, a process, software, or a design. Inspection topics include requirements, design requirements, detailed design requirements, source code, test plans, procedures, manual standards, and plans.

Inspection is a very structured process which requires that team members, who are involved because of their technical expertise, be sincerely interested in the software product. Rather than being viewed as a fault-finding mission, the inspection should be considered a tool to help the author identify and correct problems as early as possible in the development process. The inspection should also help to foster a team environment by emphasizing that everyone is involved to develop a high-quality product.

Metrics (minor errors discovered, major errors discovered) generated during this process are used to monitor the type of software defects discovered and to help prevent their recurrence (refs. 8–16 to 8–18).

Process overview.—Staff, procedures, development time, and training are applied to a developing software product to improve its quality. The formal seven-step program for inspection includes

(1) The planning phase: organizing for the inspection
(2) The training phase: background and details of the inspection activity given to team members
(3) The preparation phase: review of the work by individual inspectors prior to the joint inspection meeting
(4) The inspection meeting: defects identified, classified, and recorded by the team
(5) The "third hour" (cause phase): offline discussions held by programmers to get help with defects
(6) The rework phase (corrective action): defects corrected by programmers
(7) The followup phase: revisions reviewed and verified by the team

Roles.—Each person who participates in the inspection performs various tasks:

- Moderator: coordinates the inspection process, chairs the inspection meetings, and ensures that the inspection is conducted
- Reader: presents the work product to the inspection team during the meeting (the programmer (author) does not give the presentation)
- Recorder: documents all the defects, open issues, and action items brought forward during the meeting
- Inspector: helps to identify and evaluate defects (the responsibility of every person at the meeting)

Development process benefits.—Some of the benefits of formal inspection for the overall software development process are that it

- Improves quality and saves cost through early fault detection and correction
- Provides a technically correct base for the following development phases
- Contributes to project tracking
- Improves communication between developers
- Aids in project education of personnel
- Provides structure for in-process reviews

Inspection also benefits the software developer in a number of ways:

- Reduces defects made by the author because they are identified early in the product life cycle
- Identifies efficiently any omissions in the requirements
- Provides constructive criticism of and guidance to the programmer by the inspection team in private rather than by tearing down software in open public project design reviews
- Provides a constructive atmosphere for the entire team because of lessons learned from others' mistakes
- Implements improved project tracking with inspection milestones embedded in the project
- Improves understanding of the overall project and engenders communication and teamwork by bringing together project persons from varied backgrounds
- Trains new members of the software development team by working with senior team members

Figure 4 presents the waterfall flowchart of the software development process (based on phases in MIL–STD–498, Defense System Software Development, ref. 8–14). The following acronyms are used:

CDR	critical design review
CSCI	computer software configuration item (major computer software PROGRAM)
CSU	computer software unit (program module)
FCA	functional configuration audit
I	software inspections
IV&V	independent verification and validation activity

PCA physical configuration audit
PDR preliminary design review
SDR system design review
SRR system requirements review
SSR software specification review
SW computer software
TRR test readiness review
V&V verification and validation activity

Basic rules of inspection.—These basic rules must be followed if the software inspection process is to be effective:

(1) Inspections are in-process reviews conducted during the development of a product in contrast to milestone reviews conducted between development phases.

(2) Inspections are conducted by a small peer team, each member of which has a special interest in the project success.

(3) Managers are not involved in the inspection and its results are not used as a tool to evaluate developers.

(4) The moderator leads the inspection and must have received formal training to do so.

(5) Each team member, in addition to being an inspector, is assigned a specific role.

(6) The inspection is spelled out in detail and no step of the process is omitted.

(7) The overall time of the inspection is preset to aid in meeting the schedule.

(8) Checklists are used to help identify defects.

(9) Inspection teams should work at an optimal rate, the object of the meeting being to identify as many defects as possible—not to cover as many pages as possible.

(10) Inspection metrics are defect type, number, and time spent on inspections. These metrics are used to improve the development process and the work product and to monitor the inspection.

Results of software inspections.—Formal inspections save costs because fixing defects early in the development cycle is less costly than removing them later; they train team members and provide them with a valuable development tool as lessons learned from their participation in the bug identification and removal process; and they improve developer and development efficiency and lead to higher quality.

Fixing a defect found through inspections costs on the average less than 1 hour per defect; fixing a defect found during software testing typically takes from 5 to 18 hours. Another cost factor is that defects tend to amplify. One defect in requirements or design may impact multiple lines of code. For example, a small study conducted by the Jet Propulsion Laboratory (JPL) found an amplification rate of 1 to 15, which means that 1 defect in the requirements impacts 15 source lines of code (SLOC), as seen in figure 5 (information taken from ref. 8–19).

Inspections were also used at IBM Federal Systems to develop software for the space shuttle. The original defect rate of 2.25 defects per thousand lines of code (KLOC) was unacceptable. Over a 3-year period, inspections were applied on requirements, design, code and test plans, specifications, and procedures. The goal for this effort was 0.2 defect per KLOC. With inspections, the project was able to surpass the goal and attain a defect rate of 0.08 defect per KLOC.

One of the most essential lessons learned from the initial implementation of the inspection process is that all inspection participants require some type of training. Everyone needs to understand the purpose and focus of inspections and the resources required to support the process. Adequate time has to be provided for inspections in the software development process. Furthermore, using metrics from inspections provides an excellent basis for monitoring both the inspection and development process and for evaluating process improvements.

Another lesson learned is that a formal inspection requires projects to have an established development life cycle, an established set of documents produced during the phases of the life cycle, programming standards, and software development standards (e.g., NASA Software Assurance Standard, NASA–STD–2201–93, which states that "Software verification and validation activities shall be performed during each phase of the software life cycle and shall include formal inspections.").

Additional benefits of formal inspections to the project are that they can be used with any development methodology because no matter which development process or life cycle is used, products being produced can be inspected; they are applied during the development of work products and are a compliment to milestone or formal reviews but are not intended to replace them; they are recommended by the NASA Software Assurance Standard and can be applied to the work products called out in the NASA Software Documentation Standard (refs. 8–20 to 8–22).

Additional Recommendations

On the basis of an evaluation of the space shuttle software development process, the following recommendations were made (ref. 8–13):

(1) Verification and validation (V&V) inspections by contractors should pay close attention to off-nominal cases (crew and/or ground error, hardware failure, software error conditions), should focus on verifying the consistency in the levels of descriptions for modules with the consistency in module requirements and the design platform, should assure correctness with respect to the hardware and software platforms, and should maintain the real independence of independent verification and validation (IV&V).

(2) The project should have sufficient personnel trained in system reliability and quality assurance (SR&QA) to support

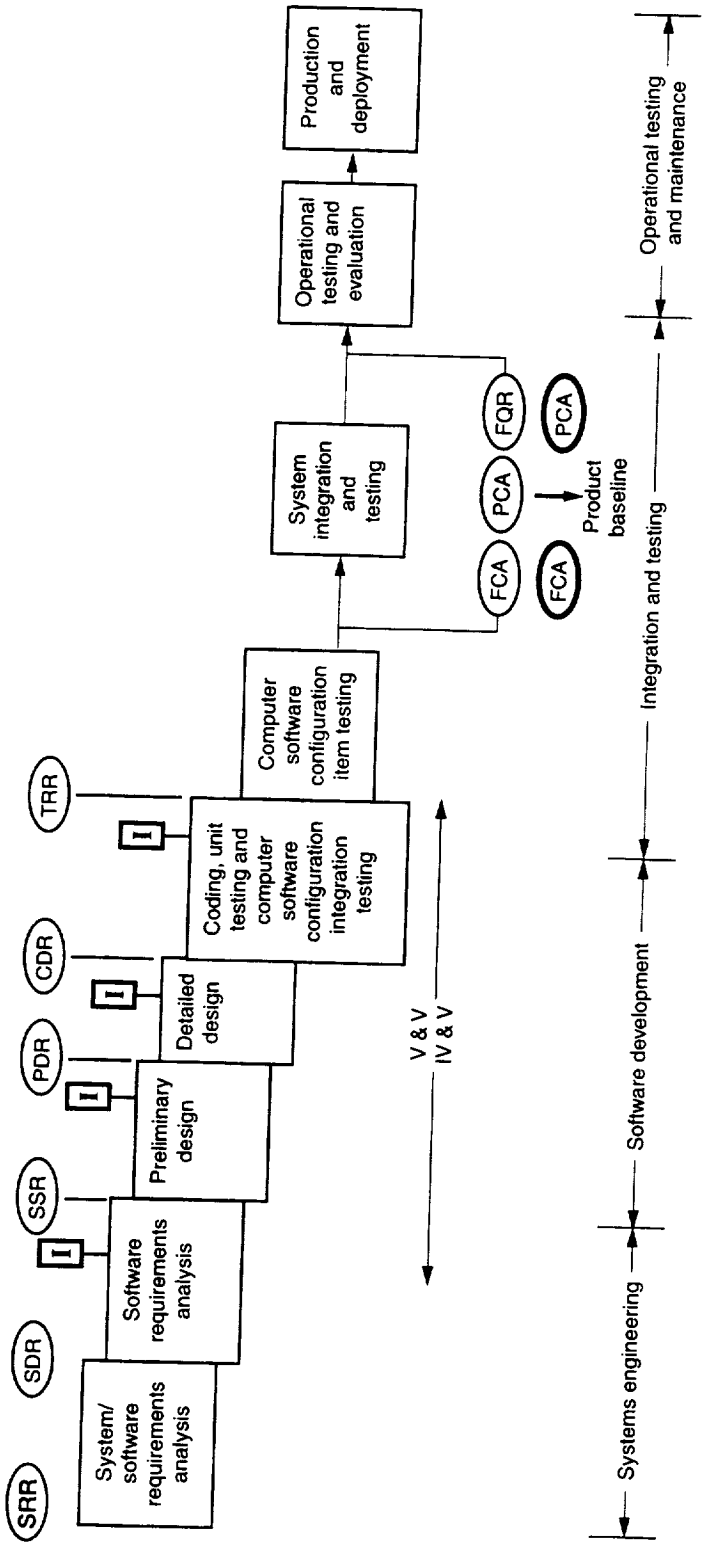


Figure 8–4.—Flowchart for software development process.

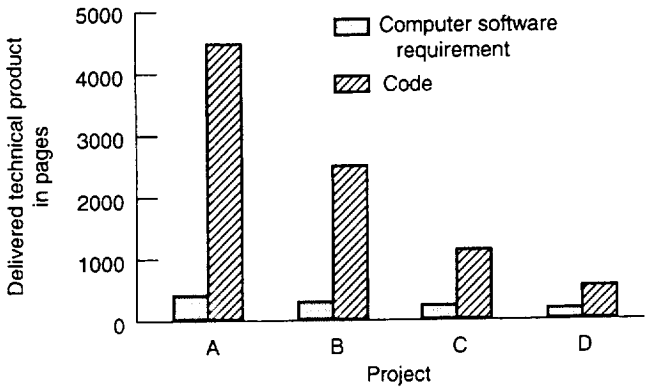


Figure 8–5.—Amplification of requirements into source code. Average amplification ratio, 1:15 (from ref. 8–19).

software-related activities and provide oversight and evaluation of software development activities by the individual SR&QA offices.

(3) The same standards and procedures should be provided and enforced for multiple centers on the same program. Consistent software development coding guidelines should be provided to contractors.

(4) Visibility for potential software problems should be provided by defining detailed procedures to report software reliability, quality assurance (QA), or safety problems to the program-level organization.

(5) Accepted policies and guidelines should be provided for the development and implementation of software V&V, IV&V, assurance, and safety. These should include a well-documented maintenance and upgrade process.

(6) Sufficient resources, personnel, and expertise should be provided to develop the required standards. Also, sufficient resources, manpower, and authority should be used to compel development contractors to verify that proper procedures were followed.

(7) Lessons learned in the development, maintenance, and assurance of software should be recorded for use by other programs (refs. 8–23 to 8–26).

(8) The information that each development and oversight contractor is responsible for making available to the community as a whole should be precisely identified. Mechanisms should be in place to ensure that programs be given all information needed to make intelligent implementations of software oversight functions.

Conclusions

The overall software design process will be improved by carefully constructing the initial documentation to generate real and usable requirements. Requirements must be capable of being verified by inspection and test.

Software product assurance activities include formal inspection, production-quality metrics, software inspection training, a code "walkthrough," verification and validation, and independent verification and validation. These activities are making NASA projects more successful.

References

8–1. Halfhill, T.R.: The Truth Behind the Pentium Bug. BYTE, vol. 20, no. 3, Mar. 1995.

8–2. Gray, E.M.; and Thayer, R.H.: Requirements. Aerospace Software Engineering, A Collection of Concepts, C. Anderson and M. Dorfman, eds., AIAA, Washington, DC, 1991, pp. 89–121.

8–3. F–22 Software on Track With Standard Processes. Avia. Week Space Technol., vol. 143, no. 4, July 1995, pp. 53–54.

8–4. Norris, M.; and Rigby, P.: Software Engineering Explained. Wiley, New York, NY, 1992.

8–5. Beatson, J.: Is America Ready to 'Fly by Wire'? Washington Post, sec. C, April 2, 1989, p.1.

8–6. Mecham, M.: Autopilot Go-Around Key to CAL Crash. Aviat. Week Space Technol., vol. 140, no. 19, May 1994, pp. 31–32.

8–7. System Safety Program Requirements. MIL–STD–882C, Jan. 1993.

8–8. Asker, J.R.; and Lenorovitz, J.M.: Computer Woes Spoil Clementine Flyby Plan. Aviat. Week Space Technol., vol. 140, no. 20, May 1994, p. 58.

8–9. Begley, Sharon: Mystery Stories at 10,000 Feet. Newsweek, vol. 122, July 1993, p. 50.

8–10. Addy, E.A.: A Case Study on Isolation of Safety-Critical Software. Proc. 6th Annual Conference on Computer Assurance, NIST/IEEE, 1991, pp. 75–83.

8–11. Leveson, N.: Safeware: System Safety and Computers. Addison-Wesley, Reading, MA, 1995.

8–12. Knight, J.C.; and Leveson, N.G.: An Experimental Evaluation of the Assumption of Independence in Multiversion Programming. IEEE Trans. Software Eng., vol. SE–12, no. 1, Jan. 1986, pp. 96–109.

8–13. Soistman, E.C.; and Ragsdale, K.B.: Impact of Hardware/Software Faults on System Reliability. Report No. OR 18, 173, Griffiths Air Force Base, Rome Air Development Center, Rome, NY, 1985.

8–14. Defense System Software Development. MIL–STD–498, Dec. 1994.

8–15. Software Engineering Handbook. General Electric Company, Corporate Information Systems, McGraw-Hill, New York, NY, 1986.

8–16. Brooks, F.P., Jr.: No Silver Bullet: Essence and Accidents of Software Engineering. Comput., vol. 20, no. 4, 1987, pp. 10–19.

8–17. Fagan, M.E.: Advances in Software Inspections. IEEE Trans. Software Eng., vol. SE–12, no. 7, July 1986, pp. 123–152.

8–18. Clardy, R.C.: Sneak Analysis: An Integrated Approach. Third International System Safety Conference Proceedings, 1977, pp. 377–387.

8–19. Kelly, John C.; Sherif, Joseph S.; and Hops, Jonathan: An Analysis of Defect Densities Found During Software Inspection. J. Systems Software, vol. 17, no. 2, Feb. 1992, pp. 111–117.

8–20. Software System Safety Handbook. AFISC SSH 1–1, U.S. Air Force Inspection and Safety Center, Norton Air Force Base, CA, 1985.

8–21. Leveson, N.: Safety Analysis Using Petri Nets. IEEE Trans. Software Eng., 1986.

8–22. Mattern, S.: Confessions of a Modern-Day Software Safety Analysis. Proceedings of the Ninth International System Safety Conference, Long Beach, California, June 1989.

8–23. Goel, A.L.; and Okumoto, Kayudina: A Time Dependent Error Rate Model for Software Reliability and Other Performance Measures. IEEE Transaction on Reliability, no. R–28, 1979, pp. 206–211.

8–24. Sagols, G.; and Albin, J.L.: Reliability Model and Software Development: A Practical Approach. Software Engineering: Practice and Experience, E. Girard, ed., Oxford Publishing, Oxford, England, 1984.
8–25. Reliability Qualification and Production Approval Tests. MIL–STD–781, Oct. 1986.
8–26. Myers, G.: Software Reliability: Principles and Practice. Wiley, New York, NY, 1976.

Reliability Training

Read the "Reference Document for Inspection: 'Big Bird's' House Concept" (found at the end of ch. 7). The class meeting exercise explains what has to be done and the reference document explains the system requirements. The "'Big Bird's' Requirements Checklist" gives the classifications for the inspection. Complete the "'Big Bird's' Formal Inspection Subsystems Requirements," and send it to the instructor to grade. A score of 70 percent correct will qualify you for a certificate (e.g., item 1, 2-acceptable, item 3-squak, a cubic is about 17 inches, major, wrong, correctness, system).

Chapter 9
Software Quality Assurance

Concept of Quality

Let us first look at the concept of quality before going on to software quality. The need for quality is universal. The concepts of "zero defects" and "doing it right the first time" have changed our perspective on quality management from that of measuring defects per unit and acceptable quality levels to monitoring the design and cost-reduction processes. The present concepts indicate that quality is not free. One viewpoint is that a major improvement in quality can be achieved by perfecting the process of developing a product. Thus, we would characterize the process, implement processes to achieve customer satisfaction, correct defects as soon as possible, and then strive for total quality management. The key to achieving quality appears to have a third major factor in addition to product and process—the environment. People are important because they make the process or the product successful. Figure 9–1 represents the union of these three factors.

The term "software quality" is defined and interpreted differently by the many companies involved in producing programming products. To place the subject in perspective, we present principles and definitions for software quality from several source materials:

(1) The purpose of software quality assurance is to assure the acquisition of high-quality software products on schedule, within cost, and in compliance with the performance requirements (ref. 9–1).

(2) The developer of a methodology for assessing the quality of a software product must respond to various needs. There can be no single quality metric (ref. 9–2).

(3) The process of assessing the quality of a software product begins when specific characteristics and certain of the metrics are selected (ref. 9–3).

(4) Software quality can be defined as (a) the totality of features and characteristics of a software product that bear on its ability to satisfy needs (e.g., conform to specifications), (b) the degree to which software possesses a desired combination of attributes, (c) the degree to which a customer or user perceives that software meets his or her expectations, and (d) the composite characteristics of software that determine the degree to which the software in use will meet the expectations of the user.

We can infer from these statements and other source materials that software quality metrics (e.g., defects per 1000 lines of code per programmer year, 70 percent successful test cases for the first 4 weeks, and zero major problems at the preliminary design review) may vary more than hardware quality metrics (e.g., mean time between failures (MTBF) or errors per 1000 transactions). In addition, software quality management has generally focused on the process whereas software reliability management has focused on the product. Since processes differ for different software products, few comparative benchmarks are available. For hardware in general, benchmarks have been available for a long time (i.e., MIL–HDBK–217E series (ref. 9–4) for reliability). Recently, Rome Air Development Center (RADC), the sponsor of MIL–HDBK–217E, sponsored a software reliability survey that was intended to give software quality the same status as that of hardware.

The next step is to discuss the process of achieving quality in software and how quality management is involved. The purpose of quality management for programming products is to ensure that a preselected software quality level be achieved on schedule and in a cost-effective manner. In developing a quality management system, the programming product's critical life-cycle phase reviews provide the reference base for tracking the achievement of quality objectives. The guidelines for reliability and maintainability management of the International Electrotechnical Commission (IEC) system life-cycle phases follow:

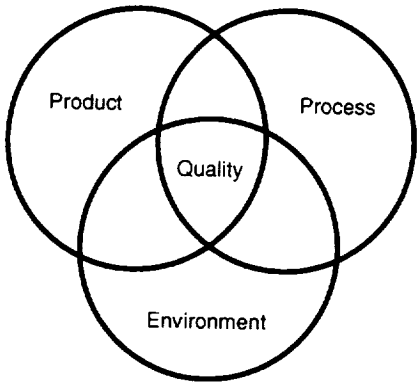


Figure 9–1.—Quality diagram.

(1) Concept and definition: The need for the product is decided and its basic requirements defined, usually in the form of a product specification agreed upon by the manufacturer and user.

(2) Design and development: The product hardware and software are created to perform the functions described in the product specification. This phase will normally include the assembly and testing of a prototype product under laboratory-simulated conditions or in actual field trials and the formulation of detailed manufacturing specifications and instructions for operation and maintenance.

(3) Manufacturing, installation, and acceptance: The design is put into production. In the case of large, complex products, the installation of the product on a particular site may be regarded as an extension of the manufacturing process. This phase will normally conclude with acceptance testing of the product before it is released to the user.

(4) Operation and maintenance: The product is operated for the period of its useful life. During this phase, essential preventive and corrective maintenance is performed, product enhancements are made, and product performance is monitored. The useful life of a product ends when its operation becomes uneconomical because of increasing repair costs, it becomes technically obsolete, or other factors make its use impractical.

(5) Disposal: The product reaches the end of its planned useful life or the requirement no longer exists for the product, so it is disposed of, destroyed, or modernized, if economically feasible.

The quality of the programming product can be controlled in the first three life-cycle phases to achieve the expected level of performance of the final product. When the fourth phase (operation and maintenance) has been entered, the quality of the software is generally fixed. With these five life-cycle phase boundaries in place, we can conceptualize what can be implemented as "programming quality measurement." If the phases and activities are the *X*- and *Y*-coordinates, the individual quality metrics can be placed on the *Z*-axis as shown in figure 9–2.

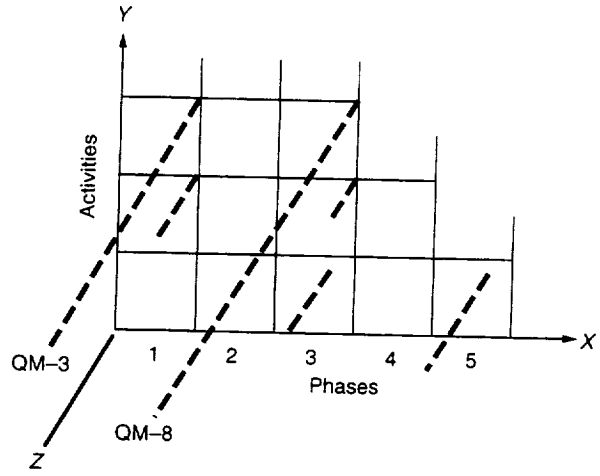


Figure 9–2.—Programming quality measurement map.

Without stating the specific activities for each phase, we can discuss the generalities of software quality and its cost. The cost of implementing quality increases with distance along the *X*-axis. Activities can be arranged along the *Y*-axis so that the cost of quality increases with distance along the *Y*-axis. With this arrangement, we can establish rigorous quality standards for the individual quality metrics as a function of cost effectiveness (e.g., error seeding—the statistical implanting and removal of software defects—may be expensive). Other quality metrics (e.g., test case effectiveness) may cost significantly less and could be selected.

In general, for a programming product, the higher the level of quality, the lower the costs of the product's operation and maintenance phase. This fact produces an incentive for implementing quality metrics in the early design phases. The programming industry has traditionally required large maintenance organizations to correct programming product defects. Figure 9–3 presents a typical phase-cost curve that shows the increased costs of correcting programming defects in the later phases of the programming product's life cycle. Note that the vertical axis is nonlinear.

Software Quality

The next step is to look at specific software quality items. Software quality is defined in reference 9–4 as "the achievement of a preselected software quality level within the costs, schedule, and productivity boundaries established by management." However, agreement on such a definition is often difficult to achieve. In practice, the quality emphasis can change with respect to the specific product application environment. Different perspectives of software product quality have been presented over the years. However, in today's literature, there is general agreement that the proper quality level for a particular software product should be determined in the

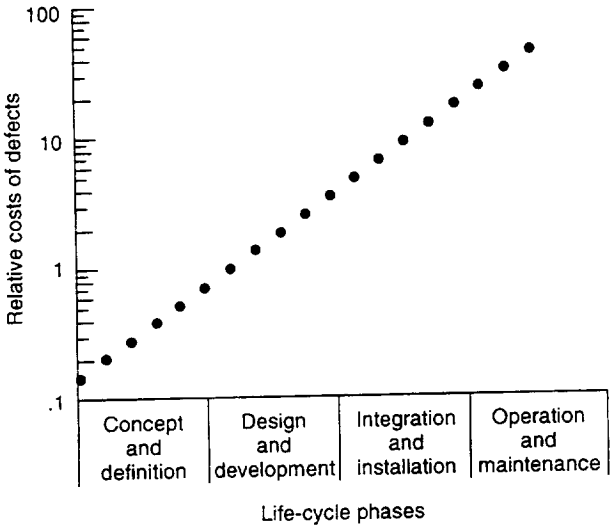


Figure 9–3.—Increasing costs of programming defects.

concept and definition phase and that quality managers should monitor the project during the remaining life-cycle phases to ensure the proper quality level.

The developer of a methodology for assessing the quality of a software product must respond to the specific characteristics of the product. There can be no single quality metric. The process of assessing the quality of a software product begins with the selection of specific characteristics, quality metrics, and performance criteria.

The specifics of software quality can now be addressed with respect to these areas:

(1) Software quality characteristics
(2) Software quality metrics
(3) Overall software quality metrics
(4) Software quality standards

Areas (1) and (2) are applicable during the design and development phase and the operation and maintenance phase. In general, area (2) is used during the design and development phase before the acceptance phase for a given software product.

Software Quality Characteristics

A software quality characteristic tree is presented in reference 9–5. The authors assume that different software products require different sets of quality characteristics. A product that has a rigorous constraint on size may sacrifice the maintainability characteristic of the software to meet its operational program size goals. However, this same product may need to be highly portable for use on several different processors. In general, the primary software quality characteristics are

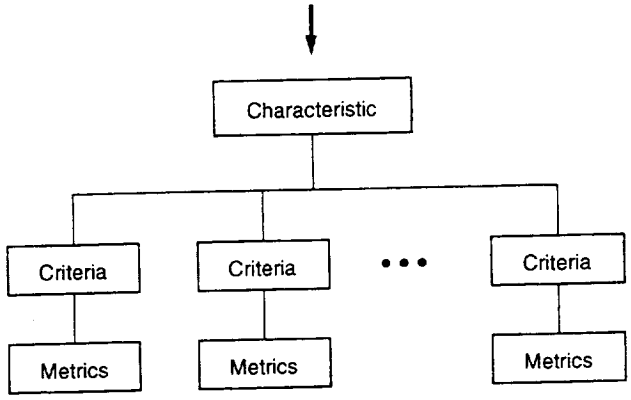


Figure 9–4.—Management's view of quality.

TABLE 9–1.—APPLICATION-DEPENDENT SOFTWARE QUALITY CHARACTERISTICS

Characteristic	Application	Importance
Maintainability	Aircraft	High
	Management information systems	Medium
	Testbeds	Low
Portability	Spacecraft	Low
	Testbeds	High

(1) Maintainability
(2) Portability
(3) Reliability
(4) Testability
(5) Understandability
(6) Usability
(7) Freedom from error

Management's view of software quality is the quality characteristics. Established criteria for these characteristics will provide the level of quality desired. The quantitative measures (metrics) place the quality at the achieved level. This concept is shown in figure 9–4.

Software quality criteria and metrics are directly related to the specific product. Too often, establishing the characteristic and the metric in the early life-cycle phases without the proper criteria leads to defective software. An example of the characteristics and their importance for various applications is presented in table 9–1.

Software Quality Metrics

The entire area of software measurements and metrics has been widely published and discussed. Two textbooks (refs. 9–6 and 9–7) and the establishment of the Institute for Electrical and Electronics Engineers (IEEE) Computer Society's working group on metrics, which has developed a guide for software reliability measurement, are three examples of such activity. Software metrics cannot be developed before

TABLE 9–2.—MEASUREMENT OF SOFTWARE QUALITY CHARACTERISTICS

Characteristic	Software life-cycle phase				
	3	4	5	7	9
	Product definition	Top-level design	Detailed design	Testing and integration	Maintenance and enhancements
Maintainability	---	(a)	(a)	----	(b)
Portability				----	----
Reliability	(a)			(b)	(b)
Testability	---				
Test case completion	---				
Estimate of bugs remaining	---				
Understandability	(a)				
Usability	(a)		-----		
Freedom from error	---	---	(a), (c)	(a), (c)	

[a]Where quality characteristic should be measured.
[b]Where impact of poor quality is realized.
[c]Metric can take form of process indicator.

TABLE 9–3.—MEASUREMENTS AND PROGRAMMING PRODUCT LIFE CYCLE

System life-cycle phase	Software life-cycle phase	Order of precedence	
		Primary	Secondary
Concept and definition	Conceptual planning (1)	----------	----------
	Requirements definition (2)	----------	----------
	Product definition (3)	Quality metrics[a]	----------
Design and development	Top-level design (4)	Quality metrics	Process indicators
	Detailed design (5)	Quality metrics	Process indicators
	Implementation (6)	Process indicators[b]	Quality metrics
Manufacturing and installation	Testing and integration (7)	Process indicators	Performance measures
	Qualification, installation, and acceptance (8)	Performance measures[c]	Quality metrics
Operation and maintenance	Maintenance and enhancements (9)	Performance measures	----------
Disposal	Disposal (10)	----------	----------

[a]Metrics—qualitative assessment, quantitative prediction, or both.
[b]Indicators—month-by-month tracking of key project parameters.
[c]Measures—quantitative performance assessment.

the cause and effect of a software defect have been established for a given product with relation to its product life cycle.

Table 9–2 is a typical cause-and-effect chart for a software product and includes the process indicator concept. At the testing stage of product development, the evolution of software quality levels can be assessed by characteristics such as freedom from error, completion of a successful test case, and estimate of the software bugs remaining. These process indicators can be used to predict slippage of the product delivery date, the inability to meet original design goals, or other development problems.

When the programming product enters the qualification, installation, and acceptance phase and continues into the maintenance and enhancements phase, the concept of performance is important in the quality characteristic activity. This concept is shown in table 9–3, where the aforementioned 5 IEC system life-cycle phases have been expanded into 10 software life-cycle phases:

(1) Conceptual planning: The functional, operational, and economic context of the proposed software is understood and documented in a product proposal.

(2) Requirements definition: A product proposal is expanded into specific product requirements and the requirements, such as performance and functional capabilities, are analyzed and translated into unambiguous developer-oriented terms.

(3) Product definition phase: Software engineering principles, technical information, and creativity are used to describe the architecture, interfaces, algorithms, and data that will satisfy the specified requirements.

(4) Top-level design: The functional, operational, and performance requirements are analyzed and designs for system architecture, software architecture, interfaces, and data are created and documented to satisfy requirements.

(5) Detailed design: The functional, operational, and performance requirements are analyzed and designs for system architecture, software architecture, components, interfaces, and data are further created, documented, and verified to satisfy requirements.

(6) Implementation: The software product is created or implemented from the software design and the faults are detected and removed.

(7) Testing and integration: Software elements, hardware elements, or both are combined into an overall system or an element of a system, and the elements are tested in an orderly process until the entire system has been evaluated, integrated, and tested.

(8) Qualification, installation, and acceptance: A software product is formally tested to assure the customer or the customer's representative that the product meets its specified requirements. This phase includes all steps necessary to deliver, install, and test a specific release of the system software and its deliverable documentation.

(9) Maintenance and enhancements: The product is ready for serving its designated function, is monitored for satisfactory performance, and is modified as necessary to correct problems or to respond to changing requirements.

(10) Disposal: The product reaches the end of its planned useful life or the requirement no longer exists for the product and it is disposed of, destroyed or modernized, if economically feasible.

Overall Software Quality Metrics

Several overall software quality metrics have been put into practice and have effectively indicated software quality. Jones (ref. 9–8) presents an overall quality metric called defect removal efficiency. The data collected for the overall quality metric are simplified to the more practical expression of "defects per 1000 lines of source code."

A second overall quality metric is based on the concept of quality prisms (refs. 9–9 and 9–10), which considers the extent of effort with which a given quality characteristic has been implanted into a product and the degree of effort for quality that has occurred in each life-cycle phase. An example of the extent and degree of effort is presented in table 9–4 for any given quality characteristic. From the table,

(1) Each quality characteristic can have a matrix similar to this with a specific quality program tailored to a company's products.

(2) The quality effort is extended to each of the product's life-cycle phases to the degree desired by the company.

(3) For each level, as the complexity and difficulty of a characteristic requirement increase, the intensity of the test and verification program effort increases.

(4) This matrix will change for each characteristic in accordance with company emphasis.

(5) Traditionally, the quality levels of a product correspond to degrees of effort. However, this matrix extends the effort to all phases of the product's life cycle.

As an example of using the matrix shown in table 9–4, a characteristic such as reliability may be targeted to reach service level 2. Then throughout planning, design, testing, integration, and installation, the reliability should achieve at least level 2. These indicators are tied to the proper major phase review points of a product's life cycle. For most characteristics, the planning level should be achieved after the preliminary design review (PDR); the design level, after the development phase or at the critical design review (CDR); the integration level, after integration at the qualification testing; and the service level, during the operational service reviews.

Now, quality management can apply this matrix to each characteristic in a manner depending on how critical it is to ensure achievement of the characteristic. For example, the reliability goal for a key system may be 10 or fewer mishandled calls per week, but the reliability goal for a private branch exchange (PBX) may be only 5 mishandled calls per month. These objectives may cause quality management to define a planning 2, design 2, integration 2, and service 2 program for the key system and a more demanding planning 4, design 3, integration 3, and service 3 program for the PBX.

In this manner, the quality characteristics are clearly identified by detailed criteria that set the scope of and limit the required objectives. Once these objectives are identified, a quality program can be determined to define the specific required definition, design, test, and measurement efforts. No longer are nebulous measurements made against vague objectives in the service phase of a product's life cycle in a last-minute attempt to improve quality.

The program for pursuing quality characteristics must be established early. If a particular quality characteristic is not pursued to a reasonable extent in the planning and design phases, a maximum degree of effort (4) may not realistically be achieved in the service phase. Conversely, the more uniformly and consistently a quality characteristic is pursued, the more achievable and figuratively stable the characteristic. This is graphically represented for a single characteristic in

TABLE 9–4.—QUALITY CHARACTERISTIC DEGREE/EXTENT MATRIX

Product phase	Service level					
	0	1	2	3	4	
Planning	No activity	General high level required	Specific detailed requirements definition	Highly complex required definition and support model	Difficult or complex required definition and prototype	Extent of effort
Design and test	No activity	General architecture consideration; general test and measurement program	Detailed architecture structure impact; language impact; test program extended	Extensive architecture and structure consideration; tailored language, operating system, man-machine interface impact, etc.; code walkthroughs; detailed documentation	Separate quality teams to verify design; detailed test facility; extensive qualification test plans and procedure	
Integration and installation	No activity	General quality management program; acceptance test; nominal change control quality program	Extensive qualification test plans and procedure to verify characteristics; above-nominal-quality-requirement verification testing	Quality teams formed; detailed quality configuration control release program; extensive data collection, verification, and analysis	Specialized quality integration, manufacturing, and installation programs to ensure achievement of quality characteristics by separate quality organization	
Service	No activity	General quality tracking and redesign program to achieve quality objectives and requirements	Formal data collection and analysis program to verify quality objectives; quality redesign effort	Detailed measurements, data analysis, and modeling program to verify high-level quality objectives; extensive redesign to obtain quality	Extensive measures and modeling, vigorous data analysis, and specialized tests to ensure high-level achievement of detailed quality requirements; extensive change program	
	No quality	First level of quality	Second level of quality	Third level of quality	Fourth level of quality	
Degree of effort						

figures 9–5 to 9–7, where the quality item is shown as either stable, unstable, or extremely costly to stabilize.

In figure 9–5 an optimum tradeoff of stability and productivity is portrayed. The base of the prism is secure, supporting the platform by properly balancing quality versus cost. In figure 9–6 schedule pressures have established an unstable prism to support the platform. In this example, the decision was made to send the product into the field at service level 1 even though it initially had reached a more extensive degree of quality (3) in the planning phase (considerable effort to define quality objectives in the planning phase but no followup). Figure 9–7 presents the extremely costly view of upgrading a programming product in the field to service level 4 (after passing the first three phases only to the first degree). Note the increasing amount of time and effort to achieve service levels 1, 2, or 3. Service level 4 in this example is usually extremely difficult and expensive, if not impossible, to achieve. The measured productivity of such a product will most likely be low.

An excellent example of the need for this type of quality management process occurred many years ago, but the lessons still apply today. An automated program was proposed to generate from 160 fields of input data per customer, a centralized data base that would control a table-driven, wired logic system. It was estimated that 13 weeks of design time would be required to construct this table generator by using a nominal amount of computer support time. A representative of the design group was assigned to define the input and output requirements for the support program and verify its operation. The program was initially written in assembly language. It was later redesigned and split into three separate programs written in a high-level language. These programs could then be separately designed, verified, and maintained. The main consideration became the verification process. An input and output test was written to check the extensive program paths. The project dragged along for a year as verification testing attempted to meet a zero-defect objective (imposed after the initial design had been completed). Costs increased and the schedule became critical as the customer became impatient (fig. 9–7). As the program began to function more successfully, deciding the degree of testing required for verification became a serious problem. Confrontation developed between the design and marketing departments over the commercial release of the program. The testing continued without agreement on the required degree of effort. Eventually, the customer became disillusioned and turned to another firm to provide the table generator.

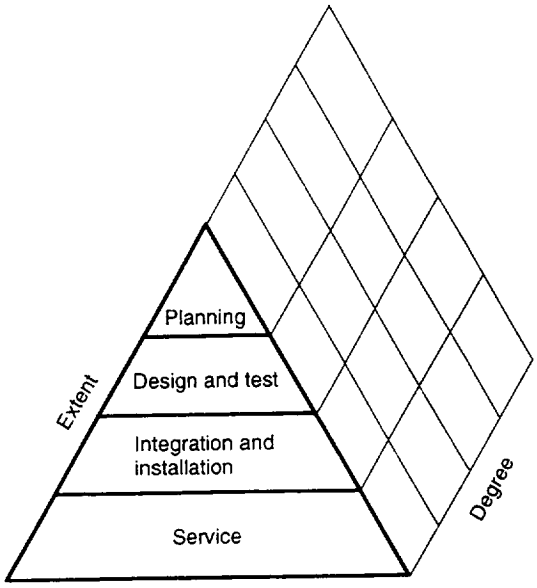


Figure 9–5.—Stability in quality and cost.

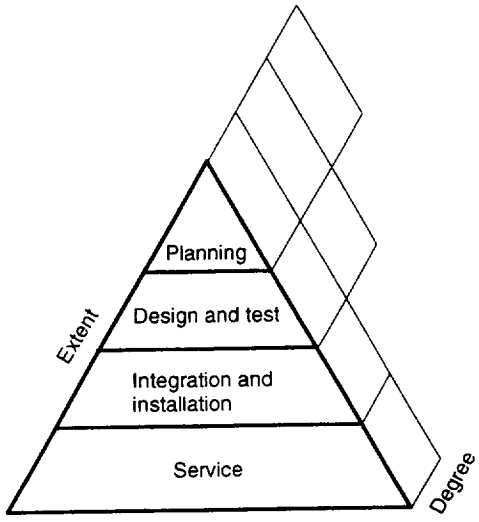


Figure 9–6.—Instability due to scheduling decisions.

Had a clear quality management decision been made in the planning phase and tracked throughout the development on the degree of error-free "verified" operation, the quality characteristic objectives for its design architecture and structure, the language required for changes, and so forth, a more realistic projection (and control) of schedule and people could have been achieved. Several releases to the customer may have been required as the program designs and operation were verified to a predetermined extent within the various life-cycle phases. Had this procedure been followed, both the customer and the supplier would have been more satisfied.

This example offered an excellent opportunity to first determine the type and degree of quality desired. Then management could have constructed a quality process, in terms of the extent

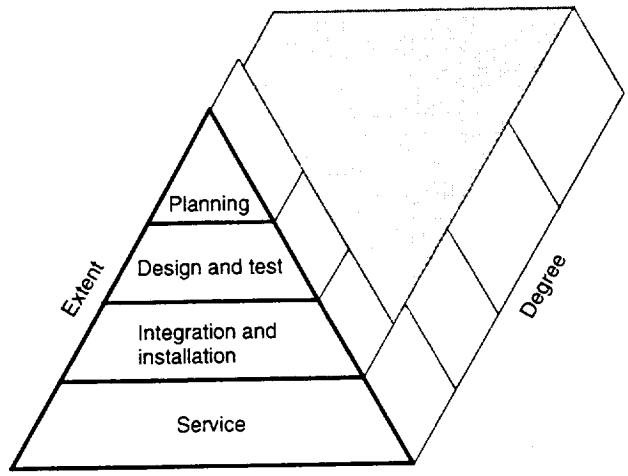


Figure 9–7.—Extremely costly programming products.

and degree of each desired characteristic, with an elastic compromise between the schedule, resources, and design activity needed to achieve it. In this case, many of the "ilities," changeability, usability, maintainability, and reliability, were subsequently more critically identified. These considerations could have been translated into the initial requirements for structural design, program segmentation, extensive documentation, type of language, amount of code walkthrough, number of subfunctional tests, amount of error acceptable at first release, depth of verification reviews, and so on. From this form of planning, the quality prisms could have been established to define the extent and degree (such as service level 2, 3, or 4) to which each of these characteristics should have been pursued in terms of project cost restraints that depended on user willingness to pay and wait for a quality product.

A figuratively secure prismatic base for the programming product is presented in figure 9–5. This security is developed through execution of an extensive quality program, as progressively shown in figures 9–8 to 9–10. A product's quality objective is usually composed of more than one characteristic. Previously, those have tentatively been noted as maintainability, portability, reliability, testability, understandability, usability, and freedom from error. Thus, quality management can extend the support prismatic structure to a greater depth than to just one quality characteristic. In practice, several quality prisms will be placed together to achieve a firm quality base.

It may be desirable to have a product developed that has reached service level 4 for all the aforementioned quality characteristics. However, realistic schedules and productivity goals must be considered in terms of cost. These considerations establish the need for vigorous quality management over all life-cycle phases to selectively balance the various possibilities. It would be nonsupportive, expensive, and time consuming if quality management established the structural combination of individual characteristic quality prisms graphically

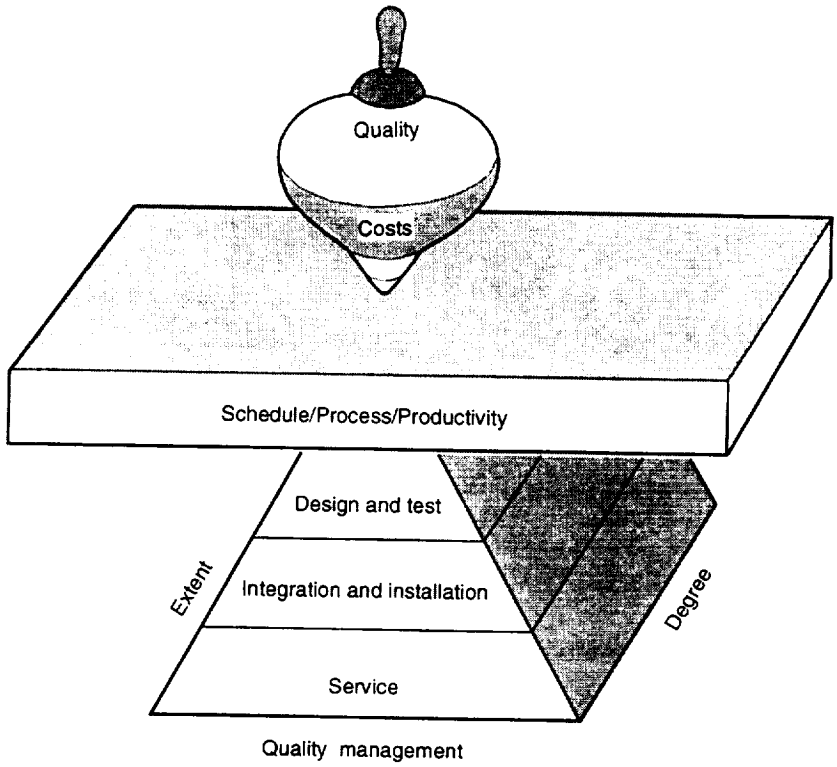


Figure 9–8.—Delicate balance—planning complete.

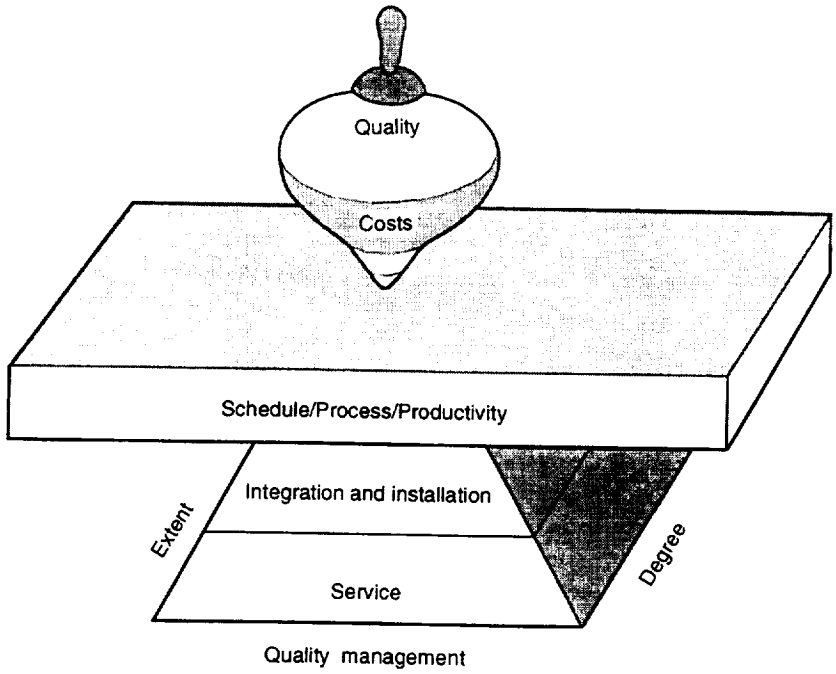


Figure 9–9.—Delicate balance—design and testing complete.

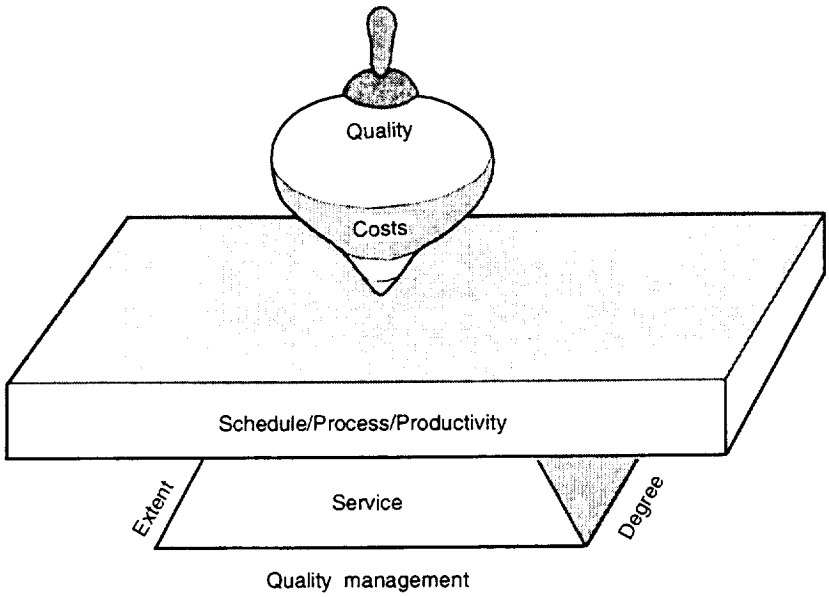


Figure 9–10.—Delicate balance—integration and installation complete.

P Planning
D Design and test
I Integration and installation
S Service

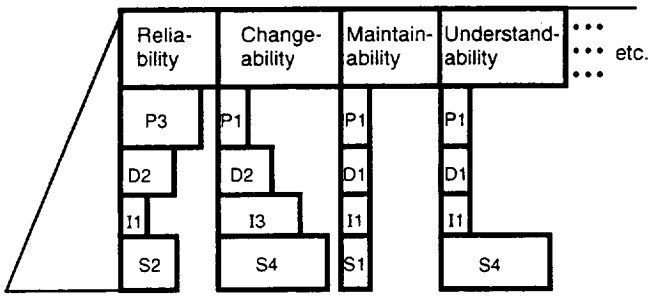


Figure 9–11.—Example of poor quality management.

P Planning
D Design and test
I Integration and installation
S Service

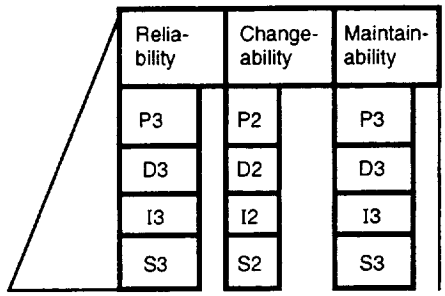


Figure 9–12.—Example of good quality management.

presented in figure 9–11. Unfortunately, this is the case for too many products. Quality management would do better to establish a more consistent support structure, like that represented in figure 9–12. The figurative result of this consistent effort is shown in the solid cost-effective base of figure 9–13.

If quality characteristics are established, monitored, measured, and verified throughout the life cycle, a realistic balance can successfully be achieved between quality costs, schedule, and productivity. However, it will require an active quality management process to establish and track these indicators. An example of such a quality management process matrix is presented in table 9–5 to quantify the extent and degree of effort needed to achieve a desired level of quality. This table can be used as a programming product quality worksheet or as both the characteristic survey data collection instrument and part of the final quality prisms planning document.

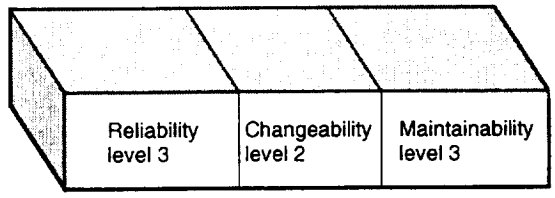


Figure 9–13.—Example of solid quality base.

As discussed, a quality management team must establish the degree of quality that a particular quality characteristic must reach throughout its life cycle. It may use specialized support tools, measurement systems, and specific product quality standards to pursue its quality objectives. A point system can give a quantitative reference for the pursuit of quality. The point system can become the basis for trading time versus cost to reach specific quality goals. Of course, a firm's quality

TABLE 9–5.—EXAMPLE OF QUALITY MANAGEMENT PROCESS MATRIX

[Number in circle denotes degree of quality selected by a quality management process.]

Product phase	Quality characteristic		
	Reliability	Changeability	Maintainability
Planning	1 ② 3 4	1 2 3 ④	1 2 ③ 4
Design and test	1 ② 3 4	1 2 3 ④	1 2 ③ 4
Integration and installation	1 ② 3 4	1 ② 3 4	1 2 3 ④
Service	1 ② 3 4	1 ② 3 4	1 2 ③ 4

Extent of quality ↓

Degree of quality →

TABLE 9–6.—EXAMPLE OF PURSUIT OF QUALITY

Product phase	Quality characteristic		
	Reliability	Changeability	Maintainability
Planning	2	4	3
Design and test	↓	4	3
Integration and installation		2	4
Service	↓	2	3
Total points/ available points	8/16 (50 percent)	12/16 (75 percent)	13/16 (81 percent)
Total	(33/48)/C3, or (69 percent)/C3		

management will define their own point system. However, the following example point system will serve as an illustration for discussion purposes.

If a single characteristic's quality effort has progressed through all four levels and through each level's maximum degree, it has accumulated a maximum of 4 + 4 + 4 + 4 = 16 points. If another characteristic's effort has moved through the levels only at one-half its maximum degree, it has accumulated 2 + 2 + 2 + 2 = 8 points. If it reached three-quarters of the maximum degree of effort on all levels, it has 3 + 3 + 3 + 3 = 12 points. Management can now assign a reference value to the pursuit of quality for a programming product. This is shown in the simplified example in table 9–6. For this example the total is 9 + 12 + 13 = 33 points out of a possible 16 + 16 + 16 = 48 points, or 69 percent. (In more general terms, this can also be referred to as an overall level-3 quality effort in the 50- to 75-percent range.) Note that the real indication of the quality objectives will be the magnitude of the *X*/*Y* (33/48) values. The greater the *X*- and *Y*-values, the deeper the degree to which the characteristics have been pursued. The greater the *X*-value, the more stable the structure has become and the more quality objectives the programming product has achieved.

If this type of analysis is carried over all eight characteristics (8×16), a maximum of 128 points is possible. Products that approach this level of effort will have a considerably more stable structure than those that are only based upon a 16-point, single-character structure. The *X*-percent quality reference number should also be qualified by a factor to note how many characteristics were actually used. This could be shown as 69 percent/C3, or 33/48/C3.

Finally, some characteristics will be more complex and require greater costs to achieve than others. Thus, a weighting multiplier (WM) can be used to equalize the quality characteristics. Weighting multipliers for the preceding example are demonstrated in table 9–7. For this example, the total of 10 + 28 + 19 = 57 points out of a possible 20 + 40 + 24 = 84 points is 57/84/C3, or 68 percent/C3. This three-part programming quality ratio (e.g., 57/84/C3) can be used for reviewing quality across programming products within a corporation as a more quantitative cross reference of quality costs to quality objectives.

A quality management process matrix (table 9–5) has been presented for pursuing quality throughout a programming product's life cycle. It relates the pursuit of quality characteristics to the planning, design and testing, integration and installation, and service phases. In practice, actual implementation of this approach will require the selection of languages, walkthroughs of code, type of testing, and so forth to be specifically defined for reaching service quality level 2, 3, or 4. From this matrix, the impact on schedule and the cost of quality can be projected and monitored.

This process will also help management to compare the extent and degree of quality for products of competing compa-

TABLE 9-7.—EXAMPLE OF USE OF WEIGHTING MULTIPLIERS (WM)

Product phase	Quality characteristic		
	Reliability	Changeability	Maintainability
	Level × WM	Level × WM	Level × WM
Planning	2 × 1	4 × 2	3 × 2
Design and test	2 × 1	4 × 2	3 × 1.5
Integration and installation	2 × 1	2 × 3	4 × 1
Service	2 × 2	2 × 3	3 × 1.5
Total points/ available points	10/20 (50 percent)	28/40 (70 percent)	19/24 (79 percent)
Total	(57/84)/C3, or (68 percent)/C3		

nies or internal corporate divisions. Of course, until such a standard is developed, the quality management team will subjectively assign values and multipliers as noted in table 9–5 and relate them to their own acceptable degree of documentation, walkthrough of code, and module tests. These subjective values are extremely useful in establishing individual product quality effort goals, translating the concept of quality prisms to planning, design, and test considerations that balance schedule and cost against quality objectives. However, management will now have a more reasonable opportunity to pursue and successfully achieve the extent and degree of desired quality for their products.

The ability to specify an overall software quality metric has been addressed. Overall quality measurements can be normalized, as in the quality prisms concept, for purposes of comparison. The quality prisms concept can be used to compare the software of two or more different projects within the same company or of different companies even if the software products have unique applications or utilize different programming languages. Quality prisms can also be used to combine hardware quality and software quality into an assessment of the quality of the entire system.

Software Quality Standards

The relationship of software quality standards and software quality measurements is depicted in figure 9–14. Measurements and standards must agree. If a set of quality standards is established (e.g., zero defects) and quality measurement cannot prove it (i.e., through exhaustive testing, error seeding, etc.), the software development project must realistically set a goal so that both quality standards and measurements can be developed. The IEEE has published many articles on and general guides for formulating goal criteria. In addition, many technical papers are available on setting specific goals on the bases of life cycle and a per-delivered software product (ref. 9–11).

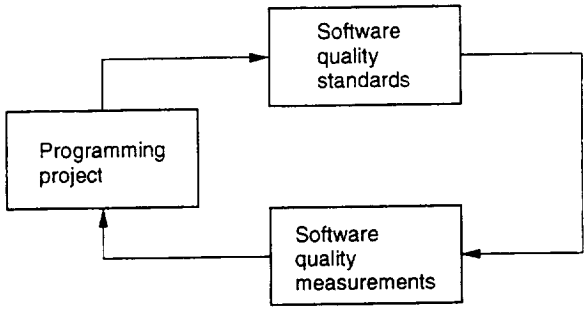


Figure 9–14.—Relationship of measurements and standards.

Concluding Remarks

This chapter has presented a snapshot of software quality assurance today and has indicated future directions. A basis for software quality standardization was issued by the IEEE. Research is continuing into the use of overall software quality metrics and better software prediction tools for determining the defect population. In addition, simulators and code generators are being further developed so that high-quality software can be produced.

Several key topics were discussed:

(1) Life-cycle phases
(2) Software quality characteristics
(3) Software quality metrics
(4) Overall software quality metrics
(5) Software quality standards
(6) Process indicators
(7) Performance measures

Process indicators are closely tied to the software quality effort and some include them as part of software development. In general, there are measures such as (1) test cases completed versus test cases planned and (2) the number of lines of code developed versus the number expected. Such process indicators can also be rolled up (all software development projects added together) to give an indication of overall company or corporate progress toward a quality software product. Too often, personnel are moved from one project to another and thus the lagging projects improve but the leading projects decline in their process indicators. The life cycle for programming products should not be disrupted.

Performance measures, which include such criteria as the percentage of proper transactions, the number of system restarts, the number of system reloads, and the percentage of uptime, should reflect the user's viewpoint. The concept of recently proposed performability combines performance and availability from the customer's perspective.

In general, the determination of applicable quality measures for a given software product development is viewed as a specific task of the software quality assurance function. The determination of the process indicators and performance measures is a task of the software quality standards function.

References

9–1. Dunn, R.; and Ulman, R.: Quality Assurance for Computer Software. McGraw-Hill, 1982, p. 265.

9–2. Boehm, B.W., et al.: Characteristics of Software Quality. North-Holland, 1978, p. 3–1.

9–3. IEEE Standard Glossary of Software Engineering Terminology. IEEE Computer Society, 1982, p. 34.

9–4. Reliability Prediction of Electronic Equipment. MIL–HDBK–217E, Jan. 1990.

9–5. Boehm, B.W.; Brown, J.R.; and Lipow, M.: Quantitative Evaluation of Software Quality, Tutorial on Models and Metrics for Software Management and Engineering, V.R. Basili, ed., IEEE Computer Society Press, 1980.

9–6. Perlis, A.J.; Sayward, F.G.; and Shaw, M., eds.: Software Metrics: An Analysis and Evaluation. MIT Press, 1981.

9–7. Basili, V.R.: Tutorial on Models and Metrics for Software Management and Engineering. IEEE Computer Society Press, 1980.

9–8. Jones, T.C.: Measuring Programming Quality and Productivity. IBM Syst. J., vol. 17, no. 1, 1978, pp. 39–63.

9–9. Heldman, R.K.; and Malec, H.A.: Quality Management Process for Telecommunications Programming Products. 1984 IEEE Global Telecommunications Conference, GlobeCom 1984, IEEE, 1984, pp. 557–565.

9–10. Malec, H.A.: An Introduction to Quality Prisms and Their Application to Software. Relectronic '85, Sixth Symposium on Reliability in Electronics, OMIKK-Technoinform, Budapest, Hungary, pp. 155–163.

9–11. Jones, D.R.; and Malec, H.A.: Communications Systems Performability: New Horizons. 1989 IEEE International Conference on Communications, vol. 1, IEEE, 1989, pp. 1.4.1–1.4.9.

Reliability Training[1]

1. What are the three factors that determine quality software?

   A. Process, material, and vibration
   B. Process, product, and environment
   C. Planning, product, and shock
   D. All of the above

2. What does software quality consist of?

   A. Various aspects of producing programming products
   B. Bar charts for process control
   C. Statistical analysis of software bugs
   D. All of the above

3. How is the term "software quality" defined?

   A. To assure the acquisition of high-quality software products on schedule, within cost, and in compliance with the performance requirements
   B. To ignore various needs
   C. To develop specifications and attributes, perceive customer needs, and meet the user's expectations
   D. All of the above

4a. What are the 10 software life-cycle phases?

   A. Conceptual; requirements; product definition; design; implementation; testing; vibration; prototypes; installation; and disposal
   B. Planning; definition; design; manufacturing; testing; acceptance; debugging; and repair
   C. Conceptual planning; requirements definition; product definition; top-level design; detailed design; implementation; testing and integration; qualification, installation, and acceptance; maintenance and enhancements; and disposal
   D. All of the above

4b. What are the IEC system life-cycle phases?

   A. Concept and research; design and plan; manufacture and debug; operation and maintenance; and wearout
   B. Concept and definition; design and development; manufacturing and installation; operation and maintenance; and disposal
   C. Research and development; design and breadboard; manufacturing and testing; operation and maintenance; and disposal
   D. All of the above

4c. How can the 10 software life-cycle phases be combined to fit in the IEC system life-cycle phases?

   A. Concept and definition: conceptual planning; requirements definition; and product definition
   B. Design and development: top-level design and detailed design
   C. Manufacturing and installation: implementation; testing and integration; qualification; and installation and acceptance
   D. Operations and maintenance: maintenance and enhancement
   E. Disposal: disposal
   F. All of the above

[1]Answers are given at the end of this manual.

5. Can there be different degrees of a quality characteristic for different life-cycle phases?

A. Yes B. No C. Do not know

6a. The definition of a lack of software quality is

A. The lack of proper planning in early life-cycle phases
B. The application of dependent software quality characteristics
C. Poorly developed software that lacks proper criteria in life-cycle phases
D. All of the above

6b. Three example characteristics of software quality are

A. Testing, integration, and portability
B. Maintainability, portability, and reliability
C. Design, implementation, and reliability
D. All of the above

7. Seven software quality characteristics are

A. Maintainability, portability, reliability, testability, understandability, usability, and freedom from error
B. Planning, definition, reliability, testing, software, hardware, usability
C. Design, implementation, integration, qualification, acceptance, enhancement, maintenance
D. All of the above

8. Management has decided that quality engineering should measure four characteristics of the *XYZ* software: maintainability, portability, reliability, and testability. The desired goals set at the beginning of the program by management for the characteristic effort were maintainability, 3.5; portability, 3.0; reliability, 3.9; and testability, 3.5. The overall goal was thus 87 percent/C4 for the extent of quality. The 2-year program gave the following results:

Characteristic	Planning	Design and test	Integration	Service
Maintainability	4.0	3.5	3.4	3.4
Portability	4.0	3.0	3.1	3.1
Reliability	3.5	3.6	3.9	3.9
Testability	4.0	3.1	3.5	3.6
Total	15.5	13.2	13.9	14.0

a. The actual extent of quality was

A. (87.5 percent)/C4 B. (88.4 percent)/C4 C. (88.8 percent)/C4 D. None of these

b. Have the management objectives been achieved?

A. Yes B. No C. Do not know

Chapter 10
Reliability Management

Roots of Reliability Management

Over the past few years the term "reliability management" has been raised to a high level of awareness. Previously, the management of reliability was concerned with eliminating failure by testing to prove reliability, and it generally complemented the design function. Quality management, on the other hand, focused on quality control and generally aligned itself with manufacturing and production. The picture began to change with the focus on customer reliability and quality concerns. Specifically, the usage and standardization by companies of reliability growth models established that the new concept of reliability management is replacing the old concept of the management of reliability. The focus is now on enlarging the area of reliability concern to all phases of the life cycle. The current thinking is that all aspects of management operations and functions must be integrated in the reliability concept and program. Thus, reliability in the manufacturing or production phase is as important as reliability in the design phase (ref. 10–1), as shown in figure 10–1.

Planning a Reliability Management Organization

Planning a reliability management organization requires that the reliability function report to a high enough level to be effective. The reporting level is too low if it does not involve top management in reliability issues. For example, many successful programs today encompass 3 to 6 hours per month at vice-

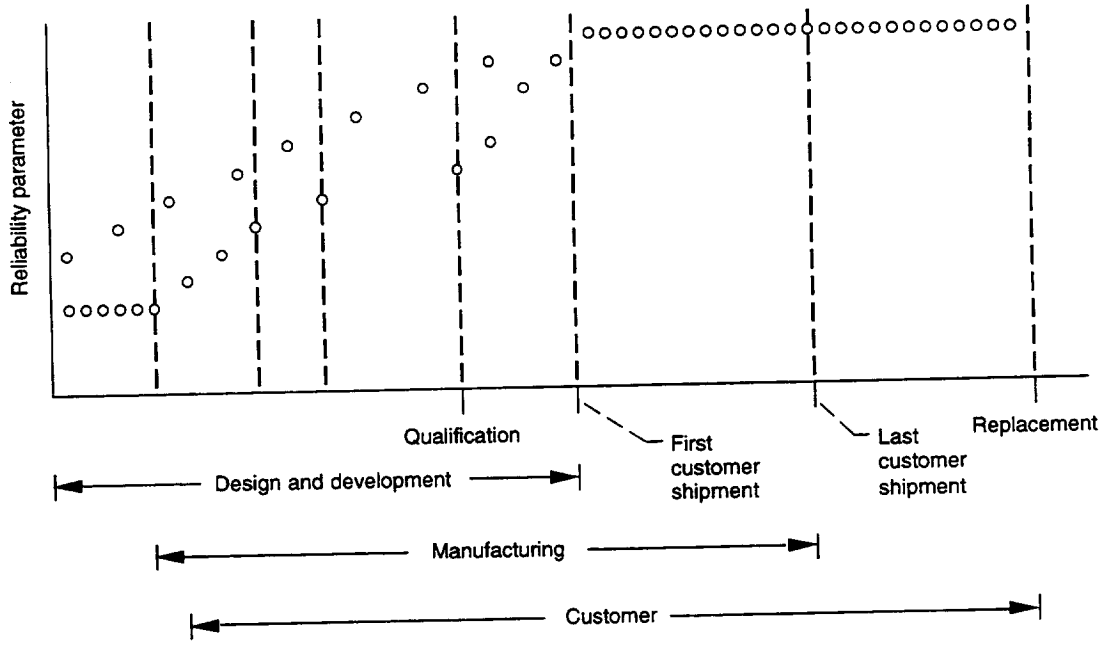


Figure 10–1.—Life–cycle reliability growth with two different parts to first customer shipment.

presidential staff meetings. Each company must find the level that makes reliability a significant issue to be addressed. A guide to reliability management is reference 10–2.

A functional organization forms groups to perform similar generic tasks such as planning, designing, testing, and reliability. Often, such an organization gets mired down with too many levels of management, and specific product priorities are often different in the many task groups. However, many benefits accrue from the concentration of talent and constant technical peer review. With today's time-to-market pressures, building such a large centralized reliability organization is often not the best choice. The team approach, distributed reliability, is often selected over functional organization.

In a team organization, people with diverse talents and backgrounds comprise the teams. Quality circles and reliability circles are based on the same organizational approach. Even though peer review is not ongoing, the cross technology knowledge of today's personnel appears to fully compensate for the lack of constant peer review. In the software development world, several types of team organization exist. For instance, the first type, the project team, is typical and is a hierarchical organization in which programmers with less experience are assigned to work for programmers with more experience. The project team is designed to fit the company organization rather than to fit project requirements. The second type is the chief programmer team, which employs a highly skilled person who performs most of the programming while providing technical direction. A third type is the Weinberg programming team, which is composed of groups of 10 or fewer programmers with complementary skills. Group consensus and leadership role shifts are characteristic of this type. Each of these team organizations has advantages depending on the size of the project, the newness of the technology being implemented, and so on.

The fourth type of team organization, the matrix, is a hybrid approach that combines functional talent to put teams together, but it can be a reliability disaster especially if time-to-market pressures exist. Often the technology is masked by middle management procedural meetings because these teams report to one manager. Individual contributors are added to work on one or more tasks of a given project or product development. These projects usually report to middle management.

A fifth possible type of team organization is based on the theory stated in reference 10–3: reliability is actively pursued by involvement starting on the vice-presidential level and proceeds throughout the organization. This new style of reliability involves establishing a reliability council, dedicating a full-time diagnostic person or team, and generally making an upward change in the reliability reporting level. Figure 10–2 presents this concept. The reliability council's responsibilities are to

(1) Endorse the annual reliability plan
(2) Regularly review reliability status
(3) Approve reliability improvement projects
(4) Set priorities on resources

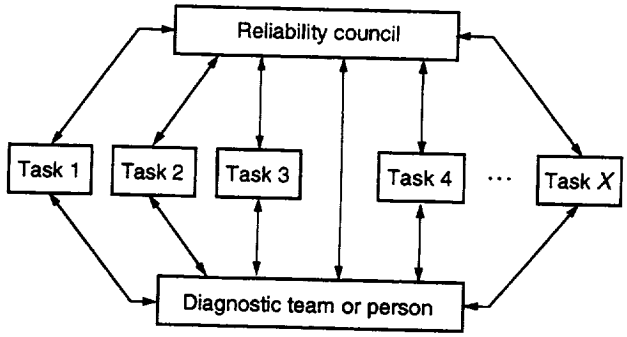


Figure 10–2.—Reliability organization.

(5) Assign tasks
(6) Regularly review tasks
(7) Participate in reliability improvement awards

The reliability council membership may consist of the

(1) Vice president of the company or division as chairman
(2) Vice president's staff
(3) Vice president's business partners
(4) Corporate engineering director
(5) Corporate manufacturing director
(6) Corporate customer services director

The diagnostic team's or person's functions are to

(1) Review the internal reliability status
(2) Review reliability as perceived by customers
(3) Recommend tasks to the reliability council
(4) Diagnose problems
(5) Design experiments
(6) Collect and analyze data

The diagnostic team's or person's concerns include

(1) Reliability, quality, and statistics
(2) Engineering and manufacturing engineering
(3) Product development and process optimization
(4) Product assembly and test strategies
(5) Customer perception

This is a new dynamic approach for establishing reliability management at the proper level in a corporation while optimizing its effectiveness.

General Management Considerations

Program Establishment

To design for successful reliability and continue to provide customers with a reliable product, the following steps are necessary:

(1) Determine the reliability goals to be met.
(2) Construct a symbolic representation (e.g., block diagram or Petri net, ref. 10–4).
(3) Determine the logistics support and repair philosophy.
(4) Select the reliability analysis procedure.
(5) Select the source or sources of the data for failure rates and repair rates.
(6) Determine the failure rates and the repair rates.
(7) Perform the necessary calculations.
(8) Validate and verify the reliability.
(9) Measure reliability until customer shipment.

This section will address the first three steps in detail.

Goals and Objectives

Goals must be placed into the proper perspective. They are often examined by using models that the producer develops. However, one of the weakest links in the reliability process is the modeling. Dr. John D. Spragins, an editor for the IEEE Transaction on Computers, places this fact in context (ref. 10–3) with the following statement:

> Some standard definitions of reliability or availability, such as those based on the probability that all components of a system are operational at a given time, can be dismissed as irrelevant when studying large telecommunication networks. Many telecommunication networks are so large that the probability they are operational according to this criterion may be very nearly zero; at least one item of equipment may be down essentially all of the time. The typical user, however, does not see this unless he or she happens to be the unlucky person whose equipment fails; the system may still operate perfectly from this user's point of view. A more meaningful criterion is one based on the reliability seen by typical system users. The reliability apparent to system operators is another valid, but distinct, criterion. (Since system operators commonly consider systems down only after failures have been reported to them, and may not hear of short self-clearing outages, their estimates of reliability are often higher than the values seen by users.)

Reliability objectives can be defined differently for various systems. An example from the telecommunications industry (ref. 10–5) is presented in table 10–1. We can quantify the objectives, for example, for a private automatic branch exchange (PABX) (ref. 10–6) as shown in table 10–2, which presents the reliability specifications for a wide variation of PABX sizes (from fewer than 120 lines to over 5000 lines).

TABLE 10–1.—RELIABILITY OBJECTIVES FOR TELECOMMUNICATIONS INDUSTRY

Module or system	Objective
Telephone instrument	Mean time between failures
Electronic key system	Complete loss of service Major loss of service Minor loss of service
PABX	Complete loss of service Major loss of service Minor loss of service Mishandled calls
Traffic service position system (TSPS)	Mishandled calls System outage
Class 5 office	System outage
Class 4 office	Loss of service
Class 3 office	Service degradation

Symbolic Representation

Chapter 3 presents reliability diagrams, models that are the symbolic representations of the analysis. The relationship of operation and failures can be represented in these models. Redundancy (simple and compound) is also discussed. Performance estimates and reliability predictions are now being performed simultaneously by using symbolic modeling concepts such as Petri nets.

In 1966, Carl Adam Petri published a mathematical technique for modeling. Known as a Petri net, it is a tool for analyzing systems and their projected behavior. In 1987, he delivered the keynote address at the international workshop on Petri nets and performance models (ref. 10–7). Many applications were discussed: the use of timed models for determining the expected delay in complex sequences of actions, the use of methods to determine the average data throughput of parallel computers, and the average failure rates of fault-tolerant computer designs. Correctness analysis and flexible manufacturing techniques were also described. Timed Petri nets show promise for analyzing throughput performance in computer and communications systems.

A Petri net is an abstract and formal graphical model used for systems that exhibit concurrent, asynchronous, or nondeterministic behavior. The Petri net model provides accurate system information when it validly represents the system and the model solution is correct. A Petri net is composed of four parts: a set of places, a set of transitions, an input function, and an output function. The input and output functions relate to transitions and places. In general, graphics are used to represent the Petri net structures and to show the concepts and the problems. A circle represents a place, a bar represents a transition, and directed arcs connect transitions to places or places to transitions. The state of a Petri net is called the PN marking and is defined by the number of "tokens" contained in each place.

TABLE 10–2.—RELIABILITY SPECIFICATION FOR PABX

	Number of lines							
	<120	200	400	600	800	1200	3000	5000
Common control performance:								
Mean time between catastrophic failures, yr	10	----	----	----	----	----	----	----
System outage time per 20 yr, hr	----	----	----	----	----	1	1	1
Mean time between outages, yr	----	----	----	----	----	>5	>5	>5
Mean time between complete losses of service, yr	5	10	40	40	40	----	----	----
Service level:								
Mean time between major losses of service, days	200	400	300	200	150	365	365	----
Mean time between minor losses of service, days	60	60	50	40	30	30	15	----
Degradation of service, hr/yr	----	----	----	----	----	----	----	1
Mishandled calls, percent	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.02

TABLE 10–3.—SPARES POLICY

Subsystem	Onsite spares ?	Subdepot spares ?	Turnaround time[a] of subdepot spares, days	Depot spares ?	Turnaround time[a] of depot spares, days
Common control and memory	Yes	Yes	2	Yes	15
Network	No	↓	↓	↓	30
Line and trunk units	Yes				30
Peripheral equipment	No				30
Test equipment	No	No	----		5

[a]For replacing spares.

A place is an input to a transition if an arc exists from the place to the transition and an output if an arc exists from the transition to the place. Enabled transitions can be "fired" by removing one token from each input place and adding one token to each output place. The firing of a transition causes a change of state and produces a different PN marking. Reference 10–8 contains additional information. Petri nets are a useful reliability modeling tool.

Logistics Support and Repair Philosophy

The logistics support plan is normally based on criteria such as (1) failure rates and repair rates of replaceable units, (2) system maturity, (3) whether the sites can be served by depots or subdepots, and (4) the rate at which additional sites are added to the depot responsibility. Since spares are the key to support, this chapter will examine them further.

The size of the spares stock depends on (1) the criticality of the replaceable unit to the system, (2) the necessary spare adequacy level, (3) the number of systems served, (4) whether the area served is rural, suburban, or urban, and (5) whether the repair facility is onsite or remote. A typical spares policy for a telecommunications system (ref. 10–9) is presented in table 10–3.

Policies can be formulated for families of systems or for multifamily geographical areas. The turnaround time depends on the replaceable units failure rate, the repair location, the repair costs, and so forth. A specific spares policy can be tailored to a given geographical area. Note that subsystems have different spares policies owing to the criticality of their failures in contrast to a blanket spares assignment without regard to functionality or survivability.

Even though the spares location and turnaround time are the same for two different subsystems, the spares adequacy can be different. Some spares adequacy levels for a telecommunications systems are presented in table 10–4.

Spares provisioning is an important part of a spares plan. Requirements must be clearly stated or they can lead to over- or undersparing. For example, a spares adequacy of 99.5 percent can be interpreted in two ways. First, six spares might be needed to guarantee that spares are available 99.5 percent of the time. Alternatively, if one states that when a failure occurs, a spare

TABLE 10–4.—SPARES ADEQUACY

Subsystem	Onsite spares?	Subdepot spares	Depot spares
		Adequacy[a]	
Common control and memory	Yes	0.9995	0.9995
Network	No	.995	.995
Line and trunk units	Yes	.999	.999
Peripheral equipment	No	.99	.99
Test equipment	No	---------	.95

[a]Probability of having spares available.

TABLE 10–6.—MAINTENANCE ACTION RECOMMENDATIONS

Action	Before busy hour	Busy hour	After busy hour	Off-shift time
Repair	Yes	Yes	Yes	Yes
Defer repair for (days)	0	0	1	1
Is second failure affecting service?	No	Yes	No	No
Probability of no similar second failure	0.95	0.90	0.82	0.60
Site failures last month	Low	High	Normal	Low
Site failures last year	Low	Low	Normal	Low
Transient error rate	Low	High	Low	Low

TABLE 10–5.—DEPOT EFFECTIVENESS FOR TYPICAL DIGITAL PABX

Foreign branch part	Control automatic trunk	Printed wiring cards for n systems					Spare printed wiring cards for n systems				
		1	2	10	50	100	1	2	10	50	100
15002	6	65	130	650	3 250	6 500	2	2	5	13	20
15003	5	16	32	160	800	1 600	1	1	2	5	7
15004	6	14	28	140	700	1 400	1	1	4	5	8
.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.
20703	8	28	56	280	1 400	2 800	2	1	4	10	15
20703	16	153	206	1 530	7 650	15 300	7	11	29	106	196
	Total	1058	2116	10 580	52 900	105 800	153	173	287	658	1001
						Spares, percent of total	14.5	8.2	2.7	1.2	0.95

must be available 99.5 percent of the time, it will be necessary to supply 6 + 1 = 7 spares.

The establishment of depot and subdepot sparing, rather than only individual site sparing, has proven to be cost effective. As an example, table 10–5 presents the depot effectiveness for a typical digital PABX. This table indicates that a 14.5-percent spares level would be required if only per-site sparing was used; however, when one depot serves 100 sites, the required spares level is less than 1 percent.

A centralized maintenance base (CMB) (ref. 10–10) is essential to a deferred maintenance concept. Deferred maintenance can be available on a real-time basis. When a failure occurs at an unattended site, the CMB would receive information on a display as to the criticality of the failure and the deferred maintenance action taken if imposed and would receive a projection indicating impending problems. The CMB would analyze the situation for the specific site configuration, the processing level in the system, and the site's failure-repair history.

Input data could consist of items such as the last similar occurrence, the next planned visit to the site, the criticality of the site to the operating network, the cumulative site failures for the last 3 months, and the probability of additional failures occurring. The data would be analyzed with a maintenance-prediction computer program to generate a table based on system loading, such as table 10–6. Often the suggested maintenance deferral time is recommended to be the next maintenance visit (NMV). The NMV will vary with the amount of equipment onsite and the projected failure frequency (ref. 10–10).

The combination of deferred maintenance and a centralized maintenance base dictates the needs for an efficient spares program. Spares planning combined with knowledge of the logistics can optimize support costs. A depot stocking plan can additionally vary because of many factors, including error coverage, system maturity, deferred repair, and maintenance familiarity. A dynamic (continuously updated) depot stocking plan would be cost effective. A dynamic depot model using Monte Carlo methods (ref. 10–11) includes unit delivery schedules, item usage per month, support personnel efficiency, and depot and base repair cycle times.

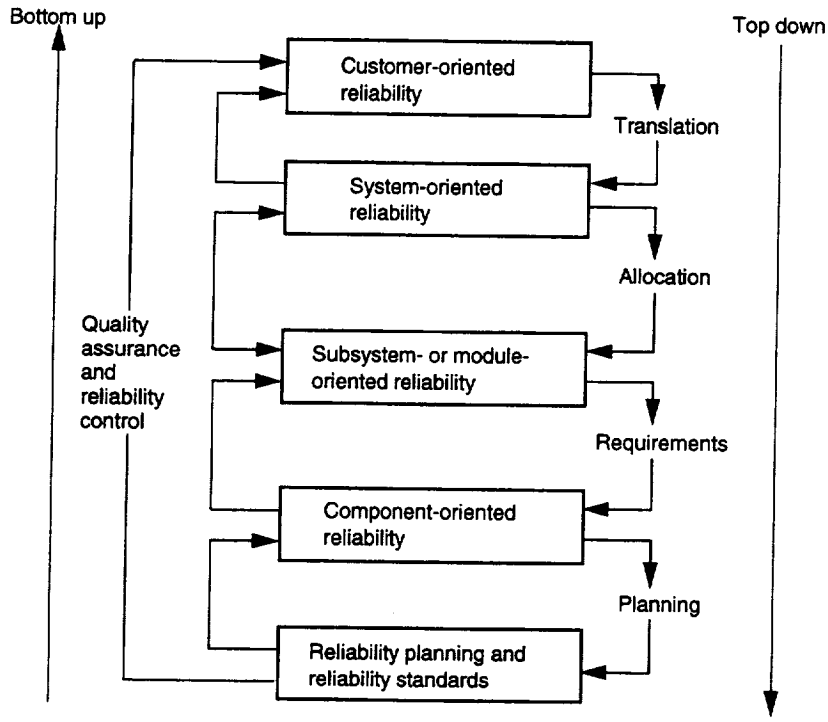


Figure 10–3.—Overall reliability process.

Reliability Management Activities

Performance Requirements

It is often difficult to translate customer performance requirements into design requirements, especially in the area of quality and reliability. Reliability encompasses both quantitative and qualitative measures. New terms in the computer industry, such as "robustness," are not formally metricized. However, we can adapt concepts for the overall performance process (ref. 10–12) to apply to reliability as presented in figure 10–3.

If a business' matrix of reliability requirements is reduced to one or more models, subjective and qualitative customer-oriented reliability measures can be translated into quantitative system-oriented reliability criteria. Figure 10–3 identifies both the top-down and bottom-up approaches to reliability validation, which include (1) translation, (2) allocation, (3) requirements, and (4) planning.

With the identification of the agreed-to system-oriented reliability criteria, designer-oriented subsystem or module reliability parameters can be allocated as shown in figure 10–3, generally by a system reliability team. The team evaluates simple versus redundant configurations, levels of fault detection and correction implementations, software considerations, and so forth. System or module reliability modeling may specify reliability requirements for specific components. An example of such modeling is a failure modes and effects analysis (FMEA) performed on a product to predict the probability of network failures due to a single failure or due to a failure after an accumulation of undetected failures.

For example, a replacement product was to use a very large-scale integration (VLSI) implementation, and the protection against network failures needed to be assessed. An investigation found no apparent standard industry FMEA method for VLSI components. Because future VLSI products may show an increasing need for FMEA, it is important that an industry standard be generated. In the network examples discussed, a single fault could directly cause a customer-oriented problem.

The bottom-up approach to reliability validation ensures customer satisfaction. The appropriate certification, process metrics, and statistical in-process tests must be designed from the customer viewpoint. A step-by-step upward certification and design review using process metrics can be designed to ensure customer-oriented reliability. In addition, we can see the need for the independent upward path from reliability planning and standards to customer-oriented reliability in figure 10–3. This is the key to success, since reliability control cannot be bypassed or eliminated from design- or performance-related issues.

Specification Targets

A system can have a detailed performance or reliability specification that is based on customer requirements. The survivability of a telecommunications network is defined as the ability of the network to perform under stress caused by cable cuts or sudden and lengthy traffic overloads and after failures including equipment breakdowns. Thus, performance and availability have been combined into a unified metric. One area of

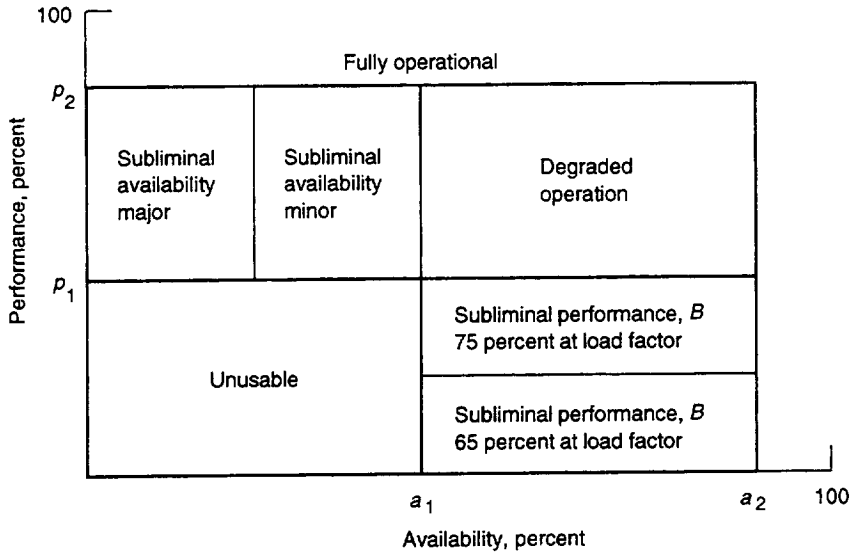


Figure 10–4.—Specification target (ref. 10–14).

telecommunications where these principles have been applied is the design and implementation of fiber-based networks. Reference 10–13 states that "the statistical observation that on the average 56 percent of the pairs in a copper cable are cut when the cable is dug up, makes the copper network 'structurally survivable.' " On the other hand, a fiber network can be assumed to be an all-or-nothing situation with 100 percent of the circuits being affected by a cable cut, failure, or other destruction. In this case study, according to reference 10–13, "cross connects and allocatable capacity are utilized by the intelligent network operation system to dynamically reconfigure the network in the case of failures." Figure 10–4 (from ref. 10–14) presents a concept for specification targets.

Field Studies

The customer may observe specific results of availability. For instance, figure 10–5 has been the basis for the proposal of an IEC technology trend document (ref. 10–15).

System reliability testing is performed today to benchmark the reliability, availability, and dependability metrics of complex new hardware and software programs. Figure 10–6 (taken from ref. 10–1) presents the traditional viewpoint of the design, development, and production community on cumulative reliability growth. It is possible that the same data generated both curves in figure 10–6. When we measure the cumulative reliability growth, the decline of production coupled with a decline of reliability is masked. If we track the product on a quarterly basis, often the product shows a relaxation of process control, incorporation of old, marginal components into the last year's product manufacture, failure to incorporate the latest changes into service manuals, knowledgeable personnel transferred to other products, and so forth. Thus, there is a need to track specific products on a quarterly basis (ref. 10–1).

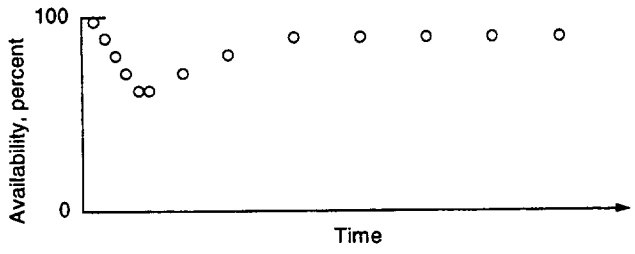


Figure 10–5.—Software availability.

Human Reliability

Analysis Methods

The major objectives of reliability management are to ensure that a selected reliability level for a product can be achieved on schedule in a cost-effective manner and that the customer perceives the selected reliability level. The current emphasis in reliability management is on meeting or exceeding customer expectations. We can view this as a challenge, but it should be viewed as the bridge between the user and the producer or provider. This bridge can be titled "human reliability." In the past, the producer was concerned with the process and the product and found reliability measurements that addressed both. Often there was no correlation between field data, the customer's perception of reliability, and the producer's reliability metrics. Surveys then began to indicate that the customer or user distinguished between reliability performance, response to order placement, technical support, service quality, and so on.

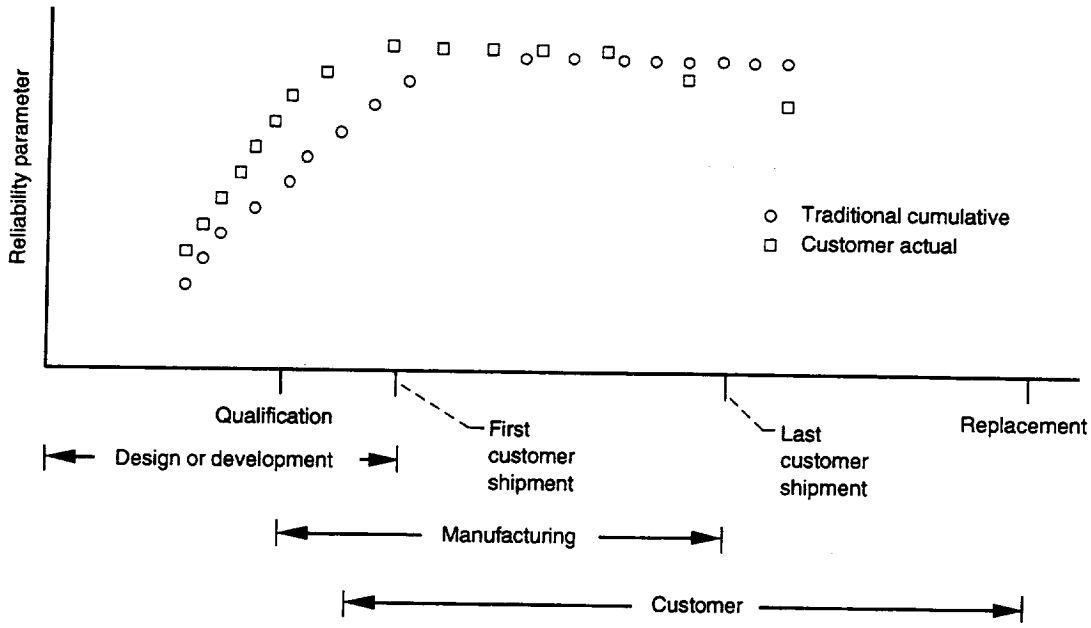


Figure 10–6.—Traditional viewpoint of reliability growth (ref. 10–1).

Human Errors

Human reliability is defined (ref. 10–16) as "the probability of accomplishing a job or task successfully by humans at any required stage in system operations within a specified minimum time limit (if the time requirement is specified)." Although customers generally are not yet requiring human reliability models in addition to the requested hardware and software reliability models, the science of human reliability is well established.

Example

Presently, the focus in design is shifting from hardware and software reliability to human reliability. A recent 2 1/2-year study by Bell Communication Research (ref. 10–17) indicated that reliability in planning, design, and field maintenance procedures must be focused on procedural errors, inadequate emergency actions, recovery and diagnostic programs, the design of preventive measures to reduce the likelihood of procedural errors, and the improvement of the human factors in the design and subsequent documentation. The study revealed the following results for outages or crashes as shown in figure 10–7. Approximately 40 percent of outage events and downtime is due to procedural problems (human error). In fact, if software recovery problems are included with procedural problems, 62 percent of the events and 68 percent of the downtime are due to human error. Therefore, human reliability planning, modeling, design, and implementation must be focused on to achieve customer satisfaction.

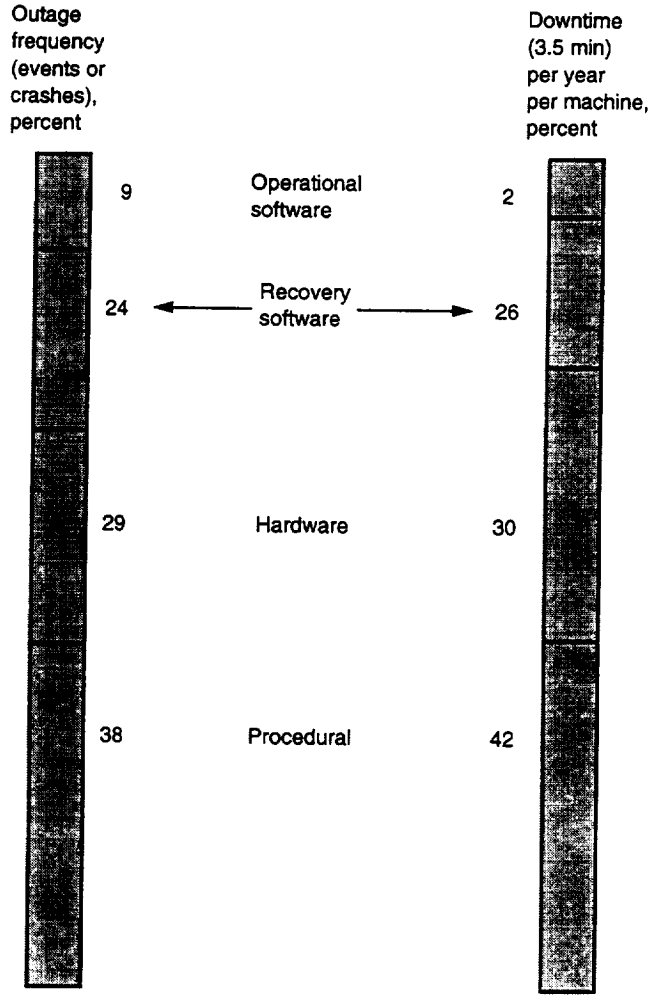


Figure 10–7.—Reliability characteristics.

Presentation of Reliability

Reliability testing usually occurs during product development and ends with the first product shipment. However, product reliability testing can be cost effectively run through the manufacturing life of the product to achieve both continued customer satisfaction and the inherent reliability of the product.

A major concern in planning reliability testing is the maturity of the specific manufacturing facility. For instance, a new plant may initially need three to five failures per week of tested product under controlled test environments to shape the manufacturing process and the product specifics. Therefore, detailed failure analysis will be conducted on 150 to 250 failed items per year. Once plant personnel begin to feel comfortable as a team and several of the plant's processes, products, or both are certified, the goal of one failure per week can be instituted in a medium-mature plant. The team in a mature plant with few failures can observe leading indicators that forewarn of possible problems and can prevent them from entering into the shipped product. Thus, in a mature plant the goal of one failure per 2 weeks can suffice as a benchmark for quality operations to achieve product reliability.

Engineering and Manufacturing

Measuring reliability in a practical way is a challenge. Reliability grows with product, process, and customer use maturity. We could measure, for example, the reliability at the first customer shipment and the reliability during a 5-year production life. An effective start may be to establish a three- to five-level reliability tier concept (ref. 10–18). For example, table 10–7 presents a five-tier reliability concept. With this concept, products can achieve the first customer shipment at a mean time between failures (MTBF) of T(min). Manufacturing and service will accept risks until T(spec) is reached. Manufacturing has a commitment to drive the MTBF of the product up to T(spec), and engineering has a commitment to provide resources for solving design problems until T(spec) is reached. The qualification team working with this process is now involved throughout the design qualification process through field feedback. Ideally, the MTBF's of tiers 2 to 5 would be equal; however, the calibration of reliability modeling tools and the accuracy of field MTBF measurements are challenges yet to be met in some corporations and industries. Thus, a three- to five-tier approach is a practical and effective solution for developing reliability measurements.

TABLE 10–7.—FIVE-TIER RELIABILITY CONCEPT

Tier	Mean time between failures	Description
1	T(min)	Minimum demonstrated MTBF before shipping (statistical test)
2	T(spec)	Specified MTBF that meets market needs and supports service pricing
3	T(design)	Design goal MTBF (calculation)
4	T(intrinsic)	Intrinsic MTBF (plant measurement)
5	T(field	Field MTBF measurement

Although the MTBF is between T(min) and T(spec), progress is tracked toward T(spec) as a goal. The point is to find and fix the problems and thus improve the reliability of the product. Teamwork and commonality of purpose with manufacturing and engineering are necessary to deal with real problems and not symptoms. After T(spec) has been achieved, an "insurance policy" is necessary to determine if anything has gone radically wrong. This can be a gross evaluation based on limited data as the "premiums" for a perfect "insurance policy" are too high. Once T(spec) has been demonstrated, a trigger can be set at the 50-percent lower MTBF limit for control purposes. Improvement plans at this level should be based on the return on investment. At maturity, T(intrinsic), dependence on reliability testing can be reduced. A few suggestions for reductions are testing fewer samples, shortening tests, and skipping testing for 1 or 2 months when the personnel feel comfortable with the product or process. With a reduced dependence on reliability testing, other manufacturing process data can be used for full control.

User or Customer

Reliability growth has been studied, modeled, and analyzed—usually from the design and development viewpoint. Seldom is the process or product studied from the customer's or user's perspective. Furthermore, the reliability that the first customer observes with the first customer shipment can be quite different from the reliability that a customer will observe with a unit or system produced 5 years later, or the last customer shipment. Because the customer's experience can vary with the maturity of a system, reliability growth is an important concept to customers and should be considered in the customer's purchasing decision.

The key to reliability growth is the ability to define the goals for the product or service from the customer's perspective while reflecting the actual situation in which the customer obtains the product or service. For large telecommunications switching systems, there has been a rule of thumb for determining reliability growth. Often systems have been allowed to operate at a lower availability than the specified availability goal for the first 6 months to 1 year of operation (ref. 10–19). In addition, component part replacement rates have often been allowed to be 50 percent higher than specified for the first 6 months of operation. These allowances accommodated craftspersons learning patterns, software patches, design errors, and so on.

TABLE 10–8.—1980 GENERIC QUALITY METRICS

[From ref. 10–20.]

Metric	Implementation phase				
	Requirements	Design	Laboratory system test	Field test	Field performance
Open questions	0	-------	-------	-------	-------
Problems fixed, per words	----	-------	1/500	1/1000	1/1000
Problems open, per words	----	1/5000	1/5000	1/2000	1/2000
Interrupts, per day	----	-------	<20	<20	<25
Audits, per day	----	0	<10	<10	<25
Service affective incidents, per office month	----	-------	0	0	1.8
Reinitializations, per month	----	-------	↓	↓	1
Cutoff calls, per 10 000	----	-------	↓	↓	<0.2
Denied calls, per 10 000	----	-------	↓	↓	<0.7
Trunk out of service, min/yr	----	-------	↓	↓	20

TABLE 10–9.—PRODUCTION LIFE-CYCLE RELIABILITY GROWTH CHART

System size	Year								
	1987				1988		...	1994	
	Quarter								
	Q1	Q2	Q3	Q4	Q1	Q2	...	Q3	Q4
Small system:									
Reliability growth, percent	5	0	0	0	0	0	...	0	0
Time to steady state, months	3	0	0	0	0	0	...	0	0
Medium system:									
Reliability growth, percent	100	50	25	10	10	10	...	10	10
Time to steady state, months	6	3	2	1	1	1		1	1
Large system:									
Reliability growth, percent	200	100	50	50	33	33	...	20	20
Time to steady state, months	12	9	6	3	3	3	...	3	3

The key to reliability growth is to have the growth measurement encompass the entire life cycle of the product. The concept is not new, only here the emphasis is placed on the customer's perspective. Reference 10–20 presents the goals of software reliability growth (table 10–8).

Table 10–8 covers a large complex system with built-in fault tolerance. Reference 10–21 regarded this system as not "technically or economically feasible to detect and fix all software problems in a system as large as No. 4 ESS [electronic switching system]. Consequently, a strong emphasis has been placed on making it sufficiently tolerant of software errors to provide successful operation and fault recovery in an environment containing software problems."

Reliability growth can be specified from "day 1" on a product development and can be measured or controlled on a product with a 10-year life until "day 5000." We can apply the philosophy of reliability knowledge generation principles, which is to generate reliability knowledge at the earliest possible time in the planning process and to add to this base for the duration of the product's useful life. To accurately measure and control reliability growth, we must examine the entire manufacturing life cycle. One method is the construction of a production life-cycle reliability growth chart.

Table 10–9 presents a chart for setting goals for small (e.g., a 60-line PABX or a personal computer), medium, and large systems. Small systems must achieve manufacturing, shipping, and installation maturity in 3 months to gain and keep a market share for present and future products. This is an achievable but difficult goal to reach. The difference in reliability growth characterization between small systems and larger systems is that the software-hardware-firmware interaction, coupled with the human factors of production, installation, and usage, limits the reliability growth over the production life cycle for most large, complex systems.

In certain large telecommunications systems, the long installation time allows the electronic part reliability to grow so that the customer observes the design growth and the production growth. Large, complex systems often offer a unique environment to each product installation, which dictates that a significant reliability growth will occur. Yet, with the difference that size and complexity impose on the resultant product reliability growth, corporations with a wide scope of product lines should not present overall reliability growth curves on a corporate basis but must present individual product line reliability growth pictures to achieve total customer satisfaction.

References

10–1. Malec, H.A.: Reliability Growth From the Customer Perspective. IEEE J. Sell Topics Commun., vol. 6, no. 8, Oct. 1988, pp. 1287–1293.

10–2. Dhillon, B.S.; and Reiche, H.: Reliability and Maintainability Management. Van Nostrand Reinhold, 1985.

10–3. Spragins, J.D., et al.: Current Telecommunication Network Reliability Models: A Critical Assessment. IEEE J. Sell Topics Commun., vol. SAC–4, no. 7, Oct. 1986, pp. 1168–1173.

10–4. PNPM '87, International Workshop on Petri Nets and Performance Models. IEEE Computer Society Press, 1987.

10–5. Malec, H.A.: Reliability Optimization in Telephone Switching Systems Design. IEEE Trans. Rel., vol. R–26, no. 3, Aug. 1977, pp. 203–208.

10–6. Petri, C.A.: Communication With Automata. Final Report, Vol. I, Supplement I, RADC TR 65–377–VOL I–SUPPL I, Applied Data Research, Princeton, NJ, Jan. 1966.

10–7. Woodside, C.M.: Innovator of Timed Petri Nets Keynotes International Workshop. Spectrum, Mar. 1988, p. 143.

10–8. Peterson, J.L.: Petri Net Theory and the Modeling of Systems. Prentice-Hall, Inc., 1981.

10–9. Malec, H.A.; and Steinhorn, D.: A New Technique for Depot and Sub-Depot Spares. IEEE Trans. Rel., vol. R–29, no. 5, Dec. 1980, pp. 381–386.

10–10. Malec, H.A.: Maintenance Techniques in Distributed Communications Switching Systems. IEEE Trans. Rel., vol. R–30, no. 3, Aug. 1981, pp. 253–257.

10–11. Murray, L.R.; and Morris, R.S.: Spare/Repair Parts Provisioning Recommendations. 1979 IEEE Annual Reliability and Maintainability Symposium, IEEE, 1979, pp. 224–230.

10–12. Gruber, J.G., et al.: Quality-of-Service in Evolving Telecommunications Networks. IEEE J. Sell Topics Commun., vol. SAC–4, no. 7, Oct. 1986, pp. 1084–1089.

10–13. Roohy-Laleh, E., et al.: A Procedure for Designing a Low Connected Survivable Fiber Network. IEEE J. Sell Topics Commun., vol. SAC–4, no. 7, Oct. 1986, pp. 1112–1117.

10–14. Jones, D.R.; and Malec, H.A.: Communications Systems Performability: New Horizons. 1989 IEEE International Conference on Communications, vol. 1, IEEE, 1989, pp. 1.4.1–1.4.9.

10–15. Decroix, A.: Analysis and Evaluation of Reliability and Availability of Software. IEC–TC–56 draft, 56/WG10 (DECROIX)02, June 1986.

10–16. Dhillon, B.S.: Human Reliability: With Human Factors. Pergamon Press, 1986.

10–17. Ali, S.R.: Analysis of Total Outage Data for Stored Program Control Switching Systems. IEEE J. Sell Topics Commun., vol. SAC–4, no. 7, Oct. 1986, pp. 1044–1046.

10–18. Malec, H.A.: Product/Process Reliability Testing. 1987 IEEE International Conference on Communications, IEEE, 1987, pp. 1198–1202.

10–19. Conroy, R.A.; Malec, H.A.; and Van Goethem, J.: The Design, Applications, and Performance of the System-12 Distributed Computer Architecture. First International Conference on Computers and Applications, E.A. Parrish and S. Jiang, eds., IEEE, 1984, pp. 186–195.

10–20. Giloth, P.K.; and Witsken, J.R.: No. 4 ESS—Design and Performance of Reliable Switching Software. International Switching Symposium (ISS '81 CIC), IEEE 1981, pp. 33A1/1–9.

10–21. Davis, E.A.; and Giloth, P.K.: Performance Objectives and Service Experience. Bell Syst. Tech. J., vol. 60, no. 6, 1981, pp. 1203–1224.

Reliability Training[1]

1. Reliability management is concerned with what phases of the life cycle?

A. Design and development B. Manufacturing C. Customer D. All of the above

2. Name a new style of organizing reliability activities.

A. Functional B. Team C. Matrix D. Council

3. What are the functions of the diagnostic team or person?

A. Review the internal reliability status
B. Review reliability as perceived by the customer
C. Recommend tasks to the reliability council
D. Diagnose problems
E. Design experiments
F. Collect and analyze data
G. All of the above

4. Name a goal category for a telephone instrument.

A. Loss of service
B. Mean time between failures
C. Mishandled calls
D. All of the above

5. A PABX with 80.0 lines has a service level reliability specification for the mean time between major losses of service (MTBF) of

A. 150 days B. 1 hour C. 0.1 percent D. All of the above

6. A Petri net is composed of which of the following parts?

A. A set of places
B. A set of transitions
C. An input function
D. An output function
E. All of the above

7. For a telecommunications system, what is the spares adequacy level for a network subsystem with spares depots?

A. 0.999 B. 0.995 C. 0.95

8. Turnaround time depends on

A. Replaceable unit failure rate
B. Repair location
C. Repair cost
D. All of the above

[1]Answers are given at the end of this manual.

9. Spares adequacy is the probability of having spares available.

A. True B. False C. Do not know

10. What is the normal maintenance action recommendation for the site to defer repair for (days) during off-shift time?

A. 0 B. 2 C. 1

11. The bottom-up approach to reliability makes use of planning, requirements, allocations, and customer orientation.

A. True B. False C. Do not know

12. Specification targets can be used to define what performance and availability requirements?

A. Fully operational
B. Subliminal availability
C. Degraded operation
D. Unusable
E. Subliminal performance
F. All of the above

13. Tracking a product on a quarterly basis often shows

A. A relaxation of process control
B. Incorporation of old marginal components
C. Failure to incorporate the latest changes into service manuals
D. Knowledgeable personnel transferred to other products
E. All of the above

14. If we consider recovery software and procedural problems as human error, human error can account for what percentage of outage and downtime problems?

a. Outage frequency, percent of events/crashes	A. 38	B. 55	C. 62
b. Downtime (3.5 min), percent per year per machine	A. 42	B. 51	C. 68

15. As a benchmark for quality operations to achieve product reliability, what is a reasonable goal (failures per week) for a mature plant?

A. 3.0 B. 1.0 C. 0.5

16. While the MTBF is between *T*(min) and *T*(spec), progress is tracked toward what goal?

A. *T*(design) B. *T*(spec) C. *T*(intrinsic)

17. The key to reliability growth is to have the growth measurement encompass

A. The design phase
B. The manufacturing phase
C. The testing phase
D. The user phase
E. The entire life cycle of the product

18. For a No. 4 ESS system in the field-test phase, the number of interrupts per day can be

A. <20 B. >20 C. 40

19. An electronic system must achieve manufacturing, shipping, and installation maturity in what period of time (months) to gain and keep market share?

a. Small system	A. 1	B. 2	C. 3
b. Medium system	A. 4	B. 6	C. 12
c. Large system	A. 12	B. 8	C. 16

Chapter 11
Designing for Maintainability and System Availability

Introduction

The final goal for a delivered system (aircraft, a car, an avionics box, or a computer) should be its availability to operate and to perform its intended function over the expected design life. Hence, in designing a system, we cannot think in terms of delivering the system and just walking away. The system supplier needs to provide support throughout the operating life of the product, which involves the concepts presented in figure 11–1. Here, supportability requires an effective combination of reliability, maintainability, logistics, operations, and safety engineering to have a system that is available for its intended use throughout the designated mission lifetime (see the Definitions section for more details). Maintainability is the key to providing effective support, upkeep, modification, and upgrading throughout the lifetime of the system.

This chapter will concentrate on maintainability and its integration into the system engineering and design process. The topics to be covered include the elements of maintainability, the total cost of ownership, and the ways that system availability, maintenance, and logistics costs plus spare parts costs affect the overall program costs. System analysis and maintainability will show how maintainability fits into the overall systems approach to project development. Maintainability processes and documents will focus on how maintainability is to be performed and what documents are typically generated for a large-scale program. Maintainability analysis shows how tradeoffs can be performed for various alternative components. Note that the majority of the mathematical analysis and examples will concentrate on maintainability analysis at the component level or below. In a highly complex and redundant system, the evaluation availability at a system level may be extremely difficult and is beyond the scope of this manual. Redundancy, switches and software that can be used to bypass failed subsystems, and other methodologies can allow a system to operate even with some system degradation. The treatment of these types of problems is beyond the scope of this manual. Finally, specific problems for hands-on training follow the concluding section.

Definitions

Reliability is the probability that an item can perform its intended functions for a specific interval under stated conditions. What is the chance that a failure will stop the system from operating? Usually the failure is random and unexpected, not predicted as with brake wearout or a clutch or fatigue failure when a given input load spectrum is known.

Availability is a measure of the degree to which an item is in the operable and commitable state at the start of the mission, when the mission is called for at an unknown (random) point in time. Also, it is the probability of system readiness over a long interval of time. Will the system be ready to operate when needed? Does it have very high reliability or very small maintenance requirements (easily maintainable and having a good supply of spare parts) or a combination of both? For example, what was the percentage of times a car started out of the total number of tries over its lifetime? Alternatively, how many days was it in the driveway ready to start as opposed to being in the garage for repairs?

Maintainability is a system effectiveness concept that measures the ease and rapidity with which a system or equipment is restored to operational status after failing. Also, it is the probability that a failed system can be restored to operating condition in a specified interval of downtime. How easy is it to diagnose the problems in a failed (or marginally operable) system and how easy is it to replace the failed components (or software) after this diagnosis has been made? If a system is not reliable and is prone to partial or complete failures, if it is difficult to find out what is causing the system to malfunction, or

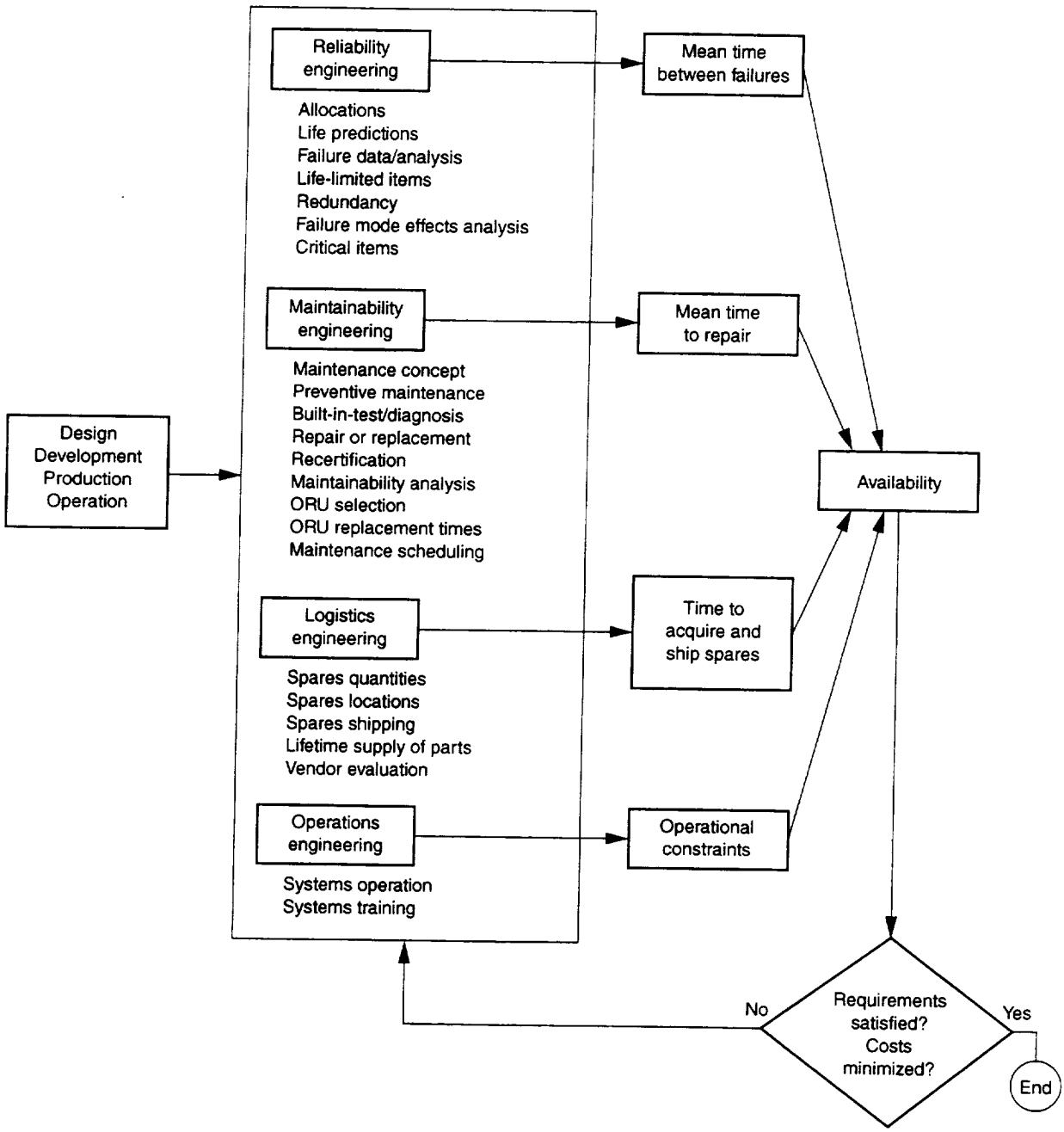


Figure 11–1.—System supportability requirements.

if it is difficult to get to and replace failed components, we have a serious problem that must be corrected (ref. 1).

Safety analysis is that which considers the possible types, reasons, and effects of operation and failures on the system as they affect the personal safety of those who operate or maintain it.

Logistics is the art and science of the management, engineering, and technical activities concerned with requirements, design, and planning and maintaining resources to support objectives, plans, and operations.

Operations defines the environment, schedule, loading, and input and output parameters a system will need to function and the tasks it will perform.

Importance of Maintainability

The importance of maintainability is further noted in figure 11–2. Too often, the performance specifications or the appearance of a product are the overriding factors in its acquisition or purchase. This attitude can be extremely detrimental, especially when the first failure occurs and it is realized that the availability of critical parts and the ease of maintenance keep critical systems operating. A large integrated system can come from the best possible design, utilizing the newest technology; it can be a work of art and outperform any competitive system, but who would want it if

- System breakdowns could not be diagnosed to a level of detail needed to pinpoint the problem in a short time.
- Spare parts were not readily available.
- Repair required extremely long lead times.
- Installing the spare parts was extremely difficult.
- Checkout and/or alignment of spare parts was difficult.

For all practical purposes, such a system is not available (operational).

Figure 11–2.—Importance of maintainability.

Elements of Maintainability

We need to consider up front in our design what must be done to maintain the system. Either the system will not fail for the entire mission or some parts of the system will fail and will need to be replaced. If we do not have a system with perfect reliability (there is wearout), the following questions (as illustrated by fig. 11–3) should be asked:

(1) What parts have high failure rates and how will their failure be diagnosed? For example, if a cathode ray tube (CRT) screen does not show a display, has the screen failed or has a power supply failed or has a computer stopped sending the screen data?

(2) Can various problems be diagnosed easily? How quickly can the problem be diagnosed? If there is an intermittent fault, can information during this anomaly be retrieved later? If a failure cannot be isolated or if insufficient diagnostic capabilities are built into the system, restoration can be a time-consuming task.

(3) How quickly can the system be repaired? Has the system been segmented into easily replaceable units? Are parts buried on top of one another with hundreds of attachment points between units? Also, can software be used to detect and route around a hardware failure and make the failure transparent to the user?

(4) Where will spare parts be stored? How many spare units should be ordered? Will parts for a unit in Washington be lost in a warehouse in Los Angeles? Will there be an oversupply of one unit and a shortage of another?

(5) Will a failed unit be discarded or repaired? If it is to be repaired, where should it be repaired? What equipment and personnel are required to do the work?

(6) Will unique parts be available to repair the unit? Will some unique part such as a traveling wave tube or a low-noise amplifier still be manufactured when it is needed to be replaced to repair a unit? Will the supplier who sold the unit repair it? If repairs are agreed to, will the supplier still be in business (logistics issues)?

When a product is planned, all these questions must be answered. Although some of these questions overlap with logistics (the science of supply and support of a system throughout its product life cycle), they must all be addressed. Early in the design phase of the product, the maintenance concept to be used for the system and the design for maintainability must be examined first. The following definitions will be helpful in making decisions in the design phase.

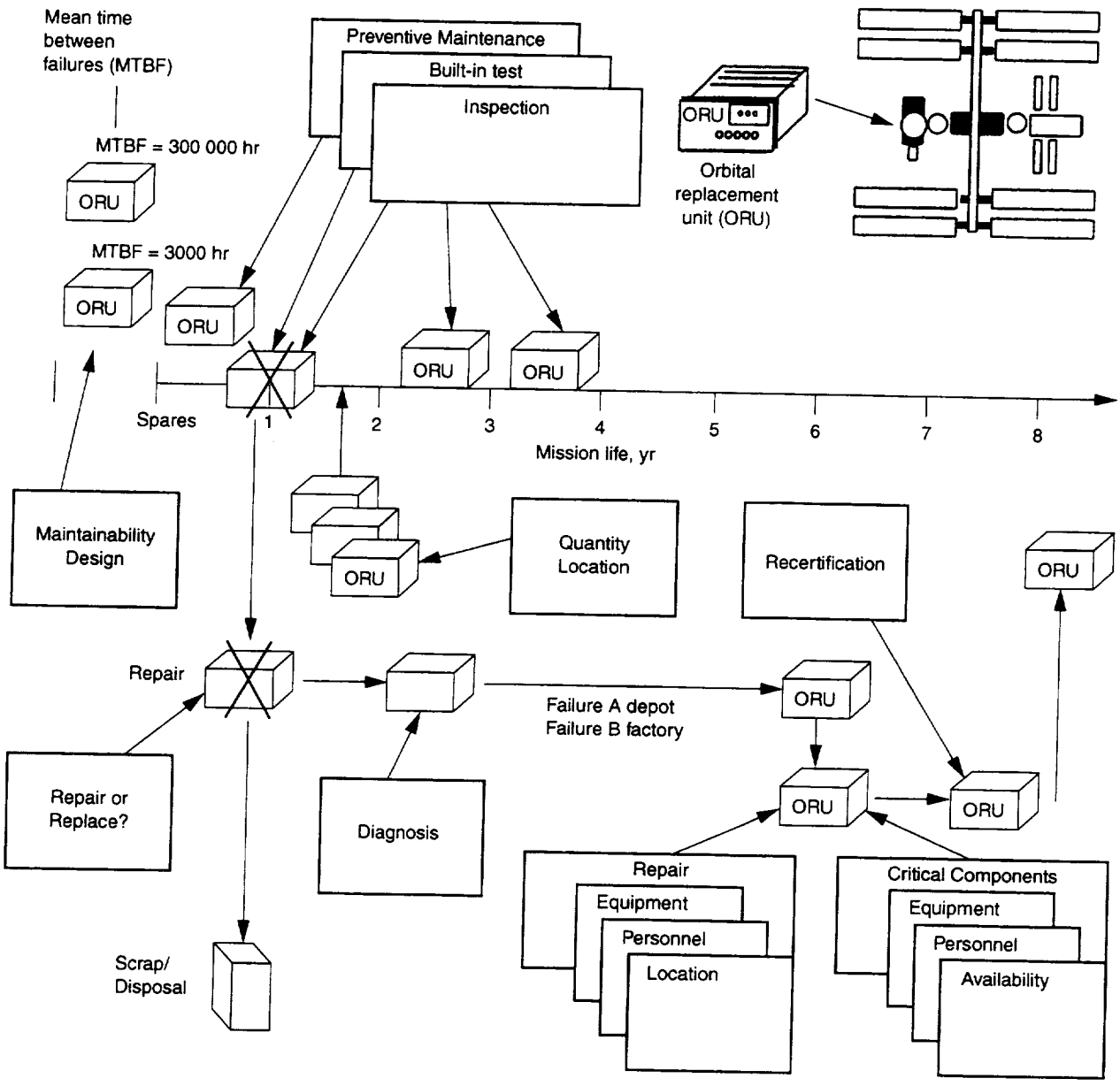


Figure 11–3.—Elements of maintainability.

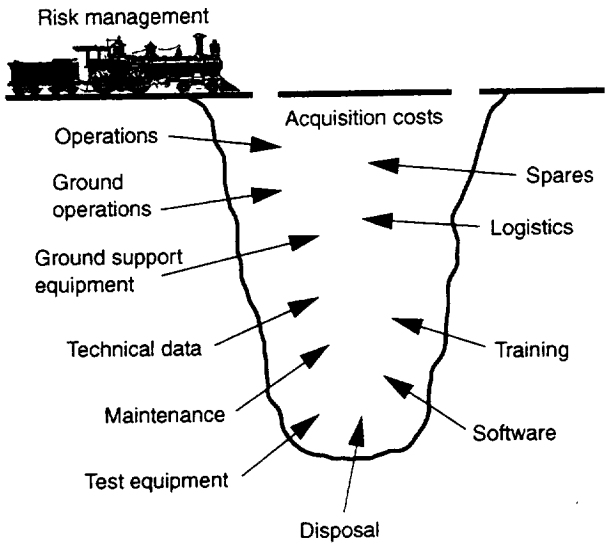


Figure 11–4.—Hidden system costs.

Total Cost of Ownership

The total life-cycle cost of a unit must be assessed when evaluating project cost. The need to support the system through an effective logistics program that includes maintainability is of paramount importance (fig. 11–4).

The project can follow a faster development course and procure less reliable hardware; however, the maintenance cost will make the project more expensive. Additionally, if the unit is not available because of lengthy maintenance processes or lack of spare parts, additional units must be procured to have the fleet strength at the desired level (whether it is delivery vehicles or research aircraft). The total cost of ownership includes

- Total life-cycle: more than just the cost of flight units and a prototype unit
- Availability of the unit: more than the advertised features when it is running (backup systems needed for excessive downtime)
- Maintenance and logistics: often 40 to 60 percent of the total system costs
- Spares: a function of reliability and speed with which the system can be maintained

Often all the costs associated with a project are not considered. Besides just the cost of producing the units, a huge amount of time and money must be expended keeping them operational throughout the mission lifetime. Total project costs are considered in table 11–1. Evident from the table is that total system costs include design and development costs and a whole host of training, operations, and maintenance costs.

As the quality and reliability of the system increase, the cost of the system classically increases. However, this increase may not necessarily occur because as the quality and reliability of the system are improved, the cost of maintenance, logistics, and spares decreases. Since total support costs are a function of maintenance costs and the cost of the total number of spares, spare repair, and spare transport, improved reliability drastically reduces the total cost of ownership, also.

TABLE 11–1.—TOTAL PROJECT COSTS

Cost item	Cost breakdown
Acquisition	Design and development Research, trades, design, analysis, prototype production and test Production
Operations	Personnel, facilities, utilities, operating supplies and other consumables, maintenance ground operations
Ground operations	Ground support engineering model and test and checkout models; maintenance for these
Ground support equipment	All test, checkout, and diagnostic equipment; purchase, storage, and calibration of ground support equipment
Technical data	All manuals, specifications, configuration management; software configuration management, data base, storage
Training	Continuous training of all operations and maintenance personnel
Maintenance	Calibration, repair, and system downtime Repair facilities Labs, depots, and others
Test equipment	Equipment used for maintenance, alignment, and calibration of the system; equipment used for recertification (e.g., flight)
Software	Maintenance, upgrades, test, and installation
Logistics	Packaging, storage, transportation, and handling; tracking support
Spares	Spare orbital replacement units and line replacement units; long-lead-time items and critical components
Disposal	Disassembling and recycling; disposing of hazardous waste

Maintainability and Systems Engineering

Figure 11–5 gives a global overview of a long-term research project, such as the space program, and shows maintainability as an integral part of it. The Horizon Mission Methodology (HMM) was developed initially for the study of breakthrough space technology. The HMM's are hypothetical space missions whose performance requirements cannot be met, even by extrapolating known space technologies. The missions serve to develop conceptual thinking and depart from simple projections and variations of existing capabilities.

The use of HMM's with breakthrough technology options (BTO's) has been an attempt to provide a systematic analytical approach to evaluate and identify technological requirements for BTO's and to assess their potential for providing revolutionary capabilities for advanced space missions.

Therefore, we can think of the space program (or other major research program) not just as a number of isolated projects but as a single unified program with a global goal (e.g., landing men on the Moon or planning a manned mission to Mars or establishing a permanent manned lunar base).

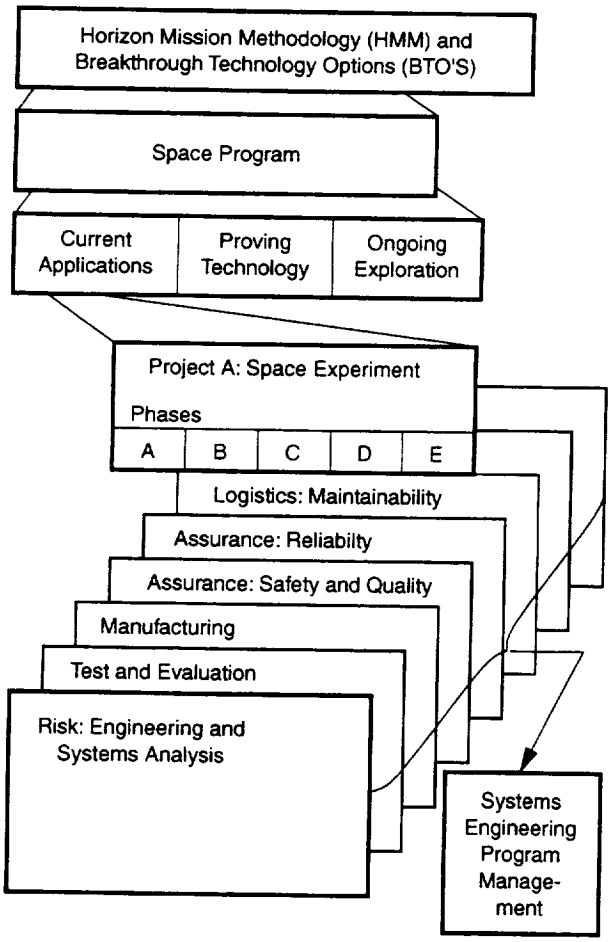


Figure 11–5.—Systems engineering and operations.

The program concept assumes a single consistent objective. It involves putting tested and proven equipment together to perform a step toward the goal. Another area of work involves developing technology and components and conducting ongoing exploration with the outer fringes of what lies ahead. At an individual project level, a number of different disciplines are brought together to design, develop, deploy, and operate the project. One of these disciplines is maintainability. Expanding the various maintainability activities over project phases gives us the chart of figure 11–6. Systems engineering at the National Aeronautics and Space Administration (NASA) uses five phases to describe a mission. Note that the maintainability program is run across all five phases. The task descriptions are also shown.

The various activities are defined in the following sections. Of great importance is that the maintainability concept of the project be introduced early in the program. Without this introduction, long-term missions will see costs rise and downtime increase. True, initial development costs may increase, but total cost will decrease. In some cases, projects have ignored maintainability and built in diagnostics to obtain budgetary approval of a new system. However, the final costs always increase as a result of this practice (ref. 2).

Finally, figure 11–7 shows the interrelationship of the various project tasks and how work and information flow between operations, reliability, and logistics functions. Basically, systems operation and mission requirements are evaluated to generate the maintainability concept. This concept is further affected by component reliability and the various reliability analyses performed. This maintenance analysis is then integrated with design engineering to develop a design that can be repaired and maintained.

Maintainability data and requirements flow to logistics to allow development of an effective support resource program. The output of the maintenance analysis is also critical to the logistics support analysis.[1] The logistics support analysis record (LSAR) and support resource development feed the plan for (1) facilities to house equipment or ground operations, (2) ground support equipment, (3) the logistics plan and other activities, (4) data (technical publication) for equipment operation and maintenance, and (5) identification of personnel and training needed to maintain, repair, and support the equipment. Finally, a maintainability demonstration is performed to evaluate the actual times needed to diagnose and physically changeout a line replaceable unit (LRU) or an orbital replaceable unit (ORU).

[1]The following general guideline distinguishes support, logistics, and maintenance for this manual. Supportability encompasses all logistics, maintainability, and sustaining engineering. Logistics is involved with all movement of orbital replaceable units (ORU's) and spare parts, the procuring and staging of spare parts, and the development of storage containers. Maintainability is responsible for (after the ORU's are located) repairing ORU's, shop replaceable units (SRU's), printed circuit boards (PCB's), which includes test and diagnostic equipment, tools; providing training, a suitable workarea, and maintenance personnel.

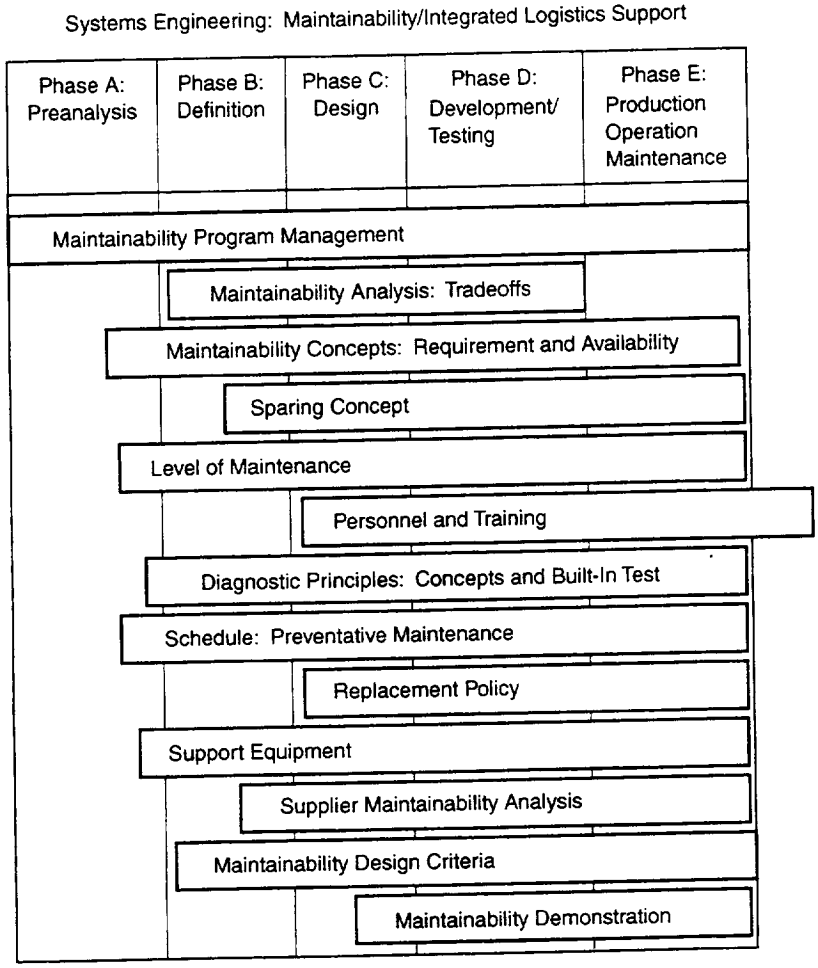


Figure 11–6.—Maintainability in system life cycle.

Maintainability Processes and Documents

The mission requirements analysis and the operational requirements of a new system are derived from the initial needs and wants of the community. Directly and simultaneously derived from this is the system maintenance concept (as described in the maintenance concept document (MCD)).

At this time, an initial draft of maintenance requirements should also be developed. Operational requirements and system requirements are funneled into the maintenance concept document, which covers every aspect of a maintenance program throughout the life of the system (see fig. 11–8) (ref. 3).

First Phase

The first phase involves planning and designing because maintainability is made a part of the design process, which includes making components easy to service. In this first step, ORU's (orbital replaceable units) or LRU's (line replaceable units) are selected. As the name implies, replaceable units can be quickly changed out to bring the system back into operation. To speed the system back into operation, it is typically divided into units that can easily be replaced on-orbit or on the flight line. A module or system is designated an ORU or an LRU if that part of the design has high modularity (can be self-contained, such as a power supply) and low connectivity (a minimum of power and data cables to other parts of the system). As we will discuss later, we must be able to diagnose that an ORU or LRU has failed. This means that maintenance on-orbit (or on the flight line) will only replace these items. The system is built, tested, shipped, and put into operation. Operations and maintenance training are also conducted.

The maintainability analysis (see fig. 11–9) also uses (1) the predicted time for corrective maintenance times the number of failures, (2) the predicted times preventive maintenance (PM) times the number of scheduled PM's and predicted times changeout of limited-life items times the number of scheduled changeouts. With these times, a prediction of overall maintenance time per period is made. Assuming that the system is shut down during maintenance, we can then predict availability.

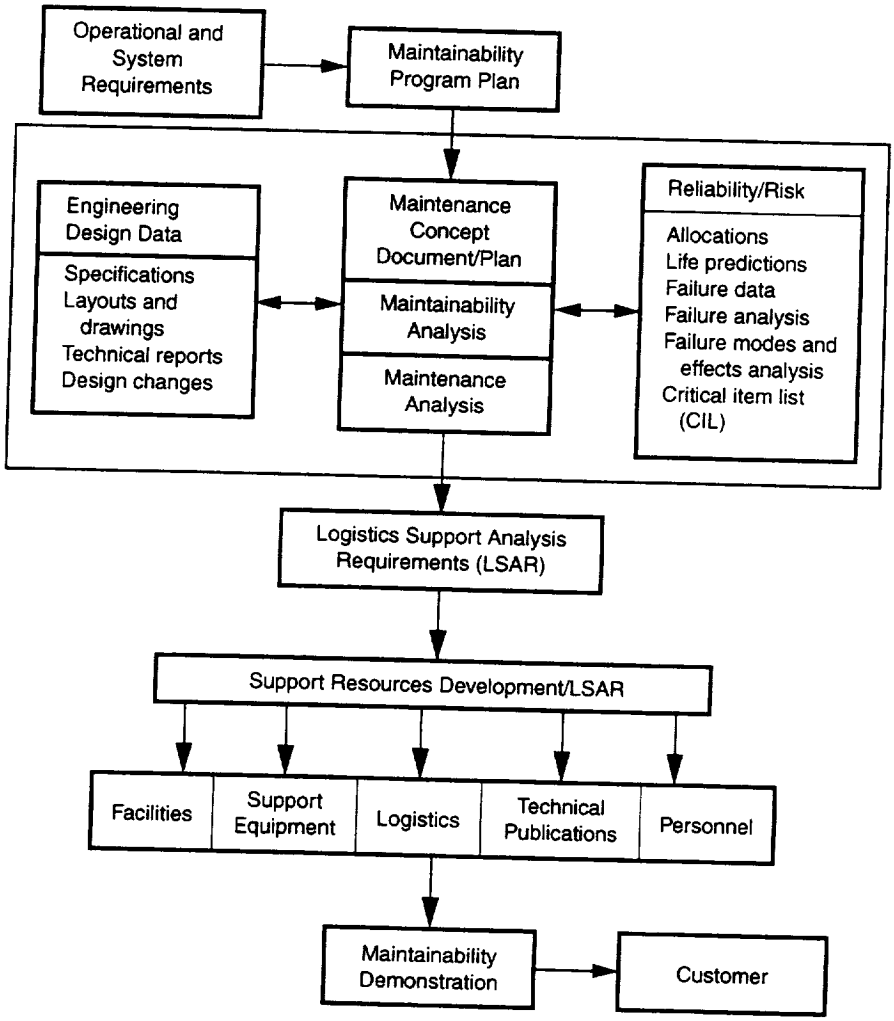


Figure 11–7.—Maintainability in systems engineering process.

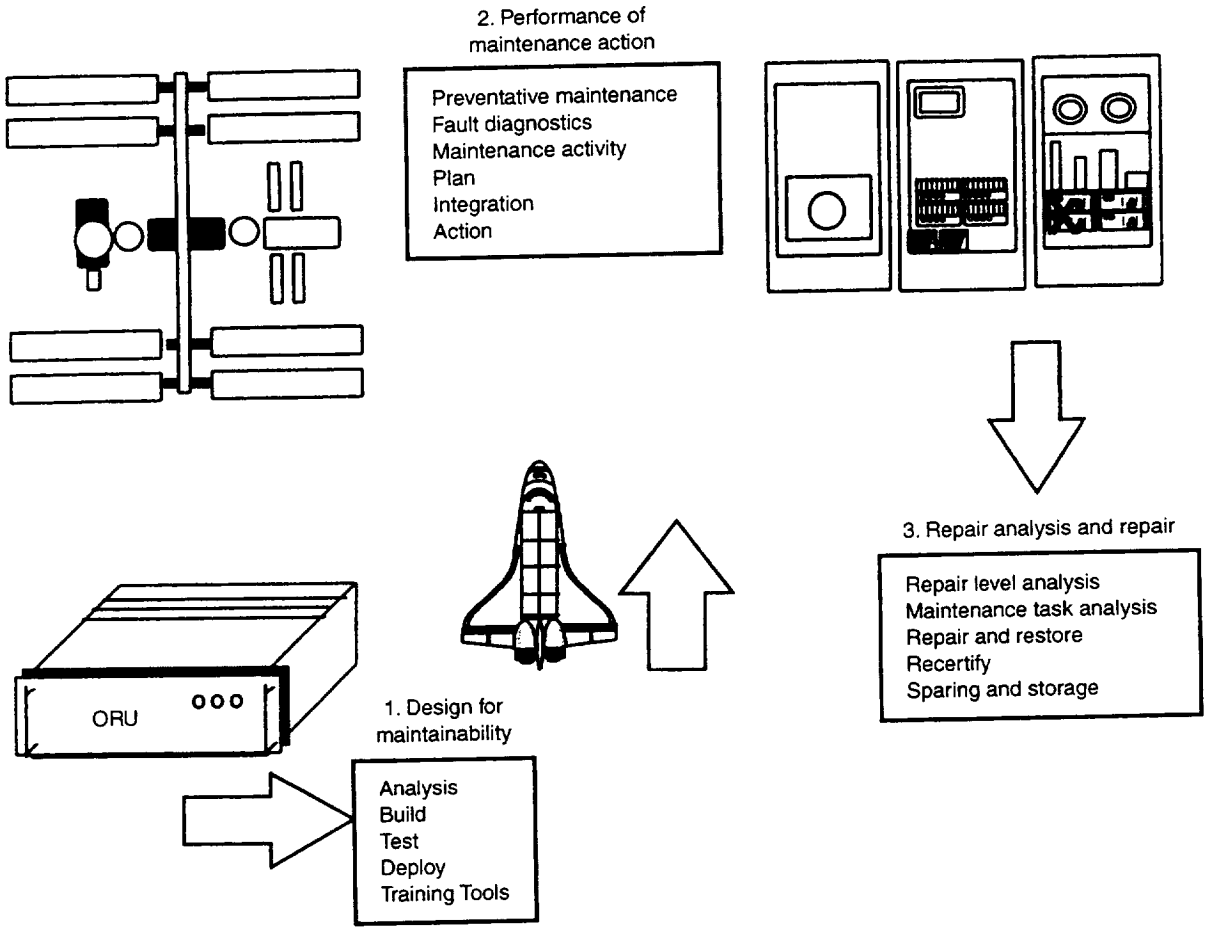


Figure 11–8.—Maintainability activities.

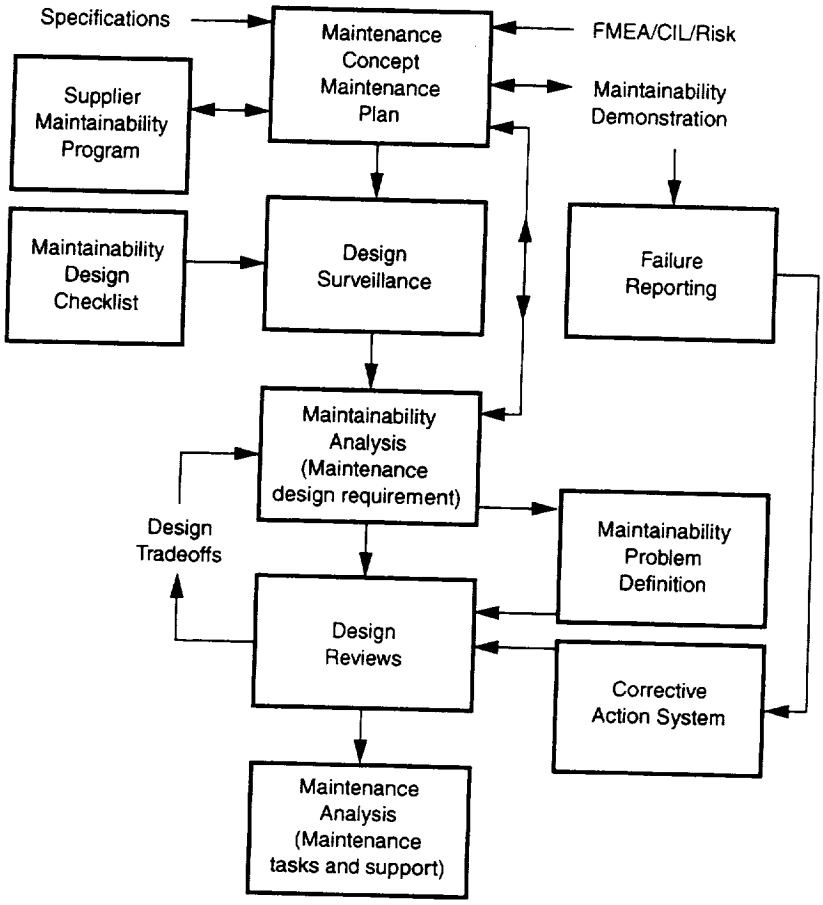


Figure 11–9.—Maintainability analysis process.

As the design matures and the failure mode and effects analysis/critical items list (FMEA/CIL) and supplier maintainability program data mature, the overall availability (as well as other maintainability figures of merit) is recalculated. The data generated by the maintainability analysis serves to appraise project management of the overall maturity of the design and the ability of the design to meet program objectives.

Second Phase

The second phase of maintenance is handling failures, performing preventive maintenance, and replacing life-limited items. Eventually the deployed unit breaks down. The failure must be detected and isolated from the actual failed ORU/LRU. How is the failure detected, and how is the maintenance action planned and executed? Can it be combined with any other maintenance actions or preventive maintenance activities? The on-orbit or flight line maintenance is performed by removing and replacing the failed unit. But what do we do with the broken ORU/LRU?

Third Phase

The third phase involves the handling of failed components. Here, repair-level analysis evaluates the failed ORU or LRU to determine whether it should be repaired or replaced. If repaired, it may be done in-house (intermediate maintenance at a maintenance depot where more specialized equipment and better diagnostic instrumentation might be available) or at the factory. (The following section discusses the Maintenance Concept Document in more detail.) Then the unit needs to be recertified, retested, finally checked out, and returned to the spare parts storage area (preferably bonded storage).

Only by developing the complete maintenance concept and the maintenance requirements early in the development process will the design really be impacted by maintenance needs. The operational requirements document, the mission (or science) requirements document, and the maintainability concept document with preliminary requirements should be the design drivers. Only then can effective trade studies, systems analysis and functional analysis, and allocation be performed. Also, trade studies with reliability and maintainability alternatives

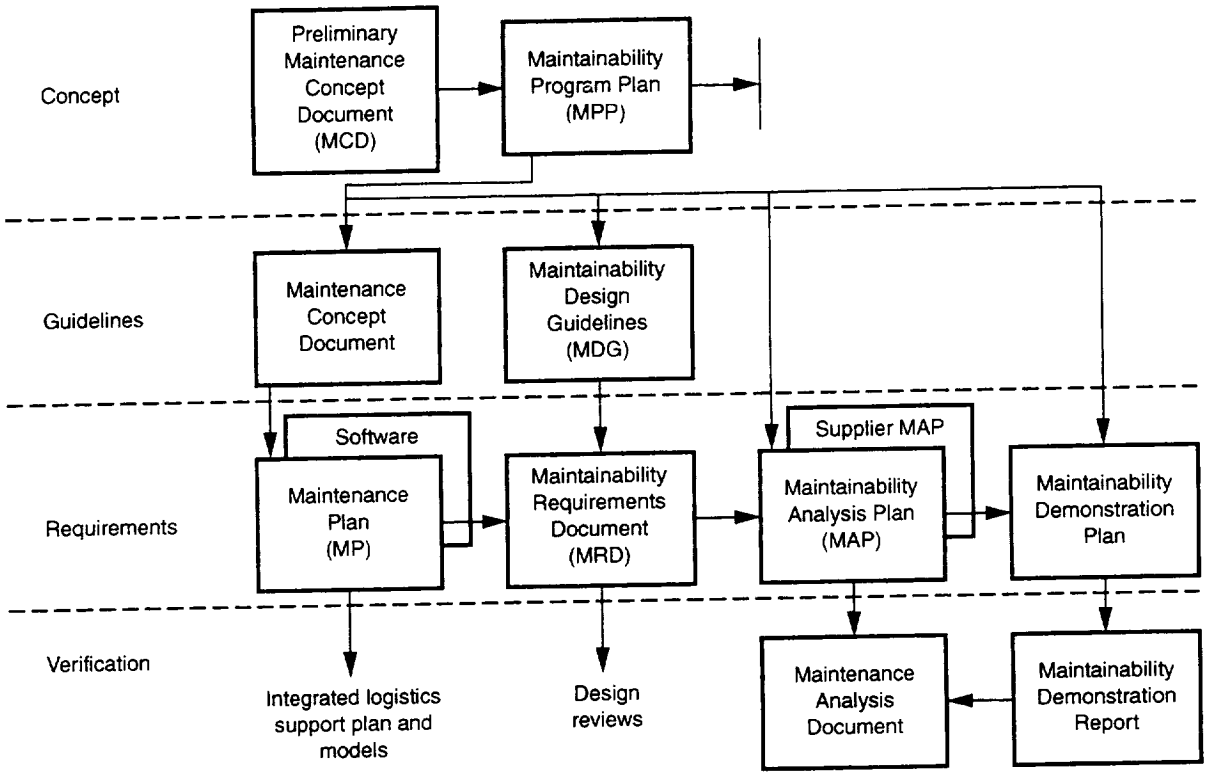


Figure 11–10.—Maintainability documentation.

can be used to evaluate total system cost. Reliability and maintainability alternative selections will drive maintenance and repair costs, shipping costs, ORU/LRU spare costs, long-lead-time components, and components manufactured by complex processes.

Documents

Several documents (fig. 11–10) typically support a large-scale engineering project (some describe the activities already discussed). They officially begin with a basic plan and the maintenance concept document (MCD). The MCD together with the operations concept document and the science requirements are the chief design and cost drivers for the future system. The individual documents are as follows:

Maintainability program plan (MPP) (required).—This document defines the overall maintainability program, activities, documents to be generated, responsibilities, interfaces with the logistics function, and the general approach to the analysis of maintenance.

Maintenance concept document (MCD) (required).—This document defines the proposed way maintenance is to be performed on the product (see fig. 11–11); gives details of the aims of the maintenance program and support locations; describes the way all maintenance actives are to be carried out (details of support and logistics may additionally be specified depending on document requirements); defines the input and output data requirements and the scheduling of maintenance activities, including the following sections:

Mission profile/system operational availability: How often and over what period of time is the system operational? What is the geographic deployment of the system and where is the location of the system that needs to be repaired?

System-level maintainability requirements: What are the allocated and actual reliability requirements and maintainability requirements (MTTR, MTBF, MLDT, MDT[2])?

Design requirements: What constitutes a maintainable element that can be removed or replaced (e.g., an orbital replaceable unit (ORU) or a line replaceable unit (LRU)?). What are the sizes and weight limits?

[2]MTTR, mean time to repair; MTBF, mean time between failures; MLDT, mean logistic delay time; MDT, maintenance downtime.

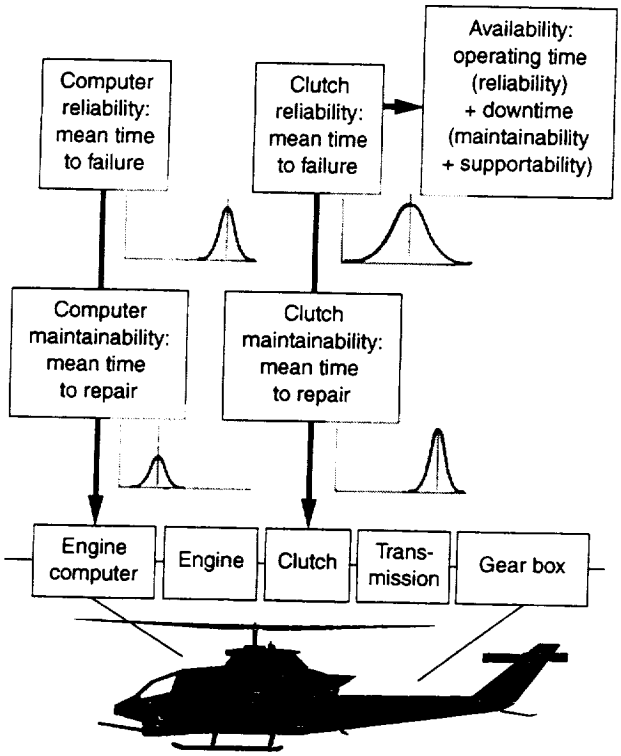


Figure 11–11.—Factors affecting maintainability.

Diagnostic principles and concepts: How will a failure be detected and isolated? How will repairs be evaluated?

Requirements for suppliers: What information about parts and components must the supplier give? How will the first-, second-, and third-tier suppliers support their products? How quickly will they be available and for how long will they be available?

Repair versus replacement policy: How is the decision made to repair or replace a unit? If repaired, how is the unit requalified?

Repair level analysis: Where will different failures be repaired? Which repairs will be made on-orbit (or on the flightline)? Which repairs will be made at an intermediate maintenance facility (depot) and which will be made at the factory?

Tools and test equipment: What diagnostic, alignment, and check-out tools will be required for each level of maintenance (repair)?

Personnel and training: What is the level of training required for the units at each level of maintenance (from simple remove and replace to detailed troubleshooting of an ORU/LRU)?

Crew considerations: What time will be allocated for preventive and corrective maintenance? How much time can a flight crew and a ground crew give to maintenance during or between missions?

Sparing concepts: Which spares will be onboard versus those delivered when needed? Will failed units be repaired or replaced? What are the general repair policies?

Elements of logistic support (optional): Where will all the test and ground support equipment and inventory control supplies be located?

Maintenance plan (MP) (required).—This document defines the actual way maintenance is to be performed on the product. The MP gives detailed requirements for repair or replacement analysis, the location for and levels of maintenance, and other detailed requirements for performing the maintenance.

Maintainability design guidelines (MDG) (optional).—This guideline contains suggestions, checklists, and descriptions of ways to make the design maintainable. Related safety and human factors and factors to consider for vendors and transportation may also be considered.

Maintainability requirements document (MRD) (required).—This document gives the specific requirements (criteria) that will facilitate maintenance or repair in the predicted environment. It contains all maintainability requirements.

Maintainability analysis plan (MAP) (required).—The maintainability analysis plan specifies how the maintainability of the system is assessed. It also documents the process that translates system operational and support requirements into detailed quantitative and qualitative maintainability requirements with the associated hardware design criteria and support requirements and provides basic analysis information on each ORU/LRU. This document includes evaluation processes for preventive, corrective, and emergency maintenance. The MAP documents the formal procedure for evaluating system and equipment design,[3] using prediction techniques failure modes and effects analysis, procedures and design data to evolve a comprehensive, quantitative description of maintainability design status, problem areas and corrective action requirements.

Supplier maintainability analysis plan (optional).—This document outlines methodology to evaluate suppliers for conformance to maintainability standards.

Maintenance analysis document (required).—This document provides the details of how each ORU/LRU is to be maintained and includes detailed maintenance tasks, maintenance task requirements, and maintenance support requirements.

Maintainability demonstration plan (optional).—This plan documents the process that translates (and verifies) system operational and support requirements into actual test plans for the maintainability of systems and subsystems. The output, the maintainability demonstration report, includes MTTR's and maintenance descriptions (ref. 4).

[3]To help the reader distinguish between the various aspects of maintainability evaluation, the following is useful. The three stages to the overall evaluation process are (1) engineering design analysis, (2) maintainability analysis, and (3) the maintainability demonstration. Engineering design analysis includes the initial trade studies and evaluation to determine the optimum ORU design configuration. Also, identified are safety hazards, reaction time constraints for critical maintenance, and an evaluation of diagnostic alternatives. Maintainability analysis includes an expanded detailed analysis of the final design to determine all maintainability system parameters. The maintainability demonstration then specifies tests to verify the data collected during the maintainability analysis.

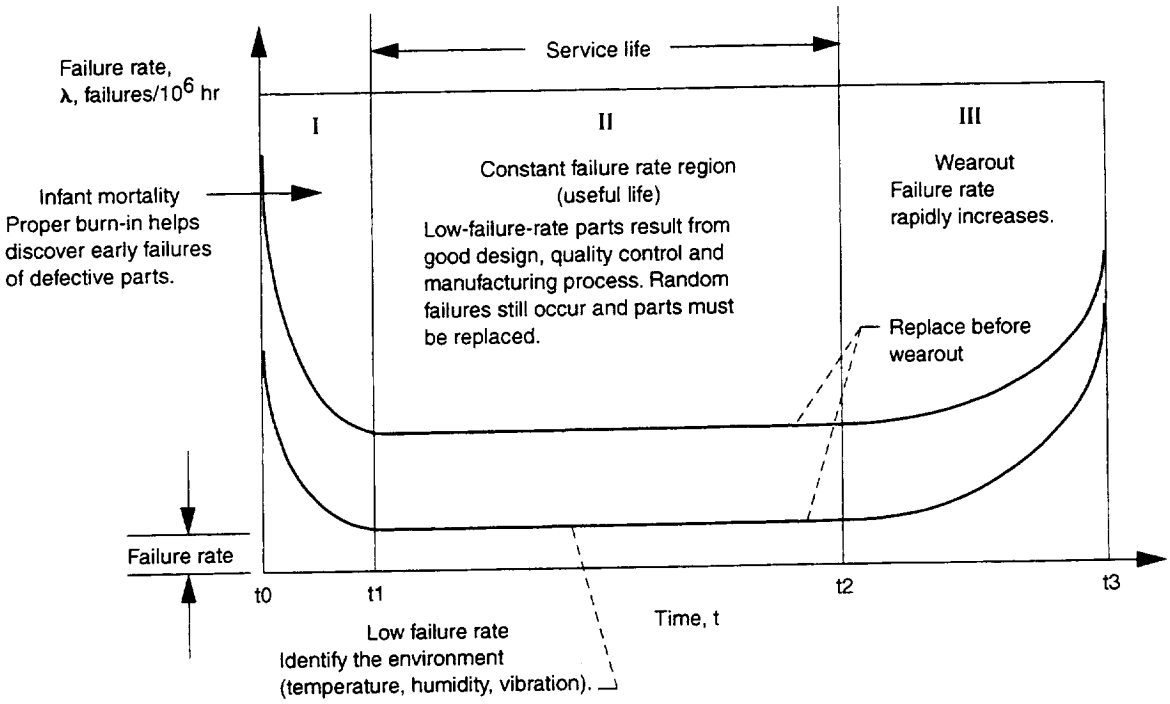


Figure 11–12.—Maintenance of limited-life items.

Maintainability Analysis Mathematics

As previously stated, the goal of system performance is to have the system available when it is need. As figure 11–11 shows, the failure rate, the mean time to repair, the time to acquire spares, and operational constraints all affect availability.

Availability requirements can be met with an extremely reliable system, one that is easy to repair and has an adequate supply of spare parts, or a combination of both. System use and mission profile also affect system availability requirements. The following list gives examples of continuous and intermittent mission requirements (ref. 5).

Is continuous operation required as for a critical life support system on a space station or an air traffic control system? If so, the reliability has to be very high and/or backup systems may be needed:

- Continuous operation
  - ° Spacecraft (LEO)
  - ° Space station
  - ° Air traffic control system
- Intermittent operation (on demand)
  - ° Emergency vehicle
  - ° Research fighter
  - ° Shipboard gattling gun
- Intermittent operation (scheduled)
  - ° Space experiment
  - ° CAT scan or MRI equipment in hospital
  - ° Space Shuttle main engines

An intermittent operation requirement is different. If availability is on demand, the built-in-test/built-in-test-equipment (BIT/BITE) and preventive maintenance functions have to be perfected and evaluated (through accumulating many hours on similar units). However, downtime for preventive maintenance has to be accounted for with spare systems. If there is scheduled intermittent operation, critical components can be replaced or continuously monitored (ref. 6).

For the mathematical analysis that follows, we will assume that we have a system that requires continuous operation except for scheduled preventive maintenance, that a temporary backup system exists, or that the system can be down for short periods. Once the system is put into operation, it might experience periods when not all features are operating but the failures can be tolerated until the next scheduled preventive maintenance (e.g., failure of a monitoring sensor or a BIT/BITE function).

Maintenance includes (1) corrective maintenance, the replacement of failed components or ORU's and LRU's; (2) preventive maintenance,[4] scheduled maintenance identified in the design phase as solution, alignment, calibration, or replacement of wear items such as clutches, seals, or belts: (3) replacement of life-limited items such as those illustrated in figure 11–12. Distinctions must be made between the availability calculated from the MTBF that is only valid in region II and the availability once a component enters its wearout region. Here the failure rate may increase exponentially, and it is more

[4]Preventive maintenance can also include software. Fixing corrupted tables, updating data bases, and loading revisions of software are an important part of scheduled maintenance.

difficult to predict. The generally accepted practice is to replace life-limited items before they enter their wearout period. If the mission life extends into region III (wearout), the part is a life-limited component and will be replaced before the beginning of the wearout stage at time $t2$. If the mission life is somewhere in region II, the component will only be replaced if it fails randomly. No scheduled replacement time will be made.

Availability can be calculated as the ratio of operating time to total time, where the denominator, total time, can be divided into operation time (uptime) and downtime. System availability depends on any factor that contributes to downtime. Underpinning system availability, then, are the reliability and maintainability of the system design; however, support factors, particularly logistics delay time, also play a critical role especially when a long supply line exists (such as with the International Space Station (ISS)). Assuming these factors remain the same, the following availability figures of merit can be calculated:

$$\text{Inherent availability} = \frac{\text{MTBF}}{\text{MTBF} + \text{MTTR}}$$

where MTBF is the mean time between failures and MTTR is the mean time to repair. Inherent availability considers only maintenance of failed units.

$$\text{Achieved availability} = \frac{\text{MTTMA}}{\text{MTTMA} + \text{MMT}}$$

where MTTMA is the mean time to a maintenance action (corrective, preventive, and replacement of limited-life items) and MMT is the mean (active) maintenance time (corrective, preventive, and replacement of limited-life items). Achieved availability includes inherent availability plus consideration for time spent for preventive maintenance and maintenance of life-limited items.

$$\text{Operational availability} = \frac{\text{MTTMA}}{\text{MTTMA} + \text{MMT} + \text{MLDT} + \text{MADT}}$$

where MLDT is the mean logistics delay time (includes downtime due to waiting time for spares or waiting for equipment or supplies). Maintenance downtime is the time spent waiting for a spare part to become available or time waiting for test equipment, transportation or a facility area to perform maintenance. For this discussion, it does not include local delivery such as going to a local storage location and returning to the work sight and returning the used part to a location for transport to a repair facility. MADT is the mean administrative delay time and includes downtime due to administrative delays, waiting for maintenance personnel, time when maintenance is delayed due to personnel being assigned elsewhere, filling out forms and signing out the part. Operational availability includes achieved availability plus consideration for all delay times.

Availability measures can also be calculated for a point in time or for an average over a period of time. Availability can

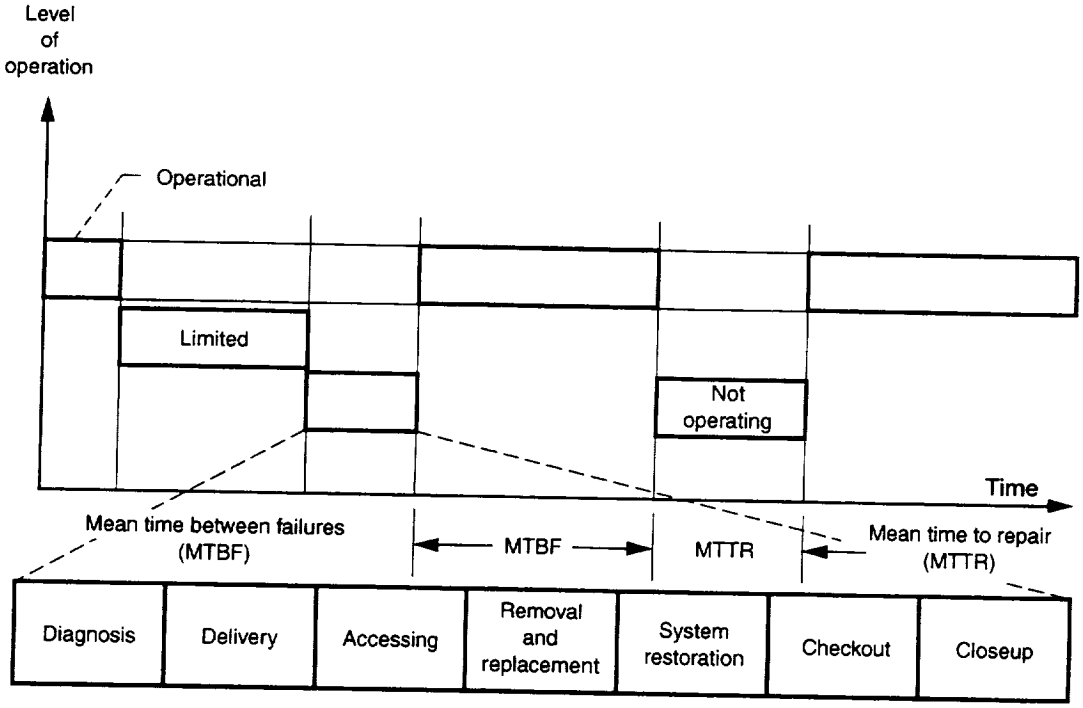


Figure 11–13.—Maintainability during system operation.

also be evaluated for a degraded system. For the remainder of our discussion, we will assume average availability and maintainability factors.

Other important factors in calculating availability include (1) maximum allowable time to restore, (2) proportions of faults and percentage of time detected as a function of failure mode, (3) maximum false alarm rate for built-in test equipment, and (4) maximum allowable crew time for maintenance activities.

We also want to look in detail at an individual corrective maintenance action. A number of elements make up a maintenance action and once they are combined, other factors must be considered before the overall impact on crew hours, maintenance hours, and other maintenance parameters are determined (fig. 11–13). These elements are (ref. 7)

(1) Maintainability prediction using the most effective methods available emphasizes an estimation of the time to restore at the ORU/LRU level. For a failed unit, the time to restore is the total corrective maintenance time T in minutes for each ORU:

$$T = \mathrm{DI} + \mathrm{DL} + \mathrm{GA} + \mathrm{RR} + \mathrm{SR} + \mathrm{CK} + \mathrm{CU}$$

where

DI diagnostic time to detect and isolate a fault to the ORU level, min
DL local delivery of spare ORU/LRU as opposed to shipping in from a remote location, min
GA time required to gain access to the failed ORU, min
RR time required to remove and replace the defective ORU, min
SR time required to restore system (including alignment, checkout, and calibration), min
CK time required to complete system checkout, min
CU time required to close up system, min

(2) The mean time to repair (MTTR) the ORU (on-orbit) follows. For this exercise, assume a crew size of one for all repair operations:

$$\mathrm{MTTR_{ORU}} = \frac{(T \times Z)}{60}$$

where MTTR is in hours and Z is the conversion factor for 1 to 10^{-6} g.

(3) The Mean time to a maintenance action (MTTMA) based on a yearly average is

$$\mathrm{MTTMA} = \frac{\left(\mathrm{MMHY}_p + \mathrm{MMHY}_p + \mathrm{MMHY}_l\right)}{8640}$$

where MMHY is the preventive maintenance hours per year, the subscripts p and l denote preventive and life-limited replacement, respectively, and 8640 is the number of hours in one year.

(4) The maintenance hours per year (MMHY) for corrective (c), preventive (p) and life-limited replacement (l) follow:

$$\mathrm{MMHY}_c = \mathrm{DC} \times \mathrm{MTTR_{ORU}} \times K \times \left(\frac{8640}{\mathrm{MTBF}}\right)$$

$$\mathrm{MMHY}_p = \mathrm{MMP} \times \mathrm{F(P)}$$

$$\mathrm{MMHY}_l = \frac{\mathrm{MTTR_{ORU}}}{T_l}$$

where

DC duty cycle of ORU, percent
MTBF mean time between failures, hr
MTBM mean time between maintenance, hr
MMP mean hours to perform preventive task, hr
F(P) preventive task frequency per year
K MTBF to MTBM conversion factor
T_l life limit for ORU, hr

(5) Maximum corrective maintenance time M_{max} is the +90 percent time for a normal distribution. It is assumed that since this is a manual operation and not the subject of wearout, the normal distribution will apply:

$$M_{max} = \mathrm{MTTR_{ORU}} + (1.61 \times \sigma)$$

where σ is the standard deviation of the repair time.

Plots of typical inherent availability are presented in figure 11–14 as a function of MTTR and MTBF. Here, solving the expression

$$\text{Inherent availability} = \frac{\mathrm{MTBF}}{\mathrm{MTBF} + \mathrm{MTTR}}$$

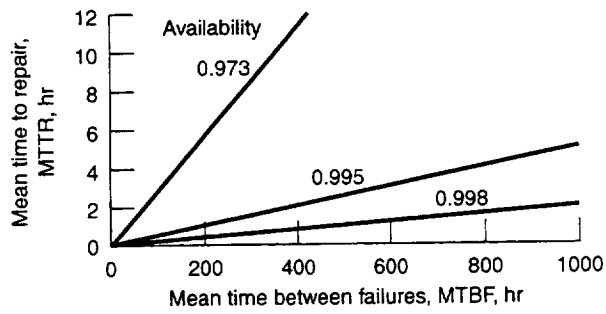


Figure 11–14.—Relationship of MTTR and MTBF to availability.

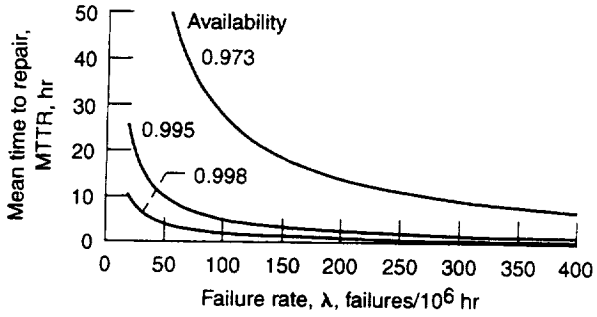


Figure 11–15.—Relationship of MTTR and failure rate to availability.

gives

$$\text{MTTR} = (1 - \text{inherent availability}) \times \text{MTBF}$$

Figure 11–15 shows MTTR as a function of failure rate (assuming an exponential rate). For an exponential distribution, the failure rate λ is 1/MTBF. Substituting this into the above expression for inherent availability and solving for MTTR yields the results shown.

Additional Considerations

As previously mentioned, to speed the system back into operation, it is typically divided into units (ORU's/LRU's) that can be easily replaced, either on-orbit or on the flight line. This means that maintenance on-orbit (or on the flight line) will usually only replace these items. The following are important questions we need to ask for our maintainability analysis (ref. 8):

- How much downtime is acceptable?
- What will be replaced on the flight line (what should be designated an LRU or an ORU)?
- How will a failure be diagnosed and isolated to an ORU/LRU, a BIT/BITE, manual processes, software, or a combination?
- Will the failed units be scrapped or repaired?
- If repaired, what should be repaired for each type of failure? Where should it be repaired (depot, lab, factory) and by what skill level?
- What preventive maintenance needs to be performed?
- What kind of maintenance tests need to be performed?
- Can all components be inspected for structural defects?
- How will structural defects be detected and tracked?
- Have acceptable damage limits been specified?
- Are safety-related components easy to replace?
- Are there safety issues that occur during maintenance?
- How is corrosion controlled?
- Are limited-life items tracked for maintenance?

A combination of built-in testing and diagnostic procedures (with the needed tools and instruments) must be available to diagnose a fault or failure to at least one ORU/LRU level. If it cannot be determined with that fidelity, the wrong item might be replaced. The built-in test procedures begin with specific questions:

- Do we know what is going to fail?
  - ° Do maintenance records allow preventive maintenance where critical items are replaced at a known percentage of life?
  - ° Do smart diagnostic features sense impending failures?

- Do we know what has failed?
  - ° Does built-in test equipment quickly diagnose the problems?
  - ° Does readily available external test equipment quickly diagnose the problems?

- Do we know how we are going to handle each failure?
  - ° Has a repair analysis been performed on all likely failures?
  - ° How will each failure be diagnosed and repaired?
  - ° Has the failure modes and effects analysis (FMEA) been evaluated for failures and corrective actions?

The questions that remain are Can all plausible and probable failure modes (based on the FMEA/CIL) be diagnosed with BIT/BITE? and Can the necessary diagnostic procedures be carried out by a crew member or technician on the flight line? The answers to these questions determine the design concept for maintainability. The aim of this analysis is to reduce downtime.

Requirements and Maintainability Guidelines for ORU's

Other requirements to evaluate ORU's/LRU's follow.

(1) On-orbit replacements of ORU's should not require calibrations, alignments, or adjustments. Replacements of like items in ORU's should be made without adjustments or alignments (this will minimize maintenance time).

(2) Items that have different functional properties should be identifiable and distinguishable and should not be physically interchangeable. Provisions should be incorporated to preclude installation of the wrong (but physically similar) cards, components, cables, or ORU's with different internal components or engineering, revision number, and so forth. Reprogramming, changing firmware, and changing internal switch settings may be allowed with special procedures and safeguards.

(3) All replaceable items should be designed so that it will be physically impossible to insert them incorrectly. This is a basic maintainability and safety requirement.

Additional maintainability considerations that should be incorporated in the design are

(1) Any ORU, shop replaceable unit (SRU)[5], their subcomponents, or cards that are physically identical should be interchangeable (excluding cables and connectors). Identical hardware (e.g., a signal conditioning card) shall not be made unique. Different software and switch settings do not affect identity. The ability to replace ORU's with an identical unit from an inactive rack will improve availability.

(2) Standardization should be incorporated to the maximum extent through the design. In the interest of developing an efficient supply support capability and in attaining the availability goals, the number of different types of spares should be held to a minimum.

(3) The ORU should be designed from standard off-the-shelf components and parts.

(4) The same items and/or parts should be used in similar ORU's with similar applications (e.g., boards, fasteners, switches, and other human interface items; fuses, cable color designations, and connectors (except to avoid improper hook-ups)).

(5) Equipment control panel positions and layouts (from panel to panel) should be the same or similar when a number of panels are incorporated and provide comparable functions.

Related Techniques and Disciplines

Some disciplines that relate to basic maintainability analysis are now discussed (ref. 9).

Supportability.—This is a global term that covers all maintenance and logistics activities. The unit can be supported if it can be maintained and if spare parts can be delivered to it.

Reliability-centered maintenance (RCM).—This maintenance process is based on the identification of safety-critical failure modes and deterioration mechanisms through engineering analyses and experience. Thus, the consequences of the failure can be determined on the basis of severity level so that maintenance tasks can be allocated according to severity level and risk. The RCM logic process considers maintenance task relative to (1) hard-time replacements in which degradation because of age or usage is prevented and maintenance is at predetermined intervals; (2) on-condition maintenance in which degradation is detected by periodic inspections and (3) conditional maintenance in which degradation prior to failure is detected by instrumentation and/or measurements.

Integrated logistics support.—This includes the distribution, maintenance, and support functions for systems and products: (1) maintenance, (2) supportability, (3) test and support equipment, (4) personnel training, (5) operations facilities, (6) data (manuals), (7) computer resources (for maintenance of equipment and software), and (8) disposal. Personnel considerations involve analyzing what level of expertise is needed for each level of maintenance (on the flight line, in a depot (intermediate repair facility), or in the factory) to effectively perform the repairs.

[5]A part or component that is designed and/or designated to be replaced in a depot or at the manufacturer. For instance, it may be highly modular but its failure cannot be easily detected on-orbit or on the flight line.

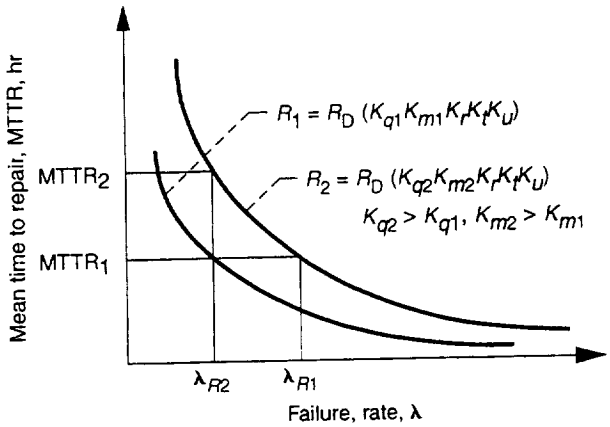


Figure 11–16.—Effect of quality on maintainability.

Maintainability, quality, and reliability.—Figure 11–16 shows the relationship between the three. As quality and manufacturing techniques improve, reliability increases. Therefore, for the same availability, MTTR may increase and a higher availability may be attained. The reliability of the product is given by $R_{product}$ where the design stage reliability R_D is modified by various K factors. These denote probabilities that the design-stage reliability will not be degraded by any given factor. The K factors are external contributors to product failure:

$$R_{product} = R_D\left(K_q K_m K_r K_l K_u\right)$$

where

K_m	manufacturing, fabrication, assembly techniques
K_q	quality test methods and acceptance criteria
K_r	reliability fault control activities
K_l	logistics activities
K_u	user or customer activities

Manufacturing processes or assembly techniques that are not statistically controlled can greatly affect reliability. Special cause variation, change in raw materials, or lack of adherence to manufacturing procedures can dramatically reduce product reliability. Poor test methods may allow substandard components to be used in a product that would fail final test screenings and enter the operating population. Poor packing, shipping practices, storage, and so on will raise the failure rate. The user or customer may abuse the product by using it for what it was not intended or using it in a new unspecified environment. All these problems require that the system be maintainable during operation.

TABLE 11–2.—MAINTAINABILITY FIGURES OF MERIT

Weight of orbital replacement and line replacement units, kg
Volume, m^3
Power requirement, W
Definition of partial operation
Mean time between failures, hr
Life and wearout, hr
Mean time to repair, hr/repair
Failure modes and effects analysis, hr
Manifest time, hr
Operation time, hr
Operation period, hr
Spare location, sec
Maintenance cost, dollars
Repair cost, dollars
Transportation, dollars
Built-in test capabilities
Tools required
Preventive maintenance
Supportability
Availability

Maintainability Problems

The maintainability, reliability, and cost data items in table 11–2 represent the information required to perform a maintainability analysis. We will consider how these items interact and how maintainability trades can be made. First, consider examples 1 and 2 (for the basic formulas, refer to the section Maintainability Analysis Mathematics).

Example 1

Five pressure transducers (model c–4) were tested and failed after an average of 2257 hr (time for first failure t_f). Time studies have shown that it takes 5.5 hr to diagnose, remove, replace, and check out a unit (MTTR). Assuming continuous use and an exponential failure rate, what is the failure rate λ, the reliability for a mission time t_m of 50 hr, the MTBF, and the availability? First, determine the failure rate:

$$\lambda = \left(\frac{\text{Failure}}{\text{hr}}\right) \text{ or } \left(\frac{\text{Failures}}{10^6\,\text{hr}}\right)$$

$$= \frac{1}{\text{MTBF}} = \frac{1}{2257}$$

$$= 0.000443 \text{ failures / hr or } 443 \text{ failures / } 10^6\,\text{hr}$$

Determine the reliability:

$$\text{Reliability} = \exp(-\lambda t_m)$$

$$= \exp(-0.000443 \times 50) = 0.9780$$

Determine the availability:

$$\text{Availability} = \frac{\text{MTBF}}{(\text{MTBF} + \text{MTTR})}$$

$$= \frac{2257}{(2257 + 5.5)} = 0.9976$$

Example 2

Five RTD temperature sensors (model RTD–A–7) were tested and failed after an average of 4026 hr (time for first failure t_f). Time studies have shown that it takes 52 hr to diagnose, remove, order, receive, replace, and check out a unit (MTTR). Assuming continuous use and an exponential failure rate, what is the failure rate λ, the reliability for a mission time t_m of 50 hr, the MTBF, and the availability. Determine the failure rate:

$$\lambda = \frac{1}{\text{MTBF}} = \frac{1}{4026}$$

$$= 0.000248 \text{ failures / hr}$$

The reliability is

$$\text{Reliability} = \exp(-\lambda t)$$

$$= \exp(-0.000248 \times 50) = 0.9876$$

The availability is

$$\text{Availability} = \frac{\text{MTBF}}{(\text{MTBF} + \text{MTTR})}$$

$$= \frac{4026}{(4026 + 52)} = 0.9872$$

Problem Solving Strategy

One way to assess tradeoffs is to first evaluate conformance to minimum maintainability requirements and then to calculate the effects that the alternatives have on costs by following these steps: (1) determine screens, minimum or maximum acceptable values for a system or component; (2) determine

TABLE 11-3.—SYSTEM AND MISSION PARAMETERS AND COSTS

Requirement	Parameter	Questions in determination of screen
System	(1) Availability minimum, 0.990	Based on the MTTR and MTBF for each unit. Is the availability greater than or equal to the requirement (0.990)?
	(2) Mean time to repair (MTTR) maximum, 5.0 hr	What is the maximum repair time that can be allowed? How long can the system be down?
	(3) Mean time between failures (MTBF) minimum, 300 hr	What is the minimum reliability goal of the system?
	(4) Logistics delay time (LDT) + administrative delay time (ADT), 0.3 hr	What is the maximum LDT allowable? For a single repair action, how long does it take to deliver a replacement part from the warehouse or factory (for the total mission, turnaround time for repair of boards also needs to be considered)? What is the ADT? How long will it take to process an order for spares and how long will it take to do other paperwork? (ADT may not affect system availability but it will affect total crew maintenance time used to repair the system.)
Mission	(5) Total mission time, 520 wk	What is the total time that the unit will be in the system and available for operation?
	(6) System operating time per week, 4 hr	How many hours per week does the unit operate and in what modes (operational, standby, partial, off)?
	(7) Maximum resource allocation for maintenance, 0.1 hr/wk	Are crews available for maintenance and operation of the unit? Is the MTTR reasonable so that the crew will have time to do maintenance?
	(8) Operational requirement, 6 hr/wk	Are there limits on how long an item can take to be repaired? (Often, if a system is difficult to repair, it may be neglected in favor of a more easily maintained system.)
	(9) Total mission time, 87 360 hr	What are the total clock hours the mission is to last (irrespective of whether the system being considered is operating)?
	(10) Total system operation, 2080.0 hr/yr	What are the total hours per year the system or board being considered is operating (6 times/wk × 52)?
Cost	(11) Board repair, $7000 dollars	What is the cost to repair a failed board?
	(12) Transportation of board, $4500	What is the cost to transport a spare board to the site of field repairs. (If the site is remote or on-orbit, the cost may be considerable.)
	(13) Maintenance on-orbit, $500/hr	What are the allocated costs for crew maintenance time on-site or on-orbit? (The cost of crew maintenance time may be considerable and significantly affect the overall trade study costs.)

which tradeoffs meet these screens; (3) of the systems that pass, calculate costs (cost of spare, cost to ship spare, cost to install spare); (4) determine the lowest cost system; and (5) examine the results for reasonableness.

This discussion presents a more detailed analysis of how tradeoffs (at the board or component level) involving maintenance and reliability may be made. This is a more complex example for which we want to determine the lowest cost solution to a maintainability problem with fixed requirements by following the above procedures.

Determining screening requirements.—The reliability and maintainability screening requirements must be determined. Here there is a maximum MTTR[6] related to maintenance crew availability, a minimum MTBF due to mission restrictions, and a specified availability requirement needed to complete the mission. The operation of the system is intermittent. A detailed list of these requirements and costs is presented in table 11–3, which gives quantitative system data needed to evaluate the model.

The availability, maintainability, and reliability screens in table 11–3 are also portrayed graphically in figure 11–17 where availability is shown as a function of F(MTBF and MTTR). The solution space described by the system and mission requirements is bounded by the 0.990 availability line, the MTBF minimum of 300 hr, and the MTTR maximum of 5 hr. Note also that in this figure the constant availability lines are generated with MTBF's and MTTR's that represent average values: MTTR and MTBF are usually considered distributed variables with an exponential or normal distribution.

Having addressed the basic requirements imposed on the system and the costs associated with a maintenance action, we will now evaluate individual boards that are being considered for a black box in the system.

First, some additional assumptions must be made. (1) Only one spare board is required and it is readily accessible on-orbit or on the flight line; (2) all spares cost the same; (3) there is no finance (carrying) cost; and (4) repair costs for each alternative board are the same.[7]

[6]Strictly speaking, we do not have a "maximum MTTR" since MTTR and also MTBF do not have distributions but are derived from a distribution. This notation is kept because we are looking at a number of MTTR's for various alternative boards and the like.

[7]A problem arises when the boards are stored on the ground or in a warehouse (for LRU's) when there are long logistic delay times. If systems were in remote sites or on-orbit (with no local storage of spares) with only three or four deliveries of spares per year (as with the space shuttle), there might be considerable periods of downtime.

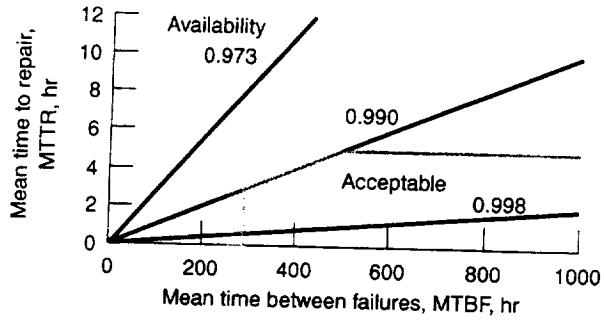


Figure 11–17.—Problem solution area on availability plot.

Determining tradeoffs that meet screens.—Data required to evaluate each potential electronic board for a particular function in the system are given in table 11–4. Board option 1 was discarded for failure to meet functional design parameters. Each remaining board (first column) was evaluated for expected MTBF or reliability (with a parts count according to MIL–HDBK–217 or possibly via testing), estimated cost to purchase the board, estimated time to repair the board (based on ease of diagnosis, built-in test circuitry or software), and estimated LDT (based on the supplier turnaround history) and administrative delay time (ADT).

The next step is to calculate the data required in table 11–5 to see if the maintainability and reliability requirements have been met.[8]

$$\text{Number of maintenance actions} = \frac{\text{Total mission time / wk} \times \text{system operating time / wk}}{\text{Mean time between failures}}$$

$$\text{Availability} = \frac{\text{MTBF}}{\text{MTBF} + \text{MTTR}}$$

$$\text{Total maintenance time} = \text{Number of maintenance actions / mission} \times (\text{MTTR} + \text{LDT} + \text{ADT})$$

$$\text{Total maintenance time (hr / wk)} = \frac{\text{Total maintenance time (hr)}}{\text{Total mission time (wk)}}$$

Note that the maintainability screens are independent and may not necessarily relate to these formulas (e.g., irrespective of the required availability and minimum MTBF, there may be a maximum maintenance time allowed). After evaluating the results, we found that options 2, 3, 4, and 5 failed the minimum MTBF and availability screens; option 4a failed the maximum MTTR screen; the remaining options 2a and 4a will be evaluated to determine which has the lower cost.

[8]The formula for column F is F = (5)(6)/B where (5) and (6) refer to items in table 11–3 and B refers to column B in table 11–4.

TABLE 11–4.—BOARD TRADEOFF OPTION DATA
[Logistics and administrative delay times, LDT + ADT, 0.3 hr.]

A Board option	B Mean time between failures, MTBF, hr	C Cost, dollars	D Mean time to repair, MTTR, hr
[a]1	—	—	—
2	195	74 100	3.7
2a	662	182 900	3.8
3	191	77 600	3.5
3a	583	130 800	3.7
4	199	76 600	3.3
4a	828	188 257	6.8
5	62	45 400	3.4

[a]Discarded for failing to meet functional design parameters.

Determining the cost of acceptable systems.—Of the systems that pass, calculate the costs of purchasing the spare and the board, repairing the failed unit, and shipping and installing the spare. These figures are shown in table 11–6.

The total mission board repair cost is equal to the cost of repairing each board (at a depot or the factory) times the total number of maintenance actions. The cost of the board repair is $7000/repair, which would theoretically be reduced by the number of spares purchased. The repair cost and turnaround time should be part of the supplier's bid for the board.

The total mission board shipping cost is equal to the cost of transporting the board times the total number of maintenance actions. The cost of shipping the board is $4500 per shipment.

The total mission board maintenance cost reflects costs to change out the board on-orbit or on the flight line. The cost to replace the board (on-orbit or on the flight line) is $500 per hr, which assumes that the board is also an ORU or an LRU. It is equal to the total number of maintenance actions times (MTTR + LDT + ADT).

The total mission board repair cost is equal to the total cost of repair, shipping, and maintenance.

The total mission board cost is equal to the total mission board repair cost plus the cost of the board and one spare board. The cost of manufacturing the board was already given in column C of table 11–4. For the present example, we will assume that we need to purchase one board and one spare board.[9]

[9]One must also consider the quantity of spares needed to have a replacement board available at all times. This is a function of the desired probability of an available spare, the time to ship the board out for repairs, to repair it, to recertify it, and to return it to a storage location. A detailed discussion of the mathematics of this evaluation is beyond the scope of this paper. Additional costs will also be incurred with parts storage.

TABLE 11–5.—MAINTAINABILITY FIGURES OF MERIT

A Board option	F Number of maintenance actions per mission	G Availability, percent,	H Total maintenance time, hr	I Total maintenance time, hr/wk
	F = (5)(6)/B	G = B/(B + D)	H = F(D + E[a])	I = H/(5)
2	10.7	0.980	42.7	0.08
2a	3.1	.994	12.9	.02
3	10.9	.980	41.4	.08
3a	3.6	.993	14.3	.03
4	10.5	.982	37.6	.07
4a	2.5	.992	17.8	.03
5	33.3	.944	123.3	.24

[a]E = logistics and administrative delay time of 0.3.

TABLE 11–6.—TOTAL MISSION COST CALCULATIONS

A Board option	J Board repair, dollars/mission	K Board shipping, dollars/mission	L Board maintenance, dollars/mission	M Repair, dollars	N Board and spare, dollars
	J = (11)F	K = (12)F	L = (13)H	M = J + K + L	N = (2C) + M
2	74 683	48 011	1861	124 554	272 754
2a	22 005	14 146	1903	38 055	403 855
3	76 216	48 996	1761	126 974	282 174
3a	24 965	16 049	1854	42 867	304 467
4	73 151	47 026	1660	121 837	275 037
4a	17 578	11 300	3403	32 280	408 794
5	233 206	149 918	1733	384 857	475 657

Determining the lowest cost system.—The solution is to pick the lowest-cost board that passed the screens. Options 2 to 4a and 5 have already failed screens. Of the remaining candidates 2a and 3a, 3a has the lowest cost.

Examining the results for reasonableness.—As always, factors other than costs must be included in the analysis. Human factors, hierarchy of repairs, ease of problem diagnosis, ability to isolate faults, ability to test the unit, manufacturer's process controls and experience, and the ability of the manufacturer to provide long-term support for the unit are some additional considerations.

Recommended Techniques

Current and future NASA programs face the challenge of achieving a high degree of mission success with a minimum degree of technical risk. Although technical risk has several elements, such as safety, reliability, and performance, a proven track record of overall system effectiveness ultimately will be the NASA benchmark that will foster the accomplishment of mission objectives within cost and schedule expectations without compromising safety or program risk. A key characteristic of system effectiveness is the implementation of appropriate levels of maintainability through the program life cycle.

Maintainability is a process for assuring the ease with which a system can be restored to operation following a failure. It is an essential consideration for any program requiring ground and/or on-orbit maintenance. The Office of Safety and Mission Assurance (OSMA) has undertaken a continuous improvement initiative to develop a technical roadmap that will provide a path to achieving the desired degree of maintainability while realizing cost and schedule benefits. Although early life-cycle costs are a characteristic of any assurance program, operational cost savings and improved system availability almost always result from a properly administered maintainability assurance program. Experience in NASA programs has demonstrated the value of an effective maintainability program initiated early in the program life cycle.

Technical Memorandum 4628 entitled "Recommended Techniques for Effective Maintainability" provides guidance for achieving continuous improvement of the life-cycle development process within NASA, having been developed from the experiences of NASA, the Department of Defense, and industry. The degree to which these proven techniques should be imposed resides with the project or program and will require an objective evaluation of the applicability of each technique. However, each applicable suggestion not implemented may represent an increase in program risk. Also, the information presented is consistent with OSMA policy, which advocates an integrated product team (IPT) approach for NASA systems acquisition. Therefore, this memorandum should be used to communicate technical knowledge that will promote proven maintainability design and implementation methods resulting

in the highest possible degree of mission success while balancing cost effectiveness and programmatic risk. The recommend techniques can be found online at http://www.hq.nasa.gov/office/codeq/doc.pdf.

Conclusion

The benefit of a system maintainability program is mission success, the goal of every NASA System Reliability and Quality Assurance (SR&QA) office.[10,11] A well-planned maintainability program gives greater availability at lower costs. A design with easily maintained (and assembled) modules results. Considering maintenance prevents the inclination to use lower-cost components at the expense of reliability unless maintainability tradeoffs justify them. Finally, maintainability analysis forces considerations of potential obsolescence and the need for upgrades[12] and reduces overall maintenance hours and the total cost of ownership.

References

11-1. Maintainability Prediction. MIL–HDBK–472, Jan. 1984.

11-2. Electronic Reliability Design Handbook. MIL–HDBK–338, June 1995.

11-3. Pillar, C.S.: Maintainability in Power Plate Design. Sixth Reliability Engineering Conference for Electric Power Industry, American Society for Quality Control, Milwaukee, 1979.

11-4. Fujiwara, H.: Logic Testing and Design for Testability. MIT Press, 1985.

11-5. Lala, P.K.: Fault Tolerant and Fault Testable Hardware Design. Prentice-Hall, 1985.

11-6. Engineering Design Handbook. Pamphlet AMCP 706–134, McGraw-Hill, 1970.

11-7. USAF R&M 2000 Process. Handbook SAF/AQ, U.S. Air Force, Washington, DC, 1987.

11-8. Testability Program for Systems and Equipment. MIL–HDBK–2165, July 1995.

11-9. Raheja, D.G.: Assurance Technologies: Principles and Practices, McGraw-Hill, 1991.

[10]NASA Glenn Research Center is designing a second-generation instrument to measure microgravity on the space station. The operating time for the instrument is expected to be 10 yr. Reliability analysis has shown low reliability for this mission even if we can get all the components to have an MTBF of 40 000 hr. Therefore, we are developing a maintenance program with an on-orbit repair time of 700 hr, which should give a suitable availability for the mission.

[11]NASA Glenn had an interesting experience with one of its space instruments. It was designed for a mission time of 18 hr and had a reliability greater than 0.90. It was suggested that we use the instrument on MIR for a 3000-hr mission. The reliability fell to 0.40 when this and other factors were considered. Maintainability was factored in with selected spare parts, software was added to perform built-in test (BIT) of the unit. The mission specialists were also trained to do repair work. The availability was returned to its previously acceptable level (with the previous level of reliability). The instrument has successfully collected data on MIR.

[12]For example, a ruggedized optical disk drive required maintenance after each flight on the space shuttle or after 450 hr of operation. This process took 4 wk, which was unacceptable to NASA when the system had to be placed on the Russian Space Station MIR. To correct the problem, the drives were replaced with another component that greatly reduced maintenance time.

Reliability Training[13]

1. Three thermostats were tested and failed after an average of 39 500 cycles. Time studies showed that diagnosis took an average 6.8 hr to remove, replace, and check out a thermostat. What is the MTBF of the unit for a mission time of 168 cycles?

A. 30 200 cycles B. 35 600 cycles C. 39 500 cycles

What is the failure rate?

A. 20.6×10^{-6} failure/hr B. 25.3×10^{-6} failure/hr C. 30.7×10^{-6} failure/hr

What is the reliability?

A. 0.976 B. 0.986 C. 0.996

What is the availability?

A. 0.979 B. 0.989 C. 0.999

2. Three air bearings were tested and failed after an average of 323 000 hr. It is estimated that it will take an average of 3200 hr to diagnose, remove, replace, and check out a bearing in low Earth orbit. What is the MTBF of a unit for a mission time of 80 000 hours?

A. 293 000 hr/failure B. 313 000 hr/failure C. 323 000 hr/failure

What is the failure rate?

A. 3.1×10^{-6} failure/hr B. 3.5×10^{-6} failure/hr C. 4.0×10^{-6} failure/hr

What is the reliability?

A. 0.68 B. 0.78 C. 0.88

What is the availability?

A. 0.79 B. 0.89 C. 0.99

[13]Answers are given at the end of this manual.

Appendix A
Reliability Information

The figures and tables in this appendix provide reference data to support chapters 2 to 6. For the most part these data are self-explanatory.

Figure A–1 contains operating failure rates for military standard parts. They relate to electronic, electromechanical, and some mechanical parts and are useful in making approximate reliability predictions as discussed in chapter 3. Their use, limitations, and validity are explained in chapter 4.

Figure A–2 provides failure rate information for making approximate reliability predictions for systems that use established-reliability parts, such as air- and ground-launched vehicles, airborne and critical ground support equipment, piloted aircraft, and orbiting satellites. The use of this figure is discussed in chapter 4.

Figure A–3 shows the relationship of operating application factor to nonoperating application factor. These data can be used to adjust failure rates for the mission condition. The use of this figure is also discussed in chapter 4.

Figure A–4 contains reliability curves for interpreting the results of attribute tests. They provide seven confidence levels, from 50 percent to 99 percent; and six test failure levels, from 0 to 5 failures. The use of these figures is discussed in chapter 5.

Table A–1 contains values of the negative exponential function e^{-x}, where $-x$ varies from 0 to –0.1999. The tabulated data make it easy to look up the reliability, where the product of failure rate l (or 1/MTBF) and operating time t are substituted for $-x$. The use of this table is discussed in chapter 3 and it is frequently referred to in chapters 4 to 6.

Table A–2 contains tolerance factors for calculating the results of mean-time-between-failure tests. It provides seven confidence levels, from 50 to 99 percent for 0 to 15 observed failures. The use of this table is explained in the table. Examples are discussed in chapter 6.

Tables A–3 to A–5 contain tabulated data for safety margins, probability, sample size, and test-demonstrated safety margins for tests to failure. They provide three confidence levels, from 90 to 99 percent, and sample sizes from 5 to 100. Values similar to these are presented on the safety margin side of the reliability slide rule; the slide rule provides six confidence levels and sample sizes from 5 to 80. The use of these tables and the slide rule is discussed in chapter 6.

More information on this subject can be found in references A–1 and A–2.

References

A–1. Reliability Modeling and Prediction. MIL–STD–756B (plus change notices), Aug. 31, 1982.

A–2. Reliability for the Engineer. Book Seven: Reliability Tables, Martin Marietta Corporation, 1965.

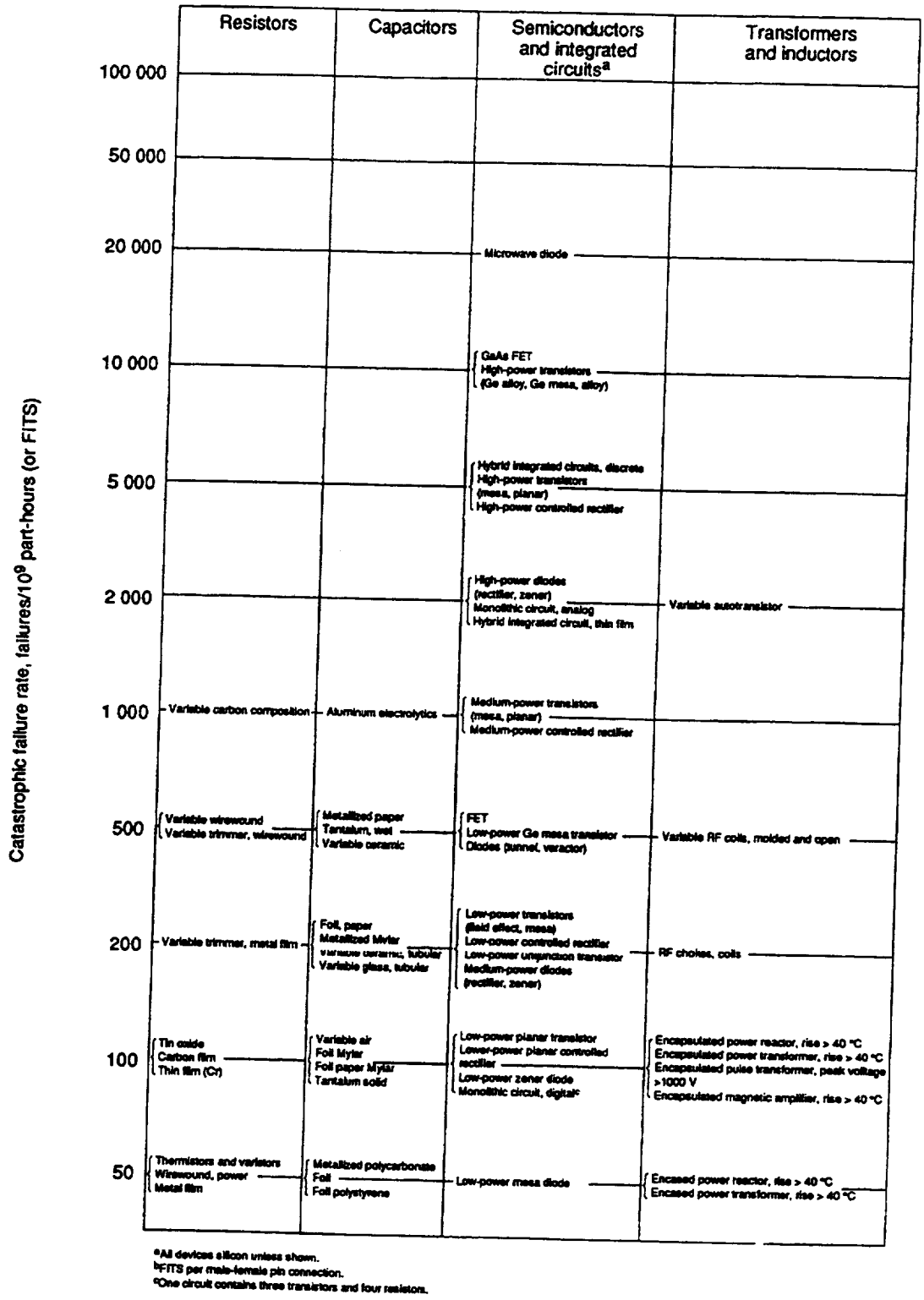


Figure A–1.—Military standard catastrophic failure rates for operating mode. (Failure rate for these parts in nonoperating mode can be estimated by dividing these values by the application factor shown in fig. A–3.)

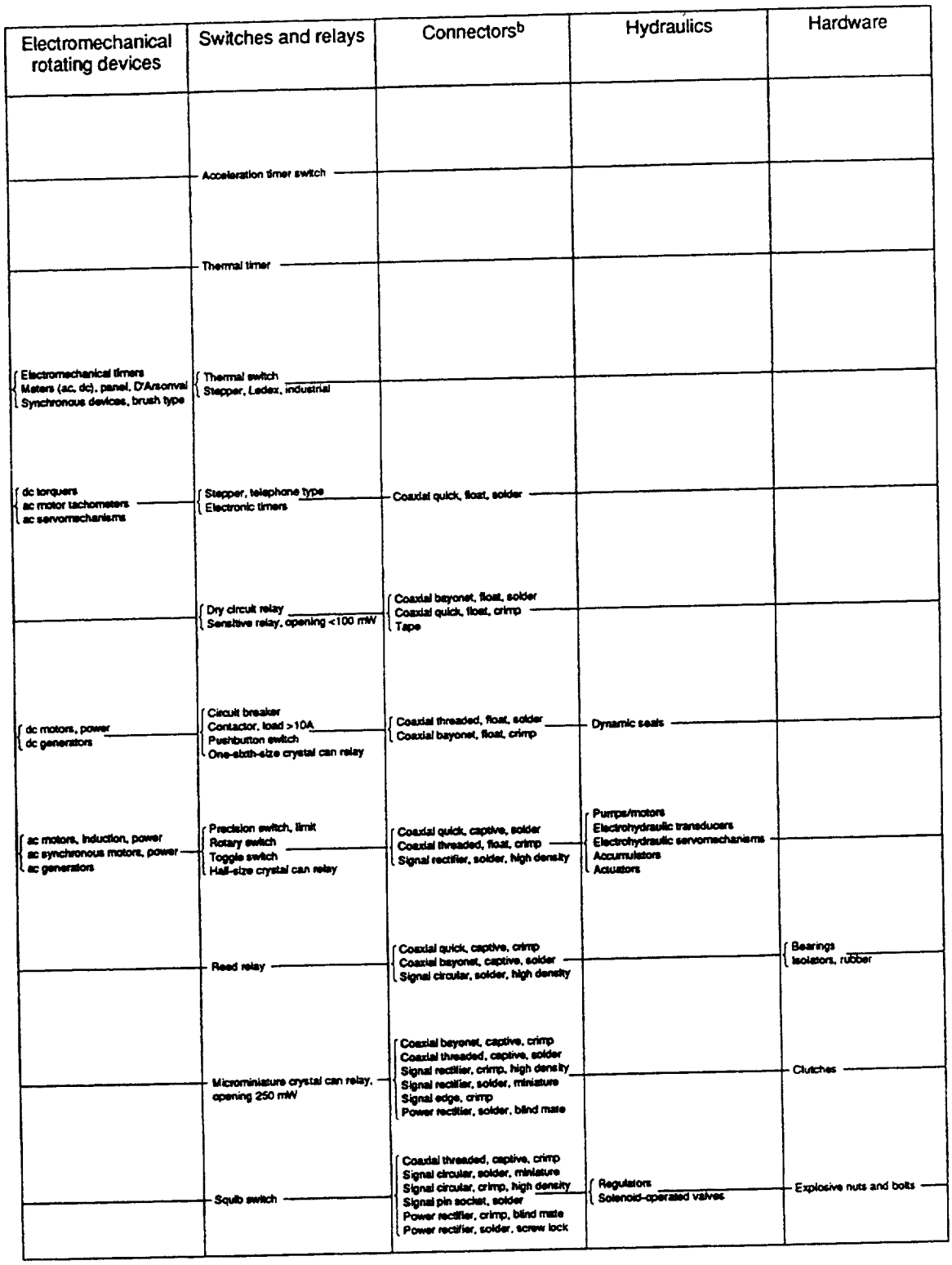


Figure A–1.—Continued.

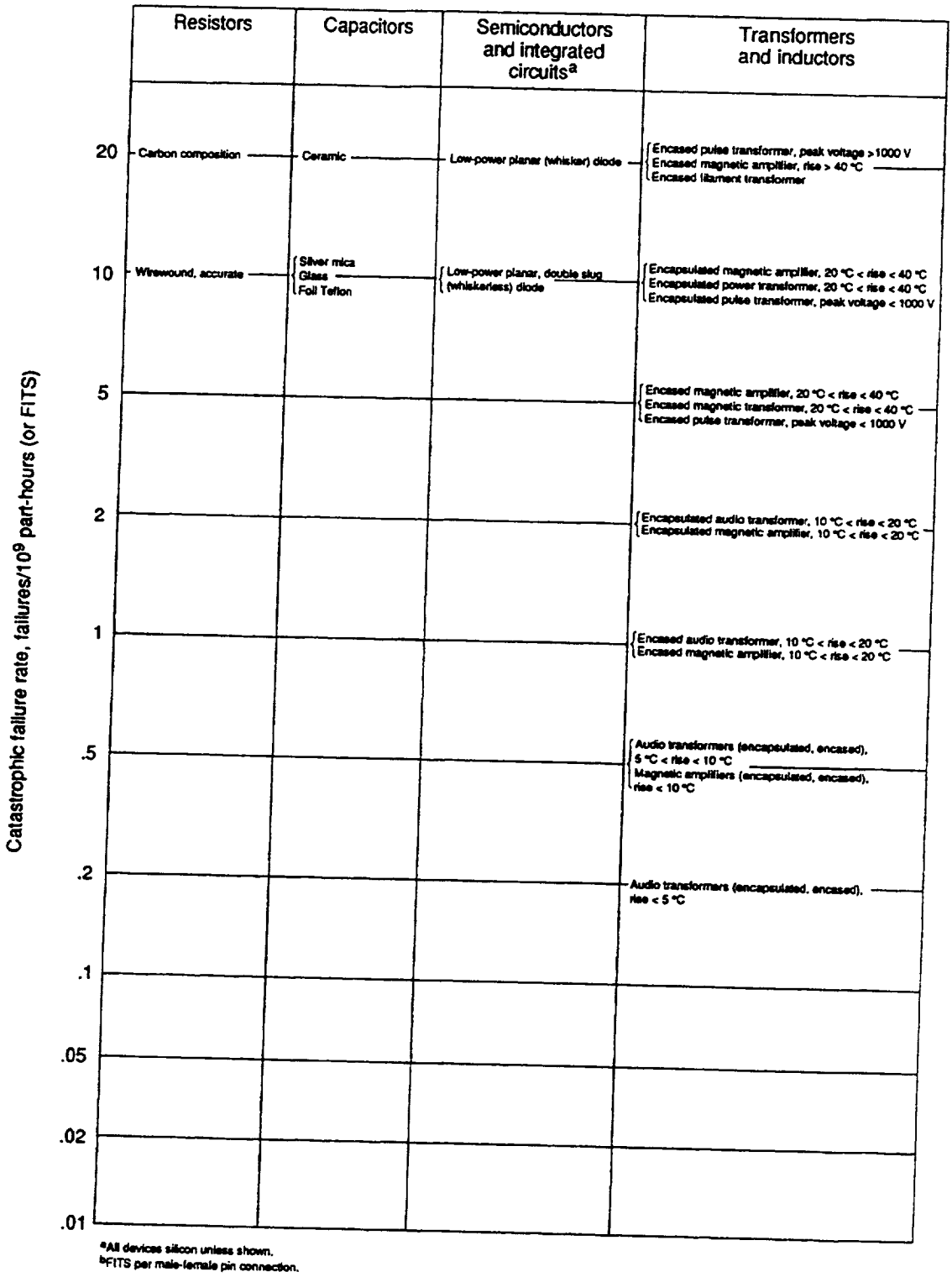


Figure A–1.—Continued.

Electromechanical rotating devices	Switches and relays	Connectors[b]	Hydraulics	Hardware
	Mercury-wetted relay	Signal rectifier, crimp, miniature Signal pin socket, crimp Power, bayonet, solder Power, threaded, solder	Self-operating valves	Gears
	Mercury switch	Signal circular, crimp, miniature Signal edge, solder Power rectifier, crimp, screw lock	Manual valves Fittings, tubing (flexible)	Strap assemblies Slings
		Power, bayonet, crimp Power, threaded, crimp	Check valves Relief valves Static seals	Containers, sealed
				O-ring pins O-ring fasteners Blind rivets Inserts Permanent fasteners
			Reservoirs Gages Fluids, lubricants Fittings, tubing (rigid) Filters	
		Connection, one: solder, weld, or wirewrap		Bolts and screws, structural Nuts Clamps and couplings
				Bolts and screws, nonstructural
				Solid rivets

Figure A–1.—Concluded.

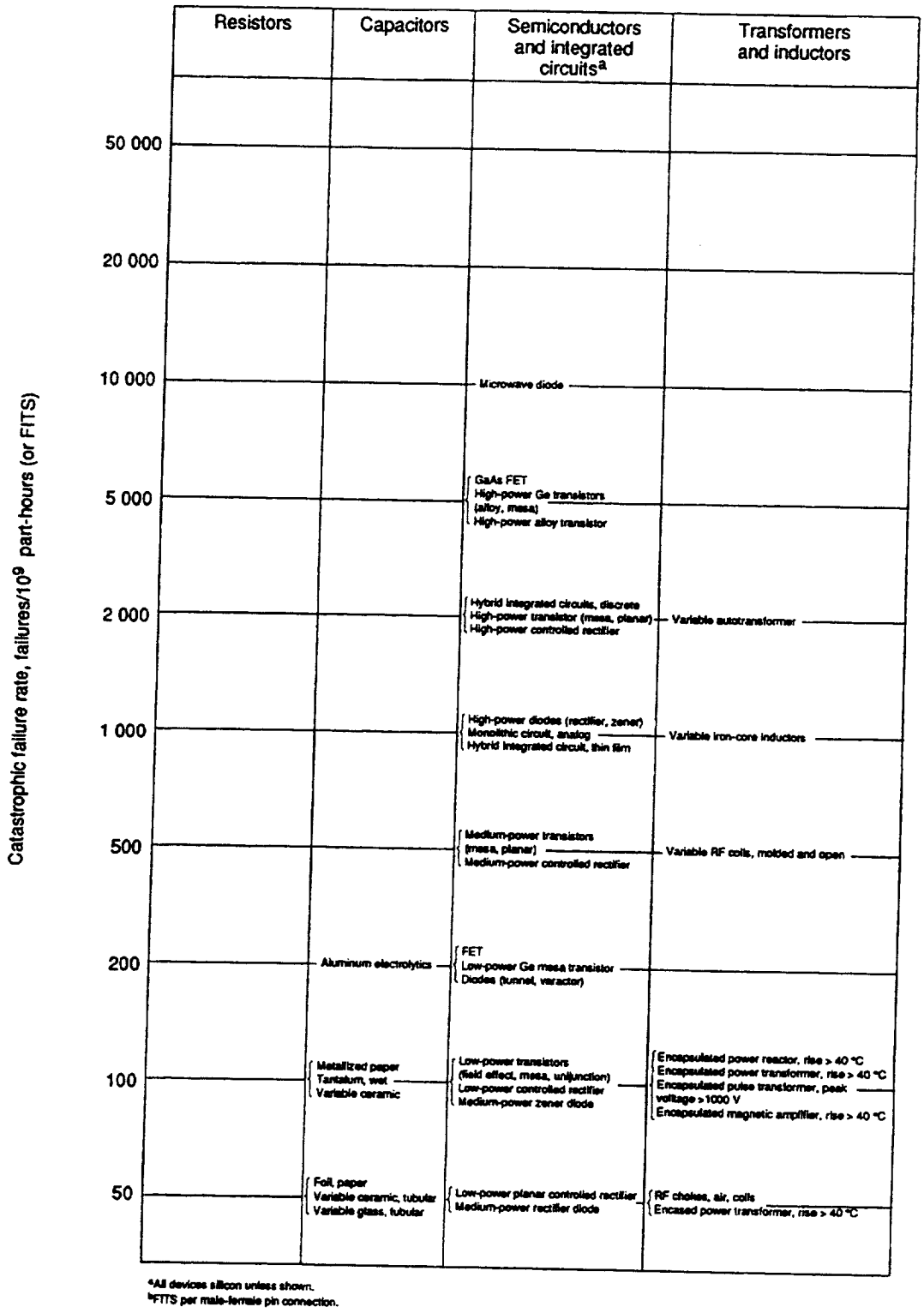


Figure A–2.—High-reliability catastrophic failure rates for operating mode. Failure rate for these parts in non-operating mode is about a factor of 10 less than values shown. (From ref. A–1).

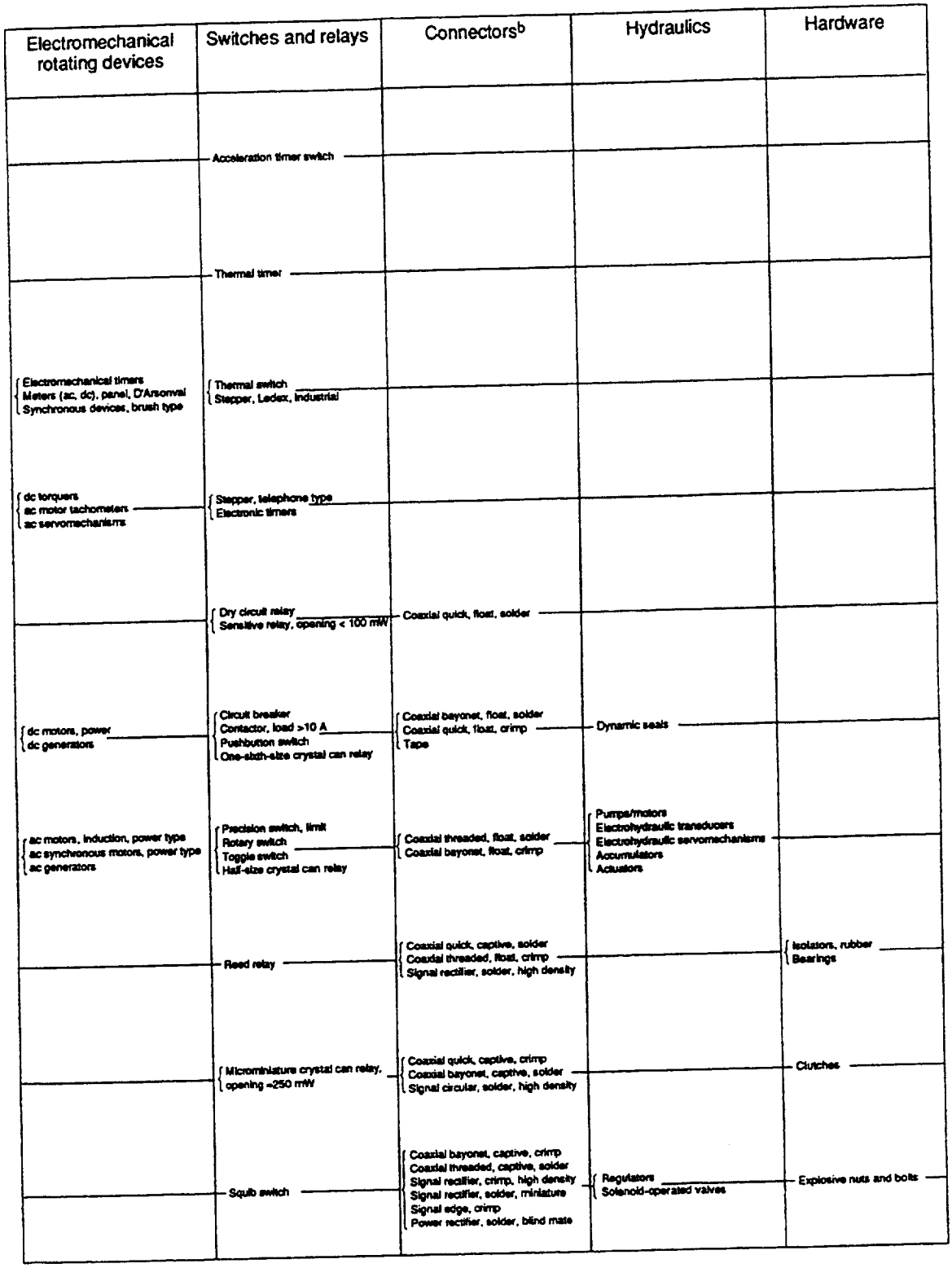
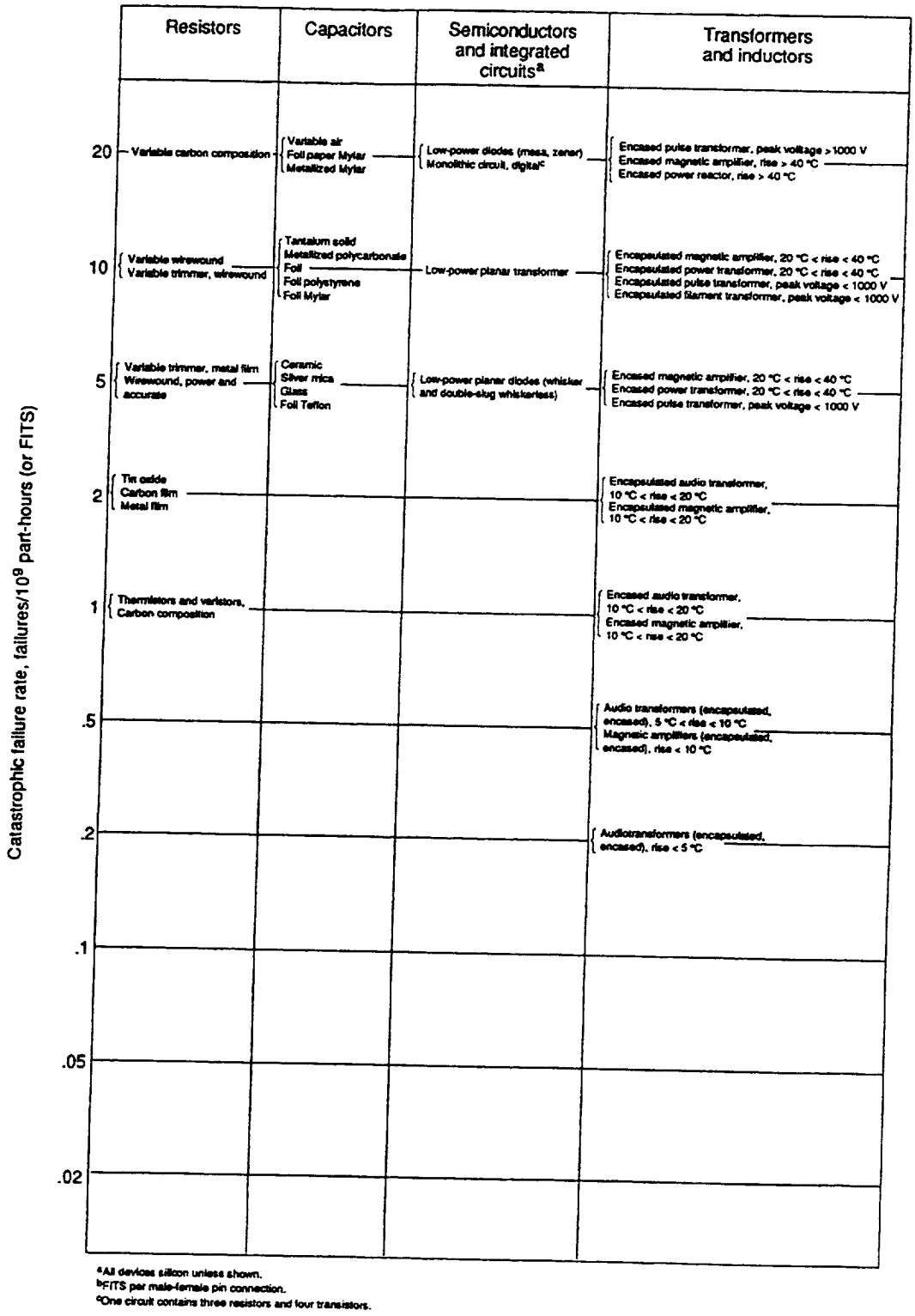


Figure A–2.—Continued.



[a]All devices silicon unless shown.
[b]FITS per male-female pin connection.
[c]One circuit contains three resistors and four transistors.

Figure A–2.—Continued.

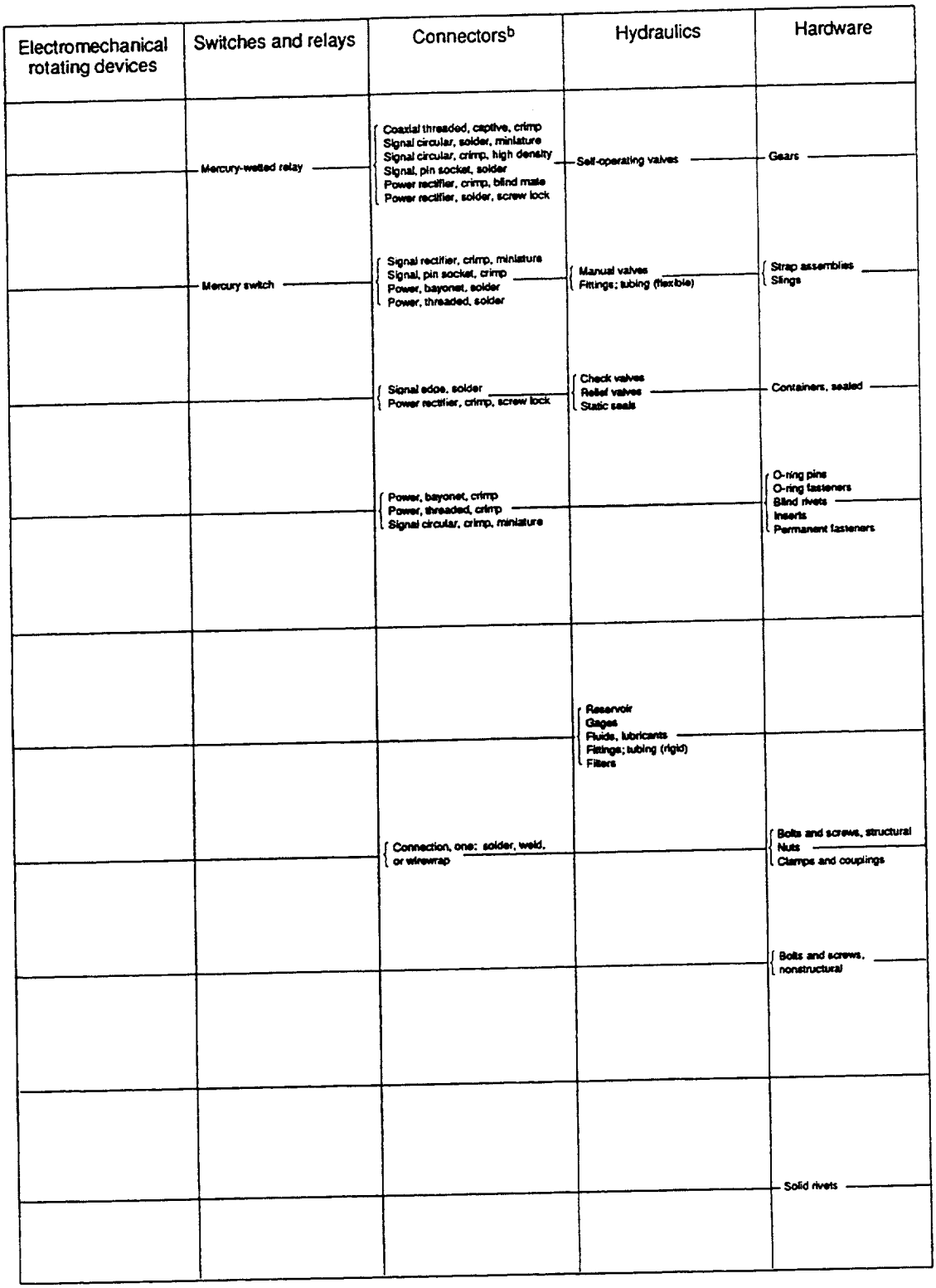


Figure A–2.—Concluded.

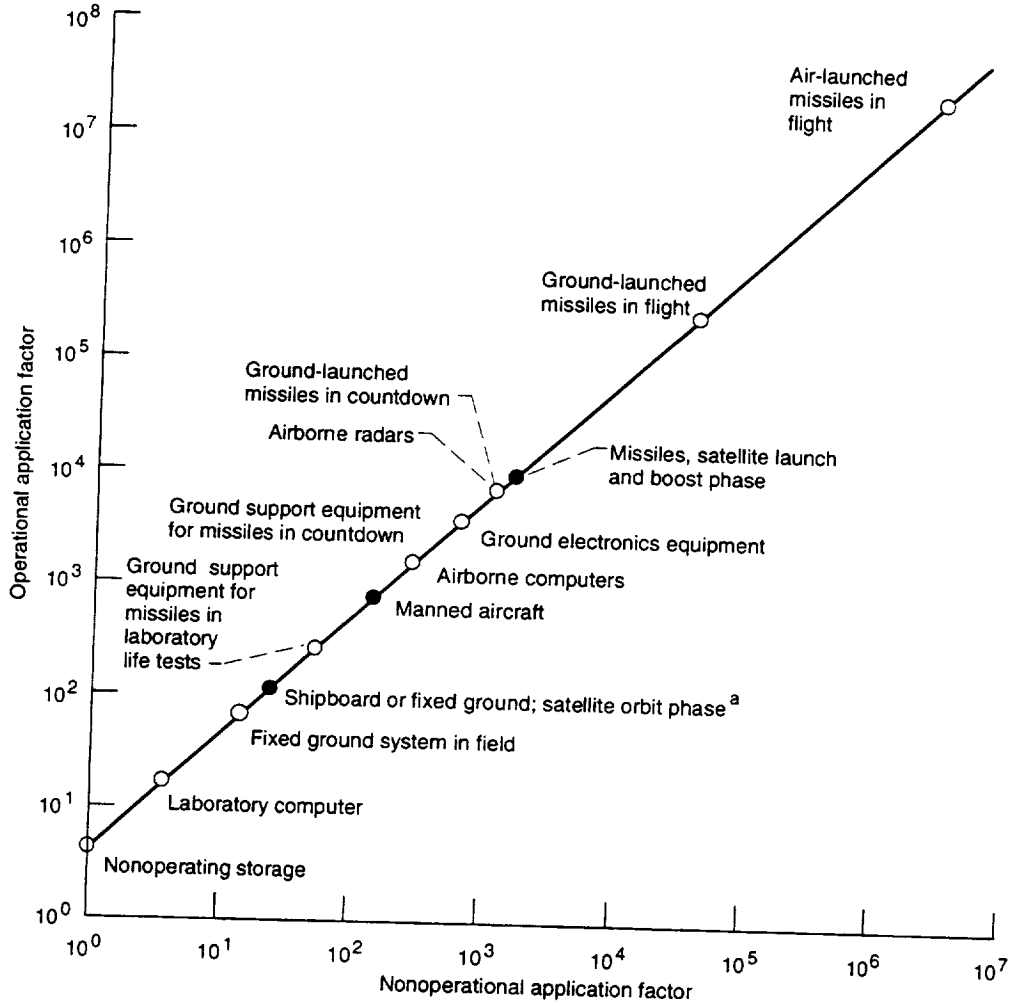


Figure A–3.—Application factor comparison for nonoperating storage of military standard electronic parts. MIL–STD–756 points (solid symbols) are given for comparison. (From ref. A–2.)

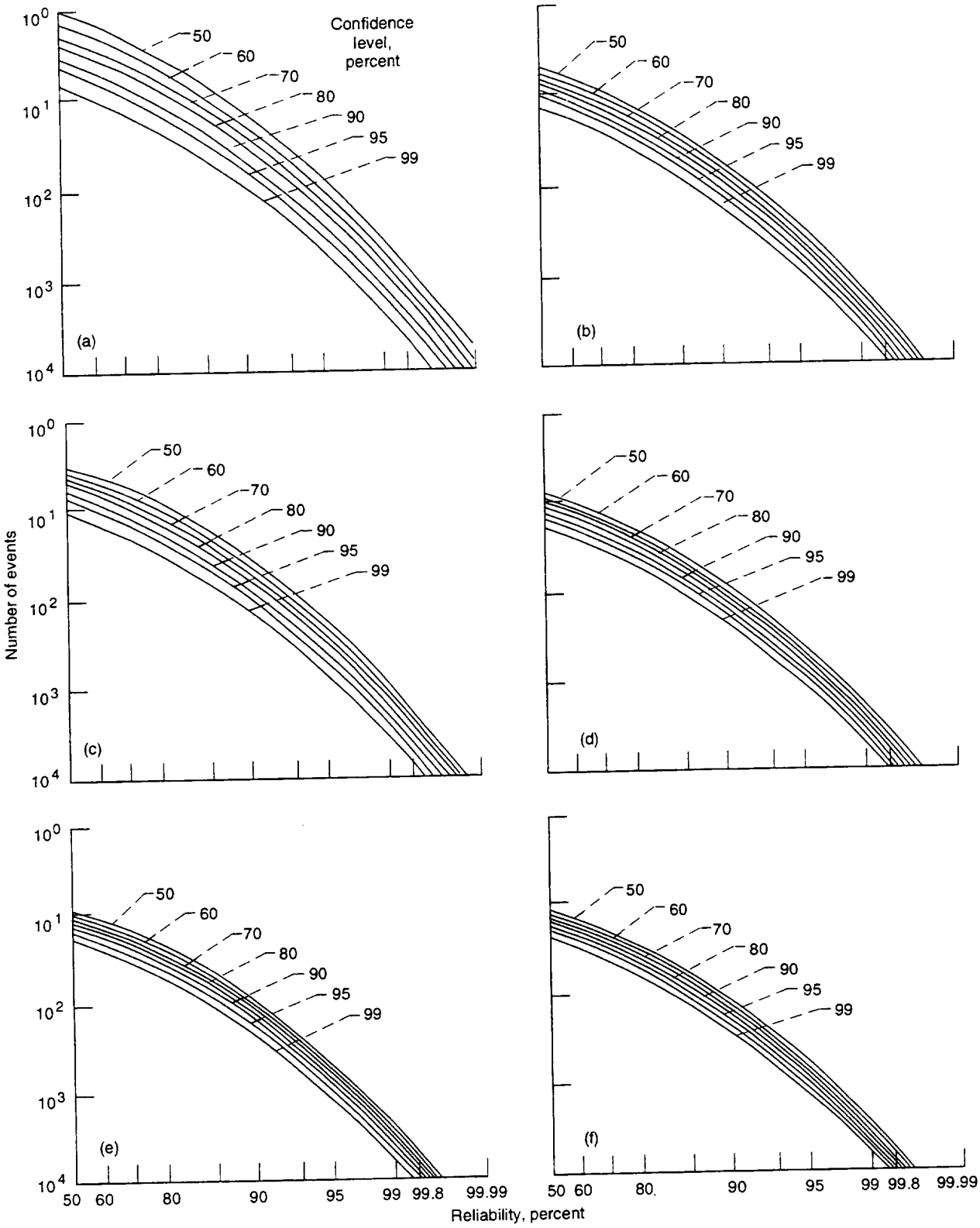


Figure A–4.—Confidence curves for attribute testing. (a) When no failures are observed. (b) When one failure is observed. (c) When two failures are observed. (d) When three failures are observed. (e) When four failures are observed. (f) When five failures are observed. (From ref. A–2.)

TABLE A-1.—VALUES OF NEGATIVE EXPONENTIAL FUNCTION e^{-x}

x	e^{-x}	x	e^{-x}	x	e^{-x}	x	e^{-x}	x	e^{-x}	x	e^{-x}
0.0000	1.00000	0.0050	0.99501	0.0100	0.99005	0.0150	0.98511	0.0200	0.98020	0.0250	0.97531
.0001	.99990	.0051	.99491	.0101	.98995	.0151	.98501	.0201	.98010	.0251	.97521
.0002	.99980	.0052	.99481	.0102	.98985	.0152	.98491	.0202	.98000	.0252	.97511
.0003	.99970	.0053	.99471	.0103	.98975	.0153	.98482	.0203	.97990	.0253	.97502
.0004	.99960	.0054	.99461	.0104	.98965	.0154	.98472	.0204	.97981	.0254	.97492
0.0005	0.99950	0.0055	0.99452	0.0105	0.98955	0.0155	0.98462	0.0205	0.97971	0.0255	0.97482
.0006	.99940	.0056	.99442	.0106	.98946	.0156	.98452	.0206	.97961	.0256	.97472
.0007	.99930	.0057	.99432	.0107	.98936	.0157	.98442	.0207	.97951	.0257	.97463
.0008	.99920	.0058	.99422	.0108	.98926	.0158	.98432	.0208	.97941	.0258	.97453
.0009	.99910	.0059	.99412	.0109	.98916	.0159	.98423	.0209	.97932	.0259	.97443
0.0010	0.99900	0.0060	0.99402	0.0110	0.98906	0.0160	0.98413	0.0210	0.97922	0.0260	0.97434
.0011	.99890	.0061	.99392	.0111	.98896	.0161	.98403	.0211	.97912	.0261	.97424
.0012	.99880	.0062	.99382	.0112	.98886	.0162	.98393	.0212	.97902	.0262	.97414
.0013	.99870	.0063	.99372	.0113	.98876	.0163	.98383	.0213	.97893	.0263	.97404
.0014	.99860	.0064	.99362	.0114	.98866	.0164	.98373	.0214	.97883	.0264	.97395
0.0015	0.99850	0.0065	0.99352	0.0115	0.98857	0.0165	0.98364	0.0215	0.97873	0.0265	0.97385
.0016	.99840	.0066	.99342	.0116	.98847	.0166	.98354	.0216	.97863	.0266	.97375
.0017	.99830	.0067	.99332	.0117	.98837	.0167	.98344	.0217	.97853	.0267	.97365
.0018	.99820	.0068	.99322	.0118	.98827	.0168	.98334	.0218	.97844	.0268	.97356
.0019	.99810	.0069	.99312	.0119	.98817	.0169	.98324	.0219	.97834	.0269	.97346
0.0020	0.99800	0.0070	0.99302	0.0120	0.98807	0.0170	0.98314	0.0220	0.97824	0.0270	0.97336
.0021	.99790	.0071	.99293	.0121	.98797	.0171	.98305	.0221	.97814	.0271	.97326
.0022	.99780	.0072	.99283	.0122	.98787	.0172	.98295	.0222	.97804	.0272	.97317
.0023	.99770	.0073	.99273	.0123	.98777	.0173	.98285	.0223	.97795	.0273	.97307
.0024	.99760	.0074	.99263	.0124	.98767	.0174	.98275	.0224	.97785	.0274	.97297
0.0025	0.99750	0.0075	0.99253	0.0125	0.98757	0.0175	0.98265	0.0225	0.97775	0.0275	0.97287
.0026	.99740	.0076	.99243	.0126	.98747	.0176	.98255	.0226	.97765	.0276	.97278
.0027	.99730	.0077	.99233	.0127	.98738	.0177	.98246	.0227	.97756	.0277	.97268
.0028	.99720	.0078	.99223	.0128	.98728	.0178	.98236	.0228	.97746	.0278	.97258
.0029	.99710	.0079	.99213	.0129	.98718	.0179	.98226	.0229	.97736	.0279	.97249
0.0030	0.99700	0.0080	0.99203	0.0130	0.98708	0.0180	0.98216	0.0230	0.97726	0.0280	0.97239
.0031	.99690	.0081	.99193	.0131	.98699	.0181	.98206	.0231	.97716	.0281	.97229
.0032	.99681	.0082	.99183	.0132	.98689	.0182	.98196	.0232	.97707	.0282	.97219
.0033	.99671	.0083	.99173	.0133	.98679	.0183	.98187	.0233	.97697	.0283	.97210
.0034	.99661	.0084	.99164	.0134	.98669	.0184	.98177	.0234	.97687	.0284	.97200
0.0035	0.99651	0.0085	0.99154	0.0135	0.98659	0.0185	0.98167	0.0235	0.97677	0.0285	0.97190
.0036	.99641	.0086	.99144	.0136	.98649	.0186	.98157	.0236	.97668	.0286	.97181
.0037	.99631	.0087	.99134	.0137	.98639	.0187	.98147	.0237	.97658	.0287	.97171
.0038	.99621	.0088	.99124	.0138	.98629	.0188	.98138	.0238	.97648	.0288	.97161
.0039	.99611	.0089	.99114	.0139	.98620	.0189	.98128	.0239	.97638	.0289	.97151
0.0040	0.99601	0.0090	0.99104	0.0140	0.98610	0.0190	0.98118	0.0240	0.97629	0.0290	0.97142
.0041	.99591	.0091	.99094	.0141	.98600	.0191	.98108	.0241	.97619	.0291	.97132
.0042	.99581	.0092	.99084	.0142	.98590	.0192	.98098	.0242	.97609	.0292	.97122
.0043	.99571	.0093	.99074	.0143	.98580	.0193	.98089	.0243	.97599	.0293	.97113
.0044	.99561	.0094	.99064	.0144	.98570	.0194	.98079	.0244	.97590	.0294	.97103
0.0045	0.99551	0.0095	0.99054	0.0145	0.98560	0.0195	0.98069	0.0245	0.97580	0.0295	0.97093
.0046	.99541	.0096	.99045	.0146	.98551	.0196	.98059	.0246	.97570	.0296	.97083
.0047	.99531	.0097	.99035	.0147	.98541	.0197	.98049	.0247	.97560	.0297	.97074
.0048	.99521	.0098	.99025	.0148	.98531	.0198	.98039	.0248	.97550	.0298	.97064
.0049	.99511	.0099	.99015	.0149	.98521	.0199	.98030	.0249	.97541	.0299	.97054

TABLE A-1.—Continued.

x	e^{-x}	x	e^{-x}	x	e^{-x}	x	e^{-x}	x	e^{-x}	x	e^{-x}
0.0300	0.97045	0.0350	0.96561	0.0400	0.96079	0.0450	0.95600	0.0500	0.95123	0.0550	0.94649
.0301	.97035	.0351	.96551	.0401	.96069	.0451	.95590	.0501	.95113	.0551	.94639
.0302	.97025	.0352	.96541	.0402	.96060	.0452	.95581	.0502	.95104	.0552	.94630
.0303	.97015	.0353	.96531	.0403	.96050	.0453	.95571	.0503	.95094	.0553	.94620
.0304	.97006	.0354	.96522	.0404	.96041	.0454	.95562	.0504	.95085	.0554	.94611
0.0305	0.96996	0.0355	0.96512	0.0405	0.96031	0.0455	0.95552	0.505	0.95075	0.0555	0.94601
.0306	.96986	.0356	.96503	.0406	.96021	.0456	.95542	.0506	.95066	.0556	.94592
.0307	.96977	.0357	.96493	.0407	.96012	.0457	.95533	.0507	.95056	.0557	.94582
.0308	.96967	.0358	.96483	.0408	.96002	.0458	.95523	.0508	.95047	.0558	.94573
.0309	.96957	.0359	.96474	.0409	.95993	.0459	.95514	.0509	.95037	.0559	.94563
0.0310	0.96948	0.0360	0.96464	0.0410	0.95983	0.0460	0.95504	0.0510	0.95028	0.0560	0.94554
.0311	.96938	.0361	.96454	.0411	.95973	.0461	.95495	.0511	.95018	.0561	.94544
.0312	.96928	.0362	.96445	.0412	.95964	.0462	.95485	.0512	.95009	.0562	.94535
.0313	.96918	.0363	.96435	.0413	.95954	.0463	.95476	.0513	.94999	.0563	.94526
.0314	.96909	.0364	.96425	.0414	.95945	.0464	.95466	.0514	.94990	.0564	.94516
0.0315	0.96899	0.0365	0.96416	0.0415	0.95935	0.0465	0.95456	0.0515	0.94980	0.0565	0.94507
.0316	.96889	.0366	.96406	.0416	.95925	.0466	.95447	.0516	.94971	.0566	.94488
.0317	.96879	.0367	.96397	.0417	.94916	.0467	.95437	.0517	.94961	.0567	.94488
.0318	.96870	.0368	.96387	.0418	.95906	.0468	.95428	.0518	.94952	.0568	.94478
.0319	.96860	.0369	.96377	.0419	.95897	.0469	.95418	.0519	.94942	.0569	.94469
0.0320	0.96851	0.0370	0.96368	0.0420	0.95887	0.0470	0.95409	0.0520	0.94933	0.0570	0.94450
.0321	.96841	.0371	.96358	.0421	.95877	.0471	.95399	.0521	.94923	.0571	.94450
.0322	.96831	.0372	.96348	.0422	.95868	.0472	.95390	.0522	.94914	.0572	.94441
.0323	.96822	.0373	.96339	.0423	.95858	.0473	.95380	.0523	.94904	.0573	.94431
.0324	.96812	.0374	.96329	.0424	.95849	.0474	.95371	.0524	.94895	.0574	.94422
0.0325	0.96802	0.0375	0.96319	0.0425	0.95839	0.0475	0.95361	0.0525	0.94885	0.0575	0.94412
.0326	.96793	.0376	.96310	.0426	.95829	.0476	.95352	.0526	.94876	.0576	.94403
.0327	.96783	.0377	.96300	.0427	.95820	.0477	.95342	.0527	.94866	.0577	.94393
.0328	.96773	.0378	.96291	.0428	.95810	.0478	.95332	.0528	.94857	.0578	.94384
.0329	.96764	.0379	.96281	.0429	.95801	.0479	.95323	.0529	.94847	.0579	.94374
0.0330	0.96754	0.0380	0.96271	0.0430	0.95791	0.0480	0.95313	0.0530	0.94838	0.0580	0.94365
.0331	.96744	.0381	.96262	.0431	.94782	.0481	.95304	.0531	.94829	.0581	.94356
.0332	.96735	.0382	.96252	.0432	.95772	.0482	.95294	.0532	.94819	.0582	.94346
.0333	.96725	.0383	.96242	.0433	.95762	.0483	.95285	.0533	.94810	.0583	.94337
.0334	.96715	.0384	.96233	.0434	.95753	.0484	.95275	.0534	.94800	.0584	.94327
0.0335	0.96705	0.0385	0.96223	0.0435	0.95743	0.0485	0.95266	0.0535	0.94791	0.0585	0.94318
.0336	.96696	.0386	.96214	.0436	.95734	.0486	.95256	.0536	.94781	.0586	.94308
.0337	.96686	.0387	.96204	.0437	.95724	.0487	.95247	.0537	.94772	.0587	.94299
.0338	.96676	.0388	.96194	.0438	.95715	.0488	.95237	.0538	.94762	.0588	.94289
.0339	.96667	.0389	.96185	.0439	.95705	.0489	.95228	.0539	.94753	.0589	.94280
0.0340	0.96657	0.0390	0.96175	0.0440	0.95695	0.0490	0.95218	0.0540	0.94743	0.0590	0.94271
.0341	.96647	.0391	.96165	.0441	.95686	.0491	.95209	.0541	.94734	.0591	.94261
.0342	.96638	.0392	.96156	.0442	.95676	.0492	.95199	.0542	.94724	.0592	.94252
.0343	.96628	.0393	.96146	.0443	.95667	.0493	.95190	.0543	.94715	.0593	.94242
.0344	.96618	.0394	.96137	.0444	.95657	.0494	.95180	.0544	.94705	.0594	.94233
0.0345	0.96609	0.0395	0.96127	0.0445	0.95648	0.0495	0.95171	0.0545	0.94696	0.0595	0.94224
.0346	.96599	.0396	.96117	.0446	.95638	.0496	.95161	.0546	.94686	.0596	.94214
.0347	.96590	.0397	.96108	.0447	.95628	.0497	.95151	.0547	.94677	.0597	.94205
.0348	.96580	.0398	.96098	.0448	.95619	.0498	.95142	.0548	.94667	.0598	.94195
.0349	.96570	.0399	.96089	.0449	.95609	.0499	.95132	.0549	.94658	.0599	.94186

TABLE A-1.—Continued.

x	e^{-x}	x	e^{-x}	x	e^{-x}	x	e^{-x}	x	e^{-x}	x	e^{-x}
0.0600	0.94176	0.0650	0.93707	0.0700	0.93239	0.0750	0.92774	0.0800	0.92312	0.0850	0.91851
.0601	.94167	.0651	.93697	.0701	.93230	.0751	.92765	.0801	.92302	.0851	.91842
.0602	.94158	.0652	.93688	.0702	.93221	.0752	.92756	.0802	.92293	.0852	.91833
.0603	.94148	.0563	.93679	.0703	.93211	.0753	.92747	.0803	.92284	.0853	.91824
.0604	.94139	.0654	.93669	.0704	.93202	.0754	.92737	.0804	.92275	.0854	.91814
0.0605	0.94129	0.0655	0.93660	0.0705	0.93193	0.0755	0.92728	0.0805	0.92265	0.0855	0.91805
.0606	.94120	.0656	.93651	.0706	.93183	.0756	.92719	.0806	.92256	.0856	.91796
.0607	.94111	.0657	.93641	.0707	.93174	.0757	.92709	.0807	.92247	.0857	.91787
.0608	.94101	.0658	.93632	.0708	.93165	.0758	.92700	.0808	.92238	.0858	.91778
.0609	.94092	.0659	.93622	.0709	.93156	.0759	.92691	.0809	.92229	.0859	.91769
0.0610	0.94082	0.0660	0.93613	0.0710	0.93146	0.0760	0.92682	0.0810	0.92219	0.0860	0.91759
.0611	.94073	.0661	.93604	.0711	.93137	.0761	.92672	.0811	.92210	.0861	.91750
.0612	.94064	.0662	.93594	.0712	.93128	.0762	.92663	.0812	.92201	.0862	.91741
.0613	.94054	.0663	.93585	.0713	.93118	.0763	.92654	.0813	.92191	.0863	.91732
.0614	.94045	.0664	.93576	.0714	.93109	.0764	.92645	.0814	.92182	.0864	.91723
0.0615	0.94035	0.0665	0.93566	0.0715	0.93100	0.0765	0.92635	0.0815	0.92173	0.0865	0.91714
.0616	.94026	.0666	.93557	.0716	.93090	.0766	.92626	.0816	.92164	.0866	.91704
.0617	.94016	.0667	.93548	.0717	.93081	.0767	.92617	.0817	.92155	.0867	.91695
.0618	.94007	.0668	.93538	.0718	.93072	.0768	.92608	.0818	.92146	.0868	.91686
.0619	.93998	.0669	.93529	.0719	.93062	.0769	.92598	.0819	.92136	.0869	.91677
0.0620	0.93988	0.0670	0.93520	0.0720	0.93053	0.0770	0.92589	0.0820	0.92127	0.0870	0.91668
.0621	.93979	.0671	.93510	.0721	.93044	.0771	.92580	.0821	.92118	.0871	.91659
.0622	.93969	.0672	.93501	.0722	.93034	.0772	.92570	.0822	.92109	.0872	.91649
.0623	.93960	.0673	.93491	.0723	.93025	.0773	.92561	.0823	.92100	.0873	.91640
.0624	.93951	.0674	.93482	.0724	.93016	.0774	.92552	.0824	.92090	.0874	.91631
0.0625	0.93941	0.0675	0.93473	0.0725	0.93007	0.0775	0.92543	0.0825	0.92081	0.0875	0.91622
.0626	.93932	.0676	.93463	.0726	.92997	.0776	.92533	.0826	.92072	.0876	.91613
.0627	.93923	.0677	.93454	.0727	.92988	.0777	.92524	.0827	.92063	.0877	.91604
.0628	.93913	.0678	.93445	.0728	.92979	.0778	.92515	.0828	.92054	.0878	.91594
.0629	.93904	.0679	.93435	.0729	.92969	.0779	.92506	.0829	.92044	.0879	.91585
0.0630	0.93894	0.0680	0.93425	0.0730	0.92960	0.0780	0.92496	0.0830	0.92035	0.0880	0.91576
.0631	.93885	.0681	.93417	.0731	.92951	.0781	.92487	.0831	.92026	.0881	.91567
.0632	.93876	.0682	.93407	.0732	.92941	.0782	.92478	.0832	.92019	.0882	.91558
.0633	.93866	.0683	.93398	.0733	.92932	.0783	.92469	.0833	.92008	.0883	.91549
.0634	.93857	.0684	.93389	.0734	.92923	.0784	.92459	.0834	.91998	.0884	.91539
0.0635	0.93847	0.0685	0.93379	0.0735	0.92914	0.0785	0.92450	0.0835	0.91989	0.0885	0.91530
.0636	.93838	.0686	.93370	.0736	.92904	.0786	.92441	.0836	.91980	.0886	.91521
.0637	.93829	.0687	.93361	.0737	.92895	.0787	.92432	.0837	.91971	.0887	.91512
.0638	.93819	.0688	.93351	.0738	.92886	.0788	.92422	.0838	.91962	.0888	.91503
.0639	.93810	.0689	.93342	.0739	.92876	.0789	.92413	.0839	.91952	.0889	.91494
0.0640	0.93800	0.0690	0.93333	0.0740	0.92867	0.0790	0.92404	0.0840	0.91943	0.0890	0.91485
.0641	.93791	.0691	.93323	.0741	.92858	.0791	.92395	.0841	.91934	.0891	.91475
.0642	.93782	.0692	.93314	.0742	.92849	.0792	.92386	.0842	.91925	.0892	.91466
.0643	.93772	.0693	.93305	.0743	.92839	.0793	.92376	.0843	.91916	.0893	.91457
.0644	.93763	.0694	.93295	.0744	.92830	.0794	.92367	.0844	.91906	.0894	.91448
0.0645	0.93754	0.0695	0.93286	0.0745	0.92921	0.0795	0.92358	0.0845	0.91897	0.0895	0.91439
.0646	.93744	.0696	.93277	.0746	.92811	.0796	.92349	.0846	.91888	.0896	.91430
.0647	.93735	.0697	.93267	.0747	.92802	.0797	.92339	.0847	.91879	.0897	.91421
.0648	.93725	.0698	.93258	.0748	.92793	.0798	.92330	.0848	.91870	.0898	.91411
.0649	.93716	.0699	.93249	.0749	.92784	.0799	.92321	.0849	.91860	.0899	.91402

TABLE A-1.—Continued.

x	e^{-x}	x	e^{-x}	x	e^{-x}	x	e^{-x}	x	e^{-x}	x	e^{-x}
0.0900	0.91393	0.0950	0.90937	0.1000	0.90484	0.1050	0.90032	0.1100	0.89583	0.1150	0.89137
.0901	.91384	.0951	.90928	.1001	.90475	.1051	.90023	.1101	.89574	.1151	.89128
.0902	.91375	.0952	.90919	.1002	.90466	.1052	.90014	.1102	.89565	.1152	.89119
.0903	.91366	.0953	.90910	.1003	.90457	.1053	.90005	.1103	.89557	.1153	.89110
.0904	.91357	.0954	.90901	.1004	.90448	.1054	.89996	.1004	.89548	.1154	.89101
0.0905	0.91347	0.0955	0.90892	0.1005	0.90439	0.1055	0.89987	0.1105	0.89539	0.1155	0.89092
.0906	.91338	.0956	.90883	.1006	.90429	.1056	.89978	.1106	.89530	.1156	.89083
.0907	.91329	.0957	.90874	.1007	.90420	.1057	.89969	.1107	.89521	.1157	.89074
.0908	.91320	.0958	.90865	.1008	.90411	.1058	.89960	.1108	.89512	.1158	.89065
.0909	.91311	.0959	.90855	.1009	.90402	.1059	.89951	.1109	.89503	.1159	.89056
0.0910	0.91302	0.0960	0.90846	0.1010	0.90393	0.1060	0.89942	0.1110	0.89494	0.1160	0.89048
.0911	.91293	.0961	.90837	.1011	.90384	.1061	.89933	.1111	.89485	.1161	.89039
.0912	.91284	.0962	.90828	.1012	.90375	.1062	.89924	.1112	.89476	.1162	.89030
.0913	.91274	.0963	.90819	.1013	.90366	.1063	.89915	.1113	.89467	.1163	.89021
.0914	.91265	.0964	.90810	.1014	.90357	.1064	.89906	.1114	.89458	.1164	.89012
0.0915	0.91256	0.0965	0.90801	0.1015	0.90348	0.1065	0.89898	0.1115	0.89449	0.1165	0.89003
.0916	.91247	.0966	.90792	.1016	.90339	.1066	.89889	.1116	.89440	.1166	.88994
.0917	.91238	.0967	.90783	.1017	.90330	.1067	.89880	.1117	.89431	.1167	.88985
.0918	.91229	.0968	.90774	.1018	.90321	.1068	.89871	.1118	.89422	.1168	.88976
.0919	.91220	.0969	.90765	.1019	.90312	.1069	.89862	.1119	.89413	.1169	.88967
0.0920	0.92111	0.0970	0.90756	0.1020	0.90303	0.1070	0.89853	0.1120	0.89404	0.1170	0.88959
.0921	.91201	.0971	.90747	.1021	.90294	.1071	.89844	.1121	.89395	.1171	.88950
.0922	.91192	.0972	.90737	.1022	.90285	.1072	.89835	.1122	.89387	.1172	.88941
.0923	.91183	.0973	.90728	.1023	.90276	.1073	.89826	.1123	.89378	.1173	.88932
.0924	.91174	.0974	.90719	.1024	.90267	.1074	.89817	.1124	.89369	.1174	.88923
0.0925	0.91165	0.0975	0.90710	0.1025	0.90258	0.1075	0.89808	0.1125	0.89360	0.1175	0.88914
.0926	.91156	.0976	.90701	.1026	.90249	.1076	.89799	.1126	.89351	.1176	.88905
.0927	.91147	.0977	.90692	.1027	.90240	.1077	.89790	.1127	.89342	.1177	.88896
.0928	.91138	.0978	.90683	.1028	.90231	.1078	.89781	.1128	.89333	.1178	.88887
.0929	.91128	.0979	.90674	.1029	.90222	.1079	.89772	.1129	.89324	.1179	.88878
0.0930	0.91119	0.0980	0.90665	0.1030	0.90213	0.1080	0.89763	0.1130	0.89315	0.1180	0.88870
.0931	.91110	.0981	.90656	.1031	.90204	.1081	.89754	.1131	.89306	.1181	.88861
.0932	.91101	.0982	.90647	.1032	.90195	.1082	.89745	.1132	.89297	.1182	.88852
.0933	.91092	.0983	.90638	.1033	.90186	.1083	.89736	.1133	.89288	.1183	.88843
.0934	.91083	.0984	.90629	.1034	.90177	.1084	.89727	.1134	.89279	.1184	.88834
0.0935	0.91074	0.0985	0.90620	0.1035	0.90168	0.1085	0.89718	0.1135	0.89270	0.1185	0.88825
.0936	.91065	.0986	.90611	.1036	.90159	.1086	.89709	.1136	.89261	.1186	.88816
.0937	.91056	.0987	.90601	.1037	.90150	.1087	.89700	.1137	.89253	.1187	.88807
.0938	.91046	.0988	.90592	.1038	.90141	.1088	.89691	.1138	.89244	.1188	.88799
.0939	.91037	.0989	.90583	.1039	.90132	.1089	.89682	.1139	.89235	.1189	.88790
0.0940	0.91028	0.0990	0.90574	0.1040	0.90123	0.1090	0.89673	0.1140	0.89226	0.1190	0.98781
.0941	.91019	.0991	.90565	.1041	.90114	.1091	.89664	.1141	.89217	.1191	.88772
.0942	.91010	.0992	.90556	.1042	.90105	.1092	.89655	.1142	.89208	.1192	.88763
.0943	.91001	.0993	.90547	.1043	.90095	.1093	.89646	.1143	.89199	.1193	.88754
.0944	.90992	.0994	.90538	.1044	.90086	.1094	.89637	.1144	.89190	.1194	.88745
0.0945	0.90983	0.0995	0.90529	0.1045	0.90077	0.1095	0.89628	0.1145	0.89181	0.1195	0.88736
.0946	.90974	.0996	.90520	.1046	.90068	.1096	.89619	.1146	.89172	.1196	.88728
.0947	.90965	.0997	.90501	.1047	.90059	.1097	.89610	.1147	.89163	.1197	.88719
.0948	.90955	.0998	.90502	.1048	.90050	.1098	.89601	.1148	.89154	.1198	.88710
.0949	.90946	.0999	.90493	.1049	.90041	.1099	.89592	.1149	.89146	.1199	.88701

TABLE A-1.—Continued.

x	e^{-x}	x	e^{-x}	x	e^{-x}	x	e^{-x}	x	e^{-x}	x	e^{-x}
0.1200	0.88692	0.1250	0.88250	0.1300	0.87810	0.1350	0.87372	0.1400	0.86936	0.1450	0.86502
.1201	.88683	.1251	.88241	.1301	.87801	.1351	.87363	.1401	.86927	.1451	.86494
.1202	.88674	.1252	.88232	.1302	.87792	.1352	.87354	.1402	.86918	.1452	.86485
.1203	.88665	.1253	.88223	.1303	.87783	.1353	.87345	.1403	.86910	.1453	.86476
.1204	.88657	.1254	.88214	.1304	.87774	.1354	.87337	.1404	.86901	.1454	.86468
0.1205	0.88648	0.1255	0.88206	0.1305	0.87766	0.1355	0.87328	0.1405	0.86892	0.1455	0.86459
.1206	.88639	.1256	.88197	.1306	.87757	.1356	.87319	.1406	.86884	.1456	.86450
.1207	.88630	.1257	.88188	.1307	.87748	.1357	.87310	.1407	.86875	.1457	.86442
.1208	.88621	.1258	.88179	.1308	.87739	.1358	.87302	.1408	.86866	.1458	.86433
.1209	.88612	.1259	.88170	.1309	.87731	.1359	.87283	.1409	.86858	.1459	.86424
0.1210	0.88603	0.1260	0.88161	0.1310	0.87722	0.1360	0.87284	0.1410	0.86849	0.1460	0.86416
.1211	.88595	.1261	.88153	.1311	.87713	.1361	.87276	.1411	.86840	.1461	.86407
.1212	.88586	.1262	.88144	.1312	.87704	.1362	.87267	.1412	.86832	.1462	.86398
.1213	.88577	.1263	.88135	.1313	.87695	.1363	.87258	.1413	.86823	.1463	.86390
.1214	.88568	.1264	.88126	.1314	.87687	.1364	.87249	.1414	.86814	.1464	.86381
0.1215	0.88559	0.1265	0.88117	0.1315	0.87678	0.1365	0.87241	0.1415	0.86806	0.1465	0.86373
.1216	.88550	.1266	.88109	.1316	.87669	.1366	.87232	.1416	.86797	.1466	.86364
.1217	.88541	.1267	.88100	.1317	.87660	.1367	.87223	.1417	.86788	.1467	.86355
.1218	.88533	.1268	.88091	.1318	.87652	.1368	.87214	.1418	.86779	.1468	. 86347
.1219	.88524	.1269	.88082	.1319	.87643	.1369	.87206	.1419	.86771	.1469	.86338
0.1220	0.88515	0.1270	0.88065	0.1320	0.87634	0.1370	0.87197	0.1420	0.86762	0.1470	0.86329
.1221	.88506	.1271		.1321	.87625	.1371	.87188	.1421	.86753	.1471	.86321
.1222	.88497	.1272	.88056	.1322	.87617	.1372	.87180	.1422	.86745	.1472	.86312
.1223	.88488	.1273	.88047	.1323	.87608	.1373	.87171	.1423	.86736	.1473	.86304
.1224	.88479	.1274	.88038	.1324	.87599	.1374	.87162	.1424	.86727	.1474	.86295
0.1225	0.88471	0.1275	0.88029	0.1325	0.87590	0.1375	0.87153	0.1425	0.86719	0.1475	0.86286
.1226	.88462	.1276	.88021	.1326	.87582	.1376	.87145	.1426	.86710	.1476	.86278
.1227	.88453	.1277	.88012	.1327	.87573	.1377	.87136	.1427	.86701	.1477	.86269
.1228	.88444	.1728	.88003	.1328	.87564	.1378	.87127	.1428	.86693	.1478	.86260
.1229	.88435	.1279	.87994	.1329	.87555	.1379	.87119	.1429	.86684	.1479	.86252
0.1230	0.88426	0.1280	0.87985	0.1330	0.87547	0.1380	0.87110	0.1430	0.86675	0.1480	0.86243
.1231	.88418	.1281	.87977	.1331	.87538	.1381	.87101	.1431	.86667	.1481	.86234
.1232	.88409	.1282	.87968	.1332	.87529	.1382	.87092	.1432	.86658	.1482	.86226
.1233	.88400	.1283	.87959	.1333	.87520	.1383	.87084	.1433	.86649	.1483	.86217
.1234	.88391	.1284	.87950	.1334	.87511	.1384	.87075	.1434	.86641	.1484	.86209
0.1235	0.88382	0.1285	0.87941	0.1335	0.87503	0.1385	0.87066	0.1435	0.86632	0.1485	0.86200
.1236	.88373	.1286	.87933	.1336	.87494	.1386	.87058	.1436	.86623	.1486	.86191
.1237	.88364	.1287	.87924	.1337	.87485	.1387	.87049	.1437	.86615	.1487	.86183
.1238	.88356	.1288	.87915	.1338	.87477	.1388	.87040	.1438	.86606	.1488	.86174
.1239	.88347	.1289	.87906	.1339	.87468	.1389	.87031	.1439	.86597	.1489	.86166
0.1240	0.88338	0.1290	0.87897	0.1340	0.87459	0.1390	0.87023	0.1440	0.86589	0.1490	0.86157
.1241	.88329	.1291	.87889	.1341	.87450	.1391	.87014	.1441	.86580	.1491	.86148
.1242	.88320	.1292	.87880	.1342	.87442	.1392	.87005	.1442	.86571	.1492	.86140
.1243	.88311	.1293	.87871	.1343	.87433	.1393	.86997	.1443	.86563	.1493	.86131
.1244	.88303	.1294	.87862	.1344	.87424	.1394	.86988	.1444	.86554	.1494	.86122
0.1245	0.88294	0.1295	0.87853	0.1345	0.87415	0.1395	0.86979	0.1445	0.86545	0.1495	0.86114
.1246	.88285	.1296	.87845	.1346	.87407	.1396	.86971	.1446	.86537	.1496	.86105
.1247	.88276	.1297	.87836	.1347	.87398	.1397	.86962	.1447	.86528	.1497	.86097
.1248	.88267	.1298	.87827	.1348	.87389	.1398	.86953	.1448	.86520	.1498	.86088
.1249	.88256	.1299	.87818	.1349	.87380	.1399	.86945	.1449	.86511	.1499	.86079

TABLE A-1.—Continued.

x	e^{-x}	x	e^{-x}	x	e^{-x}	x	e^{-x}	x	e^{-x}	x	e^{-x}
0.1500	0.86071	0.1550	0.85642	0.1600	0.85214	0.1650	0.84789	0.1700	0.84366	0.1750	0.83946
.1501	.86062	.1551	.85633	.1601	.85206	.1651	.84781	.1701	.84358	.1751	.83937
.1502	.86054	.1552	.85624	.1602	.85197	.1652	.84772	.1702	.84350	.1752	.83929
.1503	.86045	.1553	.85616	.1603	.85189	.1653	.84764	.1703	.84341	.1753	.83921
.1504	.86036	.1554	.85607	.1604	.85180	.1654	.84755	.1704	.84333	.1754	.83912
0.1505	0.86028	0.1555	0.85599	0.1605	0.85172	0.1655	0.84747	0.1705	0.84324	0.1755	0.83904
.1506	.86019	.1556	.85590	.1606	.85163	.1656	.84739	.1706	.84316	.1756	.83895
.1507	.86010	.1557	.85582	.1607	.85155	.1657	.84730	.1707	.84307	.1757	.83887
.1508	.86002	.1558	.85573	.1608	.85146	.1658	.84722	.1708	.84299	.1758	.83879
.1509	.85993	.1559	.85564	.1609	.85138	.1659	.84713	.1709	.84296	.1759	.83870
0.1510	0.85985	0.1560	0.85556	0.1610	0.85129	0.1660	0.84705	0.1710	0.84282	0.1760	0.83862
.1511	.85976	.1561	.85547	.1611	.85121	.1661	.84696	.1711	.84274	.1761	.83853
.1512	.85968	.1562	.85539	.1612	.85112	.1662	.84688	.1712	.84265	.1762	.83845
.1513	.85959	.1563	.85530	.1613	.85104	.1663	.84679	.1713	.84257	.1763	.83837
.1514	.85950	.1564	.85522	.1614	.85095	.1664	.84671	.1714	.84248	.1764	.83828
0.1515	0.85942	0.1565	0.85513	0.1615	0.85087	0.1665	0.84662	0.1715	0.84240	0.1765	0.83820
.1516	.85933	.1566	.85505	.1616	.85078	.1666	.84654	.1716	.84231	.1766	.83811
.1517	.85925	.1567	.85496	.1617	.85070	.1667	.84645	.1717	.84223	.1767	.83803
.1518	.85916	.1568	.85488	.1618	.85061	.1668	.84637	.1718	.84215	.1768	.83795
.1519	.85907	.1569	.85479	.1619	.85053	.1669	.84628	.1719	.84206	.1769	.83786
0.1520	0.85899	0.1570	0.85470	0.1620	0.85044	0.1670	0.84620	0.1720	0.84198	0.1770	0.83778
.1521	.85890	.1571	.85462	.1621	.85036	.1671	.84611	.1721	.84189	.1771	.83770
.1522	.85882	.1572	.85453	.1622	.85027	.1672	.84603	.1722	.84181	.1772	.83761
.1523	.85873	.1573	.85445	.1623	.85019	.1673	.84595	.1723	.84173	.1773	.83753
.1524	.85864	.1574	.85436	.1624	.85010	.1674	.84586	.1724	.84164	.1774	.83744
0.1525	0.85856	0.1575	0.85428	0.1625	0.85002	0.1675	0.84578	0.1725	0.84156	0.1775	0.83736
.1526	.85847	.1576	.85412	.1626	.84993	.1676	.84569	.1726	.84147	.1776	.83728
.1527	.85839	.1577	.85411	.1627	.84985	.1677	.84561	.1727	.84139	.1777	.83719
.1528	.85830	.1578	.85402	.1628	.84976	.1678	.84552	.1728	.84131	.1778	.83711
.1529	.85822	.1579	.85394	.1629	.84968	.1679	.84544	.1729	.84122	.1779	.83703
0.1530	0.85813	0.1580	0.85385	0.1630	0.84959	0.1680	0.84535	0.1730	0.84114	0.1780	0.83694
.1531	.85804	.1581	.85376	.1631	.84951	.1681	.84527	.1731	.84105	.1781	.83686
.1532	.85796	.1582	.85368	.1632	.84942	.1682	.84518	.1732	.84097	.1782	.83678
.1533	.85787	.1583	.85359	.1633	.84934	.1683	.84510	.1733	.84089	.1783	.83669
.1534	.85779	.1584	.85351	.1634	.84925	.1684	.84502	.1734	.84080	.1784	.83661
0.1535	0.85770	0.1585	0.85342	0.1635	0.84917	0.1685	0.84493	0.1735	0.84072	0.1785	0.83652
.1536	.85761	.1586	.85334	.1636	.84908	.1686	.84485	.1736	.84063	.1786	.83644
.1537	.85753	.1587	.85325	.1637	.84900	.1687	.84476	.1737	.84055	.1787	.83636
.1538	.85744	.1588	.85317	.1638	.84891	.1688	.84468	.1738	.84046	.1788	.83627
.1539	.85736	.1589	.85308	.1639	.84883	.1689	.84459	.1739	.84038	.1789	.83619
0.1540	0.85727	0.1590	0.85300	0.1640	0.84874	0.1690	0.84451	0.1740	0.84030	0.1790	0.83611
.1541	.85719	.1591	.85291	.1641	.84866	.1691	.84442	.1741	.84021	.1791	.83602
.1542	.85710	.1592	.85283	.1642	.84857	.1692	.84434	.1742	.84013	.1792	.83594
.1543	.85701	.1593	.85274	.1643	.84849	.1693	.84426	.1743	.84004	.1793	.83586
.1544	.85693	.1594	.85266	.1644	.84840	.1694	.84417	.1744	.83996	.1794	.83577
0.1545	0.85684	0.1595	0.85257	0.1645	0.84832	0.1695	0.84409	0.1745	0.83988	0.1795	0.83569
.1546	.85676	.1596	.85248	.1646	.84823	.1696	.84400	.1746	.83979	.1796	.83560
.1547	.85667	.1597	.85240	.1647	.84815	.1697	.84392	.1747	.83971	.1797	.83552
.1548	.85659	.1598	.85231	.1648	.84806	.1698	.84383	.1748	.83962	.1798	.83544
.1549	.85650	.1599	.85223	.1649	.84798	.1699	.84375	.1749	.83954	.1799	.83535

TABLE A-1.—Concluded.

x	e^{-x}	x	e^{-x}	x	e^{-x}	x	e^{-x}
0.1800	0.83527	0.1850	0.83110	0.1900	0.82696	0.1950	0.82283
.1801	.83519	.1851	.83102	.1901	.82688	.1951	.82275
.1802	.83510	.1852	.83094	.1902	.82679	.1952	.82267
.1803	.83502	.1853	.83085	.1903	.82671	.1953	.82259
.1804	.83494	.1854	.83077	.1904	.82663	.1954	.82251
0.1805	0.83485	0.1855	0.83069	0.1905	0.82655	0.1955	0.82242
.1806	.83477	.1856	.83061	.1906	.82646	.1956	.82234
.1807	.83469	.1857	.83052	.1907	.82638	.1957	.82226
.1808	.83460	.1858	.83044	.1908	.82630	.1958	.82218
.1809	.83452	.1859	.83036	.1909	.82622	.1959	.82209
0.1810	0.83444	0.1860	0.83027	0.1910	0.82613	0.1960	0.82201
.1811	.83435	.1861	.83019	.1911	.82605	.1961	.82193
.1812	.83427	.1862	.83017	.1912	.82597	.1962	.82185
.1813	.83419	.1863	.83002	.1913	.82588	.1963	.82177
.1814	.83410	.1864	.82994	.1914	.82580	.1964	.82168
0.1815	0.83402	0.1865	0.82986	0.1915	0.82572	0.1965	0.82160
.1816	.83393	.1866	.82978	.1916	.82564	.1966	.82152
.1817	.83385	.1867	.82969	.1917	.82555	.1967	.82144
.1818	.83377	.1868	.82961	.1918	.82547	.1968	.82135
.1819	.83368	.1869	.82953	.1919	.82539	.1969	.82127
0.1820	0.83360	0.1870	0.82944	0.1920	0.82531	0.1970	0.82119
.1821	.83352	.1871	.82936	.1921	.82522	.1971	.82111
.1822	.83343	.1872	.82928	.1922	.82514	.1972	.82103
.1823	.83335	.1873	.82919	.1923	.82506	.1973	.82094
.1824	.83327	.1874	.82911	.1924	.82498	.1974	.82086
0.1825	0.83318	0.1875	0.82903	0.1925	0.82489	0.1975	0.82078
.1826	.83310	.1876	.82895	.1926	.82481	.1976	.82070
.1827	.83302	.1877	.82886	.1927	.82473	.1977	.82062
.1828	.83293	.1878	.82878	.1928	.82465	.1978	.82053
.1829	.83285	.1879	.82870	.1929	.82456	.1979	.82045
0.1830	0.83277	0.1880	0.82861	0.1930	0.82448	0.1980	0.82037
.1831	.83268	.1881	.82853	.1931	.82440	.1981	.82029
.1832	.83260	.1882	.82845	.1932	.82432	.1982	.82021
.1833	.83252	.1883	.82837	.1933	.82423	.1983	.82012
.1834	.83244	.1884	.82828	.1934	.82415	.1984	.82004
0.1835	0.83235	0.1885	0.82820	0.1935	0.82407	0.1985	0.81996
.1836	.83227	.1886	.82812	.1936	.82399	.1986	.81988
.1837	.83219	.1887	.82803	.1937	.83391	.1987	.81980
.1838	.83210	.1888	.82795	.1938	.82382	.1988	.81971
.1839	.83202	.1889	.82787	.1939	.82374	.1989	.81963
0.1840	0.83194	0.1890	0.82779	0.1940	0.82366	0.1990	0.81955
.1841	.83185	.1891	.82770	.1941	.82358	.1991	.81947
.1842	.83177	.1892	.82762	.1942	.82349	.1992	.81939
.1843	.83169	.1893	.82754	.1943	.82341	.1993	.81930
.1844	.83160	.1894	.82746	.1944	.82333	.1994	.81922
0.1845	0.83152	0.1895	0.82737	0.1945	0.82325	0.1995	0.81914
.1846	.83144	.1896	.82729	.1946	.82316	.1996	.81906
.1847	.83135	.1897	.82721	.1947	.82308	.1997	.81898
.1848	.83127	.1898	.82712	.1948	.82300	.1998	.81889
.1849	.83119	.1899	.82704	.1949	.82392	.1999	.81881

TABLE A-2.—TOLERANCE FACTORS FOR OBSERVED MTBF[a]

Confidence level, percent	Number of observed failures															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
99	4.6	6.6	8.4	10.1	11.6	13.1	14.6	16.0	17.4	18.7	20.2	21.5	22.8	24.1	25.4	26.8
95	3.0	4.7	6.3	7.8	9.1	10.5	11.8	13.1	14.4	15.7	17.0	18.2	19.4	20.7	21.9	23.1
90	2.3	3.9	5.3	6.7	8.0	9.2	10.5	11.7	13.0	14.2	15.4	16.6	17.8	19.0	20.2	21.3
80	1.6	3.0	4.3	5.5	6.7	7.9	9.0	10.2	11.4	12.5	13.7	14.8	15.9	17.0	18.1	19.2
70	1.2	2.4	3.6	4.8	5.9	7.0	8.1	9.2	10.3	11.4	12.5	13.5	14.6	15.7	16.8	17.8
60	.9	2.0	3.1	4.2	5.2	6.3	7.4	8.4	9.4	10.5	11.5	12.5	13.6	14.7	15.7	16.7
50	.7	1.7	2.7	3.7	4.7	5.7	6.7	7.7	8.7	9.7	10.7	11.7	12.7	13.7	14.7	15.7

[a]To use this table

1. Calculate total test hours, $T = \sum_{i=1}^{n} N_i t_i$ where N is the i^{th} unit tested, t_i is the test time of N_i, and n is the total number of units tested.
2. Enter table under number of observed failures at desired confidence level to find tolerance factor.
3. Lower confidence limit of MTBF = T/(Tolerance factor).

TABLE A-3.—SAFETY MARGINS AT 99-PERCENT CONFIDENCE LEVEL

(a) Sample sizes 5 to 12

Safety margin, S_M	Probability, P_x	Sample size, N							
		5	6	7	8	9	10	11	12
−5.0	0	−2.6271	−2.7679	−2.8843	−2.9789	−3.0590	−3.1327	−3.1958	−3.2521
−4.0	0	−2.0487	−2.1655	−2.2612	−2.3404	−2.4052	−2.4667	−2.5188	−2.5652
−3.0	.0013	−1.4523	−1.5466	−1.6226	−1.6880	−1.7376	−1.7878	−1.8294	−1.8664
−2.0	.0227	.8028	.8810	.9415	.9923	−1.0351	−1.0740	−1.1071	−1.1364
−1.0	.1586	.0434	.0500	.1235	.1762	.2227	.2579	.2893	.3168
−0	.5000	1.6808	1.3681	1.1900	1.0602	.9617	.8914	.8320	.7833
.1	.5398	1.9138	1.5664	1.3628	1.2168	1.1126	1.0351	.9703	.9175
.2	.5792	2.1557	1.7665	1.5439	1.3850	1.2706	1.1844	1.1137	1.0563
.3	.6179	2.4041	1.9747	1.7328	1.5608	1.4352	1.3389	1.2617	1.1994
.4	.6554	2.6582	2.1986	1.9285	1.7380	1.6061	1.4975	1.4138	1.3463
.5	.6914	2.9406	2.4294	2.1304	1.9206	1.7775	1.6602	1.5697	1.4970
.6	.7257	3.2293	2.6662	2.3378	2.1082	1.9522	1.8270	1.7295	1.6512
.7	.7580	3.5232	2.9083	2.5500	2.3002	2.1309	1.9977	1.8927	1.8085
.8	.7881	3.8217	3.1551	2.7665	2.4961	2.3133	2.1719	2.0591	1.9689
.9	.8159	4.1244	3.4059	2.9869	2.6956	2.4989	2.3493	2.2285	2.1320
1.0	.8413	4.4425	3.6604	3.2107	2.8988	2.6875	2.5295	2.4005	2.2975
1.1	.8643	4.7756	3.9183	3.4375	3.1115	2.8846	2.7118	2.5745	2.4650
1.2	.8849	5.1124	4.1791	3.6672	3.3269	3.0842	2.8962	2.7506	2.6344
1.3	.9031	5.4524	4.4467	3.9006	3.5445	3.2860	3.0827	2.9285	2.8056
1.4	.9192	5.7952	4.7243	4.1431	3.7582	3.4851	3.2713	3.1083	2.9785
1.5	.9331	6.1405	5.0042	4.3877	3.9736	3.6855	3.4616	3.2897	3.1528
1.6	.9452	6.4881	5.2861	4.6340	4.1908	3.8874	3.6533	3.4724	3.3284
1.7	.9554	6.8377	5.5698	4.8820	4.4094	4.0907	3.8463	3.6563	3.5051
1.8	.9640	7.1891	5.8550	5.1279	4.6311	4.2953	4.0404	3.8412	3.6828
1.9	.9712	7.5422	6.1417	5.3723	4.8570	4.5010	4.2353	4.0269	3.8613
2.0	.9772	7.8966	6.4295	5.6180	5.0840	4.7077	4.4310	4.2135	4.0405
2.1	.9821	8.2524	6.7186	5.8647	5.3119	4.9153	4.6277	4.4008	4.2205
2.2	.9860	8.6094	7.0086	6.1125	5.5406	5.1238	4.8251	4.5889	4.4012
2.3	.9892	8.9675	7.2996	6.3612	5.7701	5.3330	5.0232	4.7776	4.5825
2.4	.9918	9.3265	7.5914	6.6107	6.0003	5.5429	5.2219	4.9670	4.7644
2.5	.9937	9.6865	7.8840	6.8609	6.2311	5.7534	5.4212	5.1568	4.9468
2.6	.9953	1.0472	8.1772	7.1119	6.4624	5.9646	5.6210	5.3472	5.1296
2.7	.9965	1.4011	8.4694	7.3635	6.6943	6.1762	5.8213	5.5380	5.3129
2.8	.9974	1.7549	8.7588	7.6191	6.9248	6.3894	6.0221	5.7292	5.4965
2.9	.9981	11.1094	9.0488	7.8753	7.1549	6.6040	6.2232	5.9207	5.6804
3.0	.9986	11.4647	9.3395	8.1319	7.3855	6.8191	6.4247	6.1126	5.8647
3.1	.9990	11.8207	9.6307	8.3889	7.6165	7.0345	6.6266	6.3047	6.0492
3.2	.9993	12.1773	9.9223	8.6463	7.8479	7.2502	6.8287	6.4972	6.2340
3.3	.9995	12.5345	1.2145	8.9040	8.0796	7.4662	7.0312	6.6900	6.4191
3.4	.9996	12.8922	1.5070	9.1620	8.3117	7.6825	7.2339	6.8830	6.6044
3.5	.9997	13.2505	1.8000	9.4203	8.5440	7.8990	7.4368	7.0762	6.7899
3.6	.9998	13.6092	11.0933	9.6789	8.7767	8.1157	7.6400	7.2697	6.9756
3.7	.9998	13.9684	11.3870	9.9377	9.0096	8.3326	7.8435	7.4633	7.1616

TABLE A-3.—Continued.

(a) Concluded.

Safety margin, S_M	Probability, P_x	Sample size, N							
		5	6	7	8	9	10	11	12
3.8	0.9999	14.3280	11.6809	10.1968	9.2427	8.5497	8.0471	7.6572	7.3477
3.9		14.6880	11.9752	10.4560	9.4761	8.7671	8.2500	7.8512	7.5340
4.0		15.0488	12.2698	10.7155	9.7097	8.9845	8.4548	8.0454	7.7204
4.1		15.4090	12.5646	10.9751	9.9435	9.2022	8.6590	8.2397	7.9069
4.2		15.7700	12.8597	11.2350	10.1775	9.4200	8.8632	8.4342	8.0937
4.3		16.1313	13.1550	11.4950	10.4116	9.6379	9.0677	8.6288	8.2805
4.4		16.4929	13.4505	11.7551	10.6459	9.8559	9.2722	8.8235	8.4674
4.5		16.8547	13.7463	12.0154	10.8804	10.0741	9.4769	9.0184	8.6545
4.6		17.2168	14.0422	12.2758	11.1150	10.2924	9.6817	9.2134	8.8417
4.7		17.5792	14.3383	12.5364	11.3497	10.5108	9.8866	9.4084	9.0289
4.8		17.9417	14.6346	12.7970	11.5846	10.7293	10.0917	9.6036	9.2163
4.9	1.0000	18.3045	14.9310	13.0578	11.8196	10.9479	10.2968	9.7989	9.4038
5.0		18.6674	15.2276	13.3187	12.0547	11.1666	10.5020	9.9942	9.5913
5.1		19.0306	15.5243	13.5797	12.2900	11.3854	10.7073	10.1896	9.7789
5.2		19.3939	15.8212	13.8408	12.5253	11.6043	10.9127	10.3851	9.9666
5.3		19.7574	16.1182	14.1020	12.7607	11.8232	11.1181	10.5807	10.1543
5.4		20.1210	16.4153	14.3632	12.9962	12.0422	11.3237	10.7764	10.3422
5.5		20.4848	16.7125	14.6246	13.2318	12.2613	11.5293	10.9721	10.5300
5.6		20.8488	17.0099	14.8860	13.4675	12.4804	11.7350	11.1679	10.7180
5.7		21.2129	17.3073	15.1475	13.7033	12.6996	11.9407	11.3638	10.9060
5.8		21.5771	17.6049	15.4091	13.9391	12.9189	12.1465	11.5597	11.0941
5.9		21.9414	17.9025	15.6707	14.1751	13.1382	12.3524	11.7556	11.2822
6.0		22.3059	18.2003	15.9324	14.4110	13.3576	12.5583	11.9516	11.4703
6.1		22.6705	18.4981	16.1941	14.6471	13.5770	12.7643	12.1477	11.6585
6.2		23.0351	18.7960	16.4560	14.8832	13.7965	12.9703	12.3438	11.8468
6.3		23.3999	19.0940	16.7178	15.1194	14.0160	13.1763	12.5400	12.0351
6.4		23.7648	19.3921	16.9797	15.3556	14.2356	13.3825	12.7362	12.2234
6.5		24.1298	19.6902	17.2417	15.5919	14.4552	13.5886	12.9324	12.4118
6.6		24.4948	19.9884	17.5037	15.8282	14.6748	13.7948	13.1287	12.6002
6.7		24.8600	20.2867	17.7658	16.0646	14.8945	14.0011	13.3250	12.7887
6.8		25.2252	20.5850	18.0279	16.3011	15.1143	14.2074	13.5214	12.9771
6.9		25.5905	20.8834	18.2901	16.5375	15.3340	14.4137	13.7177	13.1657
7.0		25.9559	21.1819	18.5523	16.7741	15.5538	14.6200	13.9142	13.3542
7.1		26.3214	21.4804	18.8145	17.0106	15.7736	14.8264	14.1106	13.5428
7.2		26.6869	21.7789	19.0768	17.2472	15.9935	15.0328	14.3071	13.7314
7.3		27.0525	22.0776	19.3391	17.4839	16.2134	15.2393	14.5036	13.9200
7.4		27.4182	22.3762	19.6014	17.7206	16.4333	15.4458	14.7002	14.1087
7.5		27.7839	22.6749	19.8638	17.9573	16.6533	15.6523	14.8967	14.2974
7.6		28.1497	22.9737	20.1262	18.1940	16.8732	15.8588	15.0933	14.4861
7.7		28.5155	23.2725	20.3886	18.4308	17.0932	16.0654	15.2900	14.6748
7.8		28.8814	23.5714	20.6511	18.6676	17.3133	16.2720	15.4866	14.8636
7.9		29.2474	23.8702	20.9136	18.9045	17.5333	16.4786	15.6833	15.0524
8.0		29.6134	24.1692	21.1761	19.1414	17.7534	16.6852	15.8800	15.2412

TABLE A-3.—Continued.

(b) Sample sizes 13 to 20

Safety margin, S_M	Probability, P_x	Sample size, N							
		13	14	15	16	17	18	19	20
−5.0	0	−3.3027	−3.3485	−3.3903	−3.4287	−3.4642	−3.4972	−3.5280	−3.5567
−4.0	0	−2.6069	−2.6447	−2.6792	−2.7109	−2.7401	−2.7673	−2.7927	−2.8163
−3.0	.0013	−1.8997	−1.9299	−1.9573	−1.9826	−2.0058	−2.0275	−2.0477	−2.0666
−2.0	.0227	−1.1628	−1.1866	−1.2083	−1.2281	−1.2464	−1.2633	−1.2791	−1.2937
−1.0	.1586	−.3411	−.3628	−.3823	−.4000	−.4162	−.4309	−.4445	−.4571
0	.5000	.7424	.7074	.6770	.6503	.6265	.6049	.5854	.5677
.1	.5398	.8733	.8357	.8031	.7745	.7491	.7260	.7052	.6863
.2	.5792	1.0085	.9679	.9328	.9022	.8751	.8503	.8281	.8079
.3	.6179	1.1477	1.1039	1.0662	1.0332	1.0042	.9777	.9540	.9325
.4	.6554	1.2905	1.2433	1.2028	1.1675	1.1364	1.1081	1.0827	1.0598
.5	.6914	1.4369	1.3862	1.3428	1.3049	1.2717	1.2414	1.2143	1.1898
.6	.7257	1.5867	1.5324	1.4859	1.4454	1.4099	1.3775	1.3485	1.3224
.7	.7580	1.7393	1.6810	1.6312	1.5880	1.5500	1.5155	1.4846	1.4569
.8	.7881	1.8947	1.8324	1.7792	1.7331	1.6926	1.6558	1.6230	1.5935
.9	.8159	2.0527	1.9862	1.9294	1.8803	1.8372	1.7981	1.7632	1.7319
1.0	.8413	2.2130	2.1422	2.0818	2.0295	1.9838	1.9423	1.9053	1.8721
1.1	.8643	2.3752	2.3000	2.2359	2.1805	2.1320	2.0880	2.0489	2.0138
1.2	.8849	2.5393	2.4596	2.3917	2.3330	2.2817	2.2352	2.1939	2.1568
1.3	.9031	2.7050	2.6208	2.5491	2.4871	2.4329	2.3839	2.3403	2.3012
1.4	.9192	2.8722	2.7833	2.7077	2.6424	2.5853	2.5337	2.4877	2.4466
1.5	.9331	3.0408	2.9472	2.8675	2.7988	2.7388	2.6845	2.6362	2.5930
1.6	.9452	3.2106	3.1122	3.0285	2.9563	2.8932	2.8363	2.7856	2.7403
1.7	.9554	3.3815	3.2782	3.1905	3.1147	3.0486	2.9890	2.9360	2.8885
1.8	.9640	3.5534	3.4452	3.3533	3.2740	3.2049	3.1425	3.0870	3.0374
1.9	.9712	3.7259	3.6129	3.5168	3.4340	3.3617	3.2966	3.2387	3.1870
2.0	.9772	3.8992	3.7812	3.6810	3.5946	3.5193	3.4513	3.3910	3.3370
2.1	.9821	4.0732	3.9503	3.8459	3.7559	3.6774	3.6066	3.5438	3.4876
2.2	.9860	4.2479	4.1200	4.0113	3.9177	3.8360	3.7625	3.6972	3.6388
2.3	.9892	4.4232	4.2902	4.1773	4.0800	3.9952	3.9188	3.8510	3.7904
2.4	.9918	4.5990	4.4610	4.3438	4.2429	4.1548	4.0756	4.0052	3.9424
2.5	.9937	4.7753	4.6322	4.5107	4.4061	4.3149	4.2327	4.1599	4.0947
2.6	.9953	4.9520	4.8038	4.6781	4.5697	4.4753	4.3903	4.3149	4.2474
2.7	.9965	5.1291	4.9759	4.8458	4.7337	4.6361	4.5482	4.4702	4.4005
2.8	.9974	5.3066	5.1482	5.0138	4.8980	4.7971	4.7063	4.6258	4.5538
2.9	.9981	5.4843	5.3208	5.1820	5.0625	4.9584	4.8647	4.7816	4.7073
3.0	.9986	5.6624	5.4937	5.3505	5.2273	5.1198	5.0232	4.9376	4.8610
3.1	.9990	5.8407	5.6668	5.5193	5.3922	5.2816	5.1820	5.0938	5.0149
3.2	.9993	6.0192	5.8402	5.6882	5.5575	5.4435	5.3411	5.2502	5.1690
3.3	.9995	6.1981	6.0138	5.8575	5.7229	5.6056	5.5003	5.4069	5.3234
3.4	.9996	6.3771	6.1876	6.0269	5.8885	5.7680	5.6597	5.5637	5.4779
3.5	.9997	6.5564	6.3617	6.1965	6.0544	5.9305	5.8193	5.7207	5.6325
3.6	.9998	6.7358	6.5359	6.3663	6.2204	6.0932	5.9790	5.8778	5.7873
3.7	.9998	6.9154	6.7103	6.5363	6.3865	6.2560	6.1389	6.0351	5.9423

TABLE A-3.—Continued.

(b) Concluded.

Safety margin, S_M	Probability, P_x	Sample size, N							
		13	14	15	16	17	18	19	20
3.8	0.9999	7.0952	6.8849	6.7064	6.5528	6.4190	6.2990	6.1925	6.0974
3.9		7.2752	7.0596	6.8767	6.7193	6.5822	6.4591	6.3501	6.2526
4.0		7.4553	7.2344	7.0471	6.8859	6.7454	6.6194	6.5077	6.4079
4.1		7.6356	7.4094	7.2176	7.0526	6.9088	6.7798	6.6655	6.5633
4.2		7.8159	7.5845	7.3883	7.2194	7.0723	6.9403	6.8234	6.7189
4.3		7.9964	7.7597	7.5590	7.3863	7.2359	7.1010	6.9814	6.8745
4.4		8.1771	7.9351	7.7299	7.5533	7.3995	7.2617	7.1394	7.0302
4.5		8.3578	8.1105	7.9009	7.7204	7.5633	7.4225	7.2976	7.1860
4.6		8.5386	8.2861	8.0719	7.8877	7.7272	7.5833	7.4558	7.3419
4.7		8.7195	8.4617	8.2431	8.0550	7.8911	7.7443	7.6141	7.4978
4.8		8.9005	8.6374	8.4143	8.2223	8.0551	7.9053	7.7725	7.6538
4.9	1.0000	9.0816	8.8132	8.5856	8.3898	8.2192	8.0664	7.9310	7.8099
5.0		9.2628	8.9890	8.7570	8.5573	8.3834	8.2276	8.0895	7.9660
5.1		9.4440	9.1650	8.9284	8.7249	8.5476	8.3888	8.2480	8.1222
5.2		9.6253	9.3410	9.0999	8.8925	8.7119	8.5501	8.4067	8.2785
5.3		9.8067	9.5170	9.2715	9.0602	8.8763	8.7114	8.5654	8.4348
5.4		9.9881	9.6932	9.4431	9.2280	9.0406	8.8728	8.7241	8.5912
5.5		10.1696	9.8694	9.6148	9.3958	9.2051	9.0843	8.8829	8.7476
5.6		10.3512	10.0456	9.7865	9.5637	9.3696	9.1958	9.0417	8.9040
5.7		10.5328	10.2219	9.9583	9.7316	9.5341	9.3573	9.2006	9.0605
5.8		10.7145	10.3983	10.1302	9.8996	9.6987	9.5189	9.3595	9.2171
5.9		10.8962	10.5746	10.3020	10.0676	9.8634	9.6805	9.5184	9.3736
6.0		11.0779	10.7511	10.4740	10.2356	10.0280	9.8422	9.6774	9.5302
6.1		11.2598	10.9276	10.6459	10.4037	10.1928	10.0039	9.8365	9.6869
6.2		11.4416	11.1041	10.8179	10.5718	10.3575	10.1656	9.9955	9.8435
6.3		11.6235	11.2806	10.9900	10.7400	10.5223	10.3274	10.1546	10.0003
6.4		11.8054	11.4572	11.1621	10.9082	10.6871	10.4892	10.3138	10.1570
6.5		11.9874	11.6339	11.3342	11.0764	10.8520	10.6510	10.4729	10.3138
6.6		12.1694	11.8105	11.5063	11.2447	11.0168	10.8129	10.6321	10.4706
6.7		12.3514	11.9873	11.6785	11.4130	11.1817	10.9748	10.7913	10.6274
6.8		12.5335	12.1640	11.8507	11.5813	11.3467	11.1367	10.9505	10.7842
6.9		12.7156	12.3407	12.0230	11.7496	11.5116	11.2986	11.1098	10.9411
7.0		12.8978	12.5175	12.1952	11.9180	11.6766	11.4606	11.2691	11.0980
7.1		12.0799	12.6944	12.3675	12.0864	11.8416	11.6226	11.4284	11.2549
7.2		13.2621	12.8712	12.5398	12.2548	12.0067	11.7846	11.5877	11.4119
7.3		13.4443	13.0481	12.7122	12.4233	12.1717	11.9466	11.7471	11.5688
7.4		13.6266	13.2250	12.8846	12.5918	12.3368	12.1087	11.9065	11.7253
7.5		13.8088	13.4019	13.0569	12.7603	12.5019	12.2708	12.0659	11.8823
7.6		13.9911	13.5788	13.2294	12.9288	12.6671	12.4329	12.2253	12.0398
7.7		14.1734	13.7558	13.4018	13.0973	12.8322	12.5950	12.3847	12.1969
7.8		14.3558	13.9328	13.5742	13.2659	12.9974	12.7571	12.5442	12.3539
7.9		14.5381	14.1098	13.7467	13.4344	13.1626	12.9193	12.7036	12.5110
8.0		14.7205	14.2868	13.9192	13.6030	13.3278	13.0814	12.8631	12.6681

TABLE A-3.—Continued.

(c) Sample sizes 21 to 28

Safety margin, S_M	Probability, P_x	Sample size, N							
		21	22	23	24	25	26	27	28
−5.0	0	−3.5836	−3.6090	−3.6328	−3.6554	−3.6767	−3.6970	−3.7162	−3.7346
−4.0	0	−2.8385	−2.8594	−2.8790	−2.8976	−2.9152	−2.9318	−2.9477	−2.9628
−3.0	.0013	−2.0842	−2.1008	−2.1164	−2.1312	−2.1451	−2.1584	−2.1710	−2.1830
−2.0	.0227	−1.3075	−1.3204	−1.3325	−1.3439	−1.3548	−1.3650	−1.3748	−1.3840
−1.0	.1586	−.4688	−.4797	−.4900	−.4996	−.5087	−.5173	−.5254	−.5331
0	.5000	.5514	.5366	.5228	.5101	.4982	.4872	.4769	.4671
.1	.5398	.6691	.6533	.6387	.6253	.6128	.6011	.5902	.5800
.2	.5792	.7896	.7728	.7574	.7432	.7299	.7176	.7061	.6953
.3	.6179	.9130	.8952	.8788	.8637	.8496	.8366	.8244	.8130
.4	.6554	1.0390	1.0201	1.0026	.9866	.9717	.9579	.9450	.9330
.5	.6914	1.1677	1.1475	1.1289	1.1119	1.0961	1.0814	1.0678	1.0550
.6	.7257	1.2988	1.2773	1.2576	1.2395	1.2227	1.2071	1.1926	1.1791
.7	.7580	1.4318	1.4089	1.3880	1.3687	1.3510	1.3345	1.3191	1.3048
.8	.7881	1.5669	1.5426	1.5204	1.5000	1.4811	1.4637	1.4474	1.4323
.9	.8159	1.7036	1.6779	1.6544	1.6327	1.6128	1.5943	1.5771	1.5611
1.0	.8413	1.8421	1.8149	1.7900	1.7671	1.7460	1.7265	1.7084	1.6914
1.1	.8643	1.9821	1.9533	1.9270	1.9028	1.8806	1.8600	1.8408	1.8230
1.2	.8849	2.1234	2.0930	2.0652	2.0397	2.0163	1.9945	1.9744	1.9556
1.3	.9031	2.2659	2.2339	2.2046	2.1778	2.1531	2.1302	2.1090	2.0892
1.4	.9192	2.4095	2.3758	2.3451	2.3169	2.2909	2.2669	2.2446	2.2238
1.5	.9331	2.5540	2.5187	2.4864	2.4568	2.4296	2.4044	2.3810	2.3592
1.6	.9452	2.6995	2.6624	2.6286	2.5976	2.5690	2.5426	2.5182	2.4954
1.7	.9554	2.8457	2.8069	2.7716	2.7391	2.7093	2.6817	2.6561	2.6322
1.8	.9640	2.9927	2.9522	2.9152	2.8813	2.8501	2.8213	2.7946	2.7697
1.9	.9712	3.1403	3.0980	3.0594	3.0241	2.9915	2.9615	2.9336	2.9077
2.0	.9772	3.2884	3.2443	3.2041	3.1673	3.1334	3.1021	3.0731	3.0461
2.1	.9821	3.4370	3.3912	3.3493	3.3110	3.2758	3.2432	3.2130	3.1849
2.2	.9860	3.5862	3.5385	3.4950	3.4552	3.4186	3.3847	3.3534	3.3242
2.3	.9892	3.7357	3.6862	3.6411	3.5998	3.5618	3.5267	3.4941	3.4638
2.4	.9918	3.8857	3.8344	3.7876	3.7448	3.7053	3.6689	3.6352	3.6038
2.5	.9937	4.0360	3.9829	3.9344	3.8901	3.8492	3.8115	3.7766	3.7441
2.6	.9953	4.1867	4.1317	4.0816	4.0357	3.9934	3.9544	3.9183	3.8847
2.7	.9965	4.3377	4.2808	4.2290	4.1816	4.1379	4.0976	4.0603	4.0255
2.8	.9974	4.4890	4.4302	4.3767	4.3277	4.2826	4.2410	4.2025	4.1666
2.9	.9981	4.6404	4.5798	4.5246	4.4741	4.4276	4.3847	4.3449	4.3079
3.0	.9986	4.7920	4.7296	4.6727	4.6206	4.5727	4.5284	4.4874	4.4493
3.1	.9990	4.9439	4.8796	4.8210	4.7673	4.7180	4.6724	4.6302	4.5909
3.2	.9993	5.0959	5.0297	4.9694	4.9142	4.8634	4.8166	4.7731	4.7327
3.3	.9995	5.2482	5.1801	5.1181	5.0613	5.0091	4.9609	4.9162	4.8747
3.4	.9996	5.4006	5.3306	5.2669	5.2085	5.1549	5.1053	5.0594	5.0168
3.5	.9997	5.5532	5.4813	5.4158	5.3559	5.3008	5.2500	5.2028	5.1590
3.6	.9998	5.7059	5.6321	5.5649	5.5034	5.4469	5.3947	5.3463	5.3014
3.7	.9998	5.8587	5.7831	5.7142	5.6511	5.5931	5.5396	5.4899	5.4438

TABLE A-3.—Continued.

(c) Concluded.

Safety margin, S_M	Probability, P_x	Sample size, N							
		21	22	23	24	25	26	27	28
3.8	0.9999	6.0117	5.9342	5.8635	5.7989	5.7394	5.6845	5.6337	5.5864
3.9		6.1648	6.0854	6.0130	5.9467	5.8858	5.8296	5.7775	5.7291
4.0		6.3180	6.2367	6.1626	6.0947	6.0324	5.9748	5.9215	5.8719
4.1		6.4714	6.3881	6.3122	6.2428	6.1790	6.1201	6.0655	6.0147
4.2		6.6248	6.5396	6.4620	6.3910	6.3257	6.2654	6.2096	6.1577
4.3		6.7788	6.6912	6.6118	6.5392	6.4725	6.4109	6.3538	6.3007
4.4		6.9319	6.8428	6.7618	6.6876	6.6193	6.5564	6.4980	6.4438
4.5		7.0856	6.9946	6.9118	6.8360	6.7663	6.7020	6.6424	6.5870
4.6		7.2393	7.1464	7.0619	6.9844	6.9133	6.8476	6.7868	6.7302
4.7		7.3931	7.2983	7.2120	7.1330	7.0604	6.9933	6.9312	6.8735
4.8		7.5470	7.4503	7.3622	7.2816	7.2075	7.1391	7.0757	7.0168
4.9	1.0000	7.7010	7.6023	7.5125	7.4303	7.3547	7.2849	7.2203	7.1602
5.0		7.8550	7.7544	7.6628	7.5790	7.5019	7.4308	7.3649	7.3037
5.1		8.0090	7.9065	7.8132	7.7278	7.6492	7.5768	7.5096	7.4472
5.2		8.1632	8.0587	7.9636	7.8766	7.7966	7.7227	7.6543	7.5907
5.3		8.3173	8.2110	8.1141	8.0255	7.9440	7.8688	7.7991	7.7343
5.4		8.4716	8.3633	8.2646	8.1744	8.0914	8.0148	7.9439	7.8780
5.5		8.6258	8.5156	8.4152	8.3233	8.2389	8.1609	8.0887	8.0215
5.6		8.7801	8.6680	8.5658	8.4723	8.3864	8.3071	8.2336	8.1653
5.7		8.9345	8.8204	8.7165	8.6214	8.5340	8.4533	8.3785	8.3091
5.8		9.0889	8.9728	8.8671	8.7704	8.6815	8.5995	8.5235	8.4529
5.9		9.2433	9.1253	9.0179	8.9195	8.8292	8.7458	8.6685	8.5967
6.0		9.3978	9.2778	9.1686	9.0687	8.9768	8.8920	8.8135	8.7405
6.1		9.5523	9.4304	9.3194	8.2179	9.1245	8.0384	8.9585	8.8844
6.2		9.7068	9.5830	9.4702	9.3671	9.2722	9.1847	9.1036	9.0283
6.3		9.8614	9.7356	9.6211	9.5163	9.4200	9.3311	9.2487	9.1722
6.4		10.0160	9.8882	9.7720	9.6655	9.5677	9.4775	9.3938	9.3161
6.5		10.1706	10.0409	9.9229	9.8148	9.7155	9.6239	9.5390	9.4601
6.6		10.3252	10.1936	10.0738	9.9641	8.8634	9.7703	9.6842	9.6041
6.7		10.4799	10.3463	10.2247	10.1135	10.0112	9.9168	9.8294	9.7481
6.8		10.6346	10.4991	10.3757	10.2628	10.1591	10.0633	9.9746	9.8921
6.9		10.7893	10.6519	10.5267	10.4122	10.3070	10.2098	10.1198	10.0362
7.0		10.9441	10.8047	10.6777	10.5616	10.4549	10.3563	10.2651	10.1803
7.1		11.0988	10.9575	10.8288	10.7110	10.6028	10.5029	10.4104	10.3244
7.2		11.2536	11.1103	10.9798	10.8605	10.7507	10.6495	10.5557	10.4685
7.3		11.4084	11.2632	11.1309	11.0099	10.8987	10.7961	10.7010	10.6126
7.4		11.5632	11.4160	11.2820	11.1594	11.0467	10.9427	10.8463	10.7567
7.5		11.7181	11.5689	11.4331	11.3089	11.1947	11.0893	10.9916	10.9009
7.6		11.8729	11.7218	11.5843	11.4584	11.3427	11.2359	11.1370	11.0451
7.7		12.0278	11.8748	11.7354	11.6079	11.4907	11.3826	11.2824	11.1893
7.8		12.1827	12.0277	11.8866	11.7575	11.6388	11.5292	11.4278	11.3335
7.9		12.3376	12.1807	12.0378	11.9070	11.7868	11.6759	11.5732	11.4777
8.0		12.4926	12.3337	12.1890	12.0566	11.9349	11.8226	11.7186	11.6219

TABLE A-3.—Continued.

(d) Sample sizes 30 to 100

Safety margin, S_M	Probability, P_x	Sample size, N							
		30	40	50	60	70	80	90	100
−5.0	0	−3.7688	−3.9040	−4.0005	−4.0741	−4.1328	−4.1810	−4.2216	−4.2565
−4.0	0	−2.9910	−3.1021	−3.1815	−3.2420	−3.2901	− 3.3297	−3.3630	−3.3916
−3.0	.0013	−2.2053	−2.2935	−2.2564	−2.4042	−2.4422	− 2.4735	−2.4998	−2.5224
−2.0	.0227	−1.4013	−1.4691	−1.5172	−1.5536	−1.5825	− 1.6063	−1.6262	−1.6432
−1.0	.1586	−.5473	−.6024	−.6406	−.6692	−.6918	− .7101	−.7254	−.7385
0	.5000	.4494	.3835	.3401	.3087	.2846	.2655	.2497	.2364
.1	.5398	.5613	.4924	.4472	.4146	.3897	.3700	.3537	.3401
.2	.5792	.6756	.6033	.5560	.5221	.4963	.4758	.4590	.4449
.3	.6179	.7923	.7162	.6665	.6311	.6042	.5828	.5654	.5508
.4	.6554	.9110	.8308	.7786	.7415	.7134	.6912	.6730	.6579
.5	.6914	1.0318	.9471	.8922	.8533	.8239	.8007	.7818	.7659
.6	.7257	1.1545	1.0650	1.0073	.9664	.9356	.9113	.8915	.8750
.7	.7580	1.2788	1.1844	1.1236	1.0806	1.0483	1.0229	1.0022	.9850
.8	.7881	1.4048	1.3051	1.2411	1.1960	1.1621	1.1355	1.1138	1.0958
.9	.8159	1.5321	1.4270	1.3596	1.3122	1.2767	1.2488	1.2261	1.2073
1.0	.8413	1.6608	1.5500	1.4792	1.4294	1.3922	1.3629	1.3392	1.3195
1.1	.8643	1.7906	1.6740	1.5996	1.5474	1.5084	1.4778	1.4530	1.4324
1.2	.8849	1.9215	1.7989	1.7208	1.6662	1.6253	1.5933	1.5673	1.5458
1.3	.9031	2.0535	1.9247	1.8429	1.7856	1.7429	1.7094	1.6823	1.6598
1.4	.9192	2.1863	2.0513	1.9656	1.9057	1.8610	1.8260	1.7977	1.7742
1.5	.9331	2.3198	2.1784	2.0888	2.0262	1.9795	1.9430	1.9135	1.8890
1.6	.9452	2.4542	2.3062	2.2126	2.1473	2.0986	2.0605	2.0297	2.0042
1.7	.9554	2.5891	2.4346	2.3369	2.2688	2.2180	2.1784	2.1463	2.1198
1.8	.9640	2.7247	2.5635	2.4617	2.3908	2.3379	2.2966	2.2632	2.2356
1.9	.9712	2.8608	2.6928	2.5868	2.5130	2.4580	2.4151	2.3804	2.3517
2.0	.9772	2.9972	2.8225	2.7123	2.6356	2.5784	2.5339	2.4979	2.4681
2.1	.9821	3.1341	2.9525	2.8380	2.7584	2.6991	2.6529	2.6155	2.5846
2.2	.9860	3.2715	3.0829	2.9641	2.8815	2.8201	2.7721	2.7334	2.7014
2.3	.9892	3.4091	3.2136	3.0905	3.0049	2.9412	2.8916	2.8515	2.8184
2.4	.9918	3.5471	3.3445	3.2171	3.1285	3.0626	3.0112	2.9698	2.9355
2.5	.9937	3.6854	3.4757	3.3439	3.2523	3.1842	3.1311	3.0882	3.0528
2.6	.9953	3.8240	3.6072	3.4709	3.3763	3.3059	3.2511	3.2068	3.1702
2.7	.9965	3.9628	3.7389	3.5982	3.5005	3.4278	3.3712	3.3256	3.2878
2.8	.9974	4.1019	3.8707	3.7256	3.6248	3.5499	3.4915	3.4444	3.4055
2.9	.9981	4.2411	4.0028	3.8531	3.7493	3.6720	3.6119	3.5634	3.5233
3.0	.9986	4.3805	4.1349	3.9808	3.8738	3.7943	3.7324	3.6825	3.6412
3.1	.9990	4.5201	4.2673	4.1086	3.9985	3.9167	3.8530	3.8017	3.7592
3.2	.9993	4.6598	4.3997	4.2366	4.1234	4.0393	3.9737	3.9209	3.8773
3.3	.9995	4.7997	4.5323	4.3646	4.2483	4.1619	4.0945	4.0403	3.9954
3.4	.9996	4.9397	4.6650	4.4928	4.3733	4.2845	4.2154	4.1597	4.1137
3.5	.9997	4.0799	4.7979	4.6211	4.4984	4.4073	4.3364	4.2792	4.2320
3.6	.9998	5.2202	4.9308	4.7494	4.6236	4.5302	4.4574	4.3988	4.3503
3.7	.9998	5.3606	5.0638	4.8778	4.7489	4.6531	4.5785	4.5184	4.4688

TABLE A-3.—Concluded.

(d) Concluded.

Safety margin, S_M	Probability, P_s	Sample size, N							
		30	40	50	60	70	80	90	100
3.8	0.9999	5.5011	5.1969	5.0064	4.8742	4.7761	4.6997	4.6381	4.5872
3.9		5.6417	5.3301	5.1350	4.9996	4.8991	4.8209	4.7579	4.7058
4.0		5.7824	5.4634	5.2636	5.1251	5.0223	4.9422	4.8777	4.8243
4.1		5.9232	5.5968	5.3923	5.2506	5.1454	5.0635	4.9975	4.9430
4.2		6.0640	5.7302	5.5211	5.3762	5.2686	5.1849	5.1174	5.0615
4.3		6.2049	5.8637	5.6500	5.5019	5.3919	5.3063	5.2373	5.1803
4.4		6.3459	5.9972	5.7789	5.6275	5.5152	5.4277	5.3573	5.2991
4.5		6.4870	6.1308	5.9078	5.7533	5.6385	5.5492	5.4773	5.4178
4.6		6.6281	6.2645	6.0368	5.8791	5.7619	5.6708	5.5974	5.5367
4.7		6.7693	6.3982	6.1659	6.0049	5.8853	5.7923	5.7174	5.6555
4.8		6.9106	6.5319	6.2949	6.1307	6.0088	5.9139	5.8375	5.7744
4.9	1.0000	7.0518	6.6657	6.4241	6.2566	6.1323	6.0356	5.9577	5.8933
5.0		7.1932	6.7996	6.5532	6.3825	6.2558	6.1572	6.0778	6.0122
5.1		7.3346	6.9334	6.6824	6.5085	6.3794	6.2789	6.1980	6.1311
5.2		7.4760	7.0673	6.8116	6.6345	6.5029	6.4006	6.3182	6.2501
5.3		7.6175	7.2013	6.9409	6.7605	6.6266	6.5224	6.4384	6.3691
5.4		7.7590	7.3353	7.0702	6.8865	6.7502	6.6441	6.5587	6.4881
5.5		7.9005	7.4693	7.1995	7.0126	6.8738	6.7659	6.6790	6.6071
5.6		8.0421	7.6033	7.3288	7.1386	6.9975	6.8877	6.7993	6.7262
5.7		8.1837	7.7374	7.4582	7.2648	7.1212	7.0095	6.9196	6.8452
5.8		8.3254	7.8715	7.5876	7.3909	7.2449	7.1314	7.0399	6.9643
5.9		8.4671	8.0056	7.7170	7.5170	7.3687	7.2532	7.1603	7.0834
6.0		8.6088	8.1398	7.8464	7.6432	7.4924	7.3751	7.2806	7.2025
6.1		8.7505	8.2740	7.9759	7.7694	7.6162	7.4970	7.4010	7.3217
6.2		8.8923	8.4082	8.1054	7.8956	7.7400	7.6189	7.5214	7.4408
6.3		9.0341	8.5424	8.2348	8.0218	7.8638	7.7409	7.6418	7.5600
6.4		9.1759	8.6766	8.3644	8.1481	7.9876	7.8627	7.7622	7.6791
6.5		9.3177	8.8109	8.4939	8.2744	8.1114	7.9847	7.8827	7.7983
6.6		9.4596	8.9452	8.6234	8.4006	8.2353	8.1067	8.0031	7.9175
6.7		9.6015	9.0794	8.7530	8.5269	8.3592	8.2286	8.1236	8.0367
6.8		9.7434	9.2138	8.8826	8.6532	8.4830	8.3506	8.2440	8.1559
6.9		9.8853	9.3481	9.0122	8.7795	8.6069	8.4726	8.3645	8.2751
7.0		10.0272	9.4824	9.1418	8.9059	8.7308	8.5946	8.4850	8.3944
7.1		10.1692	9.6168	9.2714	9.0322	8.8547	8.7167	8.6055	8.5136
7.2		10.3112	9.7512	9.4010	9.1586	8.9786	8.8387	8.7260	8.6329
7.3		10.4532	9.8856	9.5307	9.2849	9.1026	8.9607	8.8465	8.7521
7.4		10.5952	10.0200	9.6603	9.4113	9.2265	8.0828	8.9670	8.8714
7.5		10.7372	10.1544	9.7900	9.5377	9.3505	9.2048	9.0876	8.9907
7.6		10.8792	10.2888	9.9197	9.6641	9.4744	9.3269	9.2081	9.1100
7.7		11.0213	10.4233	10.0494	9.7905	9.5984	9.4490	9.3287	9.2293
7.8		11.1633	10.5577	10.1791	9.9169	9.7224	9.5711	9.4492	9.3486
7.9		11.3054	10.6922	10.3088	10.0433	9.8464	9.6932	9.5698	9.4679
8.0		11.4475	10.8266	10.4385	10.1698	9.9704	9.8153	9.6904	9.5872

TABLE A-4.—SAFETY MARGINS AT 95-PERCENT CONFIDENCE LEVEL

(a) Sample sizes 5 to 12

Safety margin, S_M	Probability, P_x	Sample size, N							
		5	6	7	8	9	10	11	12
−5.0	0	−3.1600	−3.2797	−3.3759	−3.4551	−3.5230	−3.5814	−3.6328	−3.6783
−4.0	0	−2.4898	−2.5882	−2.6674	−2.7326	−2.7884	−2.8364	−2.8787	−2.9161
−3.0	.0013	−1.8066	−1.8847	−1.9477	−1.9995	−2.0438	−2.0819	−2.1155	−2.1453
−2.0	.0227	−1.0897	−1.1507	−1.1999	−1.2402	−1.2748	−1.3044	−1.3304	−1.3535
−1.0	.1586	−.2651	−.3241	−.3691	−.4062	−.4363	−.4625	−.4847	−.5043
0	.5000	.9538	.8223	.7340	.6697	.6197	.5796	.5464	.5184
.1	.5398	1.1123	.9664	.8692	.7989	.7449	.7017	.6661	.6361
.2	.5792	1.2779	1.1159	1.0094	.9328	.8741	.8273	.7889	.7567
.3	.6179	1.4509	1.2710	1.1543	1.0708	1.0071	.9562	.9148	.8803
.4	.6554	1.6309	1.4315	1.3035	1.2118	1.1428	1.0882	1.0436	1.0065
.5	.6914	1.8155	1.5966	1.4564	1.3565	1.2822	1.2231	1.1751	1.1353
.6	.7257	2.0061	1.7662	1.6130	1.5047	1.4247	1.3609	1.3093	1.2666
.7	.7580	2.2023	1.9395	1.7729	1.6562	1.5691	1.5010	1.4456	1.3999
.8	.7881	2.4022	2.1163	1.9362	1.8104	1.7162	1.6434	1.5841	1.5353
.9	.8159	2.6057	2.2960	2.1023	1.9666	1.8658	1.7878	1.7245	1.6724
1.0	.8413	2.8129	2.4788	2.2709	2.1252	2.0174	1.9343	1.8667	1.8112
1.1	.8643	3.0237	2.6647	2.4415	2.2858	2.1710	2.0822	2.0103	1.9514
1.2	.8849	3.2370	2.8528	2.6140	2.4483	2.3263	2.2317	2.1554	2.0930
1.3	.9031	3.4526	3.0427	2.7883	2.6123	2.4830	2.3826	2.3018	2.2358
1.4	.9192	3.6702	3.2340	2.9642	2.7779	2.6408	2.5345	2.4493	2.3796
1.5	.9331	3.8896	3.4268	3.1415	2.9447	2.7998	2.6876	2.5977	2.5243
1.6	.9452	4.1105	3.6211	3.3200	3.1126	2.9598	2.8416	2.7471	2.6699
1.7	.9554	4.3329	3.8165	3.4997	3.2815	3.1208	2.9965	2.8972	2.8163
1.8	.9640	4.5564	4.0134	3.6803	3.4513	3.2826	3.1522	3.0482	2.9633
1.9	.9712	4.7811	4.2113	3.8618	3.6216	3.4451	3.3085	3.1997	3.1110
2.0	.9772	5.0067	4.4101	4.0440	3.7927	3.6083	3.4654	3.3518	3.2592
2.1	.9821	5.2332	4.6097	4.2270	3.9645	3.7721	3.6229	3.5044	3.4079
2.2	.9860	5.4605	4.8099	4.4106	4.1368	3.9364	3.7810	3.6575	3.5570
2.3	.9892	5.6885	5.0108	4.5947	4.3096	4.1013	3.9394	3.8111	3.7066
2.4	.9918	5.9171	5.2122	4.7794	4.4829	4.2665	4.0983	3.9650	3.8565
2.5	.9937	6.1463	5.4142	4.9646	4.6567	4.4322	4.2576	4.1193	4.0068
2.6	.9953	6.3761	5.6166	5.1501	4.8308	4.5982	4.4172	4.2739	4.1574
2.7	.9965	6.6063	5.8193	5.3361	5.0053	4.7646	4.5771	4.4289	4.3083
2.8	.9974	6.8369	6.0223	5.5222	5.1801	4.9311	4.7373	4.5840	4.4594
2.9	.9981	7.0679	6.2256	5.7086	5.3552	5.0978	4.8977	4.7394	4.6107
3.0	.9986	7.2993	6.4292	5.8953	5.5306	5.2647	5.0584	4.8950	4.7622
3.1	.9990	7.5311	6.6332	6.0823	5.7062	5.4319	5.2193	5.0508	4.9139
3.2	.9993	7.7631	6.8374	6.2696	5.8821	5.5993	5.3803	5.2068	5.0658
3.3	.9995	7.9954	7.0418	6.4570	6.0582	5.7668	5.5416	5.3630	5.2179
3.4	.9996	8.2280	7.2465	6.6447	6.2344	5.9346	5.7030	5.5194	5.3701
3.5	.9997	8.4608	7.4514	6.8326	6.4109	6.1026	5.8646	5.6759	5.5225
3.6	.9998	8.6939	7.6565	7.0207	6.5875	6.2707	6.0264	5.8325	5.6750
3.7	.9998	8.9271	7.8618	7.2089	6.7643	6.4389	6.1882	5.9893	5.8277

TABLE A-4.—Continued.

(a) Concluded.

Safety margin, S_M	Probability, P_x	Sample size, N							
		5	6	7	8	9	10	11	12
3.8	0.9999	9.1606	8.0673	7.3973	6.9412	6.6073	6.3502	6.1462	5.9804
3.9		9.3942	8.2729	7.5858	7.1182	6.7758	6.5124	6.3032	6.1333
4.0		9.6280	8.4787	7.7745	7.2954	6.9444	6.6746	6.4603	6.2852
4.1		9.8619	8.6846	7.9633	7.4727	7.1132	6.8369	6.6175	6.4393
4.2		10.0960	8.8906	8.1522	7.6501	7.2820	6.9994	6.7748	6.5924
4.3		10.3302	9.0968	8.3413	7.8276	7.4510	7.1619	6.9322	6.7456
4.4		10.5645	9.3031	8.5304	8.0052	7.6200	7.3245	7.0896	6.8989
4.5		10.7990	9.5095	8.7196	8.1829	7.7891	7.4872	7.2472	7.0523
4.6		11.0336	9.7160	8.9090	8.3606	7.9583	7.6499	7.4048	7.2057
4.7		11.2683	9.9225	9.0984	8.5385	8.1276	7.8128	7.5625	7.3592
4.8		11.5031	10.1292	8.2879	8.7164	8.2969	7.9757	7.7202	7.5128
4.9	1.0000	11.7379	10.3359	9.4775	8.8944	8.4664	8.1386	7.8780	7.6664
5.0		11.9729	10.5428	9.6671	9.0725	8.6358	8.3017	8.0358	7.8200
5.1		12.2080	10.7497	9.8568	9.2506	8.8054	8.4647	8.1938	7.9738
5.2		12.4431	10.9567	10.0466	9.4288	8.9750	8.6279	8.3517	8.1275
5.3		12.6783	11.1637	10.2364	9.6070	9.1446	8.7910	8.5097	8.2813
5.4		12.9136	11.3708	10.4263	9.7853	9.3143	8.9543	8.6678	8.4352
5.5		13.1489	11.5780	10.6163	9.9636	9.4841	9.1176	8.8259	8.5891
5.6		13.3843	11.7852	10.8063	10.1420	9.6539	9.2809	8.9840	8.7430
5.7		13.6198	11.9925	10.9963	10.3205	9.8237	9.4442	9.1422	8.8970
5.8		13.8553	12.1998	11.1865	10.4989	9.9936	9.6076	9.3004	9.0510
5.9		14.0909	12.4072	11.3766	10.6775	10.1635	9.7710	9.4586	9.2050
6.0		14.3265	12.6146	11.5668	10.8560	10.3335	9.9345	9.6169	9.3591
6.1		14.5622	12.8221	11.7570	11.0346	10.5034	10.0980	9.7752	9.5132
6.2		14.7979	13.0296	11.9473	11.2133	10.6735	10.2615	9.9336	9.6673
6.3		15.0337	13.2372	12.1376	11.3919	10.8435	10.4251	10.0919	9.8215
6.4		15.2695	13.4447	12.3279	11.5706	11.0136	10.5887	10.2503	9.9757
6.5		15.5054	13.6524	12.5183	11.7494	11.1837	10.7523	10.4087	10.1299
6.6		15.7413	13.8600	12.7087	11.9281	11.3539	10.9160	10.5672	10.2841
6.7		15.9772	14.0677	12.8992	12.1069	11.5241	11.0796	10.7257	10.4384
6.8		16.2132	14.2755	13.0897	12.2857	11.6943	11.2433	10.8842	10.5927
6.9		16.4492	14.4832	13.2802	12.4646	11.8645	11.4070	11.0427	10.7470
7.0		16.6852	14.6910	13.4707	12.6434	12.0347	11.5708	11.2012	10.9013
7.1		16.9213	14.8988	13.6612	12.8223	12.2050	11.7345	11.3598	11.0556
7.2		17.1574	15.1067	13.8518	13.0013	12.3753	11.8983	11.5183	11.2100
7.3		17.3935	15.3146	14.0424	13.1802	12.5456	12.0621	11.6769	11.3644
7.4		17.6297	15.5225	14.2330	13.3592	12.7160	12.2259	11.8356	11.5187
7.5		17.8659	15.7304	14.4237	13.5381	12.8863	12.3898	11.9942	11.6732
7.6		18.1021	15.9383	14.6144	13.7171	13.0567	12.5536	12.1528	11.8276
7.7		18.3383	16.1463	14.8050	13.8962	13.2271	12.7175	12.3115	11.9820
7.8		18.5746	16.3543	14.9958	14.0752	13.3975	12.8814	12.4702	12.1365
7.9		18.8109	16.5623	15.1865	14.2542	13.5679	13.0453	12.6289	12.2909
8.0		19.0472	16.7703	15.3772	14.4333	13.7384	13.2092	12.7876	12.4454

TABLE A-4.—Continued.

(b) Sample sizes 13 to 20

Safety margin, S_M	Probability, P_x	Sample size, N							
		13	14	15	16	17	18	19	20
−5.0	0	−3.7191	−3.7560	−3.7895	−3.8202	−3.8485	−3.8747	−3.8990	−3.9217
−4.0	0	−2.9497	−2.9800	−3.0076	−3.0329	−3.0561	−3.0777	−3.0977	−3.1164
−3.0	.0013	−2.1719	−2.1960	−2.2179	−2.2380	−2.2564	−2.2735	−2.2894	−2.3042
−2.0	.0227	−1.3741	−1.3927	−1.4097	−1.4251	−1.4394	−1.4525	−1.4647	−1.4761
−1.0	.1586	−.5217	−.5373	−.5514	−.5642	−.5759	−.5866	−.5965	−.6057
−0	.5000	.4943	.4733	.4547	.4382	.4234	.4100	.3978	.3867
.1	.5398	.6104	.5881	.5684	.5510	.5353	.5212	.5084	.4967
.2	.5792	.7293	.7055	.6846	.6661	.6496	.6347	.6211	.6087
.3	.6179	.8509	.8256	.8033	.7837	.7661	.7503	.7359	.7228
.4	.6554	.9751	.9480	.9243	.9034	.8848	.8679	.8527	.8388
.5	.6914	1.1017	1.0727	1.0475	1.0252	1.0054	.9875	.9713	.9566
.6	.7257	1.2306	1.1996	1.1727	1.1490	1.1279	1.1089	1.0918	1.0762
.7	.7580	1.3614	1.3284	1.2997	1.2745	1.2521	1.2319	1.2137	1.1972
.8	.7881	1.4942	1.4591	1.4286	1.4018	1.3780	1.3566	1.3373	1.3197
.9	.8159	1.6286	1.5912	1.5588	1.5304	1.5052	1.4825	1.4620	1.4435
1.0	.8413	1.7647	1.7250	1.6906	1.6605	1.6338	1.6097	1.5880	1.5684
1.1	.8643	1.9021	1.8600	1.8236	1.7917	1.7635	1.7380	1.7151	1.6945
1.2	.8849	2.0407	1.9962	1.9577	1.9240	1.8942	1.8674	1.8432	1.8214
1.3	.9031	2.1806	2.1335	2.0929	2.0574	2.0260	1.9977	1.9723	1.9493
1.4	.9192	2.3213	2.2718	2.2290	2.1916	2.1585	2.1288	2.1020	2.0779
1.5	.9331	2.4630	2.4109	2.3659	2.3265	2.2918	2.2606	2.2325	2.2072
1.6	.9452	2.6055	2.5507	2.5035	2.4622	2.4258	2.3930	2.3636	2.3371
1.7	.9554	2.7487	2.6913	2.6418	2.5986	2.5605	2.5261	2.4954	2.4676
1.8	.9640	2.8926	2.8325	2.7807	2.7355	2.6957	2.6598	2.6277	2.5986
1.9	.9712	3.0370	2.9743	2.9202	2.8730	2.8313	2.7939	2.7604	2.7301
2.0	.9772	3.1820	3.1165	3.0601	3.0108	2.9674	2.9284	2.8935	2.8619
2.1	.9821	3.3274	3.2592	3.2004	3.1491	3.1040	3.0633	3.0270	2.9941
2.2	.9860	3.4733	3.4023	3.3411	3.2878	3.2409	3.1986	3.1608	3.1267
2.3	.9892	3.6196	3.5458	3.4823	3.4269	3.3781	3.3343	3.2950	3.2596
2.4	.9918	3.7662	3.6896	3.6237	3.5662	3.5156	3.4702	3.4295	3.3927
2.5	.9937	3.9131	3.8338	3.7654	3.7059	3.6535	3.6064	3.5642	3.5262
2.6	.9953	4.0604	3.9782	3.9075	3.8458	3.7915	3.7428	3.6991	3.6598
2.7	.9965	4.2079	4.1229	4.0497	3.9860	3.9298	3.8794	3.8343	3.7987
2.8	.9974	4.3557	4.2678	4.1922	4.1263	4.0684	4.0163	3.9697	3.9277
2.9	.9981	4.5036	4.4129	4.3349	4.2669	4.2071	4.1533	4.1053	4.0619
3.0	.9986	4.6518	4.5582	4.4777	4.4076	4.3459	4.2905	4.2409	4.1963
3.1	.9990	4.8001	4.7036	4.6207	4.5485	4.4849	4.4279	4.3768	4.3308
3.2	.9993	4.9486	4.8493	4.7639	4.6895	4.6241	4.5653	4.5128	4.4654
3.3	.9995	5.0973	4.9951	4.9072	4.8307	4.7634	4.7030	4.6489	4.6002
3.4	.9996	5.2461	5.1410	5.0507	4.9720	4.9028	4.8407	4.7851	4.7351
3.5	.9997	5.3950	5.2871	5.1943	5.1135	5.4024	4.9786	4.9215	4.8701
3.6	.9998	5.5441	5.4333	5.3380	5.2550	5.1820	5.1165	5.0580	5.0052
3.7	.9998	5.6933	5.5796	5.4818	5.3967	5.3218	5.2546	5.1945	5.1404

TABLE A–4.—Continued.

(b) Concluded.

Safety margin, S_M	Probability, P_x	Sample size, N							
		13	14	15	16	17	18	19	20
3.8	0.9999	5.8426	5.7260	5.6257	5.5385	5.4617	5.3928	5.3312	5.2757
3.9		5.9920	5.8725	5.7697	5.6803	5.6016	5.5310	5.4679	5.4110
4.0		6.1416	6.0191	5.9139	5.8223	5.7417	5.6694	5.6047	5.5465
4.1		6.2912	6.1658	6.0580	5.9643	5.8818	5.8078	5.7416	5.6820
4.2		6.4408	6.3126	6.2023	6.1064	6.0220	5.9463	5.8785	5.8176
4.3		6.5906	6.4594	6.3467	6.2485	6.1622	6.0848	6.0156	5.9532
4.4		6.7404	6.6063	6.4911	6.3908	6.3025	6.2234	6.1526	6.0889
4.5		6.8903	6.7533	6.6355	6.5331	6.4429	6.3621	6.2989	6.2247
4.6		7.0403	6.9003	6.7801	6.6754	6.5834	6.5008	6.4270	6.3605
4.7		7.1903	7.0474	6.9247	6.8178	6.7239	6.6396	6.5642	6.4963
4.8		7.3404	7.1946	7.0693	6.9603	6.8644	6.7784	6.7015	6.6322
4.9	1.0000	7.4906	7.3418	7.2140	7.1028	7.0050	6.9173	6.8388	6.7682
5.0		7.6408	7.4891	7.3588	7.2454	7.1456	7.0562	6.9762	6.9042
5.1		7.7910	7.6364	7.5035	7.3879	7.2863	7.1951	7.1136	7.0402
5.2		7.9413	7.7837	7.6484	7.5306	7.4270	7.3341	7.2510	7.1763
5.3		8.0916	7.9311	7.7932	7.6733	7.5677	7.4731	7.3885	7.3124
5.4		8.2420	8.0786	7.9381	7.8160	7.7085	7.6122	7.5260	7.4485
5.5		8.3924	8.2260	8.0831	7.9587	7.8494	7.7513	7.6636	7.5846
5.6		8.5429	8.3735	8.2281	8.1015	7.9902	7.8904	7.8012	7.7208
5.7		8.6934	8.5211	8.3731	8.2443	8.1311	8.0296	7.9388	7.8571
5.8		8.8439	8.6687	8.5181	8.3872	8.2720	8.1687	8.0764	7.9933
5.9		8.9944	8.8163	8.6632	8.5300	8.4129	8.3080	8.2141	8.1296
6.0		9.1450	8.9639	8.8083	8.6729	8.5539	8.4472	8.3518	8.2659
6.1		9.2956	9.1115	8.9534	8.8159	8.6949	8.5864	8.4895	8.4022
6.2		9.4463	9.2592	9.0986	8.9588	8.8359	8.7257	8.6272	8.5385
6.3		9.5969	9.4069	9.2438	9.1018	8.9770	8.8650	8.7650	8.6749
6.4		9.7476	9.5547	9.3890	9.2448	9.1180	9.0044	8.9028	8.8113
6.5		9.8983	9.7024	9.5342	9.3878	9.2591	9.1437	9.0406	8.9477
6.6		10.0491	9.8502	9.6794	9.5309	9.4002	9.2831	9.1784	9.0841
6.7		10.1998	9.9980	9.8247	9.6739	9.5413	9.4225	9.3162	9.2205
6.8		10.3506	10.1458	9.9700	9.8170	9.6825	9.5619	9.4540	9.3570
6.9		10.5014	10.2937	10.1153	9.9601	9.8236	9.7013	9.5919	9.4935
7.0		10.6522	10.4416	10.2606	10.1032	9.9648	9.8407	9.7298	9.6299
7.1		10.8031	10.5894	10.4059	10.2463	10.1060	9.9802	9.8677	9.7664
7.2		10.9539	10.7373	10.5513	10.3895	10.2472	10.1196	10.0056	9.9030
7.3		11.1048	10.8852	10.6967	10.5326	10.3884	10.2591	10.1435	10.0395
7.4		11.2557	11.0332	10.8421	10.6758	10.5296	10.3986	10.2815	10.1760
7.5		11.4066	11.1811	10.9875	10.8190	10.6709	10.5381	10.4194	10.3126
7.6		11.5575	11.3291	11.1329	10.9622	10.8122	10.6776	10.5574	10.4491
7.7		11.7084	11.4770	11.2783	11.1054	10.9534	10.8172	10.6954	10.5857
7.8		11.8594	11.6250	11.4237	11.2487	11.0947	10.9567	10.8334	10.7223
7.9		12.0104	11.7730	11.5692	11.3919	11.2360	11.0963	10.9714	10.8589
8.0		12.1613	11.9210	11.7147	11.5352	11.3773	11.2358	11.1094	10.9955

TABLE A-4.—Continued.

(c) Sample sizes 21 to 28

Safety margin, S_M	Probability, P_x	Sample size, N							
		21	22	23	24	25	26	27	28
−5.0	0	−3.9429	−3.9629	−3.9816	−3.9993	−4.0160	−4.0318	−4.0468	−4.0612
−4.0	0	−3.1338	−3.1502	−3.1656	−3.1801	−3.1938	−3.2068	−3.2192	−3.2310
−3.0	.0013	−2.3180	−2.3310	−2.3432	−2.3547	−2.3655	−2.3759	−2.3856	−2.3949
−2.0	.0227	−1.4867	−1.4966	−1.5060	−1.5148	−1.5231	−1.5310	−1.5385	−1.5456
−1.0	.1586	−.6142	−.6222	−.6297	−.6368	−.6434	−.6497	−.6556	−.6613
−0	.5000	.3764	.3669	.3581	.3499	.3422	.3350	.3283	.3219
.1	.5398	.4859	.4759	.4667	.4581	.4501	.4426	.4356	.4289
.2	.5792	.5974	.5869	.5772	.5682	.5598	.5519	.5446	.5376
.3	.6179	.7108	.6998	.6896	.6801	.6713	.6630	.6553	.6480
.4	.6554	.8261	.8145	.8037	.7937	.7844	.7757	.7676	.7599
.5	.6914	.9432	.9308	.9194	.9089	.8991	.8899	.8813	.8733
.6	.7257	1.0619	1.0489	1.0368	1.0256	1.0153	1.0056	.9966	.9881
.7	.7580	1.1821	1.1683	1.1555	1.1437	1.1328	1.1226	1.1130	1.1041
.8	.7881	1.3038	1.2891	1.2756	1.2632	1.2516	1.2408	1.2307	1.2213
.9	.8159	1.4266	1.4111	1.3968	1.3837	1.3715	1.3601	1.3495	1.3396
1.0	.8413	1.5506	1.5343	1.5192	1.5053	1.4925	1.4805	1.4693	1.4588
1.1	.8643	1.6756	1.6584	1.6425	1.6279	1.6144	1.6017	1.5900	1.5790
1.2	.8849	1.8016	1.7834	1.7667	1.7513	1.7371	1.7238	1.7114	1.6999
1.3	.9031	1.9284	1.9093	1.8918	1.8756	1.8606	1.8466	1.8337	1.8215
1.4	.9192	2.0560	2.0359	2.0175	2.0005	1.9848	1.9701	1.9565	1.9438
1.5	.9331	2.1842	2.1631	2.1438	2.1260	2.1095	2.0942	2.0799	2.0666
1.6	.9452	2.3130	2.2910	2.2707	2.2521	2.2349	2.2188	2.2039	2.1899
1.7	.9554	2.4424	2.4194	2.3982	2.3787	2.3607	2.3439	2.3283	2.3138
1.8	.9640	2.5723	2.5482	2.5262	2.5058	2.4870	2.4695	2.4532	2.4380
1.9	.9712	2.7026	2.6775	2.6545	2.6333	2.6137	2.5955	2.5785	2.5626
2.0	.9772	2.8333	2.8072	2.7832	2.7611	2.7407	2.7217	2.7041	2.6876
2.1	.9821	2.9644	2.9372	2.9123	2.8893	2.8681	2.8483	2.8300	2.8128
2.2	.9860	3.0958	3.0675	3.0416	3.0178	2.9957	2.9753	2.9562	2.9384
2.3	.9892	3.2275	3.1982	3.1713	3.1465	3.1237	3.1024	3.0827	3.0642
2.4	.9918	3.3594	3.3291	3.3012	3.2756	3.2519	3.2298	3.2094	3.1902
2.5	.9937	3.4917	3.4602	3.4314	3.4048	3.3803	3.3575	3.3363	3.3165
2.6	.9953	3.6241	3.5916	3.5617	3.5343	3.5089	3.4853	3.4634	3.4429
2.7	.9965	3.7568	3.7231	3.6923	3.6639	3.6377	3.6134	3.5907	3.5695
2.8	.9974	3.8896	3.8549	3.8231	3.7938	3.7667	3.7416	3.7182	3.6963
2.9	.9981	4.0226	3.9868	3.9540	3.9237	3.8958	3.8699	3.8458	3.8233
3.0	.9986	4.1558	4.1188	4.0850	4.0538	4.0251	3.9984	3.9735	3.9503
3.1	.9990	4.2891	4.2510	4.2162	4.1841	4.1544	4.1269	4.1013	4.0775
3.2	.9993	4.4225	4.3833	4.3475	4.3145	4.2839	4.2557	4.2293	4.2047
3.3	.9995	4.5560	4.5158	4.4789	4.4449	4.4136	4.3845	4.3574	4.3321
3.4	.9996	4.6897	4.6483	4.6104	4.5755	4.5433	4.5134	4.4856	4.4596
3.5	.9997	4.8235	4.7810	4.7420	4.7062	4.6731	4.6424	4.6138	4.5872
3.6	.9998	4.9573	4.9137	4.8738	4.8370	4.8030	4.7715	4.7422	4.7148
3.7	.9998	5.0913	5.0466	5.0056	4.9679	4.9330	4.9007	4.8707	4.8426

TABLE A–4.—Continued.

(c) Concluded.

Safety margin, S_M	Probability, P_x	Sample size, N							
		21	22	23	24	25	26	27	28
3.8	0.9999	5.2254	5.1795	5.1375	5.0988	5.0631	5.0300	4.9992	4.9704
3.9		5.3595	5.3125	5.2695	5.2298	5.1933	5.1593	5.1278	5.0983
4.0		5.4937	5.4456	5.4015	5.3609	5.3235	5.2887	5.2564	5.2262
4.1		5.6280	5.5787	5.5336	5.4921	5.4538	5.4182	5.3851	5.3542
4.2		5.7623	5.7119	5.6658	5.6233	5.5841	5.5477	5.5139	5.4823
4.3		5.8967	5.8452	5.7980	5.7546	5.7145	5.6773	5.6427	5.6104
4.4		6.0311	5.9785	5.9303	5.8859	5.8449	5.8069	5.7716	5.7386
4.5		6.1657	6.1119	6.0626	6.0173	5.9754	5.9366	5.9005	5.8668
4.6		6.3002	6.2453	6.1950	6.1487	6.1060	6.0663	6.0295	5.9951
4.7		6.4348	6.3788	6.3274	6.2802	6.2366	6.1961	6.1585	6.1234
4.8		6.5695	6.5123	6.4599	6.4117	6.3672	6.3259	6.2875	6.2517
4.9	1.0000	6.7042	6.6458	6.5924	6.5433	6.4979	6.4558	6.4166	6.3801
5.0		6.8389	6.7794	6.7250	6.6748	6.6286	6.5856	6.5457	6.5085
5.1		6.9737	6.9131	6.8575	6.8065	6.7593	6.7156	6.6749	6.6369
5.2		7.1085	7.0467	6.9902	6.9381	6.8901	6.8455	6.8041	6.7654
5.3		7.2433	7.1804	7.1228	7.0698	7.0209	6.9755	6.9333	6.8939
5.4		7.3782	7.3141	7.2555	7.2015	7.1517	7.1055	7.0625	7.0224
5.5		7.5131	7.4479	7.3882	7.3333	7.2826	7.2355	7.1918	7.1510
5.6		7.6480	7.5817	7.5209	7.4651	7.4134	7.3656	7.3211	7.2795
5.7		7.7830	7.7155	7.6537	7.5969	7.5443	7.4957	7.4504	7.4081
5.8		7.9180	7.8493	7.7865	7.7287	7.6753	7.6258	7.5797	7.5368
5.9		8.0530	7.9832	7.9193	7.8605	7.8062	7.7559	7.7091	7.6654
6.0		8.1880	8.1171	8.0521	7.9924	7.9372	7.8861	7.8385	7.7941
6.1		8.3231	8.2510	8.1850	8.1243	8.0682	8.0162	7.9679	7.9228
6.2		8.4582	8.3849	8.3179	8.2562	8.1992	8.1464	8.0973	8.0515
6.3		8.5933	8.5189	8.4508	8.3881	8.3303	8.2766	8.2267	8.1802
6.4		8.7284	8.6529	8.5837	8.5201	8.4613	8.4069	8.3562	8.3089
6.5		8.8635	8.7868	8.7166	8.6520	8.5924	8.5371	8.4857	8.4377
6.6		8.9987	8.9208	8.8496	8.7840	8.7235	8.6674	8.6152	8.5664
6.7		9.1338	9.0549	8.9825	8.9160	8.8546	8.7976	8.7447	8.6952
6.8		9.2690	9.1889	9.1155	9.0480	8.9857	8.9279	8.8742	8.8240
6.9		9.4042	9.3230	9.2485	9.1801	9.1168	9.0582	9.0037	8.9528
7.0		9.5395	9.4570	9.3815	9.3121	9.2480	9.1885	9.1333	9.0817
7.1		9.6747	9.5911	9.5146	9.4442	9.3791	9.3189	9.2628	9.2105
7.2		9.8099	9.7252	9.6476	9.5762	9.5103	9.4492	9.3924	9.3394
7.3		9.9452	9.8593	9.7807	9.7083	9.6415	9.5796	9.5220	9.4682
7.4		10.0805	9.9934	9.9137	9.8404	9.7727	9.7099	9.6516	9.5971
7.5		10.2158	10.1276	10.0468	9.9725	9.9039	9.8403	9.7812	9.7260
7.6		10.3511	10.2617	10.1799	10.1046	10.0351	9.9707	9.9108	9.8549
7.7		10.4864	10.3959	10.3130	10.2368	10.1664	10.1011	10.0404	9.9838
7.8		10.6217	10.5300	10.4461	10.3689	10.2976	10.2315	10.1700	10.1127
7.9		10.7570	10.6642	10.5792	10.5010	10.4288	10.3619	10.2997	10.2416
8.0		10.8924	10.7984	10.7123	10.6332	10.5601	10.4923	10.4293	10.3705

TABLE A–4.—Continued.

(d) Sample sizes 30 to 100

Safety margin, S_M	Probability, P_s	Sample size, N							
		30	40	50	60	70	80	90	100
−5.0	0	−4.0878	−4.1922	−4.2661	−4.3220	−4.3664	−4.4028	−4.4333	−4.4594
−4.0	0	−3.2528	−3.3386	−3.3992	−3.4452	−3.4815	−3.5113	−3.5364	−3.5578
−3.0	.0013	−2.4123	−2.4801	−2.5280	−2.5642	−2.5929	−2.6164	−2.6361	−2.6530
−2.0	.0227	−1.5589	−1.6105	−1.6469	−1.6743	−1.6960	−1.7138	−1.7286	−1.7413
−1.0	.1586	−.6717	−.7122	−.7402	−.7612	−.7777	−.7911	−.8023	−.8118
−0	.5000	.3102	.2664	.2371	.2158	.1993	.1861	.1752	.1660
.1	.5398	.4168	.3713	.3411	.3191	.3022	.2886	.2775	.2681
.2	.5792	.5249	.4776	.4462	.4235	.4060	.3921	.3806	.3710
.3	.6179	.6347	.5852	.5526	.5290	.5109	.4965	.4846	.4747
.4	.6554	.7459	.6941	.6600	.6354	.6166	.6017	.5894	.5791
.5	.6914	.8586	.8042	.7685	.7429	.7233	.7078	.6950	.6843
.6	.7257	.9726	.9154	.8781	.8512	.8308	.8146	.8014	.7903
.7	.7580	1.0877	1.0277	.9885	.9604	.9391	.9222	.9084	.8968
.8	.7881	1.2041	1.1409	1.0998	1.0704	1.0481	1.0304	1.0160	1.0039
.9	.8159	1.3214	1.2550	1.2118	1.1811	1.1577	1.1393	1.1242	1.1116
1.0	.8413	1.4398	1.3699	1.3246	1.2924	1.2680	1.2487	1.2329	1.2198
1.1	.8643	1.5589	1.4855	1.4381	1.4043	1.3787	1.3586	1.3421	1.3284
1.2	.8849	1.6788	1.6018	1.5520	1.5167	1.4900	1.4689	1.4518	1.4374
1.3	.9031	1.7994	1.7187	1.6666	1.6296	1.6017	1.5797	1.5618	1.5468
1.4	.9192	1.9206	1.8361	1.7816	1.7429	1.7138	1.6908	1.6721	1.6566
1.5	.9331	2.0423	1.9539	1.8970	1.8566	1.8262	1.8023	1.7828	1.7666
1.6	.9452	2.1645	2.0722	2.0127	1.9707	1.9390	1.9140	1.8938	1.8769
1.7	.9554	2.2873	2.1908	2.1289	2.0851	2.0521	2.0261	2.0050	1.9874
1.8	.9640	2.4104	2.3099	2.2453	2.1997	2.1654	2.1384	2.1164	2.0981
1.9	.9712	2.5338	2.4292	2.3620	2.3146	2.2789	2.2509	2.2281	2.2091
2.0	.9772	2.6576	2.5488	2.4790	2.4297	2.3927	2.3635	2.3399	2.3202
2.1	.9821	2.7817	2.6686	2.5962	2.5451	2.5066	2.4764	2.4519	2.4314
2.2	.9860	2.9061	2.7887	2.7136	2.6606	2.6207	2.5894	2.5640	2.5428
2.3	.9892	3.0307	2.9090	2.8312	2.7763	2.7350	2.7026	2.6763	2.6544
2.4	.9918	3.1555	3.0295	2.9489	2.8921	2.8494	2.8159	2.7887	2.7661
2.5	.9937	3.2905	3.1502	3.0669	3.0081	2.9640	2.9293	2.9012	2.8778
2.6	.9953	3.4058	3.2710	3.1849	3.1243	3.0787	3.0429	3.0139	2.9897
2.7	.9965	3.5311	3.3920	3.3031	3.2405	3.1935	3.1565	3.1266	3.1017
2.8	.9974	3.6567	3.5131	3.4214	3.3568	3.3083	3.2703	3.2394	3.2137
2.9	.9981	3.7824	3.6344	3.5398	3.4733	3.4233	3.3841	3.3523	3.3258
3.0	.9986	3.9082	3.7557	3.6583	3.5898	3.5384	3.4980	3.4653	3.4380
3.1	.9990	4.0341	3.8771	3.7769	3.7064	3.6535	3.6120	3.5783	3.5503
3.2	.9993	4.1601	3.9987	3.8956	3.8231	3.7687	3.7260	3.6914	3.6626
3.3	.9995	4.2863	4.1203	4.0144	3.9399	3.8840	3.8401	3.8045	3.7750
3.4	.9996	4.4125	4.2420	4.1332	4.0567	3.9993	3.9542	3.9177	3.8874
3.5	.9997	4.5388	4.3638	4.2521	4.1736	4.1147	4.0684	4.0310	3.9998
3.6	.9998	4.6652	4.4856	4.3711	4.2905	4.2301	4.1827	4.1443	4.1123
3.7	.9998	4.7917	4.6075	4.4901	4.4075	4.3456	4.2970	4.2576	4.2249

TABLE A–4.—Concluded.

(d) Concluded.

Safety margin, S_M	Probability, P_x	Sample size, N							
		30	40	50	60	70	80	90	100
3.8	0.9999	4.9182	4.7295	4.6092	4.5246	4.4611	4.4114	4.3710	4.3375
3.9		5.0448	4.8515	4.7283	4.6417	4.5767	4.5257	4.4844	4.4501
4.0		5.1715	4.9736	4.8475	4.7588	4.6923	4.6402	4.5979	4.5628
4.1		5.2982	5.0958	4.9667	4.8760	4.8080	4.7546	4.7114	4.6755
4.2		5.4250	5.2179	5.0860	4.9932	4.9237	4.8691	4.8249	4.7882
4.3		5.5519	5.3402	5.2053	5.1105	5.0394	4.9836	4.9385	4.9009
4.4		5.6788	5.4624	5.3246	5.2278	5.1551	5.0982	5.0520	5.0137
4.5		5.8057	5.5847	5.4440	5.3451	5.2709	5.2128	5.1656	5.1265
4.6		5.9327	5.7071	5.5634	5.4624	5.3867	5.3274	5.2793	5.2393
4.7		6.0597	5.8294	5.6828	5.5798	5.5025	5.4420	5.3929	5.3521
4.8		6.1867	5.9519	5.8023	5.6972	5.6184	5.5566	5.5066	5.4650
4.9	1.0000	6.3138	6.0743	5.9218	5.8146	5.7343	5.6713	5.6203	5.5779
5.0		6.4409	6.1968	6.0413	5.9321	5.8502	5.7860	5.7340	5.6908
5.1		6.5681	6.3193	6.1608	6.0495	5.9661	5.9007	5.8477	5.8037
5.2		6.6952	6.4418	6.2804	6.1670	6.0820	6.0154	5.9614	5.9166
5.3		6.8224	6.5643	6.4000	6.2845	6.1980	6.1302	6.0752	6.0296
5.4		6.9497	6.6869	6.5196	6.4021	6.3140	6.2449	6.1890	6.1425
5.5		7.0769	6.8095	6.6392	6.5196	6.4300	6.3597	6.3028	6.2555
5.6		7.2042	6.9321	6.7588	6.6372	6.5460	6.4745	6.4166	6.3685
5.7		7.3315	7.0547	6.8785	6.7548	6.6620	6.5893	6.5304	6.4815
5.8		7.4588	7.1774	6.9982	6.8723	6.7780	6.7041	6.6442	6.5945
5.9		7.5862	7.3000	7.1179	6.9900	6.8941	6.8189	6.7581	6.7075
6.0		7.7136	7.4227	7.2376	7.1076	7.0101	6.9338	6.8719	6.8205
6.1		7.8409	7.5454	7.3573	7.2252	7.1262	7.0486	6.9858	6.9336
6.2		7.9683	7.6681	7.4770	7.3429	7.2423	7.1635	7.0996	7.0466
6.3		8.0958	7.7908	7.5968	7.4605	7.3584	7.2784	7.2135	7.1597
6.4		8.2232	7.9136	7.7165	7.5782	7.4745	7.3932	7.3274	7.2727
6.5		8.3506	8.0363	7.8363	7.6959	7.5906	7.5081	7.4413	7.3858
6.6		8.4781	8.1591	7.9561	7.8136	7.7068	7.6230	7.5552	7.4989
6.7		8.6056	8.2819	8.0759	7.9313	7.8229	7.7379	7.6691	7.6120
6.8		8.7331	8.4047	8.1957	8.0490	7.9390	7.8529	7.7831	7.7251
6.9		8.8606	8.5275	8.3155	8.1667	8.0552	7.9678	7.8970	7.8382
7.0		8.9881	8.6503	8.4354	8.2844	8.1714	8.0827	8.0109	7.9513
7.1		9.1157	8.7731	8.5552	8.4022	8.2875	8.1977	8.1249	8.0644
7.2		9.2432	8.8960	8.6750	8.5199	8.4037	8.3126	8.2388	8.1776
7.3		9.3708	9.0188	8.7949	8.6377	8.5199	8.4276	8.3528	8.2907
7.4		9.4983	9.1417	8.9147	8.7554	8.6361	8.5425	8.4668	8.4038
7.5		9.6259	9.2645	9.0346	8.8732	8.7523	8.6575	8.5807	8.5170
7.6		9.7535	9.3874	9.1545	8.9910	8.8685	8.7725	8.6947	8.6301
7.7		9.8811	9.5103	9.2744	9.1088	8.9847	8.8874	8.8087	8.7433
7.8		10.0087	9.6332	9.3943	9.2266	9.1009	9.0024	8.9227	8.8564
7.9		10.1363	9.7561	9.5142	9.3444	9.2171	9.1174	9.0367	8.9696
8.0		10.2639	9.8790	9.6341	9.4622	9.3334	9.2324	9.1507	9.0828

TABLE A-5.—SAFETY MARGINS AT 90-PERCENT CONFIDENCE LEVEL

(a) Sample sizes 5 to 12

Safety margin, S_M	Probability, P_x	Sample size, N							
		5	6	7	8	9	10	11	12
−5.0	0	−3.5162	−3.6140	−3.6932	−3.7586	−3.8146	−3.8623	−3.9045	−3.9418
−4.0	0	−2.7824	−2.8627	−2.9278	−2.9816	−3.0276	−3.0669	−3.1016	−3.1323
−3.0	.0013	−2.0381	−2.1018	−2.1535	−2.1962	−2.2327	−2.2640	−2.2915	−2.3159
−2.0	.0227	−1.2682	−1.3178	−1.3578	−1.3910	−1.4192	−1.4435	−1.4647	−1.4835
−1.0	.1586	−.4225	−.4673	−.5023	−.5312	−.5548	−.5752	−.5927	−.6082
−0	.5000	.6857	.6023	.5439	.5000	.4657	.4373	.4137	.3936
.1	.5398	.8218	.7303	.6665	.6190	.5822	.5518	.5267	.5053
.2	.5792	.9632	.8623	.7930	.7412	.7014	.6689	.6420	.6193
.3	.6179	1.1098	.9984	.9229	.8664	.8234	.7885	.7598	.7355
.4	.6554	1.2615	1.1385	1.0557	.9946	.9481	.9105	.8797	.8537
.5	.6914	1.4168	1.2817	1.1912	1.1256	1.0750	1.0347	1.0016	.9739
.6	.7257	1.5762	1.4282	1.3296	1.2591	1.2043	1.1610	1.1256	1.0960
.7	.7580	1.7392	1.5777	1.4709	1.3945	1.3358	1.2894	1.2514	1.2197
.8	.7881	1.9057	1.7301	1.6147	1.5321	1.4693	1.4196	1.3789	1.3451
.9	.8159	2.0753	1.8851	1.7608	1.6715	1.6044	1.5512	1.5078	1.4717
1.0	.8413	2.2475	2.0422	1.9087	1.8127	1.7412	1.6843	1.6381	1.5997
1.1	.8643	2.4221	2.2010	2.0578	1.9558	1.8792	1.8186	1.7695	1.7288
1.2	.8849	2.5988	2.3615	2.2085	2.1003	2.0184	1.9542	1.9021	1.8590
1.3	.9031	2.7769	2.5237	2.3605	2.2460	2.1589	2.0908	2.0357	1.9901
1.4	.9192	2.9564	2.6871	2.5138	2.3923	2.3003	2.2283	2.1701	2.1220
1.5	.9331	3.1372	2.8518	2.6681	2.5396	2.4426	2.3666	2.3052	2.2546
1.6	.9452	3.3192	3.0175	2.8234	2.6878	2.5857	2.5057	2.4411	2.3879
1.7	.9554	3.5024	3.1841	2.9795	2.8367	2.7296	2.6454	2.5776	2.5217
1.8	.9640	3.6868	3.3516	3.1363	2.9863	2.8740	2.7857	2.7147	2.6562
1.9	.9712	3.8720	3.5198	3.2938	3.1366	3.0189	2.9265	2.8522	2.7910
2.0	.9772	4.0580	3.6886	3.4519	3.2873	3.1643	3.0678	2.9902	2.9262
2.1	.9821	4.2446	3.8580	3.6105	3.4386	3.3101	3.2095	3.1285	3.0619
2.2	.9860	4.4318	4.0279	3.7696	3.5903	3.4563	3.3515	3.2672	3.1979
2.3	.9892	4.6195	4.1983	3.9292	3.7424	3.6029	3.4940	3.4063	3.3341
2.4	.9918	4.8076	4.3691	4.0891	3.8948	3.7499	3.6367	3.5456	3.4707
2.5	.9937	4.9962	4.5403	4.2493	4.0476	3.8971	3.7797	3.6852	3.6076
2.6	.9953	5.1851	4.7118	4.4099	4.2006	4.0446	3.9230	3.8251	3.7446
2.7	.9965	5.3742	4.8836	4.5707	4.3539	4.1924	4.0665	3.9652	3.8819
2.8	.9974	5.5636	5.0557	4.7318	4.5075	4.3404	4.2102	4.1054	4.0194
2.9	.9981	5.7532	5.2281	4.8931	4.6613	4.4886	4.3541	4.2459	4.1570
3.0	.9986	5.9431	5.4007	5.0547	4.8152	4.6870	4.4982	4.3865	4.2948
3.1	.9990	6.1332	5.5735	5.2164	4.9694	4.7855	4.6425	4.5273	4.4328
3.2	.9993	6.3236	5.7465	5.3784	5.1237	4.9342	4.7869	4.6683	4.5709
3.3	.9995	6.5142	5.9197	5.5405	5.2782	5.0831	4.9314	4.8093	4.7091
3.4	.9996	6.7049	6.0931	5.7027	5.4328	5.2321	5.0761	4.9505	4.8475
3.5	.9997	6.8958	6.2666	5.8651	5.5876	5.3812	5.2209	5.0918	4.9859
3.6	.9998	7.0869	6.4402	6.0276	5.7425	5.5305	5.3658	5.2332	5.1245
3.7	.9998	7.2781	6.6140	6.1903	5.8975	5.6798	5.5108	5.3747	5.2631

TABLE A-5.—Continued.

(a) Concluded.

Safety margin, S_M	Probability, P_x	Sample size, N							
		5	6	7	8	9	10	11	12
3.8	0.9999	7.4694	6.7879	6.3531	6.0526	5.8293	5.6559	5.5163	5.4018
3.9		7.6609	6.9619	6.5159	6.2078	5.9788	5.8011	5.6580	5.5406
4.0		7.8525	7.1360	6.6789	6.3631	6.1284	5.9464	5.7998	5.6795
4.1		8.0442	7.3102	6.8419	6.5185	6.2781	6.0917	5.9416	5.8185
4.2		8.2360	7.4845	7.0051	6.6740	6.4279	6.2371	6.0835	5.9575
4.3		8.4279	7.6589	7.1683	6.8295	6.5778	6.3826	6.2254	6.0965
4.4		8.6199	7.8334	7.3316	6.9851	6.7277	6.5282	6.3675	6.2357
4.5		8.8120	8.0079	7.4949	7.1408	6.8777	6.6738	6.5095	6.3749
4.6		9.0041	8.1826	7.6584	7.2965	7.0277	6.8194	6.6517	6.5141
4.7		9.1963	8.3572	7.8219	7.4523	7.1778	6.9651	6.7938	6.6534
4.8		9.3886	8.5320	7.9854	7.6082	7.3280	7.1109	6.9361	6.7927
4.9	1.0000	9.5809	8.7068	8.1490	7.7641	7.4782	7.2567	7.0783	6.9321
5.0		9.7734	8.8816	8.3127	7.9200	7.6284	7.4025	7.2206	7.0715
5.1		9.9658	9.0566	8.4764	8.0760	7.7787	7.5484	7.3630	7.2109
5.2		10.1584	9.2315	8.6401	8.2320	7.9290	7.6944	7.5054	7.3504
5.3		10.3509	9.4065	8.8039	8.3881	8.0794	7.8403	7.6478	7.4900
5.4		10.5436	9.5816	8.9677	8.5442	8.2298	7.9863	7.7902	7.6295
5.5		10.7362	9.7567	9.1316	8.7004	8.3802	8.1324	7.9327	7.7691
5.6		10.9289	9.9318	9.2955	8.8566	8.5307	8.2784	8.0752	7.9087
5.7		11.1217	10.1070	9.4595	9.0128	8.6812	8.4245	8.2178	8.0483
5.8		11.3145	10.2822	9.6235	9.1691	8.8318	8.5706	8.3603	8.1880
5.9		11.5073	10.4574	9.7875	9.3253	8.9823	8.7168	8.5029	8.3277
6.0		11.7002	10.6327	9.9515	9.4817	9.1329	8.8630	8.6456	8.4674
6.1		11.8931	10.8080	10.1156	9.6380	9.2835	9.0092	8.7882	8.6071
6.2		12.0860	10.9833	10.2797	9.7944	9.4342	9.1554	8.9309	8.7469
6.3		12.2790	11.1587	10.4438	9.9508	9.5848	9.3016	9.0736	8.8866
6.4		12.4720	11.3341	10.6079	10.1072	9.7355	9.4479	9.2163	9.0264
6.5		12.6650	11.5095	10.7721	10.2636	9.8862	9.5942	9.3590	9.1662
6.6		12.8581	11.6849	10.9363	10.4201	10.0869	9.7405	9.5017	9.3061
6.7		13.0512	11.8604	11.1005	10.5765	10.1877	9.8868	9.6445	9.4459
6.8		13.2443	12.0359	11.2648	10.7330	10.3385	10.0332	9.7873	9.5858
6.9		13.4374	12.2114	11.4290	10.8896	10.4892	10.1795	9.9301	9.7257
7.0		13.6305	12.3869	11.5933	11.0461	10.6400	10.3259	10.0729	9.8656
7.1		13.8237	12.5625	11.7576	11.2026	10.7909	10.4723	10.2157	10.0055
7.2		14.0169	12.7380	11.9219	11.3592	10.9417	10.6187	10.3585	10.1454
7.3		14.2101	12.9136	12.0863	11.5158	11.0925	10.7651	10.5014	10.2853
7.4		14.4033	13.0892	12.2506	11.6724	11.2434	10.9116	10.6443	10.4253
7.5		14.5966	13.2648	12.4150	11.8290	11.3943	11.0580	10.7872	10.5652
7.6		14.7899	13.4405	12.5794	11.9857	11.5452	11.2045	10.9300	10.7052
7.7		14.9831	13.6161	12.7437	12.1423	11.6961	11.3509	11.0729	10.8452
7.8		15.1764	13.7918	12.9082	12.2990	11.8470	11.4974	11.2159	10.9852
7.9		15.3698	13.9675	13.0726	12.4556	11.9979	11.6439	11.3588	11.1252
8.0		15.5631	14.1432	13.2370	12.6123	12.1488	11.7904	11.5017	11.2652

TABLE A-5.—Continued.

(b) Sample sizes 13 to 20

Safety margin, S_M	Probability, P_x	Sample size, N							
		13	14	15	16	17	18	19	20
−5.0	0	−3.9752	−4.0054	−4.0328	−4.0579	−4.0809	−4.1023	−4.1221	−4.1406
−4.0	0	−3.1598	−3.1846	−3.2072	−3.2278	−3.2468	−3.2643	−3.2806	−3.2958
−3.0	.0013	−2.3377	−2.3574	−2.3753	−2.3916	−2.4067	−2.4206	−2.4335	−2.4455
−2.0	.0227	−1.5003	−1.5155	−1.5293	−1.5419	−1.5534	−1.5641	−1.5739	−1.5831
−1.0	.1586	−.6220	−.6343	−.6455	−.6556	−.6649	−.6734	−.6813	−.6886
−0	.5000	.3762	.3609	.3473	.3351	.3242	.3143	.3052	.2969
.1	.5398	.4869	.4707	.4564	.4436	.4321	.4217	.4123	.4036
.2	.5792	.5997	.5826	.5675	.5540	.5419	.5310	.5210	.5119
.3	.6179	.7146	.6964	.6804	.6662	.6534	.6419	.6314	.6218
.4	.6554	.8315	.8122	.7952	.7801	.7666	.7544	.7433	.7332
.5	.6914	.9502	.9296	.9116	.8956	.8813	.8684	.8566	.8460
.6	.7257	1.0707	1.0488	1.0296	1.0126	.9975	.9838	.9714	.9601
.7	.7580	1.1927	1.1694	1.1490	1.1310	1.1149	1.1004	1.0873	1.0753
.8	.7881	1.3163	1.2915	1.2698	1.2507	1.2336	1.2182	1.2044	1.1917
.9	.8159	1.4411	1.4148	1.3918	1.3715	1.3534	1.3372	1.3225	1.3091
1.0	.8413	1.5673	1.5393	1.5149	1.4935	1.4744	1.4572	1.4416	1.4275
1.1	.8643	1.6944	1.6648	1.6390	1.6163	1.5962	1.5780	1.5616	1.5467
1.2	.8849	1.8225	1.7913	1.7640	1.7401	1.7188	1.6996	1.6824	1.6667
1.3	.9031	1.9516	1.9186	1.8899	1.8646	1.8422	1.8220	1.8038	1.7873
1.4	.9192	2.0814	2.0466	2.0164	1.9898	1.9662	1.9450	1.9259	1.9086
1.5	.9331	2.2119	2.1753	2.1435	2.1156	2.0908	2.0686	2.0485	2.0303
1.6	.9452	2.3430	2.3046	2.2712	2.2420	2.2160	2.1927	2.1716	2.1526
1.7	.9554	2.4747	2.4344	2.3995	2.3688	2.3416	2.3172	2.2952	2.2753
1.8	.9640	2.6069	2.5648	2.5282	2.4962	2.4677	2.4422	2.4192	2.3984
1.9	.9712	2.7395	2.6955	2.6573	2.6238	2.5942	2.5675	2.5435	2.5218
2.0	.9772	2.8725	2.8266	2.7868	2.7519	2.7210	2.6932	2.6682	2.6456
2.1	.9821	3.0059	2.9580	2.9166	2.8802	2.8480	2.8191	2.7931	2.7696
2.2	.9860	3.1396	3.0898	3.0467	3.0089	2.9754	2.9454	2.9183	2.8939
2.3	.9892	3.2736	3.2219	3.1771	3.1378	3.1031	3.0719	3.0438	3.0184
2.4	.9918	3.4079	3.3542	3.3077	3.2670	3.2309	3.1986	3.1695	3.1432
2.5	.9937	3.5424	3.4867	3.4385	3.3963	3.3590	3.3255	3.2954	3.2681
2.6	.9953	3.6771	3.6195	3.5696	3.5259	3.4873	3.4526	3.4214	3.3932
2.7	.9965	3.8121	3.7525	3.7009	3.6557	3.6157	3.5799	3.5476	3.5185
2.8	.9974	3.9472	3.8856	3.8323	3.7856	3.7444	3.7073	3.6740	3.6439
2.9	.9981	4.0825	4.0189	3.9639	3.9157	3.8731	3.8349	3.8005	3.7695
3.0	.9986	4.2179	4.1523	4.0956	4.0459	4.0020	3.9626	3.9272	3.8952
3.1	.9990	4.3535	4.2859	4.2275	4.1763	4.1310	4.0904	4.0539	4.0209
3.2	.9993	4.4893	4.4197	4.3594	4.3067	4.2602	4.2183	4.1808	4.1468
3.3	.9995	4.6251	4.5535	4.4915	4.4373	4.3894	4.3464	4.3078	4.2728
3.4	.9996	4.7611	4.6874	4.6237	4.5680	4.5187	4.4745	4.4348	4.3989
3.5	.9997	4.8971	4.8215	4.7560	4.6988	4.6481	4.6027	4.5619	4.5251
3.6	.9998	5.0333	4.9556	4.8884	4.8296	4.7777	4.7310	4.6892	4.6513
3.7	.9998	5.1695	5.0898	5.0209	4.9605	4.9072	4.8594	4.8165	4.7775

TABLE A-5.—Continued.

(b) Concluded.

Safety margin, S_M	Probability, P_x	Sample size, N							
		13	14	15	16	17	18	19	20
3.8	0.9999	5.3059	5.2241	5.1534	5.0916	5.0369	4.9879	4.9438	4.9040
3.9		5.4423	5.3585	5.2860	5.2226	5.1666	5.1164	5.0712	5.0305
4.0		5.5788	5.4929	5.4187	5.3538	5.2964	5.2449	5.1987	5.1570
4.1		5.7153	5.6274	5.5514	5.4850	5.4263	5.3736	5.3263	5.2835
4.2		5.8519	5.7620	5.6842	5.6162	5.5562	5.5022	5.4539	5.4101
4.3		5.9886	5.8966	5.8171	5.7475	5.6861	5.6310	5.5815	5.5368
4.4		6.1253	6.0313	5.9500	5.8789	5.8161	5.7598	5.7092	5.6635
4.5		6.2621	6.1660	6.0829	6.0103	5.9461	5.8886	5.8369	5.7902
4.6		6.3989	6.3008	6.2159	6.1418	6.0762	6.0174	5.9647	5.9170
4.7		6.5358	6.4356	6.3490	6.2733	6.2064	6.1463	6.0925	6.0438
4.8		6.6727	6.5704	6.4821	6.4048	6.3365	6.2743	6.2203	6.1707
4.9	1.0000	6.8096	6.7053	6.6152	6.5364	6.4667	6.4043	6.3482	6.2975
5.0		6.9466	6.8402	6.7483	6.6680	6.5970	6.5333	6.4761	6.4245
5.1		7.0836	6.9752	6.8815	6.7996	6.7272	6.6623	6.6040	6.5514
5.2		7.2207	7.1102	7.0147	6.9313	6.8575	6.7914	6.7320	6.6784
5.3		7.3578	7.2452	7.1480	7.0630	6.9878	6.9205	6.8600	6.8054
5.4		7.4949	7.3803	7.2813	7.1947	7.1182	7.0496	6.9880	6.9324
5.5		7.6321	7.5154	7.4146	7.3264	7.2486	7.1787	7.1160	7.0594
5.6		7.7693	7.6505	7.5479	7.4582	7.3790	7.3079	7.2441	7.1865
5.7		7.9065	7.7856	7.6813	7.5900	7.5094	7.4371	7.3722	7.3136
5.8		8.0437	7.9208	7.8146	7.7218	7.63 8	7.5663	7.5003	7.4407
5.9		8.1809	8.0560	7.9480	7.8537	7.7703	7.6955	7.6284	7.5678
6.0		8.3182	8.1912	8.0815	7.9855	7.9008	7.8248	7.7566	7.6960
6.1		8.4555	8.3264	8.2149	8.1174	8.0313	7.9540	7.8847	7.8221
6.2		8.5928	8.4617	8.3484	8.2493	8.1618	8.0833	8.0129	7.9493
6.3		8.7302	8.5969	8.4818	8.3812	8.2924	8.2126	8.1411	8.0765
6.4		8.8675	8.7322	8.6153	8.5131	8.4229	8.3420	8.2693	8.2037
6.5		9.0049	8.8675	8.7488	8.6451	8.5535	8.4713	8.3975	8.3309
6.6		9.1423	9.0028	8.8824	8.7771	8.6841	8.6006	8.5258	8.4582
6.7		9.2797	9.1382	9.0159	8.9090	8.8147	8.7300	8.6540	8.5854
6.8		9.4171	9.2735	9.1495	9.0410	8.9453	8.8594	8.7823	8.7127
6.9		9.5546	9.4089	9.2830	9.1730	9.0759	8.9888	8.9106	8.8400
7.0		9.6920	9.5443	9.4166	9.3051	9.2065	9.1182	9.0389	8.9672
7.1		9.8295	9.6797	9.5502	9.4371	9.3372	9.2476	9.1672	9.0945
7.2		9.9670	9.8151	9.6838	9.5691	9.4679	9.3770	9.2955	9.2219
7.3		10.1045	9.9505	9.8175	9.7012	9.5985	9.5064	9.4238	9.3492
7.4		10.2420	10.0859	9.9511	9.8333	9.7292	9.6359	9.5521	9.4765
7.5		10.3795	10.2213	10.0847	9.9653	9.8599	9.7653	9.6805	9.6038
7.6		10.5170	10.3568	10.2184	10.0974	9.9906	9.8948	9.8088	9.7312
7.7		10.6546	10.4923	10.3521	10.2295	10.1213	10.0243	8.8372	9.8585
7.8		10.7921	10.6277	10.4857	10.3616	10.2521	10.1538	10.0656	9.9859
7.9		10.9297	10.7632	10.6194	10.4938	10.3828	10.2833	10.1940	10.1133
8.0		11.0672	10.8987	10.7531	10.6259	10.5135	10.4128	10.3223	10.2407

TABLE A-5.—Continued.

(c) Sample sizes 21 to 28

Safety margin S_M	Probability, P_x	Sample size, N							
		21	22	23	24	25	26	27	28
−5.0	0	−4.1579	−4.1740	−4.1893	−4.2036	−4.2172	−4.2300	−4.2422	−4.2538
−4.0	0	−3.3100	−3.3233	−3.3358	−3.3476	−3.3587	−3.3693	−3.3793	−3.3888
−3.0	.0013	−2.4568	−2.4673	−2.4772	−2.4865	−2.4954	−2.5037	−2.5116	−2.5192
−2.0	.0227	−1.5917	−1.5998	−1.6073	−1.6145	−1.6212	−1.6276	−1.6336	−1.6393
−1.0	.1586	−.6954	−.7017	−.7077	−.7133	−.7186	−.7235	−.7283	−.7327
0	.5000	.2893	.2821	.2755	.2694	.2636	.2582	.2531	.2483
.1	.5398	.3956	.3882	.3813	.3749	.3689	.3633	.3580	.3530
.2	.5792	.5035	.4958	.4886	.4819	.4757	.4698	.4643	.4591
.3	.6179	.6130	.6049	.5973	.5903	.5837	.5776	.5719	.5664
.4	.6554	.7239	.7153	.7074	.7000	.6931	.6867	.6807	.6750
.5	.6914	.8362	.8271	.8188	.8110	.8037	.7970	.7906	.7847
.6	.7257	.9497	.9402	.9314	.9232	.9155	.9084	.9017	.8955
.7	.7580	1.0644	1.0543	1.0450	1.0364	1.0283	1.0208	1.0138	1.0072
.8	.7881	1.1802	1.1695	1.1597	1.1506	1.1421	1.1342	1.1269	1.1199
.9	.8159	1.2969	1.2857	1.2754	1.2658	1.2568	1.2485	1.2407	1.2334
1.0	.8413	1.4146	1.4028	1.3919	1.3818	1.3724	1.3636	1.3554	1.3477
1.1	.8643	1.5331	1.5207	1.5092	1.4985	1.4886	1.4794	1.4708	1.4627
1.2	.8849	1.6524	1.6392	1.6271	1.6159	1.6055	1.5959	1.5868	1.5783
1.3	.9031	1.7723	1.7585	1.7457	1.7340	1.7231	1.7129	1.7034	1.6945
1.4	.9192	1.8927	1.8783	1.8649	1.8526	1.8411	1.8305	1.8205	1.8112
1.5	.9331	2.0137	1.9985	1.9846	1.9716	1.9596	1.9485	1.9380	1.9283
1.6	.9452	2.1352	2.1193	2.1047	2.0911	2.0786	2.0669	2.0560	2.0458
1.7	.9554	2.2571	2.2405	2.2252	2.2111	2.1979	2.1857	2.1743	2.1637
1.8	.9640	2.3794	2.3621	2.3461	2.3313	2.3176	2.3049	2.2930	2.2819
1.9	.9712	2.5021	2.4839	2.4673	2.4519	2.4376	2.4244	2.4120	2.4004
2.0	.9772	2.6250	2.6061	2.5888	2.5728	2.5579	2.5441	2.5312	2.5192
2.1	.9821	2.7482	2.7286	2.7105	2.6939	2.6784	2.6641	2.6507	2.6382
2.2	.9860	2.8716	2.8513	2.8325	2.8152	2.7992	2.7843	2.7704	2.7574
2.3	.9892	2.9953	2.9742	2.9547	2.9368	2.9202	2.9047	2.8903	2.8768
2.4	.9918	3.1192	3.0973	3.0772	3.0586	3.0413	3.0253	3.0104	2.9964
2.5	.9937	3.2433	3.2206	3.1998	3.1805	3.1627	3.1461	3.1306	3.1162
2.6	.9953	3.3676	3.3441	3.3225	3.3026	3.2842	3.2670	3.2510	3.2361
2.7	.9965	3.4920	3.4677	3.4454	3.4249	3.4058	3.3881	3.3716	3.3561
2.8	.9974	3.6165	3.5915	3.5685	3.5472	3.5276	3.5093	3.4922	3.4763
2.9	.9981	3.7412	3.7154	3.6916	3.6697	3.6495	3.6306	3.6130	3.5966
3.0	.9986	3.8660	3.8394	3.8149	3.7924	3.7714	3.7520	3.7339	3.7170
3.1	.9990	3.9909	3.9635	3.9383	3.9151	3.8935	3.8735	3.8549	3.8374
3.2	.9993	4.1160	4.0877	4.0618	4.0379	4.0157	3.9951	3.9759	3.9580
3.3	.9995	4.2411	4.2120	4.1854	4.1608	4.1380	4.1168	4.0971	4.0786
3.4	.9996	4.3663	4.3364	4.3090	4.2838	4.2603	4.2386	4.2183	4.1994
3.5	.9997	4.4916	4.4609	4.4328	4.4068	4.3828	4.3604	4.3396	4.3201
3.6	.9998	4.6169	4.5855	4.5566	4.5299	4.5053	4.4823	4.4610	4.4410
3.7	.9998	4.7423	4.7101	4.6805	4.6531	4.6278	5.6043	4.5824	4.5619

TABLE A-5.—Continued.

(c) Concluded.

Safety margin, S_M	Probability, P_x	Sample size, N							
		21	22	23	24	25	26	27	28
3.8	0.9999	4.8678	4.8348	4.8044	4.7764	4.7504	4.7263	4.7039	4.6829
3.9		4.9934	4.9595	4.9284	4.8997	4.8731	4.8484	4.8254	4.8039
4.0		5.1190	5.0843	5.0524	5.0231	4.9958	4.9706	4.9470	4.9250
4.1		5.2447	5.2091	5.1765	5.1465	5.1186	5.0927	5.0686	5.0461
4.2		5.3704	5.3340	5.3007	5.2699	5.2414	5.2150	5.1903	5.1673
4.3		5.4961	5.4590	5.4259	5.3934	5.3643	5.3372	5.3120	5.2885
4.4		5.6219	5.5840	5.5491	5.5170	5.4872	5.4595	5.4338	5.4097
4.5		5.7478	5.7090	5.6734	5.6405	5.6101	5.5819	5.5556	5.5310
4.6		5.8737	5.8340	5.7977	5.7641	5.7331	5.7043	5.6774	5.6523
4.7		5.9996	5.9591	5.9220	5.8878	5.8561	5.8267	5.7992	5.7736
4.8		6.1255	6.0843	6.0464	6.0115	5.9791	5.9491	5.9211	5.8950
4.9	1.0000	6.2515	6.2094	6.1708	6.1352	6.1022	6.0716	6.0430	6.0164
5.0		6.3775	6.3346	6.2952	6.2589	6.2253	6.1941	6.1650	6.1378
5.1		6.5035	6.4598	6.4197	6.3827	6.3484	6.3166	6.2869	6.2592
5.2		6.6296	6.5851	6.5442	6.5065	6.4716	6.4391	6.4089	6.3807
5.3		6.7557	6.7103	6.6687	6.6303	6.5947	6.5617	6.5309	6.5022
5.4		6.8818	6.8356	6.7932	6.7541	6.7179	6.6843	6.6530	6.6237
5.5		7.0080	6.9609	6.9178	6.8780	6.8411	6.8069	6.7750	6.7452
5.6		7.1341	7.0863	7.0423	7.0019	6.9644	6.9295	6.8971	6.8668
5.7		7.2603	7.2116	7.1669	7.1257	7.0876	7.0522	7.0192	6.9883
5.8		7.3865	7.3370	7.2916	7.2497	7.2109	7.1749	7.1413	7.1099
5.9		7.5127	7.4624	7.4162	7.3736	7.3342	7.2975	7.2634	7.2315
6.0		7.6390	7.5878	7.5408	7.4975	7.4575	7.4202	7.3856	7.3531
6.1		7.7652	7.7132	7.6655	7.6215	7.5808	7.5430	7.5077	7.4748
6.2		7.8915	7.8387	7.7902	7.7455	7.7041	7.6657	7.6299	7.5964
6.3		8.0178	7.9641	7.9149	7.8695	7.8275	7.7884	7.7521	7.7181
6.4		8.1441	8.0896	8.0396	7.9935	7.9508	7.9112	7.8743	7.8398
6.5		8.2704	8.2151	8.1643	8.1175	8.0742	8.0340	7.9965	7.9614
6.6		8.3967	8.3406	8.2891	8.2415	8.1976	8.1567	8.1187	8.0831
6.7		8.5231	8.4661	8.4138	8.3656	8.3210	8.2795	8.2409	8.2048
6.8		8.6494	8.5916	8.5386	8.4897	8.4444	8.4023	8.3632	8.3266
6.9		8.7758	8.7172	8.6633	8.6137	8.5678	8.5252	8.4854	8.4483
7.0		8.9022	8.8427	8.7881	8.7378	8.6912	8.6480	8.6077	8.5700
7.1		9.0285	8.9683	8.9129	8.8619	8.1847	8.7708	8.7300	8.6918
7.2		9.1549	9.0938	9.0377	8.9860	8.9381	8.8937	8.8522	8.8135
7.3		9.2814	9.2194	8.1625	8.1101	9.0616	9.0165	8.9745	8.9353
7.4		9.4078	9.3450	9.2873	9.2342	9.1850	9.1394	9.0968	9.0571
7.5		9.5342	9.4706	9.4122	9.3583	9.3085	8.2622	9.2191	9.1788
7.6		9.6606	9.5962	9.5370	9.4825	9.4320	9.3851	9.3414	9.3006
7.7		9.7871	9.7218	9.6619	9.6066	9.5555	9.5080	9.4638	9.4224
7.8		9.9135	9.8474	9.7867	9.7308	9.6790	9.6309	9.5861	9.5442
7.9		10.0400	9.9730	9.9116	9.8549	9.8025	9.7538	9.7084	9.6660
8.0		10.1665	10.0987	10.0365	9.9791	9.9260	9.8767	9.8308	9.7879

TABLE A-5.—Continued.

(d) Sample sizes 30 to 100

Safety margin, S_M	Probability, P_x	Sample size, N							
		30	40	50	60	70	80	90	100
−5.0	0	−4.2753	−4.3596	−4.4191	−4.4640	−4.4996	−4.5286	−4.5530	−4.5738
−4.0	0	−3.4065	−3.4757	−3.5245	−3.5613	−3.5905	−3.6143	−3.6343	−3.6514
−3.0	.0013	−2.5332	−2.5879	−2.6264	−2.6555	−2.6785	−2.6973	−2.7130	−2.7265
−2.0	.0227	−1.6500	−1.6916	−1.7208	−1.7427	−1.7601	−1.7742	−1.7861	−1.7962
−1.0	.1586	−.7411	−.7732	−.7954	−.8121	−.8251	−.8358	−.8446	−.8522
−0	.5000	.2394	.2061	.1837	.1673	.1547	.1445	.1361	.1290
.1	.5398	.3439	.3094	.2864	.2696	.2566	.2462	.2376	.2304
.2	.5792	.4496	.4138	.3901	.3727	.3594	.3487	.3399	.3325
.3	.6179	.5565	.5193	.4946	.4767	.4629	.4519	.4428	.4352
.4	.6554	.6646	.6257	.6000	.5814	.5671	.5557	.5464	.5385
.5	.6914	.7737	.7331	.7063	.6869	.6721	.6602	.6505	.6424
.6	.7257	.8840	.8414	.8134	.7931	.7777	.7654	.7553	.7468
.7	.7580	.9951	.9505	.9211	.9000	.8839	.8710	.8605	.8517
.8	.7881	1.1072	1.0603	1.0296	1.0075	.9906	.9773	.9663	.9571
.9	.8159	1.2201	1.1708	1.1386	1.1155	1.0979	1.0839	1.0725	1.0630
1.0	.8413	1.3337	1.2820	1.2482	1.2240	1.2056	1.1911	1.1791	1.1692
1.1	.8643	1.4480	1.3937	1.3583	1.3330	1.3138	1.2986	1.2861	1.2757
1.2	.8849	1.5628	1.5059	1.4689	1.4424	1.4223	1.4064	1.3935	1.3826
1.3	.9031	1.6782	1.6186	1.5799	1.5522	1.5312	1.5146	1.5011	1.4898
1.4	.9192	1.7941	1.7317	1.6912	1.6623	1.6404	1.6231	1.6090	1.5972
1.5	.9331	1.9105	1.8452	1.8029	1.7727	1.7499	1.7319	1.7171	1.7049
1.6	.9452	2.0272	1.9590	1.9149	1.8834	1.8596	1.8408	1.8255	1.8127
1.7	.9554	2.1442	2.0731	2.0271	1.9944	1.9696	1.9500	1.9341	1.9208
1.8	.9640	2.2616	2.1875	2.1396	2.1055	2.0797	2.0594	2.0428	2.0290
1.9	.9712	2.3793	2.3021	2.2523	2.2169	2.1901	2.1689	2.1517	2.1374
2.0	.9772	2.4972	2.4170	2.3652	2.3284	2.3006	2.2786	2.2608	2.2459
2.1	.9821	2.6154	2.5320	2.4782	2.4401	2.4112	2.3885	2.3700	2.3545
2.2	.9860	2.7337	2.6472	2.5915	2.5519	2.5220	2.4984	2.4792	2.4633
2.3	.9892	2.8523	2.7626	2.7049	2.6639	2.6329	2.6085	2.5887	2.5721
2.4	.9918	2.9710	2.8782	2.8184	2.7759	2.7439	2.7187	2.6982	2.6810
2.5	.9937	3.0899	2.9938	2.9320	2.8881	2.8550	2.8290	2.8078	2.7901
2.6	.9953	3.2089	3.1096	3.0457	3.0004	2.9663	2.9393	2.9174	2.8992
2.7	.9965	3.3280	3.2255	3.1596	3.1128	3.0776	3.0498	3.0272	3.0084
2.8	.9974	3.4473	3.3416	3.2735	3.2253	3.1889	3.1603	3.1370	3.1176
2.9	.9981	3.5667	3.4577	3.3875	3.3378	3.3004	3.2709	3.2469	3.2269
3.0	.9986	3.6861	3.5738	3.5016	3.4505	3.4119	3.3815	3.3568	3.3362
3.1	.9990	3.8057	3.6901	3.6158	3.5631	3.5234	3.4922	3.4668	3.4456
3.2	.9993	3.9253	3.8064	3.7300	3.6759	3.6351	3.6030	3.5768	3.5551
3.3	.9995	4.0451	3.9228	3.8443	3.7887	3.7467	3.7137	3.6869	3.6646
3.4	.9996	4.1649	4.0393	3.9586	3.9015	3.8585	3.8246	3.7970	3.7741
3.5	.9997	4.2847	4.1558	4.0730	4.0144	3.9702	3.9355	3.9072	3.8837
3.6	.9998	4.4047	4.2724	4.1874	4.1273	4.0820	4.0464	4.0174	3.9933
3.7	.9998	4.5247	4.3891	4.3019	4.2403	4.1939	4.1573	4.1276	4.1029

TABLE A-5.—Concluded.

(d) Concluded.

Safety margin, S_M	Probability, P_x	Sample size, N							
		30	40	50	60	70	80	90	100
3.8	0.9999	4.6447	4.5057	4.4165	4.3433	4.3057	4.2683	4.2379	4.2126
3.9		4.7648	4.6224	4.5310	4.4664	4.4177	4.3793	4.3482	4.3223
4.0		4.8849	4.7392	4.6456	4.5795	4.5296	4.4904	4.4585	4.4320
4.1		5.0051	4.8560	4.7603	4.6926	4.6416	4.6014	4.5688	4.5417
4.2		5.1253	4.9728	4.8749	4.8057	4.7536	4.7125	4.6792	4.6516
4.3		5.2456	5.0897	4.9896	4.9189	4.8656	4.8237	4.7896	4.7612
4.4		5.3659	5.2066	5.1044	5.0321	4.9776	4.9348	4.9000	4.8710
4.5		5.4862	5.3235	5.2191	5.1453	5.0897	5.0460	5.0104	4.9809
4.6		5.6066	5.4405	5.3339	5.2585	5.2018	5.1572	5.1209	5.0907
4.7		5.7270	5.5575	5.4487	5.3718	5.3139	5.2684	5.2314	5.2006
4.8		5.8474	5.6745	5.5635	5.4851	5.4260	5.3796	5.3418	5.3104
4.9	1.0000	5.9679	5.7915	5.6784	5.5984	5.5882	5.4908	5.4523	5.4203
5.0		6.0883	5.9086	5.7932	5.7117	5.6503	5.6021	5.5628	5.5302
5.1		6.2088	6.0256	5.9081	5.8250	5.7625	5.7133	5.6734	5.6401
5.2		6.3294	6.1427	6.0230	5.9384	5.8747	5.8246	5.7839	5.7500
5.3		6.4499	6.2598	6.1379	6.0518	5.9869	5.9359	5.8945	5.8600
5.4		6.5705	6.3770	6.2528	6.1651	6.0991	6.0472	6.0050	5.9599
5.5		6.6911	6.4941	6.3678	6.2785	6.2113	6.1585	6.1156	6.0799
5.6		6.8117	6.6113	6.4828	6.3919	6.3236	6.2698	6.2262	6.1898
5.7		6.9323	6.7285	6.5977	6.5054	6.4358	6.3812	6.3368	6.2998
5.8		7.0529	6.8456	6.7127	6.6188	6.5481	6.4925	6.4474	6.4098
5.9		7.1735	6.9628	6.8277	6.7322	6.6694	6.6039	6.5580	6.5198
6.0		7.2942	7.0801	6.9427	6.8457	6.7727	6.7152	6.6686	6.6298
6.1		7.4149	7.1973	7.0577	6.9592	6.8850	6.8266	6.7792	6.7398
6.2		7.5356	7.3145	7.1728	7.0726	6.9973	6.9380	6.8899	6.8498
6.3		7.6563	7.4318	7.2878	7.1861	7.1096	7.0494	7.0005	6.9598
6.4		7.7770	7.5490	7.4029	7.2996	7.2219	7.1608	7.1112	7.0699
6.5		7.8978	7.6663	7.5179	7.4131	7.3342	7.2722	7.2218	7.1799
6.6		8.0185	7.7836	7.6330	7.5266	7.4466	7.3836	7.3325	7.2899
6.7		8.1393	7.9009	7.7481	7.6401	7.5589	7.4950	7.4432	7.4000
6.8		8.2600	8.0182	7.8632	7.7537	7.6712	7.6064	7.5538	7.5100
6.9		8.3808	8.1355	7.9783	7.8672	7.7836	7.7179	7.6645	7.6201
7.0		8.5016	8.2528	8.0934	7.9807	7.8960	7.8283	7.7752	7.7302
7.1		8.6224	8.3701	8.2085	8.0943	8.0083	7.9408	7.8859	7.8402
7.2		8.7432	8.4875	8.3236	8.2078	8.1207	8.0522	7.9966	7.9503
7.3		8.8640	8.6048	8.4387	8.3214	8.2331	8.1637	8.1073	8.0604
7.4		8.9848	8.7222	8.5538	8.4349	8.3455	8.2751	8.2180	8.1705
7.5		9.1056	8.8395	8.6690	8.5485	8.4579	8.3866	8.3287	8.2806
7.6		9.2264	8.9569	8.7841	8.6621	8.5702	8.4981	8.4394	8.3906
7.7		9.3473	9.0743	8.8992	8.7756	8.6826	8.6095	8.5502	8.5007
7.8		9.4681	9.1916	9.0144	8.8892	8.7950	8.7210	8.6609	8.6108
7.9		9.5890	9.3090	9.1296	9.0028	8.9075	8.8325	8.7716	8.7209
8.0		9.7098	9.4264	9.2447	9.1164	9.0199	8.9440	8.8823	8.8310

Appendix B
Project Manager's Guide to Risk Management and Product Assurance

Introduction

This appendix provides project managers with practical information about increasing the chances for project success by using the tools of risk management and product assurance. The elements of an effective product assurance program are described along with the benefits of using a product-assurance-oriented management approach to reduce project risk. The information should be especially useful to new project managers and to others concerned with specifying product assurance requirements or developing risk management or product assurance plans.

This appendix is written from the perspective of the NASA Glenn Research Center's Office of Safety and Assurance Technologies (OSAT). It begins with a general discussion of how OSAT supports projects at Glenn, including the roles and responsibilities of the project assurance lead. Then follows relevant discussions on reliability and quality assurance (R&QA) with respect to economics and requirements, performance-based contracting, and risk management. Finally, it describes frequently applied requirements from various product assurance disciplines. For project managers needing further information, a more comprehensive treatment of risk management and product assurance can be found in the references.

Risk Management and Product Assurance at the NASA Glenn Research Center

The NASA Glenn Office of Safety and Assurance Technologies advises the various project offices on risk management, safety, and product-assurance-related issues. Also, consistent with the NASA Policy Directive on safety and mission success (ref. B–1), OSAT conducts independent assessment activities to reduce risk. Typically, it is more actively involved in flight projects where the risks of failure are often greater and potentially more severe. However, risk management and product assurance tools can be applied to ground-based projects as well.

Flight projects at Glenn normally develop risk management and product assurance plans to define how they will manage risks and address the applicable product assurance requirements. For many Glenn flight projects, product assurance requirements are specified in the Glenn Standard Assurance Requirements and Guidelines for Experiments (ref. B–2).

The Office of Safety and Assurance Technologies helps Glenn project managers develop their risk management and product assurance plans and recommends ways to mitigate risks and meet applicable product assurance requirements. To this end, OSAT developed and maintains the Glenn Product Assurance Manual (ref. B–3), which contains numerous product assurance instructions that give suggestions for system safety, quality, reliability and maintainability, software, and materials and processes. Glenn projects often use these instructions as is or tailor them to meet specific needs.

Project Assurance Lead

Role

The project assurance lead is OSAT's principal point of contact with the project and serves as an important advisor to the project manager. The lead provides guidance and advice during the preparation of project, risk management, and product assurance plans; the generation of statements of work; the

review of bidders' proposals, and final contract negotiations. The project assurance lead, normally shown in the project organization chart in a staff position reporting to the project manager, works closely with the project office to ensure that risk management and product assurance activities are consistent with the uniqueness of the project and are as cost effective as possible.

Responsibilities

The project assurance lead helps the project manager identify and mitigate risks and ensures that product assurance principals are applied to the design, manufacture, test, handling, installation, and operation of the project. The lead identifies and provides the product assurance technical support needed to ensure that applicable risk, safety, reliability, maintainability, quality assurance, materials and processes, and software requirements are satisfied.

Economics of OSAT

Classical curves in figure B–1 show the relationship of product quality cost and operational cost to product quality. To achieve a very small percentage of product defects (high quality), product quality cost becomes extremely high. Conversely, if the percentage of defects is high (poor quality) operational cost becomes extremely high. The intersection of the two cost curves gives the optimum goal from a cost viewpoint. When finalizing product assurance requirements for a project, the project manager should keep the optimum cost goal in mind. However, from an engineering perspective, there may be some critical items for which additional safeguards must be established and the need for close risk control is mandatory. In this situation, economics is still an important consideration.

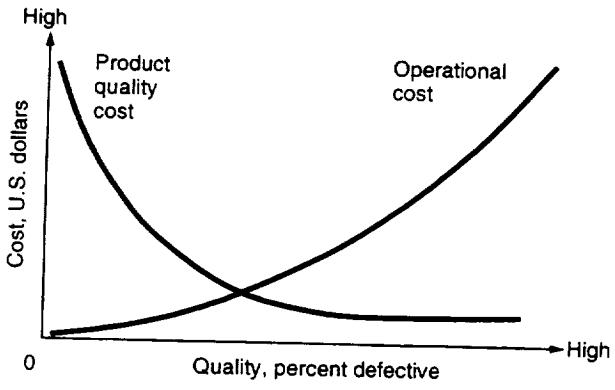


Figure B–1.—Relationship of product quality cost to operational cost.

Development of OSAT Requirements

Product assurance is a broad and diverse discipline that has overlapping authority with procurement, engineering, manufacturing, and testing. This problem has been mitigated to some degree at NASA Glenn by developing and using standard product assurance requirements where possible and by assigning experienced project assurance leads to assist projects in defining OSAT requirements.

The project assurance lead typically has an extensive OSAT background and can apply skills, training, and project experience to tailor product assurance requirements to be reasonable in scope and easily understood. In addition, the project assurance lead is responsible for assuring that the product assurance program is consistent with project objectives and that it can satisfy mission requirements.

To illustrate how product assurance requirements can be tailored, table B–1 lists the actual requirements imposed on 10 Glenn contracts and identifies the particular project phase associated with each contract.

Effect of Performance-Based Contracting

Even though the government has moved to performance-based contracting, a disciplined, organized approach to product assurance is still essential to minimize safety risks and to maximize chances for mission success. Although the government seeks to avoid imposing "how to" requirements on performance-based contractors, these contractors still should follow good product assurance practices. To verify their doing so, the government develops and implements surveillance plans to obtain information about performance. This verification is accomplished primarily through "insight" rather than through the more traditional "oversight." (Insight relies on reviewing contractor-generated data and minimizes the amount of direct government involvement; conversely, oversight is more intrusive because it normally involves direct government monitoring of contractor processes and activities.)

Risk Management and Product Assurance Plans

NASA programs and projects are required to use risk management as an integral part of their management process (ref. B–4). This requirement includes developing and implementing a risk management plan to identify, analyze, mitigate, track, and control program and/or project risks as part of a continuous risk management process (ref. B–5).

TABLE B-1.—RELIABILITY AND QUALITY ASSURANCE REQUIREMENTS IMPOSED ON VARIOUS PROGRAM TYPES

[Composite wind turbine blades, C; global air sampling program, G; lift/cruise fan, L; materials for advanced turbine engines, M; electrical power processor, P; quiet, clean, short-haul experimental engine, Q; JT8D refan engines, R; space experiments, S; variable-cycle engine, V; 200-kW wind turbine generators, W.]

Requirement	Aeronautics				Space		Energy	
	Study	Advanced technology	Development	Flight	Development	Flight	Development	Operational
Reliability program plan					P			
Reliability program control						S		
Reliability program reporting						S		
Reliability training						S		
Supplier control						S		
Reliability of Government-furnished property						S		
Design specifications					P			
Reliability prediction				G				
Failure mode and effects analysis						S		
Maintainability and human-induced failures	L							
Design reviews				G				
Failure reporting and corrective action			Q	R,G		S		
Standardization of design practices						S		
Parts program					P			W
Reliability evaluation plan						S		
Testing					P			
Reliability assessment						S		
Reliability inputs to readiness review						S		
Reliability evaluation program reviews						S		
Quality status reporting						S		
Government audits; quality program audits			Q	R				W
Quality program plan			Q	R				W
Technical documents; quality support/design reviews		M					C	
Change control			Q	R,G				
Identification control			Q	R,G		S		
Data retrieval						S		
Source selection		M	Q	R,G			C	W
Procurement documents			Q	R,G			C	
Quality assurance at source			Q	R				W
Receiving inspection		M	Q	R,G		S		
Receiving inspection records		M	Q	R,G		S		
Supplier rating system						S		
Postaward surveys						S		
Coordinate supplier inspection and tests						S		
Nonconformance information feedback								
Fabrication operations			Q	R,G				
Article and material control		M	Q	R		S	C	W
Cleanliness control							C	W
Process control			Q	R,G			C	W
Workmanship standards		M					C	

TABLE B-1.—Concluded.

Requirement	Aeronautics				Space		Energy	
	Study	Advanced technology	Development	Flight	Development	Flight	Development	Operational
Inspection and test planning			Q	R				
Inspection records; inspection and test performance		M	Q	R,G		S	C	W
Contractor quality control actions						S		
Nonconformance control		M	Q	R,G		S	C	
Nonconformance documentation		M	Q	R		S	C	
Failure analysis and corrective action		M	Q	R,G		S		
Material review			Q	R		G	C	W
Material review board			Q	R		S		
Contracting officer approval						S		
Supplier material review board						S		
Inspection of test equipment and standards								
Evaluation of standards and test equipment		M				S		
Measurement accuracy								
Calibration accuracy		M				S		
Calibration control	V	M	Q	R,G		S	C	W
Environmental requirements				R		S		
Remedial and preventive action (calibration)								
Stamp control system			Q	R				W
Stamp restriction						S		
Handling and storage			Q	R,G		S		W
Preserving, marking, packaging, and packing			Q	R		S		W,C
Shipping						S		
Sampling plans				R				
Statistical planning and analysis				G		S		
Contractor's responsibility for Government property			Q	R				W
Unsuitable Government property			Q	R,G				W

At NASA Glenn, OSAT serves as a risk management consultant to the project manager by offering OSAT risk management training, helping to prepare risk management and/or product assurance plans, conducting risk assessments, helping to track risks, and providing other valuable support to facilitate the risk management process.

An effective product assurance program is an essential ingredient for successfully managing risks. It provides the framework and discipline needed to support a structured risk management approach, a characteristic of many successful projects. The project manager can rely on an effective product assurance program to help mitigate risks in many key areas and thereby serve as an important risk management tool.

Development and Implementation of Product Assurance Plans

As part of an overall risk reduction strategy, Glenn projects and contractors develop and implement product assurance plans to define and perform the tasks necessary to satisfy

applicable product assurance requirements. The plans are intended to establish a disciplined, organized approach to product assurance, thereby minimizing safety risks and maximizing the chances for mission or project success.

The product assurance plan normally includes a description of assurance activities in the areas or disciplines discussed next.

Assurance Reviews

Assurance reviews help to ensure that the engineering development and documentation have sufficiently progressed and that the design and hardware are sufficiently mature to justify moving to the next phase of the project. These reviews ultimately require the project to demonstrate that the components, subsystems, and system can successfully perform their intended function under flightlike operating and environmental conditions.

Verification Plan

As part of its product assurance effort, the project develops a verification plan to describe the tests, analyses, and inspections to be conducted to demonstrate hardware and/or software functionality and ability to safely survive expected environmental extremes. The purpose of the verification program is to ensure that the payload and/or experiment meets all specified mission requirements. This activity includes verifying that the design complies with the requirements and that the hardware/software complies with the mission.

Verification testing includes functional and environmental tests to demonstrate the ability to meet performance requirements. Environmental tests consist of thermal cycling, random vibration, and electromagnetic interference (EMI). Note that environmental stress screening is an effective product assurance tool that project managers can use to verify the adequacy of system design and workmanship.

System Safety

System safety is a critical element in the product assurance plan. Each project must develop and implement a comprehensive system safety program to ensure project compliance with all applicable safety requirements, both flight and ground. Potential safety hazards must be identified and controlled to reduce the risk of injuring personnel or damaging equipment.

The Office of Safety and Assurance Technologies provides direct safety support or consultation to guide projects through the NASA safety review process (refs. B–6 to 11); it helps projects determine the best design solution to meet specific safety requirements, conducts hazard analyses, generates hazard reports, develops safety compliance data packages, supports safety reviews, and resolves safety issues with integration centers or payload safety review panels.

Materials and Processes

To assure safety and promote mission success, projects must exercise care in the selection, processing, inspection, and testing of materials. Prudent project managers invoke a comprehensive materials and processes (M&P) program to ensure that materials meet applicable requirements for flammability, toxic off-gassing, vacuum out-gassing, corrosion, fluid compatibility, and shelf-life control. This program and the associated M&P assurance activities are documented in the product assurance plan.

Projects prepare material identification and usage lists (MIUL's) and attempt to use compliant materials to the maximum extent possible. Regarding materials usage, projects work with and seek the advice of OSAT in several ways: justification for the use of a noncompliant material for a particular application and its selection for that application; preparation of material usage agreements (MUA's) that contain the rationale for using any noncompliant materials; assurances that fabrication and other manufacturing processes be performed in accordance with accepted practices or approved procedures; and the issuance of a materials certification letter, in concert with the applicable NASA Materials and Processes Inter-Center Agreement, when the materials and processes used by the project are shown to be acceptable.

Some applications require the certification of metallic and nonmetallic materials to assure that the chemical and physical properties of the materials are compatible with the design requirements. After materials are selected by the engineer and are precisely defined by a specification (Federal, Society of Automotive Engineers, American Society for Testing and Materials, or other available standards), the purchase order for steels, aluminum alloys, brass, welding rods, solder, metal coatings, gases, and potting compounds should require that a test report, a certificate of conformance (fig. B–2), or both accompany the vendor's shipment. In addition to the vendor's certificate, it may be necessary to conduct periodic in-house tests of metallic and nonmetallic materials to assure their continued conformance.

Quality Assurance

Quality assurance (QA), another critical element of an effective product assurance program, is documented in the product assurance plan and helps a project establish and satisfy quality requirements through all phases of the project life cycle. Quality assurance (1) promotes discipline, encouraging projects to design in quality and ensure good workmanship by using

CAST TECHNOLOGY INCORPORATED
1482 ERIE BOULEVARD
SCHENECTADY, NEW YORK 12305

CASTECH

LABORATORY REPORT OF CHEMICAL ANALYSIS AND MECHANICAL TESTS
(Job 1365)

SOLD TO
Financo Division (MS500-302)
NASA-Lewis Research Center
21000 Brookpark Road
Cleveland, Ohio 44135

SHIPPED TO
NASA-Lewis Research Center
21000 Brookpark Road
Cleveland, Ohio 44135

SPECIFICATION	PURCHASE ORDER NO.	PART NO.	DATE SHIPPED
AMS 5391 (Inco 713 LC)	NAS 3-14991	CR655729	8/27/87
SHIPPER NO. A 2987	NO. PCS. 1 pc	HEAT NO. V4271	

CHEMICAL ANALYSIS

HEAT NO.	C	Mn	Si	P	S	Ni	Cr	B	Mo	Cu	Fe	Co	Cb Ta	Ti	Al
V4271	.05	.05*	.05*	.015*	.006	Bal	12.0	.005	4.76	.05*	.05*	.08	1.96	.76	6.17
	* Less than							Zr -	.10						

MECHANICAL TESTS

HEAT NO.	TEST TEMP. °F.	TENSILE STRENGTH PSI	YIELD STRENGTH PSI	RUPTURE STRESS, PSI	RUPTURE LIFE, HRS.	E LONG. % IN 1"	R. A., %	ROCKWELL C HARDNESS AS CAST	ROCKWELL C HARDNESS AGED

As Cast
S/N 2 * OK PER PRINT & PM 9-14-87

DATE 8/27/87
Subscribed to and sworn before me
William W. Latimer
NOTARY PUBLIC

WILLIAM W. LATIMER
Notary Public in State of New York
Qualified in Schenectady County
My Commission Expires March 30, 1989

We hereby certify that the above data is a true copy of the data resulting from tests performed in our laboratory or of the data furnished us by the laboratory performing the tests.

BY Cara Mauro
C. Mauro
AUTHORIZED AGENT

CTI-22 (11-66)

Figure B–2.—Typical materical certification.

proper controls during design, fabrication, assembly and test; (2) ensures that hardware and software conform to design requirements and that documentation accurately reflect those requirements; and (3) ensures that flight hardware be maintained in a sufficiently clean environment to prevent exposure to any contaminants that could degrade performance and possibly compromise the achievement of mission objectives.

OSAT assists projects in developing effective quality management systems to address areas such as configuration control, procurement, fabrication, inspection, electrostatic discharge control, and nonconformance control. It also performs quality audits of fabrication sources, establishes inspection requirements, provides inspection and/or test monitoring services, makes dispositions for nonconforming material, and ensures that facilities maintain proper environmental controls. Project managers should be familiar with the good QA practices cited in the following sections.

Review of Drawings

Before releasing the engineering drawings to the manufacturer, design engineers may avail themselves of the technical services provided by quality engineers when developing specification callouts in the note section of the drawings (fig. B–3). Give precise information on materials, surface finish, processing, nondestructive testing, cleanliness, identification, packaging. Special instructions and notes are important in obtaining a quality product.

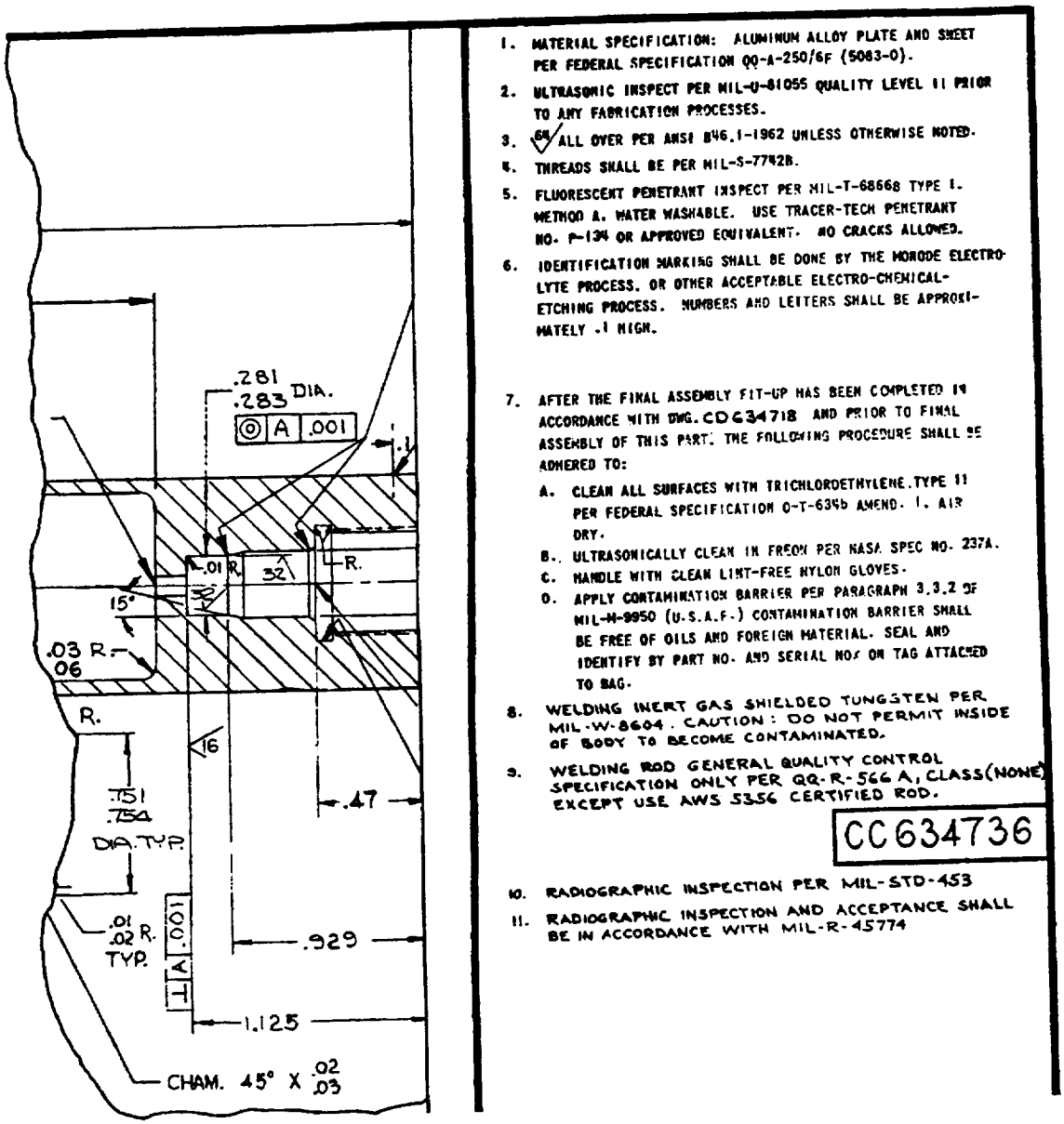


Figure B–3.—Typical drawing specifications.

Changes in Engineering Documents

Early in the design phase, establish a system to control changes (fig. B–4) in engineering documents and to remove obsolete documents. Changes in released drawings, specifications, test procedures, and related documents can be critical, particularly during the building and testing phases. For this reason, process the latest engineering data early to expedite their distribution to the participating line organizations.

Use of a Process Plan

Identify in a plan (fig. B–5) the manufacturing operations that must be performed in a particular sequence. The most commonly used processes are machining, mechanical fastening, grinding, brazing, welding, soldering, polishing, coating, plating, radiography, ultrasonics, fluorescent penetrate inspection, magnetic particle inspection, painting, bonding, heat treating, identification marking, and safety wiring.

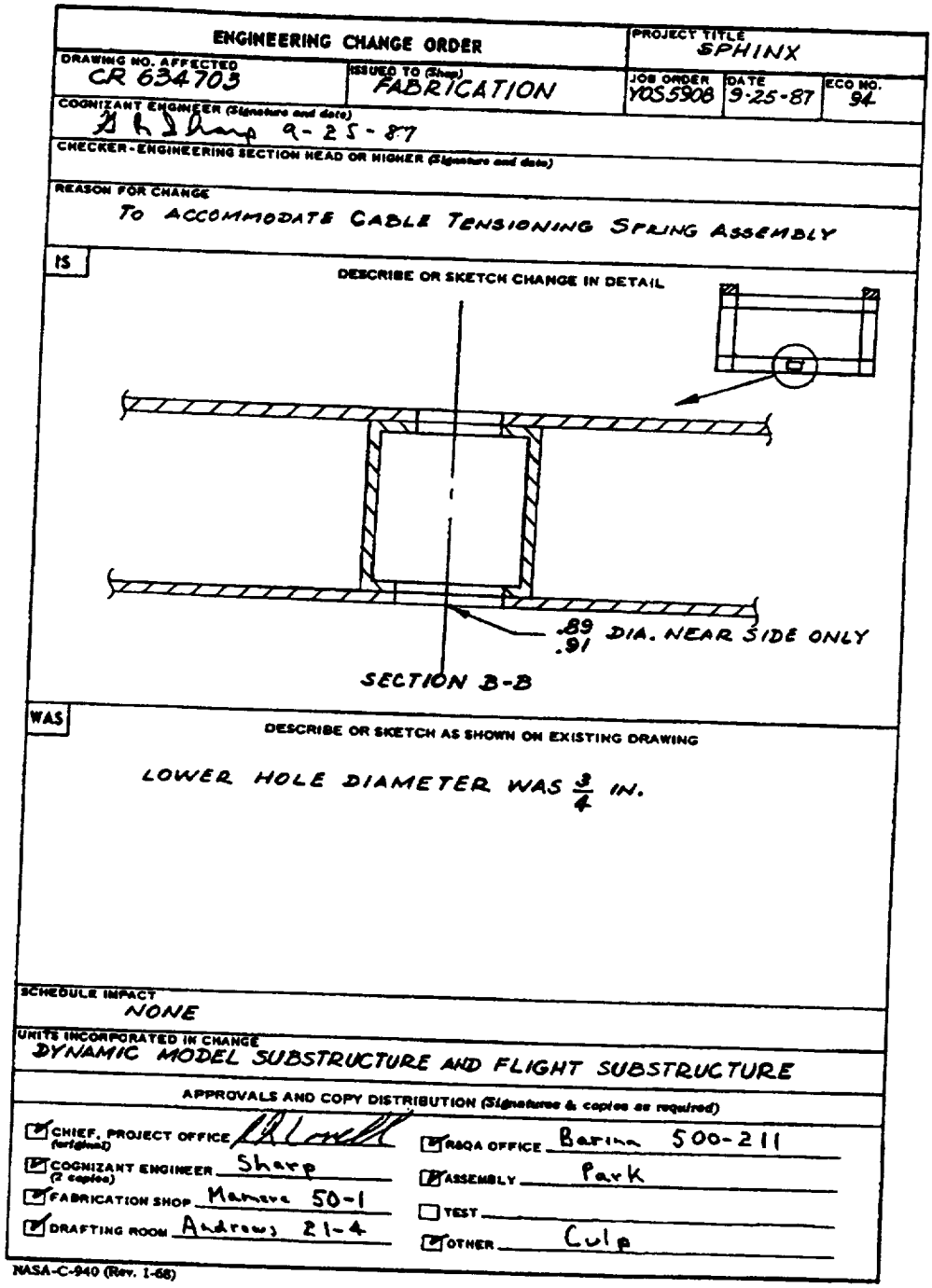
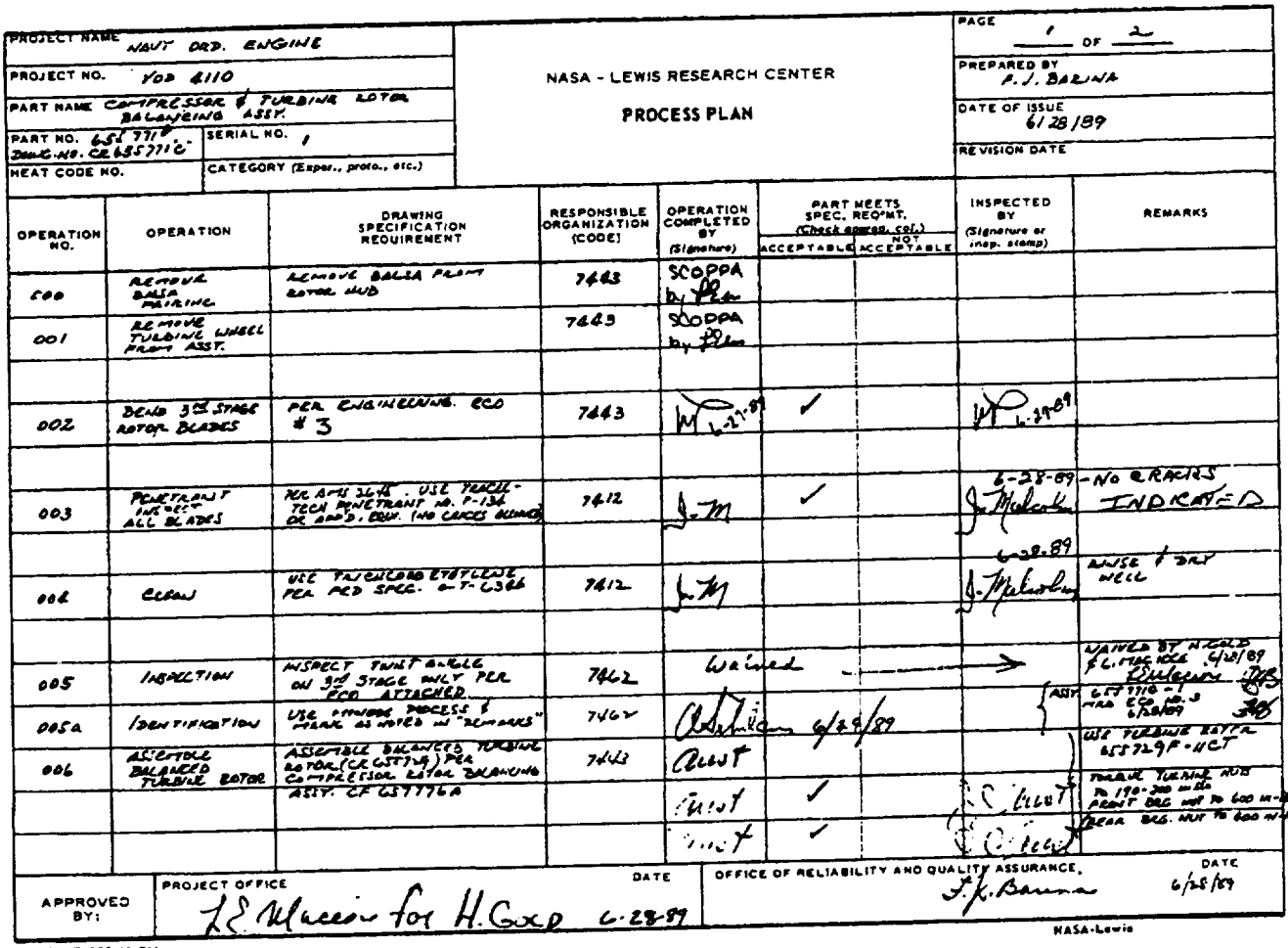


Figure B–4.—Typical engineering change order.



NASA - LEWIS RESEARCH CENTER

PROCESS PLAN

PROJECT NAME: NAVY ORD. ENGINE
PROJECT NO.: YOD 4110
PART NAME: COMPRESSOR & TURBINE ROTOR BALANCING ASSY.
PART NO.: 655 771 / DWG. NO. CR 655771C
SERIAL NO.: 1
HEAT CODE NO.:
CATEGORY (Exper., proto., etc.):
PAGE: 1 OF 2
PREPARED BY: A. J. BARINA
DATE OF ISSUE: 6/28/89
REVISION DATE:

OPERATION NO.	OPERATION	DRAWING SPECIFICATION REQUIREMENT	RESPONSIBLE ORGANIZATION (CODE)	OPERATION COMPLETED BY (Signature)	PART MEETS SPEC. REQ'MT. (Check appropr. col.) ACCEPTABLE	NOT ACCEPTABLE	INSPECTED BY (Signature or insp. stamp)	REMARKS
000	REMOVE BALSA MARKING	REMOVE BALSA FROM ROTOR HUB	7443	SCOPPA by [illegible]				
001	REMOVE TURBINE WHEEL FRONT ASST.		7443	SCOPPA by [illegible]				
002	BEND 3RD STAGE ROTOR BLADES	PER ENGINEERING ECO # 3	7443	[illegible] 6-29-89	✓		[illegible] 6-29-89	
003	PENETRANT INSPECT ALL BLADES	PER AMS 2648. USE TURCITE-TECH PENETRANT NO. P-136 OR APPD. EQUIV. (NO CARCO ALLOWED)	7412	J. M	✓		6-28-89 J. Malcolm	6-28-89 - NO CRACKS INDICATED
							6-28-89	
004	CLEAN	USE TRICHLOROETHYLENE PER MCD SPEC. C-T-C386	7412	J. M			J. Malcolm	RINSE & DRY WELL
005	INSPECTION	INSPECT TUNT BALLS ON 3rd STAGE BLT PER ECO ATTACHED	7462	Waived	→			WAIVED BY N. GOLD F.C. 6/28/89 [illegible]
005A	IDENTIFICATION	USE STANDARD PROCESS & MARK AS NOTED IN "REMARKS"	7462	[illegible] 6/29/89			ASST. 655 771C-1 PER ECO NO. 3 6/28/89	
006	ASSEMBLE BALANCED TURBINE ROTOR	ASSEMBLE BALANCED TURBINE ROTOR (CR 655771C) PER COMPRESSOR ROTOR BALANCING ASSY. CF 655771A	7443	[illegible]			[illegible]	USE TURBINE ROTOR 655729F-11CT
				[illegible]	✓		[illegible]	MARK TURBINE HUB IN 190-200 in-lbs. FRONT BRG. NUT TO 600 in-lbs. REAR BRG. NUT TO 600 in-lbs
				[illegible]	✓		[illegible]	

APPROVED BY: PROJECT OFFICE [illegible] for H. Gold 6-28-89 DATE

OFFICE OF RELIABILITY AND QUALITY ASSURANCE, F. K. Barina DATE 6/28/89

NASA-Lewis

NASA-C-902 (2-71)

Figure B–5.—Typical process plan.

Calibration of Measuring Devices

Calibrate instruments when physical quantities are to be measured with any degree of accuracy. Calibration includes repair, periodic (recall) maintenance, and determination of the accuracy (adjustments made as required) of the measuring devices as compared with known standards from the National Institute of Standards and Technology. Figure B–6 shows a typical certificate of calibration.

Inspection of Hardware Items

Quality control inspectors check in-process items against acceptable quality standards and engineering documents (fig. B–7). Minor deviations from good quality practices are normally resolved at the worksite; otherwise they are brought to the attention of the inspection supervisor. If the quality standard being violated is not contained in an engineering document, the supervisor may review the inspector's decision if risks are involved. If the discrepancy is a characteristic defined by an engineering document, the final decision is made by material review engineering and product assurance representatives or the material review board.

Nonconformance of Hardware

When hardware is to be built, some provision must be made for the orderly review and disposition of all items that are determined by inspection or test as not conforming to the drawing, specification, or workmanship requirements. The system most frequently used comprises two procedures:

(1) An engineer or a product assurance representative is authorized to review and decide whether hardware can be reworked into a conforming condition without an engineering change, an instruction, or both.

(2) The material review board reviews hardware that cannot be reworked to meet the engineering specifications. The board consists of engineering, product assurance, and when required, government representatives. In difficult situations, the board members consult with other organizations and persons to arrive at the minimum-risk decision.

WESTERN AUTOMATIC TEST SERVICES
991 Commercial Street
Palo Alto, California, 94303
(415) 328-6096

CERTIFICATE OF CALIBRATION

TO:Litton Industries
960 Industrial Way
San Carlos, CA

DATE: 21 July 1988

Reference: Your Order No. 49721

WATS Order No. 8526

TO WHOM IT MAY CONCERN:

The equipment listed below has been duly calibrated by Wavecom Industries/WATS Group per your instructions.

Wavecom Industries/WATS Group calibration measurements are traceable to the National Bureau of Standards to the extent allowed by the Bureau's Calibration facilities.

WAVECOM INDUSTRIES/WATS Group

Quantity	Description	Serial No.
1	NASA INPUT SYSTEM	#1
1	NASA INPUT SYSTEM	#2
1	NASA OUTPUT SYSTEM	#1
1	NASA OUTPUT SYSTEM	#2

LITTON 57

N. Cazares
July 22, 1988

PROVISION OF WAVECOM INDUSTRIES, SOLID STATE CIRCUITS DIVISION 687 N. PASTORIA, SUNNYVALE CALIFORNIA 94086 •245-6031

Figure B–6.—Typical certificate of calibration.

L-5394 S/N 2029 — BODY SECTIONS ASSEMBLY — PROCEDURE CFPA-171

NASA 200 WATTS — CENTER SECTION ASSY DATA SHEET — PAGE 6 OF 6

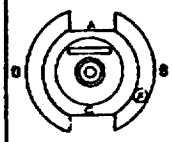
PARA NO.	DESCRIPTION	DATE	TECH	ENGR	QC
1	DEBURRING AND INSPECTION UNDER SCOPE OF ALL PARTS	1/20/5	BJ		
2	LAYOUT OF CIRCUIT PARTS ON CIRCUIT LAYOUT SHEET	1/24/5	BJ		NC
3	COLD TEST DATA (A) RETURN LOSS FREQUENCY @ 5db DOWN 12019.4 MHz (B) NOMINAL RETURN LOSS 23 db (C) WORST SPIKE 22 db @ 12100 MHz (D) I_L @ 12038 MHz 3.2 db, I_L @ 12080 MHz 2.2 db I_L @ 12123 MHz 1.8 db, I_L = 20 db @ 11989.7 MHz COMMENTS: (X) MARK AT OUTPUT	1/24/5	BJ	CLJ	NC
4	CLEAN PARTS PER LBPC-171	2-3-5	DG		
5	INSPECT PARTS BEFORE STACKING	2-3-5	DG		NC
6	STACK CIRCUIT PARTS ON BRAZING FIXTURE	2-3-5	DG		
7	MEASUREMENT OF CIRCUIT HEIGHTS BEFORE BRAZE (WITHOUT ALLOY)  A 1.7235 B 1.7236 ↑ C 1.7232 ↓ D 1.7232 ↓ Δ = .0004	2-3-5	DG		JD
8	REMOVE 0.048" ϕ CERAMIC ROD QC VERIFY ORIENTATION AND BRAZING FIXTURE NUMBERS FURNACE TYPE Lindy H2 NO 9 SOAK DURATION 1.5 MIN. TAP POSITION 5	2-5-5 2-5-5 2-5-5	DG DG		JD
9	CONDITION OF ALLOY AFTER BRAZE GOOD MEASUREMENT OF CIRCUIT HEIGHTS AFTER BRAZE Δ = .0006 A 1.7239 ↓ B 1.7241 C 1.7245 ↑ D 1.7241 RECORD THE DIFFERENCE BETWEEN PARAGRAPH 7 AND 9 A +.0004 B +.0005 C +.0013 D +.0009	2-5-5	DG		NC
10	SIZE OF MANDREL DROPPED THROUGH BEAM HOLE .048 + INCH	2-5-5	DG		JD
11	VERIFY PERPENDICULARITY .002 INCHES OFF VERTICAL @ .650 INCHES. MAXIMUM RUN OUT .001 INCHES @ .470 INCHES FROM TOP OF SPACER (X MARK UP)	2-4-5	DG		NC
12	LEAK CHECK (N/A)	2/4/5	MM		
13	FINAL COLD TEST DATA (A) RETURN LOSS FREQUENCY @ 5db DOWN 12004.3 MHz (B) NOMINAL RETURN LOSS 21 db (C) WORST SPIKE 19 db @ 12100 MHz (D) I_L @ 12038 MHz 2.0 db I_L @ 12080 MHz 1.5 db I_L @ 12123 MHz 1.5 db I_L = 20 db @ 11965.0 MHz COMMENTS: (X) MARK AT OUTPUT	2/4/5	BJ	CLJ	NC
14	DISPOSITION OF ASSEMBLY USE: ✓ REJECT: DISPOSITION IF REJECT: LITTON 5	2/4/5		CLJ	NC

Figure B–7.—Typical mandatory quality control inspection points.

Documentation of Equipment Discrepancies

In a design, certain characteristics are distinct, describable, and measurable in engineering units. Critical characteristics are generally identified by engineering documents and are closely controlled by quality assurance personnel. Whenever a design characteristic is determined to be nonconforming to released engineering requirements, one of the following reporting procedures must be followed:

(1) A minor discrepancy is recorded in a discrepancy log (fig. B–8). A disposition must be made by an engineer, an inspector, or both if the condition is a minor discrepancy (e.g., a scratch on a metal surface or excess material) that does not adversely affect form, fit, or function and the hardware can be used "as is" or reworked to engineering requirements.

(2) A failure discrepancy report is written and a disposition is obtained through the engineering review board (ERB) if a mechanical, electrical, or electronic system or subsystem has failed to perform within the limits of a critical characteristic identified by an engineering drawing, specification, test procedure, or related engineering document.

Quality Assurance Documentation of Production, Inspection, and Test Operations

Manufacturing, inspecting, testing, and related operations for major assemblies and subassemblies should be documented for several reasons. Such documentation can provide a status record of the work in progress and the work completed. Also, it can become a part of the permanent record of production, inspection, and test operations. The sophistication of the format and the entries in the log can be adjusted to suit the type of contract—research, development, or production. The chronological entries in the log can be summarized and included in an acceptance data package, which contains information helpful to review during a contractor's acceptance of a supplier's equipment or during final Government acceptance of a contract end item. Figure B–9 shows a checklist used to determine if an item conforms to specifications.

VEHICLE SERIAL #6632 SUBSYSTEM TELEMETRY SHEET 1 OF 10

ITEM NO.	PA STAMP	DATE DISC.	DESCRIPTION OF DISCREPANCY	RESP CODE	STA ASSEMBLY	QUAD	METHOD OF CORRECTION	COMPL DATE	ORG STAMP	PA STAMP
1	LMSC 57	4/27/66	Accelerometer P/N 161596-1 (S/N 109) cable breakout leads have insufficient bend radii. Read Assurance Std 1-6.4 requires a min. radius of 3 times outside diameter of lead.	44-52	102	IV	Remove five spot ties at breakout points with diagonal cutting pliers tool # re-dress to conform with PA Std 1-6.4	4/28/66	4/28/66	LMSC 63

FORM LMSC 1257 S

LOCKHEED MISSILES & SPACE COMPANY

LOG SHEET CLEARED BY: Otto Evans ORGN. 66-53

Figure B–8.—Typical discrepancy log.

2.0 Quality assurance checklist for conformance to specifications of Communications Technology Satellite (CTS) output stage tube (OST)

OST S/N: 2021 Classification: QTM-2(QF-2)

2.1 Overall efficiency

Specification: 50 percent minimum over CTS band of 12.038 to 12.123 GHz, at saturation	Actual: 40.7 percent minimum at 12.040 GHz. Out of specification. (Waiver required.)

2.2 Center frequency

Specification: 12.0805 GHz	Actual: 12.0805 GHz

2.3 RF power output

Specification: 200 W minimum at saturation over CTS band of 12.038 to 12.123 GHz	Actual: 170 W minimum at 12.040 GHz. Out of specification. (Waiver required)

2.4 Small signal bandwidth

Specification: 3 dB maximum peak to peak measured at 10 dB below peak saturation over the CTS band, 12.038 to 12.123 GHz	Actual: 2.4 dB maximum peak to peak

Figure B–9.—Checklist for item conformance to specifications.

Safety and Mission Assurance for Suppliers of Materials and Services

Materials and services acquired by the user from outside sources must satisfy contract, Government, or company reliability and quality assurance requirements. The user's system of control should involve

(1) Selecting acceptable or qualified sources
(2) Performing surveys and audits of the supplier's facilities
(3) Inspecting the received supplier's products
(4) Reporting and taking corrective action for problems that occur

Reliability and Maintainability

An effective reliability and maintainability program (R&M) can ensure that a project's hardware and software meet mission design life and availability requirements. The R&M program is documented in the project's product assurance plan and includes tests, analyses, and other assurance activities to demonstrate that the project can meet the reliability and availability goals established. The program may also include maintainability analyses or demonstrations to show that equipment can be adequately maintained based on expected component failure rates.

Several ways that OSAT assists and works with projects to ensure that hardware and software meet R&M requirements are by conducting failure mode, effects, and criticality analyses (see the next section); developing reliability models; making reliability predictions; conducting reliability trade studies; providing component selection and control design guidelines; conducting analyses to identify the root causes of failures; implementing design changes to improve reliability and maintainability; developing maintenance concepts; performing spare parts analyses; and developing plans (e.g., preventative maintenance) to address maintainability requirements.

The fundamental objective of a failure mode, effects, and criticality analysis is to identify the critical failure areas in a design. To accomplish this identification, each functional component (or higher level if adequate to attain the intended purpose) is sequentially assumed to fail, and the broad effects of each such failure on the operation of the system (fig. B–10) are traced. More details on this subject are available in the LeR–W0510.060 ISO Work Instruction.

Solar Array Failure Mode and Effects Analysis of Mounting and Mechanical Deployment Assembly for Space Electric Rocket Test II

Component	Failure mode	Cause	Effect	Criticality	Action	Status
Actuator assembly	Binding	Needle valve plugged	Degraded deployment	Minor	Spring stiffness adequacy and tolerances reviewed; tests carefully evaluated	Completed
	Operation is erratic	Tolerance buildup; O-ring damage; workmanship	Partial deployment	Major	Workmanship inspected	Specified
	Actuation stops	Spring failure	No deployment	Critical	Data packages will be prepared	Planned
Linkage (mechanism assembly)	Motion stops prematurely	Binding and lockup	Partial deployment	Major	Kinematics study disclosed source of binding; redesigned	Completed
		Design weakness; poor workmanship; damage	Slow deployment	Minor	Confidence tests will verify elimination of failure mode	Planned
Pin-puller assembly	Tie-rod is not released	Excessive load; squib failure; corrosion of pin puller; jamming of catch	Solar array does not deploy	Critical	Need study to develop alternative design with adequate redundancy	Open
Mechanical assembly	Attachment point of solar arrays to Agena bends or breaks	Excessive loads	Partial deployment	Major	Cold gas attitude control system to be programmed; low mode to avoid excessive load	Planned
	Hinges bind spring	Workmanship	Slow deployment	Minor	Confidence tests	Planned
		Tolerance stackup			Tolerances reviewed	Completed

Figure B–10.—Typical failure mode and effects analysis.

EEE Parts Control

The electronic, electrical, and electromechanical (EEE) parts used by a project can have a major impact on its safety and reliability. The project must be sure that the EEE parts selected and used are appropriate for their application and offer the lowest safety risk and greatest chance for mission success based on cost and schedule constraints. Projects must plan and implement an EEE parts control program consistent with reliability requirements and good engineering practice.

The OSAT helps projects select parts and develop EEE parts identification lists. Also, it verifies that parts selected comply with de-rating guidelines and other requirements (e.g., radiation); conducts Alert searches in conjunction with the Government Industry Data Exchange Program and NASA Parts Advisories to identify and deal with potentially unreliable parts; and assists with parts screening, ensuring traceability and analyzing part failures (see the following sections).

Selection and Screening

The costs incurred during subsystem and system testing are inversely proportional to the money spent for examining and testing parts. Success is directly related to the part screening costs. For example, the exceptional operational life of the Space Electric Rocket Test II satellite is no doubt attributable to the extensive parts selection and screening program.

Other factors influence parts selection and screening: the criticality of the hardware application, unusual environments, contractor experience, and in-house resources. The selection can range from a high-reliability part (identified in a Government- or industry-preferred parts handbook) to an off-the-shelf commercial part. Screening is a selective process as called out in the source control document.

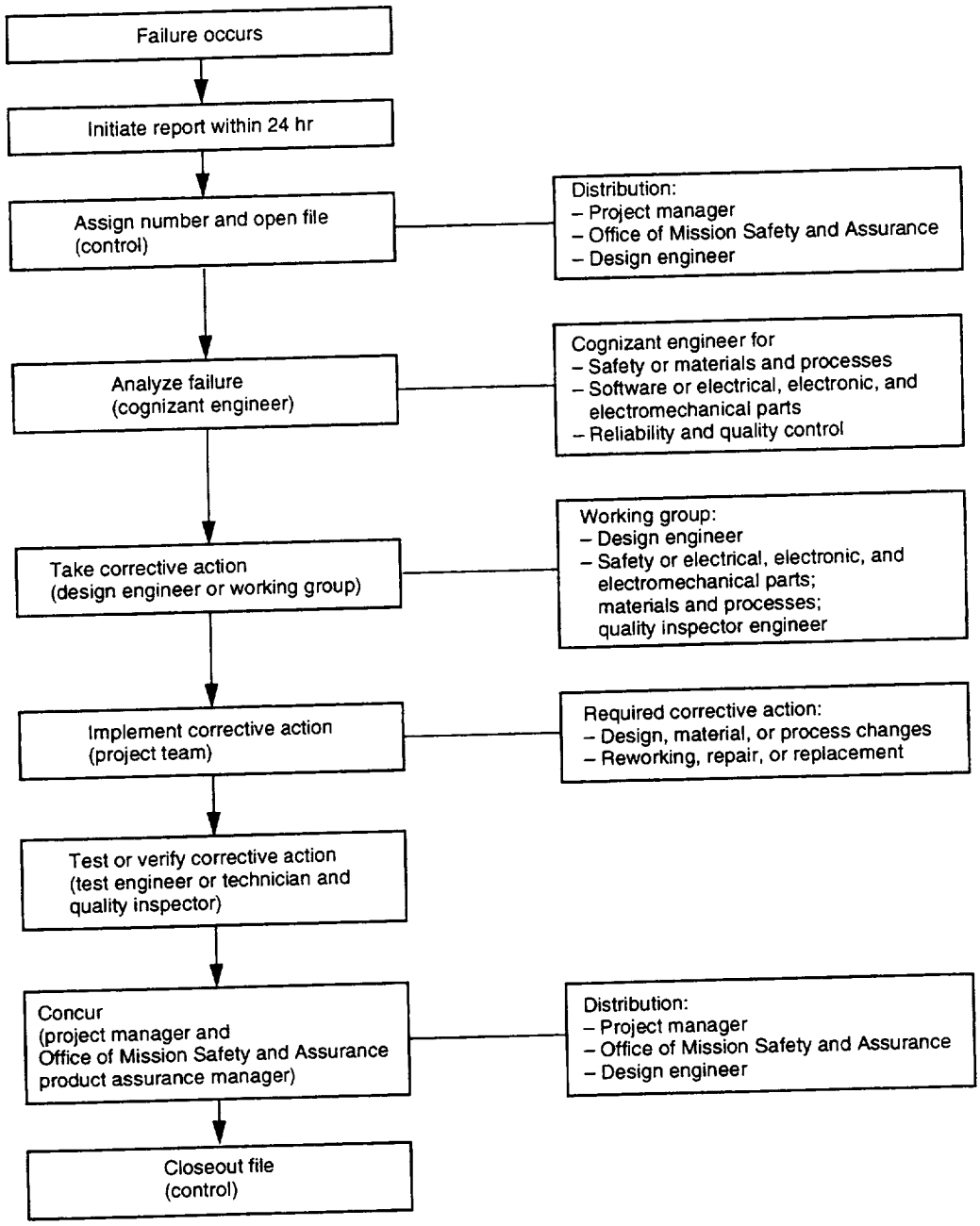


Figure B–11.—Failure report, analysis, and corrective action flowchart.

Materials Identification

Good engineering practice identifies parts, components, and materials with a part number, a screening serial number, a date code, and the manufacturer. Furthermore, the marking on parts and components should be affixed in a location that is easily seen when the item is installed in an assembly. The identification method and location on the item are included on a drawing, a specification, or other associated engineering document (fig. B–3, note 6). During the period of fabrication, assembly, and testing, the system of marking and recordkeeping should provide a way to trace backward from an end item to the part or material level.

Failure Analysis

Some failed parts are analyzed and investigated to determine the cause of the failure (fig. B–11). Corrective action is taken to assure that the problem does not recur and then the action is verified by testing. The problem is closed by ERB review. Sometimes corrective action may change a component

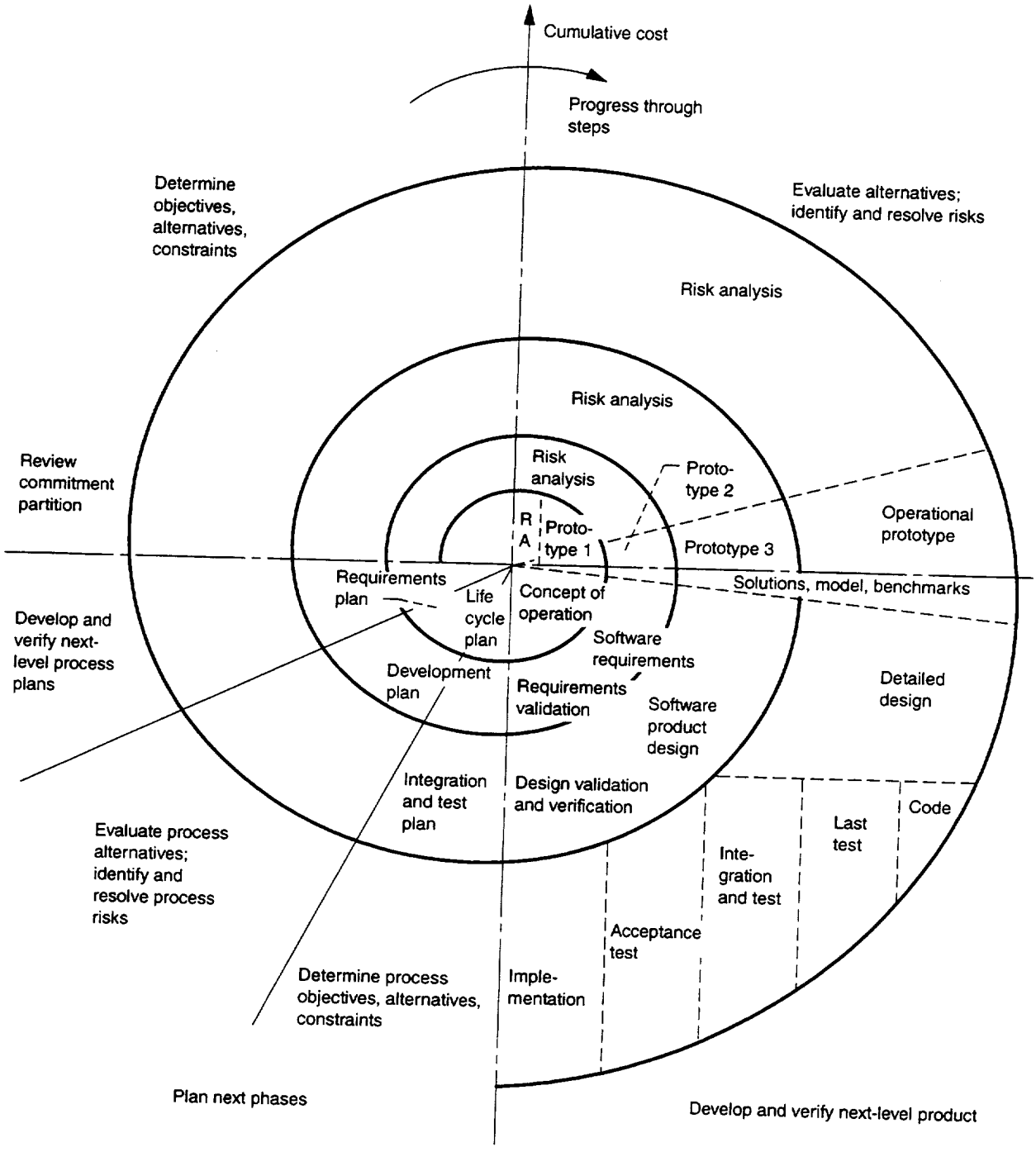


Figure B–12.—Spiral software development life cycle.

application criterion, improve a packaging technique, or revise a test procedure. Often the detailed physical and chemical examination reveals that a refinement is needed in the materials used during the manufacturing of a part or that an improvement in the parts screening process is necessary.

Software Product Assurance

Software is generally a critical element in the safety and success of a project. Project managers are therefore wise to establish an effective software assurance program to ensure the safety, reliability, and quality of their software systems. Such a program includes a software assurance plan (typically part of the product assurance plan) to address software quality standards, configuration management, testing, problem reporting, performance verification, certification process, and mission simulation.

The software product assurance (SPA) effort is intended to ensure that all software hazards be identified and controlled, that the software be capable of meeting mission availability and design life requirements, that the software meet all performance requirements for the mission simulation, and that software documentation accurately reflect those requirements (fig. B–12).

The OSAT can help projects develop and implement an effective SPA process. For example, it can prepare SPA plans and conduct software hazard analyses, failure tolerance analyses, and audits. It ensures that projects follow proper software configuration management practices. In addition, it witnesses or monitors software tests and verifies that results conform to expectations.

Conclusion

Project managers can realize many benefits by using risk management tools and a product-assurance-oriented approach to their projects. By applying effective product assurance techniques throughout the project life cycle, projects can achieve the highest level of safety, quality, and reliability for the available resources. The investment that project managers make to apply risk management and product assurance to their projects offers the probable return of increased mission safety and a greater probability of success. Experienced project managers consider this to be a wise investment.

References

B–1. NASA Policy for Safety and Mission Success. NPD 8700.1, NASA, June 12, 1997.

B–2. Standard Assurance Requirements and Guidelines for Experiments. LeR-M 0510.002, NASA Glenn Research Center, Apr. 1996.

B–3. NASA Glenn Product Assurance Manual. LeR-M 0510.001, NASA Glenn Research Center, Dec. 18, 1996.

B–4. NASA Program and Project Management Processes and Requirements. NPG 7120.5A, NASA, Apr. 3, 1998.

B–5. Continuous Risk Management Guidebook, Carnegie Mellon University, Pittsburgh, PA, 1997.

B–6. NASA Johnson Space Center Payload Safety Home Page (http://wwwsrqa.jsc.nasa.gov/pce/pcehome.htm).

B–7. Safety Policy and Requirements for Payloads Using the Space Transportation System. NSTS 1700.7B, (Change No. 5, Oct. 12, 1998).

B–8. Safety Policy Requirements for Payloads Using the International Space Station. NSTS 1700.7B ISS Addendum, Dec. 1995.

B–9. Space Shuttle Payload Ground Safety Handbook. KHB 1700.7 Rev. B, Sept. 1992.

B–10. Payload Safety Review and Data Submittal Requirements. NSTS/ISS 13830 Rev C, July 1998.

B–11. Guidelines for the Preparation of Flight Safety Data Packages and Hazard Reports for Payloads Using the Space Shuttle. JSC 26943, Feb. 1995.

Appendix C
Reliability Testing Examples

A great deal of work has been done by various researchers to develop probabilistic methods suitable for reliability problems (ref. C–1). Probabilistic methods that apply discrete and continuous random variables to user problems are not as well covered in the literature.

This appendix concentrates on four useful functions: (1) failure $f(t)$, (2) reliability $R(t)$, (3) failure rate λ, and (4) hazard rate λ'. Because we usually need to know how well a point estimate has been defined, some consideration is given to confidence intervals for these functions. The appendix also explains methods for planning events at the critical delivery milestone and closes with a brief explanation of two reliability case histories.

Useful Distribution Functions

The failure function $f(t)$, which defines failure as a function of time or number of cycles, is important knowledge obtained from reliability testing. Failure records are kept on a particular piece of hardware to obtain a histogram of failures against time. This histogram is studied to determine which failure distribution fits the existing data best. Once a function $f(t)$ is obtained, reliability analysis can proceed. In many cases, sufficient time is not available to obtain large quantities of failure density function data. In these cases, experience can be used to determine which failure frequency function best fits a given set of data. Table C–1 lists seven distributions, five continuous and two discrete. These distributions can be used to describe the time-to-failure functions for various components. The derivation of the four reliability functions for the seven listed distributions is explained in the next section (ref. C–2).

Derivation of $Q(t)$, $R(t)$, λ, and λ' functions.—The unreliability function $Q(t)$ is the probability that in a random trial the random variable is not greater than t; hence,

$$Q(t) = \int_0^t p(t)\,\mathrm{d}t$$

When time is the variable, the usual range is 0 to t, implying that the process operates for some finite time interval. This integral is used to define the unreliability function when failures are being considered.

The reliability function $R(t)$ is given by

$$R(t) = 1 - Q(t)$$

In integral form $R(t)$ is given by

$$R(t) = \int_t^\infty p(t)\,\mathrm{d}t$$

Differentiation yields

$$\frac{\mathrm{d}\,R(t)}{\mathrm{d}\,t} = \frac{\mathrm{d}\,Q(t)}{\mathrm{d}\,t} = -p(t)$$

The a posteriori probability of failure p_f in a given time interval, t_1 to t_2, can be calculated by using these equations and is given by

$$p_f = \frac{1}{R(t_1)}\left[\int_{t_1}^{t_2} p(t)\,\mathrm{d}t\right]$$

$$= \frac{1}{R(t_1)}\left[\int_{t_1}^{\infty} p(t)\,\mathrm{d}t - \int_{t_2}^{\infty} p(t)\,\mathrm{d}t\right]$$

TABLE C-1.—FIT DATA FOR FAILURE FUNCTIONS

Distribution	Failure fit
Continuous distribution	
Exponential	Complex electrical systems
Normal	Mechanical systems subject to wear
Weibull	Mechanical, electromechanical, or electrical parts: bearings, linkages with fatigue loads, relays, capacitors, and semiconductors. Reduces to exponential distribution if $\alpha = t$, $\beta = 1$, and $\gamma = 0$
Gamma	Combined mechanical and electrical systems
Log normal	Mechanical parts under stress rupture loading
Discrete distribution	
Poisson	One-shot parts
Binomial	Complex electrical systems for probability of N_f defects

Substituting and simplifying gives

$$p_f = 1 - \frac{R(t_2)}{R(t_1)}$$

The rate at which failures occur in a time interval is defined as the ratio of the probability of failure in the interval to the interval length. Thus, the equation for failure rate λ is given by

$$\lambda = \frac{R(t_1) - R(t_2)}{(t_2 - t_1)R(t_1)} = \frac{1}{t_2 - t_1}\left[1 - \frac{R(t_2)}{R(t_1)}\right]$$

Substituting $t_1 = t$ and $t_2 = t + h$ into this equation gives

$$\lambda = \frac{R(t) - R(t+h)}{(t+h-t)R(t)} = \frac{R(t) - R(t+h)}{hR(t)}$$

The instantaneous failure rate in reliability literature is often called the hazard rate. The hazard rate λ' is by definition the limit of the failure rate as $h \to 0$. Using a previous equation and taking the limit of the failure rate as $h \to 0$ gives

$$\lambda' = \lim_{h \to 0} \frac{R(t) - R(t+h)}{hR(t)}$$

Letting $h = \Delta t$ in this equation gives

$$\lambda' = \lim_{\Delta t \to 0} -\frac{1}{R(t)}\left[\frac{R(t+\Delta t) - R(t)}{\Delta t}\right]$$

The term in brackets is recognized from the calculus to be the derivation of $R(t)$ with respect to time, and the negative of this derivation is equal to $p(t)$. Substituting these values gives

$$\lambda' = -\frac{1}{R(t)}\left[\frac{\mathrm{d}\,R(t)}{\mathrm{d}\,t}\right] = \frac{p(t)}{R(t)}$$

As an example, consider a jet airplane traveling from Cleveland to Miami. This distance is about 1500 miles and could be covered in about 2.5 hr. The average rate of speed would be 1500 miles divided by 2.5 hr, or 600 mph. The instantaneous speed may have varied anywhere from 0 to 700 mph. The air speed at any given instant could be determined by reading the speed indicator in the cockpit. Replacing the distance continuum by failures, failure rate is analogous to average speed, 600 mph in this example, and hazard rate is analogous to instantaneous speed, the speed indicator reading in this example.

Figure C–1 presents a summary of the useful frequency functions for the failure distributions given in table C–1. These functions were derived by using the defining equations given previously. Choose any failure function and verify that $R(t)$, λ, and λ' are properly defined by going through the derivation yourself. Five reliability problems using the continuous distributions given in figure C–1 are solved in the next section.

Estimation using the exponential, normal, Weibull, gamma, and log normal distributions.—As an illustration of how to use these equations for an electrical part that experience indicates will follow the exponential distribution, consider example 1.

Example 1: Testing of a particular tantalum capacitor showed that the failure density function was exponentially distributed. For the 100 specimens tested, it was found that the mean time between failures $\bar{t}$ was 1000 hr.

(1) What is the hazard rate?

(2) What is the failure rate at 100 hr and during the next 10-hr interval?

(3) What are the failure and reliability time functions?

Solution 1:

(1) Using the equations given in figure C–1 for exponential distribution, the hazard rate is given by

$$\lambda' = \frac{1}{\bar{t}} = \frac{1}{1000 \text{ hr/failure}}$$

or

$$\lambda' = 1 \times 10^{-3} \text{ failure/hr}$$

(2) The failure rate is given by

$$\lambda = \frac{1}{h}\left(1 - \frac{e^{-t_2/t}}{e^{-t_1/t}}\right)$$

For this case the time interval is given by

$$h = t_2 - t_1 = 110 - 100 = 10 \text{ hr}$$

The necessary reliability functions are given by

$$e^{-t_2/t} = e^{-110/1000} = e^{-0.11} = 0.896$$

and

$$e^{-t_1/t} = e^{-110/1000} = e^{-0.1} = 0.905$$

Substituting these values gives

$$\lambda = \frac{1}{10}\left(1 - \frac{0.896}{0.905}\right) = 1 \times 10^{-3} \text{ failure/hr}$$

This is to be expected for the exponential case because the failure rate is constant with time and is always equal to the hazard rate.

(3) The failure and reliability time functions are given by

$$p(t) = \frac{1}{1000} e^{-t/1000}$$

$$R(t) = e^{-t/1000}$$

As an illustration of how to use the equations given in figure C–1 for mechanical parts subject to wear using the normal distribution, consider example 2.

Example 2: A gimbal actuator is being used where friction, mechanical loading, and temperature are the principal failure-causing stresses. Assume that tests to failure have been conducted on the mechanical parts, resulting in the data shown in table C–2.

(1) What is the mean time between failures and the standard deviation?

(2) What are the hazard rate at 85 300 hr and the failure rate during the next 10 300-hr interval?

(3) What are the failure and reliability time functions?

Solution 2:

(1) The mean time between failures is given by

TABLE C-2.—TEST DATA FOR GIMBAL ACTUATORS

Ordered sample number	Time to failure, t_f, hr	Time to failure squared, t_f^2, $(10^3 \text{ hr})^2$
1	60×10^3	3600
2	65	4225
3	68	4624
4	70	4900
5	75	5625
6	75	5625
7	80	6400
8	83	6889
9	85	7225
10	90	8100
Total	750×10^3	57 213

$$\bar{t} = \frac{\sum_{f=1}^{n} t_f}{n}$$

where

$\bar{t}$ mean time between failures, hr
t_f time to failure, hr
n number of observations

Therefore, using the data from table C–2,

$$\bar{t} = \frac{750\,000}{10} = 75\,000 \text{ hr}$$

The unbiased standard deviation σ is given by

$$\sigma = \left[\frac{\sum_{f=1}^{n} t_f^2 - \frac{\left(\sum_{f=1}^{n} t_f\right)^2}{n}}{n-1}\right]^{1/2}$$

The sum terms required for this calculation are given by

$$\sum_{f=1}^{n} t_f^2 = 57\,213 \left(10^3 \text{ hr}\right)^2 \text{(column 3, table C–2)}$$

Distribution	$p(t)$	$R(t)$
Exponential	$\frac{1}{\bar{t}} \exp(-t/\bar{t})$	$\exp(-t/\bar{t})$
Normal	$\frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{t-\bar{t}}{\sigma}\right)^2\right]$	$\frac{1}{\sigma\sqrt{2\pi}} \int_t^\infty \exp\left[-\frac{1}{2}\left(\frac{t-\bar{t}}{\sigma}\right)^2\right] dt$
Weibull	$\frac{\beta}{\alpha}\left(\frac{t-\gamma}{\alpha}\right)^{\beta-1} \cdot \exp\left[-\left(\frac{t-\gamma}{\alpha}\right)^\beta\right]$	$\exp\left[-\left(\frac{t-\gamma}{\alpha}\right)^\beta\right]$
Gamma	$\frac{1}{\alpha^\beta \Gamma(\beta)} (t-\gamma)^{\beta-1} \exp\left[\frac{-(t-\gamma)}{\sigma}\right]$	$\frac{1}{\alpha^\beta \Gamma(\beta)} \int_t^\infty (t-\gamma)^{\beta-1} \exp\left[\frac{-(t-\gamma)}{\sigma}\right] dt$
Log normal	$\frac{1}{t'\sigma_{t'}\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{t'-\bar{t}'}{\sigma t'}\right)^2\right]$	$\frac{1}{\sigma_{t'}\sqrt{2\pi}} \int_{t'}^\infty \exp\left[-\frac{1}{2}\left(\frac{t'-\bar{t}'}{\sigma_{t'}}\right)^2\right] dt'$
Distribution	$p(N_f)$	$R(N_f)$
Poisson	$\frac{(t/\bar{t})^{N_f} \exp(-t/\bar{t})}{N_f!}$	$\sum_{j=N_f}^{0} \frac{(t/\bar{t})^j \exp(-t/\bar{t})}{j!}$
Binomial	$\frac{n!}{(n-N_f)!\,N_f!} p^{N_f} g^{n-N_f}$	$\sum_{j=N_f}^{n} \frac{n!}{(n-j)!\,j!} p^j g^{n-j}$

Figure C–1.—Summary of useful frequency functions.

λ	λ'	Remarks
$\frac{1}{h}\left[1-\frac{\exp\left(\frac{-t_2}{\bar{t}}\right)}{\exp\left(\frac{-t_1}{\bar{t}}\right)}\right]$	$1/\bar{t}$	$h = t_2 - t_1$ Complex electrical systems
$\frac{1}{h}\left[1-\frac{R(t_2)}{R(t_1)}\right]$	$\frac{\text{Normal ordinate at } t}{\text{Normal area } t_1 \text{ to } \infty}$	Mechanical systems
$\frac{1}{h}\left\{1-\frac{\exp\left[-\frac{(t_2-\gamma)^\beta}{\alpha}\right]}{\exp\left[-\frac{(t_1-\gamma)^\beta}{\alpha}\right]}\right\}$	$\frac{\beta}{\alpha}(t-\gamma)^{\beta-1}$	α = scale parameter β = shape parameter γ = location parameter Mechanical or electrical systems. If $\alpha = t$, $\beta = 0$, and $\gamma = 0$, reduces to exponential. If $\beta = 3.5$, approximates normal.
$\frac{1}{h}\left\{1-\frac{(t_2-\gamma)^{\beta-1}\exp\left[\frac{-(t_2-\gamma)^\beta}{\alpha}\right]}{(t_1-\gamma)^{\beta-1}\exp\left[\frac{-(t_1-\gamma)^\beta}{\alpha}\right]}\right\}$	$\frac{\text{Gamma ordinate at } t}{\text{Gamma area } t_1 \text{ to } \infty}$	Same as Weibull parameters but may be harder to use. $\Gamma(\beta) = \int_0^\infty t^{\beta-1}e^{-t}\,dt$ $\Gamma(\beta) = (\beta-1)\Gamma(\beta-1)$ Combined mechanical and electrical systems
$\frac{1}{h}\left[1-\frac{R(t_2)}{R(t_1)}\right]$	$\frac{\text{Log normal ordinate at } t}{\text{Log normal area } t_1 \text{ to } \infty}$	Mechanical parts that fail due to some wearout mechanism
λ	λ'	Remarks
Not applicable	Not applicable	N_f = number of failures One-shot devices
Not applicable	Not applicable	p = defectives g = effectives n = trials (sample size) Complex systems for probability of N_f defects

Figure C–1.—Concluded.

and

$$\left(\sum_{f=1}^{n} t_f\right)^2 = (750)^2 = 562\,500\left(10^3 \text{ hr}\right)^2$$

$$\sigma = \left(\frac{57\,213 - 56\,250}{9}\right)^{1/2}\left(\frac{963}{9}\right)^{1/2} = 10\,300 \text{ hr}$$

(2) The hazard rate λ' is given by

$$\lambda' = \frac{\text{Scaled ordinate at 85 300 hr}}{\text{Normal area from 85 300 hr to } \infty}$$

Let Y_1 be the normal ordinate at 85 300 hr and Z_1 be the standardized normal variable, which is given by

$$Z_1 = \frac{t - \bar{t}}{\sigma} = \frac{(85\,300 - 75\,000)\text{ hr}}{10\,300 \text{ hr}}$$

Existing tables for the normal ordinate values for $Z = 1.0$ gives $Y_1' = 0.242$. The scale constant K_s to modify this ordinate value for this problem is given by (ref. C–3)

$$K_s = \frac{n\theta}{\sigma}$$

where θ is the class interval. Substituting values and solving for Y_1 gives

$$Y_1 = f(t_1) = K_s Y_1' = \frac{10 \times 1 \text{ failures}}{10\,300 \text{ hr}} \times 0.242$$

$$= 2.35 \times 10^{-4} \text{ failure/ hr}$$

Note that the denominator required to calculate λ' is $R(t_1)$, which is the normal area from 85 300 hr to ∞. Existing tables for the normal area for $Z_1 = 1.0$ (ref. C–3) give the area from $-\infty$ to Z_1, so that the unreliability $Q(t_1)$ is given by

$$Q(t_1) = 0.841 \times (\text{Area from } -\infty \text{ to } Z_1)$$

Because $Q(t_1) + R(t_1) = 1.000$,

$$R(t_1) = 1.000 - 0.841 = 0.159$$

and the hazard rate is given by

$$\lambda' = \frac{2.35 \times 10^{-4} \text{ failure/ hr}}{1.59 \times 10^{-1}} = 1.47 \times 10^{-3} \text{ failure/hr}$$

The failure rate is given by

$$\lambda = \frac{1}{h}\left[1 - \frac{R(t_2)}{R(t_1)}\right]$$

In this case h is given as 10 300 hr. The reliability at 95 600 hr is given by

$$R(t_2) = \text{Normal area from 95 600 hr to } \infty$$

Using the preceding procedure results in

$$R(t_2) = 0.023$$

Substituting values gives

$$\lambda = \frac{1}{10\,300 \text{ hr}}\left(1 - \frac{0.023}{0.159}\right) = \frac{8.56 \times 10^{-1}}{1.03 \times 10^4}$$

$$= 8.31 \times 10^{-5} \text{ failure/ hr}$$

(3) The constants required to write expressions for $p(t)$ and $R(t)$ are calculated as follows:

$$\frac{1}{\sigma(2\pi)^{1/2}} = \frac{1}{\left(1.03 \times 10^4\right) \times 2.52} = 3.87 \times 10^{-5}$$

$$2\sigma^2 = 2 \times \left(1.03 \times 10^4\right)^2 = 2.12 \times 10^8$$

Using the constants and substituting values gives

$$p(t) = 3.87 \times 10^{-5}\, e^{-\left(t - 7.5 \times 10^4\right)^2 / 2.12 \times 10^8}$$

$$R(t) = 3.87 \times 10^{-5} \int_t^{\infty} e^{-\left(t - 7.5 \times 10^4\right)^2 / 2.12 \times 10^8}$$

As an illustration for the Weibull distribution, consider example 3.

Example 3: A lot of 100 stepping motors was tested to see what their reliability functions were. A power supply furnished electrical pulses to each motor. Instrumentation recorded the

TABLE C-3.—WEIBULL DATA FOR STEPPING MOTORS

Number of steps to failure	Cumulative number of failures		Median rank	5-Percent rank	95-Percent rank
	Problem 3	Problem 9	Scaled time to failure, $\hat{t}_i$		
0.2×10^3	2	1	6.70	0.51	25.89
.4	4	2	16.23	3.68	39.42
.9	5	3	25.86	8.73	50.69
4.0	16	4	35.51	15.00	60.66
10.0	20	5	45.17	22.24	69.65
18.0	50	6	54.83	30.35	77.76
30.9	90	7	64.49	39.34	85.00
50.0	97	8	74.14	49.30	91.27

number of continuous steps a motor made before it failed to step even though a pulse was provided. All testing was stopped at 1×10^6 steps. The step failure data are given in table C–3.

(1) Calculate the frequency functions.
(2) Plot the hazard rate function on log-log paper.
(3) What conclusions can be drawn from this graph?

Solution 3: Because there are 100 motors in this lot, the data give ordered plotting positions suitable for plotting on Weibull probability paper. Figure C–2 shows a plot of these data. From the shape of the data in figure C–2, it appears as though two straight lines are necessary to fit this failure density function. This means that different frequency functions exist at different times. These frequency functions are said to be separated by a partition parameter δ.

From figure C–2 the Weibull scale, shape, and location parameters can be estimated by following these steps:

(1) Estimate the partition parameter δ. This estimate can be obtained directly from figure C–2. The two straight lines that best fit the given data intersect at point *f*. Projecting this point down to the abscissa gives a failure age of 10 000 cycles for the partition parameter δ.

(2) Estimate the location parameter γ. This parameter is used as a straightener for $p(t)$. Because $p(t-0)$ is already a straight line for both regions, it is clear that $\gamma_1 = \gamma_2 = 0$. In general, several tries at straightening may be required before the one yielding a straight line for $p(t-\gamma)$ is found.

(3) Estimate the shaping parameter β. The intercept point *a* for line *b*, drawn parallel to line *c* and passing through point *d*, where $\ln(t-\gamma) = 1$ is equal to β. Thus, $\beta_1 = 0.75$ and $\beta_2 = 1.50$.

(4) Estimate the scale parameter α. At point *e* for line *c*,

$$\ln\alpha = -\ln\ln\frac{1}{1-Q(t)}$$

so that

$$\alpha = \exp\left[-\ln\ln\frac{1}{1-Q(t)}\right]$$

Therefore,

$$\alpha_1 = e^{2.75} = 15.7$$

$$\alpha_2 = e^{4.6} = 100$$

By using the parameters just estimated and the equations given in figure C–1 for the Weibull distribution, the following failure frequency functions can be expressed: The partition limits on the number of steps *c* are $0 \le c \le 10$ and $c > 10$. The frequency functions are given by

$$f(c) = \frac{\beta}{\alpha}(c-\gamma)^{\beta-1}\, e^{-(c-\gamma)^{\beta/\alpha}}$$

Substituting values results in

$$f_1(c) = \frac{0.75}{15.7}c^{0.75-1}\, e^{-(c/15.7)^{0.75}}$$

or

$$f_1(c) = 0.47c^{-0.25}\, e^{-c^{0.75/15.7}} \qquad \text{for } 0 \le c \le 10$$

Similarly,

$$f_2(c) = 0.015c^{-0.5}\, e^{-c^{1.50/100}} \qquad \text{for } c > 10$$

The reliability functions are given by

$$R(c) = e^{-(c-\gamma)^{\beta/\alpha}}$$

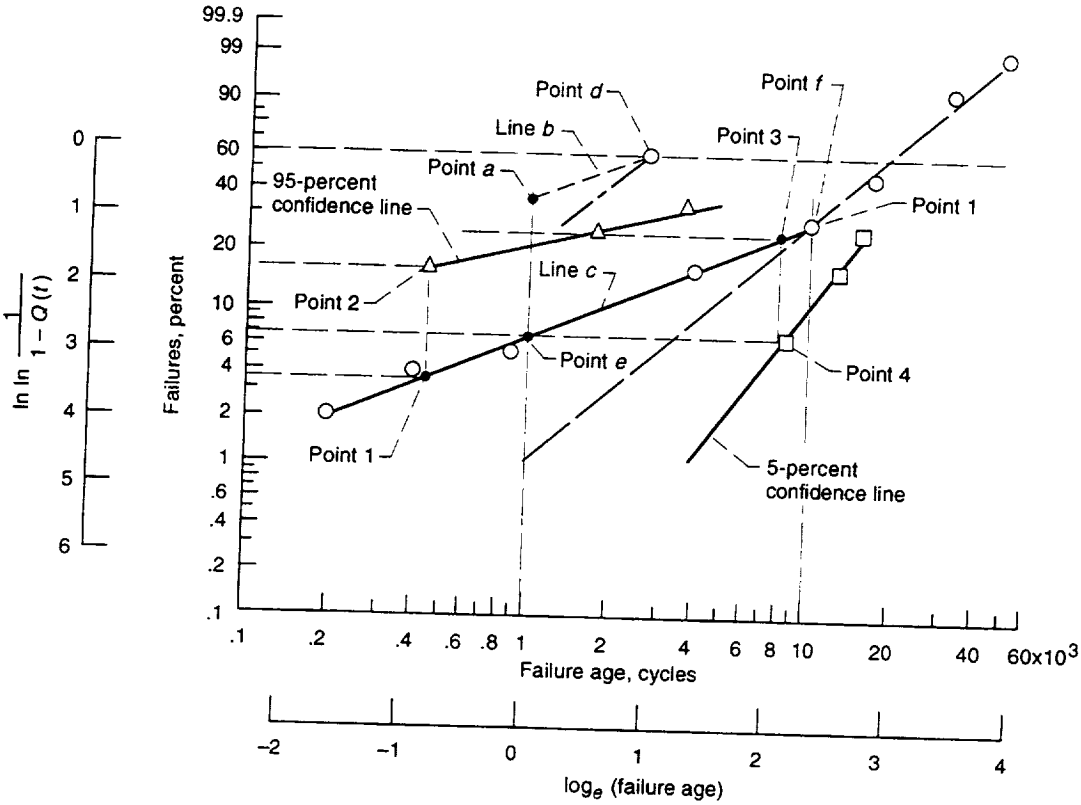


Figure C–2.—Weibull plot for stepping motors.

Therefore, substituting values gives

$$R_1(t) = e^{-c^{0.75/15.7}} \qquad \text{for } 0 \le c \le 10$$

and

$$R_2(t) = e^{-c^{1.5/100}} \qquad \text{for } c > 10$$

The failure rate functions are given by

$$\lambda = \frac{1}{h}\left[1 - \frac{e^{-(c_2-\gamma_1)^{\beta_1/\alpha_1}}}{e^{-(c_1-\gamma_1)^{\beta_1/\alpha_1}}}\right]$$

Therefore, substituting values gives

$$\lambda_1 = \frac{1}{h}\left[1 - \frac{e^{-(c_2)^{0.75/100}}}{e^{-(c_1)^{0.75/100}}}\right] \qquad \text{for } 0 \le c \le 10$$

and

$$\lambda_2 = \frac{1}{h}\left[1 - \frac{e^{-(c_2)^{1.5/100}}}{e^{-(c_1)^{1.5/100}}}\right] \qquad \text{for } c > 10$$

The hazard rate functions are given by

$$\lambda' = \frac{\beta}{\alpha}(c-\gamma)^{\beta-1}$$

Therefore, substituting values gives

$$\lambda_1' = 0.047c^{-0.25} \qquad \text{for } 0 \le c \le 10$$

and

$$\lambda_2' = 0.015c^{-0.5} \qquad \text{for } c > 10$$

(2) By using two-cycle log-log paper and the following calculation method, a graph of λ' against c can be obtained:

$$\lambda_1' = 0.047c^{-0.25}$$

Taking logarithms to the base 10 gives

$$\log \lambda_1' = \log 0.047 + (-0.25)\log c$$

Useful corollary equations are

$$10^x = y$$

$$x = \log Y$$

$$10^0 = 1$$

and

$$\log 0.047 = \log 4.7 \times 10^{-2} = \log 4.7 + (-2)\log 10$$
$$= \bar{2}.672, \text{ or } 8.672 - 10$$

For $c = 1$,

$$\log \lambda_1' = \log 0.047 + (-0.25)\log 1$$

$$\lambda_1' = 0.047$$

For $c = 10$,

$$\log \lambda_1' = \log 0.047 + (0.25)\log 10 = 2.672 - 0.25 = 2.422$$

$$\lambda_1' = 0.0264$$

In a similar manner solving for λ_2' gives the data points shown in table C–4. These data are plotted in figure C–3.

TABLE C–4.—HAZARD RATE DATA FOR STEPPING MOTORS

Number of steps, c	Failures per cycle, λ'
1×10^3	0.047
10	.026
10	.015
100	.150

(3) Figure C–3 indicates that the hazard rate is decreasing by 0.25 during the first interval and is increasing by 0.50 during the second interval for each logarithmic unit change of c. It appears that step motors, for first misses, jump from the "infant mortality" stage into the wearout stage without any transition period of random failures with a constant failure rate (ref. C–4).

As an illustration of combined mechanical and electrical systems that follow the gamma distribution, consider example 4:

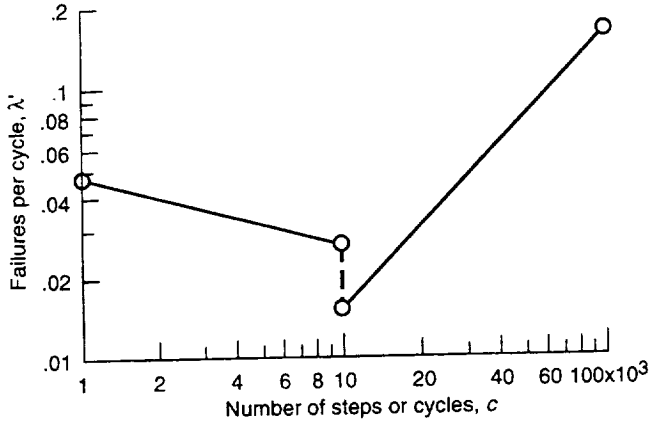


Figure C–3.—Hazard rate plot for stepping motors.

TABLE C-5.—ELECTRIC ROCKET RELIABILITY DATA

Ordered sample number	Time to failure, t_f, hr	Median rank	Scaled time to failure	Linear scale rank
		Scaled time to failure, $\hat{t}_i$		
1	1 037.8	6.70	7.2	5.0
2	1 814.4	16.23	12.6	15.0
3	2 332.8	25.86	16.3	25.0
4	3 124.8	35.51	21.7	35.0
5	3 614.4	45.71	25.1	45.0
6	4 579.2	54.83	31.8	55.0
7	5 342.4	64.49	37.1	65.0
8	6 292.8	74.14	43.7	75.0
9	7 920.0	83.77	55.0	85.0
10	11 404.8	93.30	79.2	95.0

Example 4: Environmental testing of 10 electric rockets with associated power conditioning has resulted in the ordered time-to-failure data given in table C–5.

(1) What is the mean time between failures?
(2) Write the gamma failure and the reliability functions.
(3) What is the hazard rate at 5000 hr?
(4) What is the failure rate at 5000 hr during the next 1000-hour interval?

Solution 4: The essential steps for the graphical solution of this problem follow (ref. C–5):

(1) Obtain the median ranks for each ordered position; see table C–5.
(2) Plot on linear graph paper (10 × 10 to the inch) median rank against time to failure for the range around 80-percent median rank.
(3) Fit a straight line to the plotted points. For a median rank of 80 read the corresponding time to failure t_{80} in hours. Figure C–4 gives a t_{80} of 7200 hr.

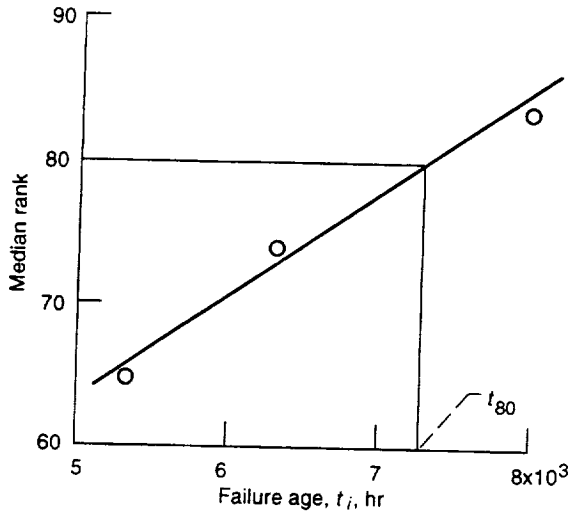


Figure C–4.—Electric rocket life.

(4) The time-to-failure data are scaled by using the equation

$$\hat{t}_i = \frac{50}{t_{80}} t_i$$

where

$\hat{t}_i$ i^{th} scaled time to failure
t_{80} rough estimate of 80-percent failure time
t_i i^{th} time to failure, hr

Table C–5 gives $\hat{t}_i$ for each ordered sample.

(5) Plot on linear graph paper (10 × 10 to the inch) median rank against scaled time to failure $\hat{t}_i$. Figure C–5 shows the plotted data points for this problem.

(6) These data points fit the gamma curve well with a β estimate of 2.0; hence, it appears as though a two-parameter gamma distribution is required with the location parameter γ equal to zero. The nonzero location parameter case is covered in the literature (ref. C–5).

(7) Overlay the linear axis (10 spaces to the inch) of a sheet of five-cycle semilog paper corresponding to a β of 2.0. Plot on this special graph paper the linear scale rank against time-to-failure data given in table C–5.

(8) Fit a straight line through the plotted points. Figure C–6 shows the plot for these data. Two additional straight lines are shown in this figure: line 1 was obtained by plotting two known points (0.5,1) and (20,8) (ref. C–5); line 2 has one point at (0.5,1) with a slope m. If line 1 were coincident with line 2, the β estimate would be sufficiently accurate.

(9) Because the two lines are not coincident, a closer approximation for β is obtained by taking a new midpoint coordinate estimate of 6.8 from figure C–6. Using existing charts gives β = 2.25, which satisfies the slope criteria (ref. C–5).

(10) For a shape parameter β of 2.25, a linear scale rank of 20 percent applies. Entering figure C–6 at this point on the ordinate gives a scale parameter α of 2400 hr.

With these graphical construction aids, the solution to the problem is readily achieved:

(1) The mean time between failures is given by

$$\bar{t} = \alpha\beta = 2.4 \times 10^3 \text{ hr} \times 2.25 = 5.4 \times 10^3 \text{ hr}$$

(2) The gamma failure and reliability functions are given by

$$p(t) = \frac{1}{\alpha^{\beta}\Gamma(\beta)}(t-\gamma)^{\beta-1} e^{-1(t-\gamma)/\alpha}$$

It has been shown that γ = 0; the other constants are calculated as follows:

$$\alpha^{\beta} = \left(2.4 \times 10^3\right)^{2.25}$$

Using logarithms, $\log \alpha^{\beta} = 2.25(\log 2.4 + \log 10^3)$; performing the indicated operations gives $\log \alpha^{\beta} = 7.61$; hence, $\alpha^{\beta} = 4.25 \times 10^7$.

The second required constant is $\Gamma(\beta) = \Gamma(2.25)$. Using the identity $\Gamma(x+1) = x!$, then $\Gamma(2.25) = \Gamma(1.25+1) = 1.25!$. Using Sterling's formula, $x! = x^x e^{-x}(2\pi x)^{1/2}$. Taking logarithms gives

$$\log x! = x \log x + (-x)\log_e + \left(\frac{1}{2}\right)[\log 2\pi + \log x]$$

$$= \left(x + \frac{1}{2}\right)\log x - 0.434x + 0.399$$

$$\log(1.25!) = 1.75 \log 1.25 - 0.434 \times 1.25 + 0.399 = 0.026$$

Substituting and forming the product gives $\alpha^{\beta}\Gamma(\beta) = (4.24 \times 10^7) \times 1.06 = 4.5 \times 10^7$. Using these constants and substituting values gives

$$p(t) = \frac{1}{4.5 \times 10^7} t^{1.25} e^{-t/2.4 \times 10^3}$$

and

$$R(t) = \frac{1}{4.5 \times 10^7} \int_t^{\infty} t^{1.25} e^{-t/2.4 \times 10^3} dt$$

(3) The hazard rate function at 5000 hr is given by

$$\lambda' = \frac{p(t_1)}{R(t_1)}$$

Here

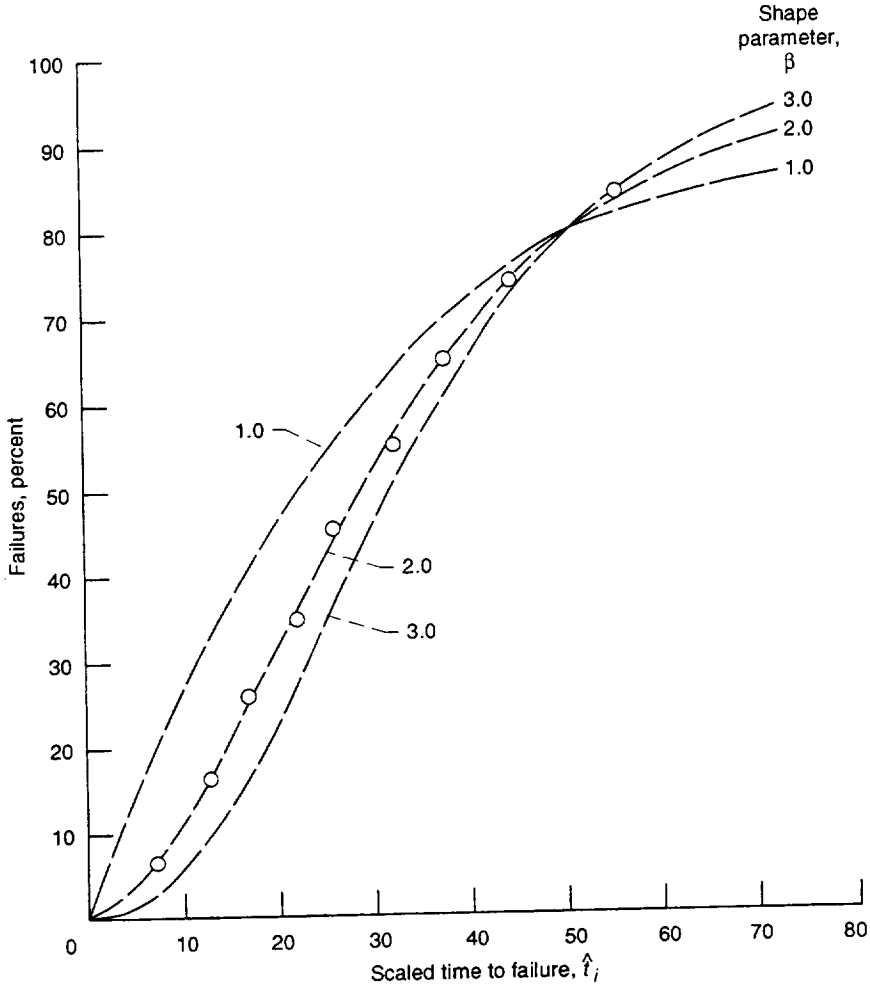


Figure C–5.—Electric rocket shape parameter curves.

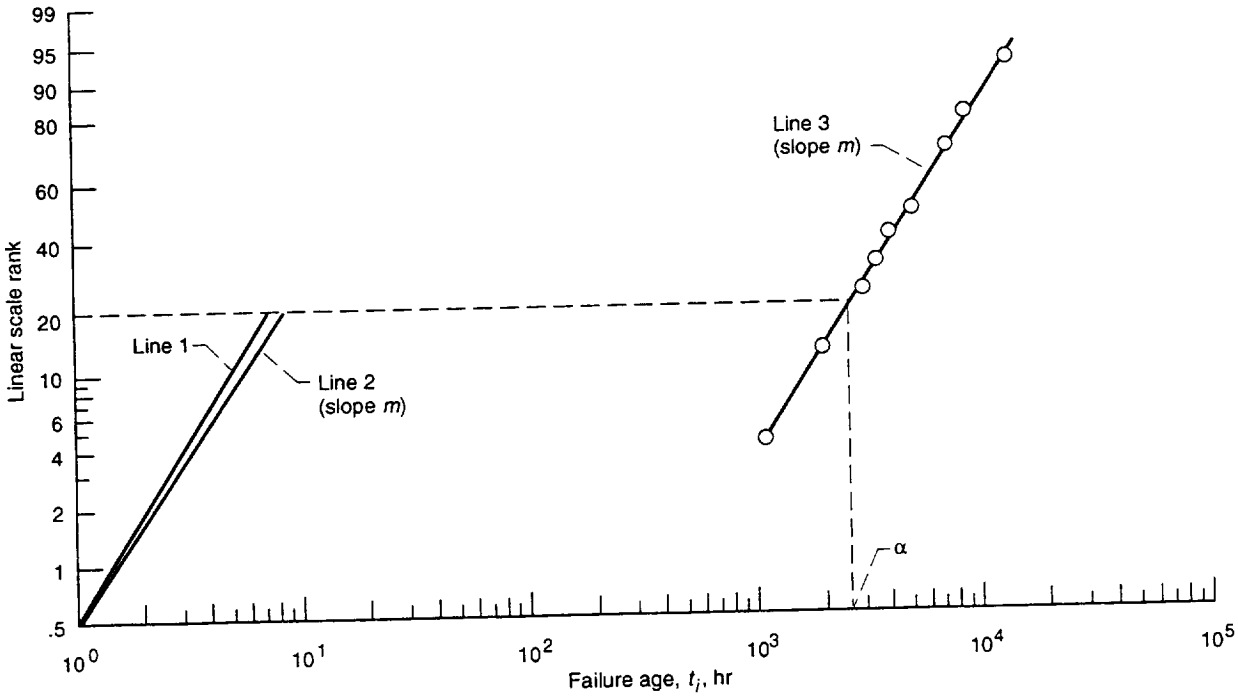


Figure C–6.—Electric rocket parameter diagram.

$$p(t_1) = \frac{1}{4.5\times10^7}\left(5\times10^3\right)^{1.25} e^{-5\times10^3/2.4\times10^3}$$

Performing the indicated operations gives

$$p(t_1) = \frac{\left(4.21\times10^4\right)\times\left(1.25\times10^{-1}\right)}{4.5\times10^7} = 1.17\times10^{-4}$$

We can obtain $R(t_1)$ either analytically by using this integral equation or graphically from figure C–6. Enter figure C–6 at a failure age of 5000 hr. Draw a vertical line to line 3. Project the intersection of $f(t)$ and 5000 hr over to the linear scale rank (0.605). Using a previous identity,

$$R(t_1) = 1 - 0.605 = 0.395$$

Substituting values gives

$$\lambda' = \frac{1.17\times10^{-4}}{3.95\times10^{-1}} = 2.71\times10^{-4} \text{ failure/hr}$$

(4) The failure rate function at 5000 hr during the next 1000-hr interval is given by

$$\lambda = \frac{1}{t_2 - t_1}\left[1 - \frac{R(t_2)}{R(t_1)}\right]$$

Following the procedure given previously and substituting values gives

$$R(t_2) = 1 - 0.710 = 0.290$$

and

$$\lambda = \frac{1}{10^3}\left(1 - \frac{0.290}{0.395}\right) = 2.65\times10^{-4} \text{ failure/hr}$$

As an illustration of mechanical parts, consider example 5:

Example 5: A cable used as guy supports for sail experiments in wind tunnel testing exhibited the time-to-failure performance data given in table C–6.

(1) Write the failure and reliability functions.
(2) What is the hazard rate at 5715 hr?
(3) What is the failure rate during the next 3000 hr?

Solution 5:

(1) The essential steps for solving this problem are

TABLE C–6.—TEST DATA FOR GUY SUPPORTS

Ordered sample number	Time to failure, t_f, hr	Median rank	5-Percent rank	95-Percent rank
1	1 100	6.7	0.5	25.9
2	1 890	16.2	3.7	39.4
3	2 920	25.9	8.7	50.7
4	4 100	35.5	15.0	60.7
5	5 715	45.2	22.2	69.7
6	8 720	54.8	30.3	77.8
7	12 000	64.5	39.3	85.0
8	17 500	74.1	49.3	91.3
9	23 900	83.3	60.6	96.3
10	46 020	93.3	74.1	99.5

(a) Obtain the median rank for each ordered position (see table C–6).

(b) Plot median rank against time to failure on log-normal probability graph paper (probability times two log cycles), as shown in figure C–7.

(c) If a straight line can be fitted to these plotted points, the time-to-failure function is log normal.

(d) The mean time between failures is calculated by $t' = \ln(\bar{t})$, where t = 6970 hr as shown in figure C–7 for a median rank of 50 percent; hence $\bar{t}' = 8.84$.

(e) The standard deviation is given by

$$\sigma_{t'} = \left[\frac{\ln t'_U - \ln t'_L}{3}\right]$$

where $t'_U = 49\,500$ hr and $t'_L = 1020$ hr as shown in figure C–7 for a median rank and a 1 – rank of 93.3 percent; hence, $\sigma_{t'} = (10.81 - 6.93)/3 = 1.28$.

With these constants the expressions for $p(t)$ and $R(t)$ are written as

$$p(t) = \frac{3.21\times10^{-1}}{t'} e^{-(t'-8.84)^2/3.28\times10}$$

and

$$R(t) = 3.21\times10^{-1}\int_{\ln(t)}^{\infty} e^{-(t'-8.84)^2/3.28\times10} dt$$

(2) The log-normal ordinate required for λ' can be calculated by using the standardized normal variable table as in example 2. The log-normal standardized variable is given by

$$Z_2 = \frac{t' - \bar{t}'}{\sigma_{t'}} = \frac{8.66 - 8.84}{1.28} = 0.143$$

From the normal-curve ordinate tables

$$Y_2' = 0.395$$

and

$$Y_2 = \frac{NY_2'}{\sigma_{t'}} = \frac{10 \times 0.395}{1.28} = 3.09 \text{ failures}$$

Substituting values gives

$$p(t')\frac{Y_2}{t} = \frac{3.09}{5.715 \times 10^3} = 5.40 \times 10^{-4} \text{ failure/hr}$$

The log-normal area from t' to infinity can be obtained directly from figure C–7 by using the 1 – rank scale. Enter the time-to-failure ordinate at 5715 hr; project over to the log-normal file function $f(t)$ and down to the 1 – rank abscissa value of 0.638. Therefore, the hazard rate λ' at 5715 hr is given by

$$\lambda' = \frac{5.40 \times 10^{-4}}{6.38 \times 10^{-1}} = 8.46 \times 10^{-4} \text{ failure/hr}$$

(3) The failure rate during the next 3000 hr is calculated by knowing that $R(t_1) = -0.638$ at a time to failure of 5715 hr and by obtaining $R(t_2) = 0.437$ from figure C–7 at 8715 hr. Therefore, the failure rate is given by

$$\lambda = \frac{1}{3 \times 10^3}\left(1 - \frac{0.437}{0.638}\right) = 1.05 \times 10^{-4} \text{ failure/hr}$$

Determination of confidence limits.—In the preceding sections, statistical estimates of various parameters have been made. Here we determine the methods for defining the confidence to be placed in some of these estimates. In example 1, tantalum capacitors with a one-parameter exponential distribution were studied. For an exponentially distributed population, additional estimates follow the chi-squared distribution. As an illustration of how to determine confidence limits for an exponentially distributed estimate, consider example 6.

Example 6: One hundred tantalum capacitors were tested for 15 000 hr, during which time 15 parts failed.

(1) What is the mean time between failures?

(2) What are the upper and lower confidence limits at 98-percent confidence level?

Solution 6:

(1) The mean time between failures is given by

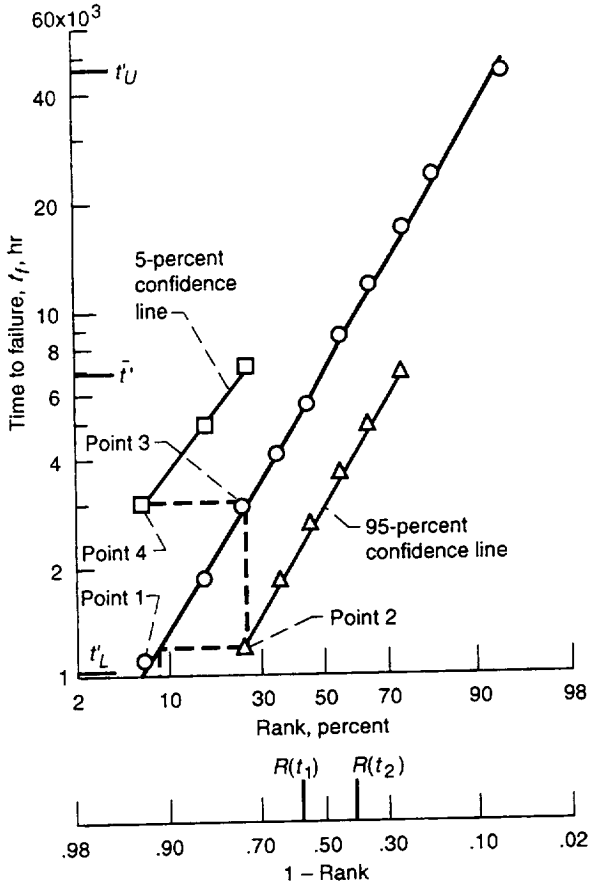


Figure C–7.—Guy support life.

$$\bar{t} = \frac{T}{r} = \frac{15\,000 \text{ hr}}{15 \text{ failures}} = 1000 \text{ hr/failure}$$

(2) The upper and lower confidence limits at some confidence level are given by

$$UCL = \left(\frac{2r}{\chi^2_{[1-(\alpha/2)];2r}}\right)\bar{t}$$

and

$$LCL = \left(\frac{2r}{\chi^2_{(\alpha/2);2r}}\right)\bar{t}$$

where

UCL upper confidence limit, hr
LCL lower confidence limit, hr

T total observed operating time, hr
χ^2 percentage points of chi-squared distribution
r number of failures
$1-\alpha/2$ probability that $\bar{t}$ will be the calculated $\alpha/2$ interval

For the 98-percent confidence level required by this problem,

$$\frac{\alpha}{2} = 0.01$$

$$1-\frac{\alpha}{2} = 0.99$$

and

$$2r = 30$$

Therefore, the chi-squared distribution values are given by (available from many existing tables)

$$\chi^2_{0.01;30} = 50.9$$

$$\chi^2_{0.99;30} = 14.9$$

Substituting values gives

$$UCL = \frac{30 \times 1000}{14.9} = 2013 \text{ hr}$$

and

$$LCL = \frac{30 \times 1000}{50.9} = 589 \text{ hr}$$

Thus, it is known with 98-percent confidence that the limits of the time $\bar{t}$ lie between approximately 590 and 2010 hr.

Determining the percentage values for the chi-squared distribution for values of r greater than 30 may also be useful. It has been shown that when $r \geq 30$,

$$\left(2\chi^2\right)^{1/2} = \left[2(2r)-1\right]^{1/2} \pm Z$$

where Z is the area under the normal curve at the specified confidence level. Example 7 illustrates how this equation is used for confidence interval calculations.

Example 7: The tantalum capacitors of example 6 have been operated for 5000 more hr; five additional units have failed. What are the confidence limits on $\bar{t}$ at the 98-percent confidence level for this additional testing?

Solution 7: For the areas under the normal curve from $-\infty$ to Z equal to 0.98 and 0.02, existing area tables give $Z = \pm 2.06$ and $r = 15 + 5 = 20$ total failures, with $2r = 40$.

Substituting values gives

$$\left(\theta_{\chi^2}\right)^{1/2} = (2 \times 40 - 1)^{1/2} \pm 2.06$$

$$\chi^2_{0.01;40} = 59.7, \qquad \chi^2_{0.99;40} = 23.4$$

Hence,

$$UCL = \frac{40 \times 10^3}{23.4} = 1709 \text{ hr}$$

$$LCL = \frac{40 \times 10^3}{59.7} = 670 \text{ hr}$$

Thus, it can be said with 98-percent confidence that $\bar{t}$ lies between approximately 670 and 1710 hr; as the test time increases, the estimated-parameter confidence interval decreases.

In example 2 gimbal actuators that exhibited normally distributed time-to-failure data were analyzed. For a normally distributed population, additional mean estimates will also be normal. As an illustration of how to determine confidence intervals for normal estimates, consider example 8.

Example 8: Twenty-five gimbal actuators have been tested. The mean time between failures has been calculated to be 75 000 hr with a standard deviation of 10 300 hr (see example 2). What are the upper and lower confidence limits at a 90-percent confidence level?

Solution 8: The upper and lower confidence limits are given by

$$UCL = \bar{t} + K_{\alpha/2}\frac{\sigma}{n^{1/2}}$$

$$LCL = \bar{t} - K_{\alpha/2}\frac{\sigma}{n^{1/2}}$$

where

$\bar{t}$ mean time between failures, hr
$K_{\alpha/2}$ standardized normal variable
σ unbiased standard deviation
n number of samples
$1-\alpha$ probability that t will be in calculated interval

For this problem

$$1-\alpha = 0.90$$

$$\alpha = 0.10$$

$$\frac{\alpha}{2} = 0.05$$

and from existing tables for the area under the normal curve, $K_{\alpha/2} = 1.64$. Substituting values gives

$$UCL = 75\,000 + \frac{1.64 \times 10\,300}{25^{1/2}} = 78\,400 \text{ hr}$$

and

$$LCL = 75\,000 - \frac{1.64 \times 10\,300}{25^{1/2}} = 71\,600 \text{ hr}$$

This means that 90 percent of the time the mean-time-between failures estimate $\bar{t}$ for 25 gimbal actuators, rather than the original 10, will be between 71 600 and 78 400 hr. Note that the sample size n has been increased to use this technique. This reflects the usual user pressure to learn as much as possible with the least amount of testing. Try to keep $n \geq 25$ in estimating normal parameters with this technique. If $n < 25$, use Student's t distribution (ref. C–6). To determine the effects on confidence intervals of reducing sample size, rework example 2 for the smaller sample size of 10, using Student's t distribution. The upper and lower confidence limits are given by

$$UCL = \bar{t} + t_{\alpha/2}\frac{s}{n^{1/2}}$$

and

$$LCL = \bar{t} - t_{\alpha/2}\frac{s}{n^{1/2}}$$

where

$t_{\alpha/2}$ Student's t variable
s standard deviation

For this problem, $r = n - 1 = 9$, $\alpha = 0.10$, and $t_{\alpha/2}$ from existing tables is 1.83. The standard deviation is given by

$$s = \left(\frac{57\,213 - 56\,250}{10}\right)^{1/2} = 9820$$

Substituting values gives

$$UCL = 75\,000 + \frac{1.83 \times 9820}{10^{1/2}} = 80\,700 \text{ hr}$$

and

$$LCL = 75\,000 - \frac{1.83 \times 9820}{10^{1/2}} = 69\,300 \text{ hr}$$

Comparing this time interval with that calculated for a sample size of 25 shows that the smaller sample gives a larger interval of uncertainty.

In example 3 stepping motors that exhibited Weibull distributed time-to-failure data were studied. As a graphical illustration of how to determine confidence intervals for a Weibull-distributed estimate, consider example 9.

Example 9: Another group of stepping motors has been step tested as previously explained in example 3. The Weibull plot of percent failures for a given failure age is the same as that given in figure C–2. During this testing, however, only eight failures have occurred. What is the 90-percent confidence band on the reliability estimate at 4000 cycles?

Solution 9: The data needed for graphical construction of the confidence lines on the Weibull plot are given in table C–3. The following steps are necessary to construct the confidence lines in figure C–2:

(1) Enter the percent failure axis with the first 5-percent rank value hitting $f(t)$; for failure 2 the 5-percent rank is 3.68.

(2) Draw a horizontal line that intersects $f(t)$ at point 1.

(3) Draw a vertical line to cross the corresponding median rank; for failure 2 the median rank is 16.23.

(4) Draw a horizontal line at the median rank, 16.23, for failure 2. The intersection point of the line for step (3) with this line is one point on the 95-percent confidence line.

(5) Repeat steps (1) to (4) until the desired cycle life is covered, 4000 cycles in this case.

(6) The 5-percent confidence line is obtained in a similar manner. Enter the percent failure axis with the 95-percent failure rank; 25.89 for failure 1.

(7) Draw a horizontal line that intersects $f(t)$ at point 3.

(8) Draw a vertical line to cross the corresponding median rank; 6.70 for failure 1.

(9) Draw a horizontal line at the median rank, 6.70, for failure 1. The intersection point of these two lines is one point on the 5-percent confidence line.

(10) Repeat steps (6) to (9) until the desired cycle life is covered.

A 90-percent confidence interval for $f(t)$ at 4000 cycles is, from figure C–2, 1.2 to 37.5 percent. Hence, a 90-percent confidence interval for $R(t)$ at 4000 cycles is 0.998 to 0.625.

In example 5, guy supports that exhibited log-normally-distributed time-to-failure data were analyzed. As a final graphical illustration of how to determine confidence intervals for a log-normally-distributed estimate, consider example 10.

Example 10: It has been shown that the guy supports of example 5 exhibited a reliability of 0.638 at a time to failure of 5715 hr. Consider now the procedure for determining the confidence band on this log-normal estimate. The data needed for the graphical construction of the 90-percent confidence lines on the log-normal graph of figure C–7 are also given in table C–6.

Solution 10: The steps necessary to graphically construct the confidence lines in figure C–7 are as follows:

(1) Enter the rank axis with the first 5-percent rank value hitting $f(t)$, the log-normal life function shown in figure C–7; for ordered sample 3, the 5-percent rank is 8.7.

(2) Draw a vertical line to intersect $f(t)$ at point 1 as shown in figure C–7.

(3) Draw a horizontal line to cross the corresponding median rank; for ordered sample 3, the median rank is 25.9.

(4) The intersection point (point 2 in fig. C–7) of step (3) and the median-rank line is one point on the 95-percent confidence line.

(5) Repeat steps (1) to (4) until the desired time to failure is covered; 5715 hr in this case.

(6) The 5-percent confidence line is obtained in a similar manner. Enter the rank axis with the 95-percent-failure rank, 25.9, for ordered sample 1.

(7) Draw a vertical line intersecting $f(t)$ at point 3.

(8) Draw a horizontal line to cross the corresponding median rank; for ordered sample 1, the median rank is 6.7.

(9) The intersection point (point 4 in fig. C–7) of these two lines is one point on the 5-percent confidence line.

(10) Repeat steps (6) to (9) until the desired time to failure is covered.

At 5715 hr the 90-percent confidence interval for $f(t)$ is, from figure C–7, 19.7 to 69.4 percent. Hence, a 90-percent confidence interval for $R(t)$ at 5715 hr is 0.803 to 0.306. Incidentally, this graphical procedure for finding confidence intervals is completely general and can be used on other types of life test diagrams.

Estimation using the Poisson and binomial events.—The Poisson and binomial distributions are discrete functions of the number of failures N_f that occur rather than of the time t.

The Poisson distribution (fig. C–1) is a discrete function of the number of failures. When this distribution applies, it is of interest to determine the probabilities associated with a specified number of failures in the time continuum. As an illustration of a complex electrical component that follows the Poisson distribution, consider example 11.

Example 11: Ten space-power speed controllers were tested during the rotating solar dynamic development program. The time-to-failure test data are given in table C–7.

(1) Write the Poisson failure density and reliability functions.

(2) What is the probability of five failures in 10 000 hr?

TABLE C–7.—POISSON DATA FOR SPEED CONTROLLER

Ordered sample number	Time to failure, t_f, hr
1	3 520.0
2	4 671.2
3	6 729.3
4	7 010.0
5	8 510.2
6	9 250.1
7	10 910.0
8	11 220.5
9	11 815.6
10	12 226.4
Total	85 866.3

(3) What is the probability that 6, 7, 8, 9, or 10 failures will occur? What is the reliability after the fifth failure?

Solution 11:

(1) Reducing the data given in table C–7 gives the mean time between failures as

$$\bar{t} = \frac{\sum_{i=1}^{10} t_i}{N_f} = \frac{8.59 \times 10^4}{10} = 8.59 \times 10^3 \text{ hr/failure}$$

Hence, the Poisson failure density function is given by

$$p(N_f) = \frac{\left(\frac{t}{8.59 \times 10^3}\right)^{N_f}}{N_f!} e^{-t/8.59 \times 10^3}$$

The reliability function is given by

$$R(N_f) = \sum_{j=1}^{10} \frac{\left(\frac{t}{8.59 \times 10^3}\right)^{j}}{j!} e^{-t/8.59 \times 10^3}$$

(2) To calculate the probability of five failures in 10 000 hr, use the ratio

$$\frac{t}{\bar{t}} = \frac{1.0 \times 10^4}{8.59 \times 10^3} = 1.16$$

The probability of five failures in 10 000 hr is given by

$$p(5) = \frac{(1.16)^5 e^{-1.16}}{5!} = \frac{2.09 \times 0.314}{1.2 \times 10^2} = 5.47 \times 10^{-3}$$

One easy method of calculating the term $(1.16)^5$ is

$$\log(1.16)^5 = 5 \log 1.16 = 5(0.148) = 0.740$$

$$(1.16)^5 = 2.09$$

(3) The reliability from the 5th to the 10th failure is the sum of the remaining terms in the Poisson expansion. The Poisson expansion in sum form is given by

$$R(N_f) = \sum_{j=6}^{10} \frac{0.314(1.16)^j}{j!}$$

Calculating each term and summing gives

$$R(6) = 0.0013$$

The binomial distribution is given in figure C–1 as distribution 7. Considerable work has been done to develop the techniques suitable for using this powerful tool (refs. C–1 and C–3). As an illustration consider a pyrotechnic part described in example 12.

Example 12: A suspicious lot of explosive bolts is estimated to be 15 percent defective due to improper loading density as observed by neutron radiography.

(1) Calculate the probability of one defective unit appearing in a flight quantity of four.

(2) Plot the resulting histogram.

(3) What is the reliability after the first defect?

Not many failure density data are available, but past experience with pyrotechnic devices has shown that the binomial distribution applies. From the given data, the per-unit number of effectives q is 0.85, the per-unit number of defectives p is 0.15, the sample size n is 4, and the possible number of failures N_f is 0, 1, 2, 3, or 4. The frequency functions corresponding to these constants are given by

$$p(N_f) = \frac{4!}{(4 - N_f)! N_f!} p^{N_f} q^{4 - N_f}$$

and

TABLE C–8.—BINOMIAL EXPANSION COEFFICIENTS

Sample size, n	Possible number of failures	Binomial expansion coefficients
1	2	1
2	3	1 2 1
3	4	1 3 3 1
4	5	1 4 6 4 1

$$R(N_f) = \sum_{j=N_f}^{4} \frac{4!}{(4-j)! j!} p^j q^{n-j}$$

One simple method for obtaining the binomial expansion coefficients is to make use of Pascal's triangle. Pascal found that there was symmetry to the coefficient development and explained it as shown in table C–8. Pascal's triangle (dashed lines) is shown in the last column. The lower number in the dashed triangle is obtained by adding the two upper numbers (i.e., 3 + 3 = 6).

Using these constants and expanding gives $p(N_f)$ as

$$p(N_f) = q^4 + 4q^3 p + 6q^2 p^2 + 4qp^3 + p^4$$

The probability of one defective unit appearing in a flight quantity of four is given by the second term in the expansion; hence,

$$4q^3 p = 4(0.85)^3 (0.15) = 0.37$$

The resulting histogram for this distribution is shown in figure C–8. The probability that 2, 3, or 4 defects will occur, as the reliability after the first defect, is the sum of the remaining terms in the binomial expansion. This probability can be calculated by using the equation for $R(N_f)$. However, it is simpler to use the histogram graph and sum the probabilities over N_f from 2 to 4; hence,

$$R(2) = 0.096 + 0.011 + 0.0011 = 0.108$$

These explosive bolts in their present form are not suitable for use on any spacecraft because the probability of zero defects is only 0.522, much below the usually desired 0.999 for pyrotechnic spacecraft devices.

Determination of confidence limits.—When an estimate is made from discrete distributions, it is expected that additional estimates of the same parameter will be close to the original estimate. It is desirable to be able to determine upper and lower confidence limits at some stated confidence level for discrete

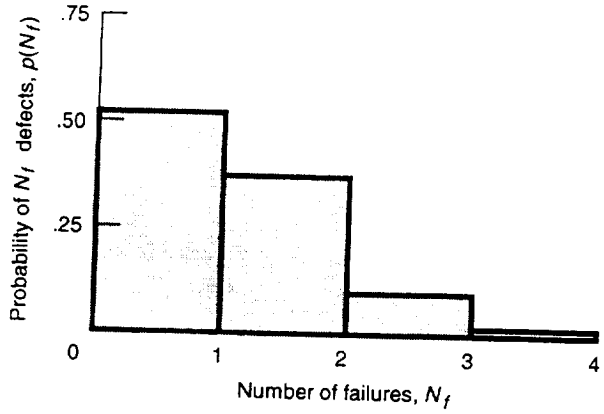


Figure C–8.—Explosive bolts histogram

distribution estimates just as is done for continuous functions of time. The analytical procedure for determining these intervals is simplified by using specially prepared tables and graphs. Useful tables for the binomial distribution are given in the literature (ref. C–3).

As an example of how confidence intervals can be obtained for Poisson estimates, consider problem 13.

Problem 13: The Poisson estimate of reliability from the 5th to the 10th failure for speed controllers was found to be 0.0013 in a previous problem. What are the upper and lower confidence limits on this estimate at a 95-percent confidence level?

The variation in $\bar{t}$ can be found by using figure C–9. Enter figure C–9 on the 5-percent α line at the left-hand end of the 5 interval. Here, $T/\bar{t}_1 = 10.5$; then $\bar{t}_1 = 10\,\bar{t}\,(T/\bar{t}_1) = 8.57\times10^4/10.5 = 8160$ hr. Using the left-hand end of the 4 interval gives $T/\bar{t}_2 = 9.25$; then $\bar{t}_2 = 8.57\times10^4/9.25 = 9530$ hr. One simple method for finding $f(5)$ is to use figure C–10 (ref. C–5). The $t/\bar{t}$ ratios of interest are 1.22, 1.16, and 1.05, respectively. For these ratios with $N_f = 5$, the values of $f(5)$ from figure C–10 are 0.997, 0.9987, and 0.99992, respectively. Because the sum of the last five terms is desired, $R(5)$ is 0.003, 0.0013, and 0.0008, respectively. This means that the probability of the 5th to the 10th failure of a speed control occurring is in the interval 0.0008 to 0.003 at a confidence level of 95 percent.

As an illustration of how confidence intervals can be obtained for a binomial distribution, consider example 14.

Example 14: The probability of one defective unit appearing in a flight quantity of four explosive bolts has been calculated to be 0.37. What are the upper and lower confidence limits on this estimate at a 90-percent confidence level?

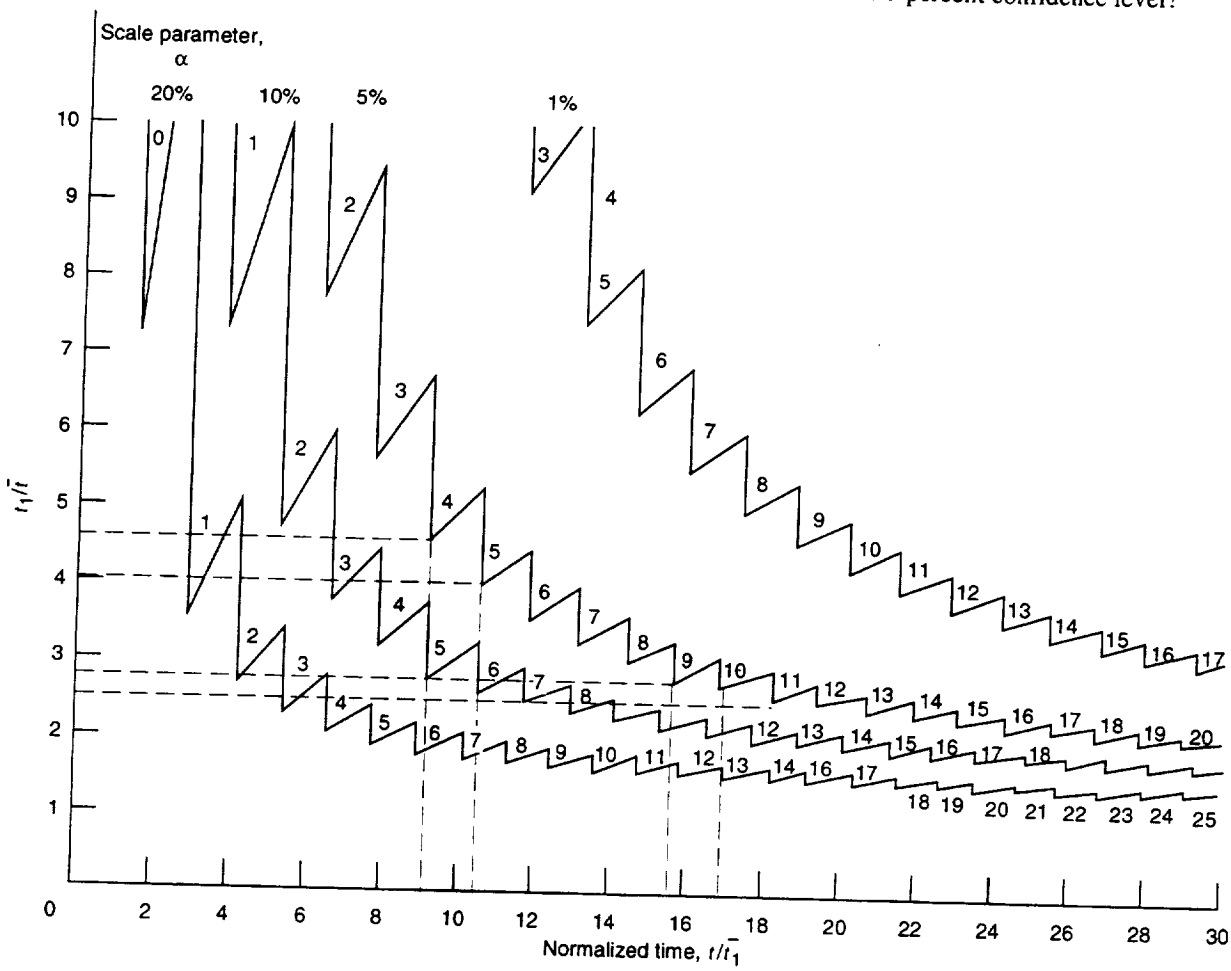


Figure C–9.—Poisson MTBF fixed test time.

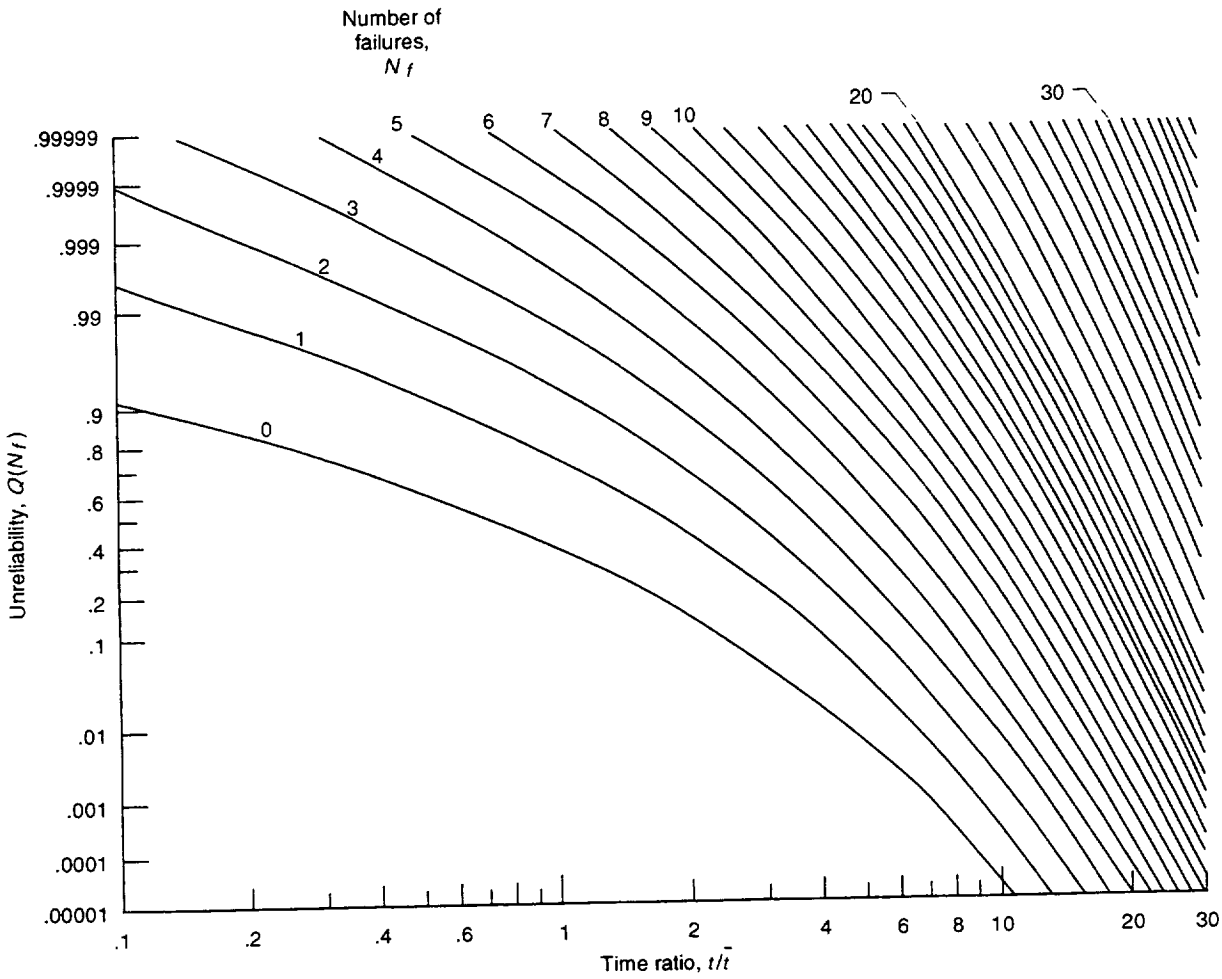


Figure C–10.—Poisson unreliability sum.

If the sample size is n, the number of defectives is r, and the confidence level is γ, this example has the following constraints: $n = 4$, $r = 1$, and $\gamma = 90$ percent. Using these constraints, the upper U and lower L confidence limits can be obtained directly from existing tables as $UCL = 0.680$ and $LCL = 0.026$. This means that with a 90-percent confidence level, the probability of one defective bolt appearing in a flight quantity of four is in the interval from 0.026 to 0.680.

Sampling

Purpose of sampling.—Sampling is a statistical method used when it is not practical to study the whole population. There are usually five reasons why sampling is necessary:

(1) Economy—It usually costs less money to study a sample of an item than to study the whole population.

(2) Timeliness—A sample can be studied in less time than the whole population can be studied, giving prompt results.

(3) Destructive nature of a test—Some tests require that the end item be consumed to demonstrate performance, leaving nothing to use afterwards.

(4) Accuracy—A sample survey conducted by well-trained researchers usually will result in accurate and valid decisions.

(5) Infinite population—In many analytical studies, an infinite population is available. If any information is to be used for decision making, it must be based on a sample.

Choosing a sample.—Good judgment must be used in choosing a sample. Subjective methods of choosing samples frequently result in bias. Bias is an expression, either conscious or subconscious, of the selector's preferences. Bias can be held to a minimum by using a nonsubjective method developed just for this purpose. Several nonsubjective sampling procedures are described:

6433	2582	0820	1460	6606	7143	9158	5114	9491	8063
3465	7348	5774	3821	6216	2148	1221	5895	7942	9971
9601	9189	0141	1377	3467	7971	0811	8309	0504	4606
2364	3260	1430	9505	3146	4815	9732	3447	7705	4532
7304	9292	4580	8160	7144	8073	8476	1896	6661	1285
3764	5460	6385	9045	7170	5831	4668	9386	3979	1116
0251	3139	4201	0578	2172	6876	4347	4288	1514	9985
2031	0919	7613	1535	1610	7491	3255	4014	3614	5599
6398	1374	1904	7490	3941	0284	5817	1630	4629	6773
0911	3930	0324	8151	3365	6685	0566	5047	8471	6166
5052	5023	3045	3433	6365	7310	5073	5416	2332	0922
9225	3984	4659	4642	7260	1383	7625	7512	8547	7343
3100	7916	9757	8869	5307	2691	0786	2701	0102	5745
4598	0065	4257	6557	4638	8418	7398	9790	5074	8018
5956	7285	0480	1411	7766	3377	5023	0227	8047	1887
9360	1041	2094	4212	2623	2384	6422	5374	0651	8673
8796	9974	1913	8309	4943	9423	9143	4683	4436	8413
7071	8254	6825	3020	9000	4673	6129	0176	3670	4836
7336	4451	5863	6559	5344	0714	1856	0451	7855	5998
1660	0222	2005	0215	2370	2687	3039	7953	1960	6579
7506	1020	8718	9665	1892	8245	7249	6023	4602	4227
5000	8237	6203	6829	5325	5784	8720	5053	6347	1112
4255	6894	8093	9191	5011	0452	6199	0009	8086	5170
5764	9837	6780	7490	5412	4869	6950	4183	8671	4008
3609	1368	9129	7113	3099	1887	0544	6415	9148	4381
7218	5939	4932	5465	6648	6365	4179	9266	9803	5572
6854	5911	1495	4940	4630	4514	0942	7218	7382	2145
4403	4263	4755	5451	8251	2652	6207	4841	3528	7665
2978	4381	2205	9638	6946	7126	9039	9194	6676	4396
1072	2292	4428	4934	8183	7385	3236	7748	4488	1351
6488	6568	9530	8316	7709	9022	8041	5564	6667	5329
9263	7756	6300	6793	7769	3099	3606	2468	2574	5230
0357	3493	0385	4451	4313	3024	8243	4920	3523	9644
5372	9351	8393	6023	2811	1744	2306	7083	4330	7278
6570	2866	7565	7871	9490	9050	4454	3475	8319	2972
8596	8251	0336	8119	1966	9115	4202	7785	5269	5941
4177	0092	4207	7386	9891	1149	3429	7062	4622	8415
8438	4892	2089	5509	2054	9024	1213	5791	2543	7863
5820	6287	7484	0339	8585	0968	3675	2440	4000	5148
7721	3804	9520	6184	9152	1853	8640	3601	5606	7218

Figure C–11.—Random digits table.

(1) Random sampling—Each item in the population has an equal and independent chance of being selected as a sample. A random-digits table (fig. C–11) has been developed to facilitate drawing random samples and has been constructed to make the 10 digits from 0 to 9 equally likely to appear at any location in the table. Adjacent columns of numbers can be combined to get various-sized random numbers.

(2) Stratified sampling—Similar items in a population are grouped or stratified, and a random sample is selected from each group.

(3) Cluster sampling—Items in a population are partitioned into clusters, and a random sample is selected from each cluster.

(4) Double sampling—A random sample is selected; then, depending on what is learned, some action is taken or a second sample is drawn. After the second random sample is drawn, action is taken on the basis of data obtained from the combination of both samples.

(5) Sequential sampling—Random samples are selected and studied one at a time. A decision on whether to take action or to continue sampling is made after each observation on the basis of all data available at that selection.

As an illustration of when to use various sampling methods consider example 15.

Example 15: Describe how a sample should be selected for three cases:

(1) Invoices numbered from 6721 to 8966 consecutively. In this case, a random sampling procedure could be used based on the four-digit table given in figure C–11. Using the given

invoice numbers, start at the top of the left column and proceed down each column selecting random digits until the desired sample size is obtained. Disregard numbers outside the range of interest.

(2) Printed circuit assemblies to compare the effectiveness of different soldering methods. If boards are all the same type, a cluster sampling procedure could be used here. Group the boards by soldering methods; select x joints from each cluster to compare the effectiveness of different soldering methods.

(3) Residual gases in a vacuum vessel to determine the partial pressure of gases at various tank locations. A stratified sampling procedure could be used in this case. Stratify the tank near existing feedthroughs into x sections; an appropriate mass run could be taken from each section at various ionizer distances from the tank walls. Analysis would tell how the partial pressures varied with ionizer depth at the feedthrough locations.

Sample size.—A completely general equation for determining sample size n is given by

$$Q(t) = 1 - R(t_t) = \frac{N_f}{n}$$

where

N_f desired number of time-to-failure points
n sample size
t_t test truncation time

This equation can be used with any of the reliability functions given in figure C–1.

As an illustration of how these equations can be applied to electrical parts, consider example 16, which is derived from example 1.

Example 16: Tantalum capacitors with a failure rate of 1×10^{-3} failure/hr are to be tested to failure. In a 1000-hr test, what sample size should be used to get 25 time-to-failure data points?

Solution 16: The truncated exponential reliability function is given by

$$R(t_t) = e^{-t_t/1000} = 0.37$$

Solving the general sample size equation for n and substituting values gives

$$n = \frac{N_f}{1 - R(t_t)} = \frac{25}{0.63} = 39.6$$

Rounding off to the nearest whole unit gives $n = 40$ pieces. This means that 40 capacitors tested for 1000 hr should give 24 time-to-failure data points.

Accelerated Life Testing

Life testing to define the time duration during which a device performs satisfactorily is an important measurement in reliability testing because it is a measure of the reliability of a device. The life that a device will exhibit is very much dependent on the stresses it is subjected to. The same devices in field application are frequently subjected to different stresses at varying times. It should be recognized then that life testing involves the following environmental factors:

(1) The use stresses may influence the life of the device and failure rate functions.

(2) The field stresses could be multidimensional.

(3) An interdependence among the stress effects exists in the multidimensional stress space.

(4) Life performance may vary because most devices operate over a range in a multidimensional stress space.

Testing objects to failure under multidimensional stress conditions is usually not practical. Even if it were, if the system were properly designed, the waiting time to failure would be quite long and therefore unrealistic. It has been shown that time-to-failure data are important to reliability testing, and now they appear difficult to obtain. These are some of the reasons why many are turning to accelerated life testing, such as compressed-time testing, advanced-stress testing, or optimum life estimates:

(1) Compressed-time testing—If a device is expected to operate once in a given time period on a repeated cycle, life testing of this device may be accelerated by reducing the operating time cycle. The multidimensional stress condition need not be changed. The stresses are being applied at a faster rate to accelerate device deterioration. Care should be taken not to accelerate the repetition rate beyond conditions that allow the device to operate in accordance with specifications. Such acceleration would move the device into a multidimensional stress region that does not exist in field conditions and would yield biased information. As an illustration of compressed time testing, consider example 17.

Example 17: The stepping motor in example 3 was being pulsed for life testing. How could this life test be accelerated?

The power supply providing the stepping pulses may have been stepping at the rate of one pulse per 10 sec, resulting in a test time of 107 sec. These motors had a frequency response allowing 10 pulses per sec. Increasing the pulse stepping rate up to the frequency response limit yields comparable time-to-failure data in 105 sec, a savings in time of 2 orders of magnitude.

(2) Advanced-stress testing—If a device is expected to operate in a defined multidimensional stress region, life testing of this device may be accelerated by changing the multidimensional stress boundary. Usually the changes will be toward increased stresses because this tends to reduce time to failure.

The two reasons why advanced stress testing is used are to save time and to see how a device performs under these stress conditions. Care should be exercised in changing stress boundaries to be sure that unrealistic conditions leading to wrong conclusions are not imposed on the device. A thorough study of the failure mechanisms should be made to ensure that proposed changes will not introduce new mechanisms that are not normally encountered. If an item has a certain failure density distribution in the rated multidimensional stress region, changing the stress boundaries should not change the failure density distribution. Some guidelines for planning advanced-stress tests are

(a) Define the multidimensional stress region for an item; nominal values should be centrally located.

(b) Study the failure mechanisms applicable to this item.

(c) On the basis of guidelines (a) and (b), decide which stresses can be advanced without changing the failure mechanisms.

(d) Specify multiple stress tests to establish trends; one point should be on the outer surface of the multidimensional region.

(e) Be sure that the specimen size at each stress level is adequate to identify the failure density function and that it has not changed from level to level.

(f) Pay attention to the types of failures that occur at various stress levels to be sure that new failure mechanisms are not being introduced.

(g) Decide whether new techniques being developed for advanced-stress testing apply to this item. Several popular techniques are described:

(i) Sensitivity testing—Test an item at the boundary stress for a given time. If failure occurs, reduce stress by a fixed amount and retest for the same time. If no failure occurs, increase stress by a fixed amount and retest for the same time. Repeat this process until 25 failures occur. This technique is used to define endurance limits for items.

(ii) Least-of-N testing—Cluster items in groups and subject each cluster to a specified stress for a given time. Stop at the first failure at each stress level. Examine failed items to ensure conformance to expected failure mechanisms.

(iii) Progressive-stress testing—Test an item by starting at the central region in stress space and linearly accelerating stress with time until failure occurs. Observe both the failure stress level and the rate of increasing stress. Vary the rate of increasing stress and observe its effect on the failure stress magnitude. Examine failed items to ensure conformance to expected failure mechanics.

As an illustration of advanced-stress testing, consider example 18.

Example 18: A power-conditioning supply was being life tested at nominal conditions with an associated electric rocket. The nominal electrical, thermal, vibration, shock, and vacuum stresses resulted in fairly long waiting periods to failure. Changing the multidimensional stress conditions by a factor of 1.25 to 2, which is usually done during development testing, tended to identify design deficiencies with shorter waiting periods without affecting the failure mechanism.

(3) Optimum life estimate—One remaining calculation for nonreplacement failure or time-truncated life test is the optimum estimate of mean time between failures $\bar{t}$. It has been shown (ref. C–1) that $\bar{t}$ given by the time sum divided by the number of failures should be modified by a censorship factor and a truncation time factor. The censorship factor K is caused by wearout failures, operator error, manufacturing errors, and so forth. The correction equation for $\bar{t}$ is given by (ref. C–1)

$$\bar{t} = \frac{\sum_{i=1}^{N_f} t_i + (n - N_f)t_t}{N_f - K}$$

where

N_f number of failures

K censorship factor

As an illustration consider example 19.

Example 19: The tantalum capacitor tested in example 1 could have been stopped when 10 capacitors (580 part-hours) out of 100 had failed at a testing time of 100 hr. What is an optimistic value for $\bar{t}$?

Solution 19: Inspection of the 10 failed capacitors showed that two units failed because of manufacturing errors. Therefore, $N_f = 10$, $K = 2$, $n = 100$ capacitors, $t_r = 100$ hr, and the sum of $t_i = 580$ hr. Substituting these values into the $\bar{t}$ correction equation gives

$$\bar{t} = \frac{580 + (100 - 10)100}{10 - 2} = 1197 \text{ hr}$$

This is an optimistic estimate for the mean time between failures, but it certainly is fair and reasonable to make these types of corrections.

Accept/Reject Decisions With Sequential Testing

A critical milestone occurs in product manufacturing at delivery time. An ethical producer is concerned about shipping a product lot that does not meet specifications. The consumer is concerned about spending money to purchase a product that does not meet specifications. A test method that permits each to have an opportunity to obtain data for decisionmaking is required.

Sequential testing constraints.—If α is the producer's risk and β is the consumer's risk, two delivery time constants valid for small risks have been defined and are given as

$$A = \frac{1-\beta}{\alpha}$$

$$B = \frac{\beta}{1-\alpha}$$

Let P_1 be the probability that N_f failures will occur in time t for a specified minimum acceptable $\bar{t}_1$, and let P_0 be the probability that N_f failures will occur in time t for an arbitrarily chosen upper value $\bar{t}_0$. Test rules using these four constants have been defined for each condition (refs. C–1 and C–5):

(1) Accept if $P_1 / P_0 \le B$.
(2) Reject if $P_1 / P_0 \ge A$.
(3) Continue testing if $B < P_1/P_0 < A$.

Exponential parameter decisionmaking.—As an illustration of how these testing constraints can be implemented for the exponential distribution, consider example 20.

Example 20: A purchased quantity of 100 000 tantalum capacitors has been received. Negotiations prior to placement of the order had established that $\alpha = \beta = 0.1$, $\bar{t}_1 = 1000$ hr, and $\bar{t}_0 = 2000$ hr and that the sequential reliability test should be truncated in 48 hr.

(1) Calculate A and B.
(2) Write the expressions for P_0 and P_1.
(3) How many units should be placed on test?
(4) Plot a sequential reliability control graph to facilitate decisionmaking at each failure time.

Solution 20:

(1) The delivery time constants are obtained by substituting values into the defining equations.

$$A = \frac{1-0.1}{0.1} = 9$$

$$B = \frac{0.1}{1-0.1} = 0.111$$

(2) Using binomial distribution from figure C–1 and substituting values gives $P_0(N_f)$ and $P_1(N_f)$ as

$$P_0(N_f) = \left(\frac{t}{2000}\right)^{N_f} \frac{e^{-t/2000}}{N_f!}$$

$$P_1(N_f) = \left(\frac{t}{2000}\right)^{N_f} \frac{e^{-t/1000}}{N_f!}$$

(3) Delivery constant B defines the acceptance criteria for P_1/P_0. Using this constraint and substituting for P_1 and P_0 gives

$$B = \frac{P_1(N_f)}{P_0(N_f)} = 2^{N_f} e^{-t/2000}$$

The minimum testing time without failure $t(0)_{min}$ is given by

$$0.111 = (2)^0 e^{-t(0)_{min}/200}$$

Solving for $t(0)_{min}$ gives

$$t(0)_{min} = 2.20 \times 2000 = 4400 \text{ unit-hr}$$

The minimum number of capacitors to be life tested for 48 hr is given by

$$n_{min} \frac{4400 \text{ unit-hr}}{48 \text{ hr}} = 91.7$$

To ensure good results, choose a sample size n that is more than twice n_{min}; for this problem, use $n = 200$ units. The required minimum testing time for 200 units is given by

$$t(0)_{min} = \frac{4400 \text{ unit-hr}}{200 \text{ units}} = 22.0 \text{ hr}$$

The test can be stopped and an accept/reject decision made at t_t, where t_t is given by

$$t_t = 48 \text{ hr} \times 20 \text{ units} = 9.6 \times 10^3 \text{ unit-hr}$$

(4) The tantalum capacitor reliability chart is constructed by using five points in the (N_f, t) plane; three of these points have already been calculated and are given by

$$t(0)_{min} = 4400, N_f = 0$$

$$t_t = 9.6 \times 10^3, N_f = 0$$

$$t = 0, N_f = 0$$

The remaining two points are calculated by using the test inequality given by

$$B < p(N_f) < A$$

In general terms the ratio $p(N_f)$ is given by

$$p(N_f) = \left(\frac{\bar{t}_0}{\bar{t}_1}\right)^{N_f} e^{-(1/\bar{t}_1 - 1/\bar{t}_0)t}$$

Taking natural logarithms of the inequality and substituting gives

$$\ln B < N_f \ln\left(\frac{\bar{t}_0}{\bar{t}_1}\right) - \left(\frac{1}{\bar{t}_1} - \frac{1}{\bar{t}_0}\right)t < \ln A$$

Adding $(1/\bar{t}_1 - 1/\bar{t}_0)t$ to each term gives

$$\ln B + \left(\frac{1}{\bar{t}_1} - \frac{1}{\bar{t}_0}\right)t < N_f \ln\left(\frac{\bar{t}_0}{\bar{t}_1}\right) < \ln A + \left(\frac{1}{\bar{t}_1} - \frac{1}{\bar{t}_0}\right)t$$

Dividing all terms by $\ln(\bar{t}_0/\bar{t}_1)$ gives

$$\frac{\ln B}{\ln\left(\frac{\bar{t}_0}{\bar{t}_1}\right)} + \left[\frac{\frac{1}{\bar{t}_1} - \frac{1}{\bar{t}_0}}{\ln\left(\frac{\bar{t}_0}{\bar{t}_1}\right)}\right]t < N_f < \frac{\ln A}{\ln\left(\frac{\bar{t}_0}{\bar{t}_1}\right)} + \left[\frac{\frac{1}{\bar{t}_1} - \frac{1}{\bar{t}_0}}{\ln\left(\frac{\bar{t}_0}{\bar{t}_1}\right)}\right]t$$

The inequality is now in the form given by

$$a + bt < N_f < c + bt$$

The constants a and c for this problem for zero failures are given by

$$a = \frac{\ln B}{\ln\left(\frac{\bar{t}_0}{\bar{t}_1}\right)} = \frac{-2.2}{0.69} = -3.18, \quad N_f = 0$$

$$c = \frac{\ln A}{\ln\left(\frac{\bar{t}_0}{\bar{t}_1}\right)} = \frac{2.2}{0.69} = 3.18, \quad N_f = 0$$

Because these boundary constraints are straight lines in the form

$$N_f = bt + (a \text{ or } c)$$

the slope b is given by

$$b = \frac{\left(\frac{1}{\bar{t}_1} - \frac{1}{\bar{t}_0}\right)}{\ln\left(\frac{\bar{t}_0}{\bar{t}_1}\right)} = \frac{5 \times 10^{-4}}{0.69} = 7.22 \times 10^{-4}$$

Figure C–12 shows the resulting tantalum capacitor reliability chart. The tantalum capacitor acceptance reliability test results in an "accept," "continue to test," or "reject" decision depending on the failure performance of the capacitors as a function of operating time in unit-hours as zoned in figure C–12.

Binomial parameters decisionmaking.—For the binomial frequency function, the procedure to set up a sequential reliability test is similar to the Poisson methodology. Because the unreliability, or number of defectives, is given by $1 - R$ for an effectiveness of R, then $P_1(N_f)$ is given in binomial form by

$$P_1(N_f) = (1 - R_1)^{N_f} (R_1)^{n - N_f}$$

where

n	$N_s + N_f$
N_s	number of successful trials
N_f	number of failed trials
R_0, R_1	chosen reliability values at some time t, $R_0 > R_1$

The ratio $P_1(N_f)/P_0(N_f)$ is given by

$$\frac{P_1(N_f)}{P_0(N_f)} = \frac{(1 - R_1)^{N_f} (R_1)^{n - N_f}}{(1 - R_0)^{N_f} (R_0)^{n - N_f}}$$

Following the steps given in example 20, give four of the points in the (N_f, t) plane:

$$N(0)_{\min} = \frac{\ln B}{\ln\left(\frac{R_1}{R_0}\right)}, \qquad N_f = 0$$

The test can be stopped and an accept/reject decision made at the number of test truncation trials N_r; N_r is given by

$$N_r = t_t N_c, \qquad N_f = 0$$

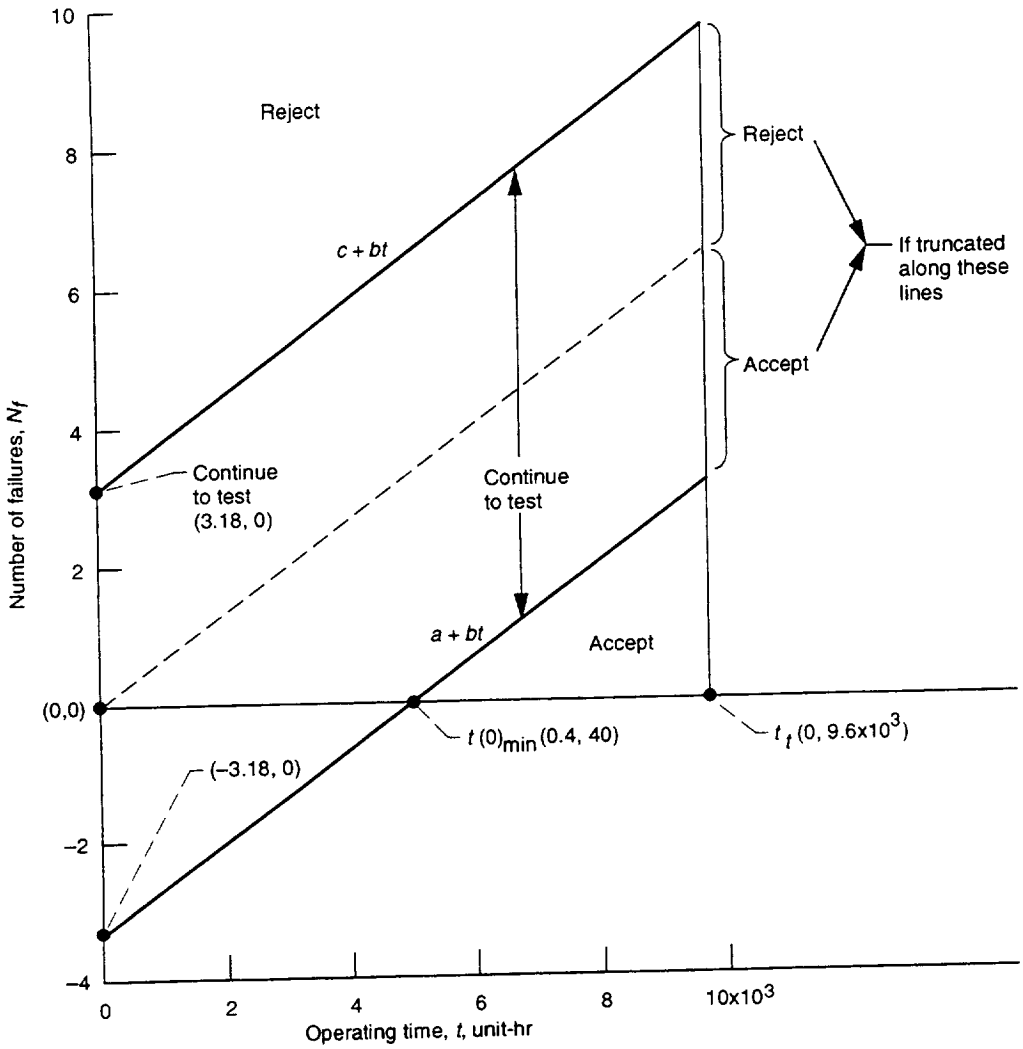


Figure C–12.—Tantalum capacitor reliability chart.

where N_c is the number of units chosen for testing:

$$n = 0, \qquad N_f = 0$$

$$a = \frac{\ln B}{\ln \dfrac{R_0(1-R_1)}{R_1(1-R_0)}}, \qquad N_f = 0$$

$$c = \frac{\ln A}{\ln \dfrac{R_0(1-R_1)}{R_1(1-R_0)}}, \qquad N_f = 0$$

The slope b is given by

$$b = \frac{\ln\left(\dfrac{R_0}{R_1}\right)}{\ln \dfrac{R_0(1-R_1)}{R_1(1-R_0)}}$$

The inequality equation for these conditions is given by

$$a + bn < N_f < c + bn$$

Accept/reject charts at delivery milestones when based on reliability sequential testing methods provide a rigorous mathematical method for deciding whether or not to accept or reject an order of components. The actual reliability value for these

TABLE C-9.—POWER SUPPLY PROBLEM DATA

Sample serial number	Number of failures	Reason for failure	Repair time, hr
1	1	A1A-2VR3 zener shorted	1.2
	1	Ground wire broke	1.4
	2	A1A2-VR3 zener shorted; A1A2-Q2 transistor shorted	5.5, 7.3
	0	In a 250-hr test no failure occurred	-------
2	0	In a 250-hr test no failure occurred	-------
	1	A3A1-C3 capacitor leaked	9.5
3	1	A3A1-C3 capacitor leaked	9.0
	0	In a 250-hr test no failure occurred	-------
4	1	A7A1-VR1 unsoldered joint	.5
		A3A1-C3 capacitor leaked	9.5
5	0	In a 250-hr test no failure occurred	-------

components is not known and neither is it wise to consider reliability assessment at this critical milestone.

Subsample f chart.—The chief advantages of a subsample *f* chart are that (1) it reduces reliability acceptance testing costs, (2) it provides for product improvements, (3) it determines if statistical control exists, and (4) it determines the mean time to repair.

Example 21: A power supply has the following data:

(1) Acceptable reliability level r_1, 0.01 failure/hr; producer's reliability risk R_α, 10 percent; specified mean time to repair, 3.0 hr

(2) Lot tolerance fractional reliability deviation, r_2, 0.005 failure/hr; consumer's reliability risk R_β, 10 percent

The product test data are given in table C–9. Use figure C–13 to analyze these data; then answer the following questions:

(1) What is a suitable time sample and rejection number for meeting the 80-percent confidence level selected by management?

(2) What are the subsample sizes and rejection numbers?

(3) What are the confidence levels for the various rejection numbers?

(4) What are the control limits on the mean time to repair?

(5) Plot these data on a subsample *f* chart.

(6) What should be done with the manufactured units?

Solution 21: Given the product data, follow these steps:

(1) Calculate the confidence level γ, the ratio of acceptable reliability level to lot tolerance fractional reliability deviation k, and the mean time between failures m:

$$\gamma = 1 - \left(R_\alpha + R_\beta\right) = 1 - (0.1 + 0.1) = 0.80, \text{ or } 80 \text{ percent}$$

$$k = \frac{r_2}{r_1} = \frac{0.005}{0.001} = 5$$

$$m = \frac{1}{r_1} = \frac{1}{1 \times 10^{-3}} = 1000 \text{ hr}$$

Looking up Z_α in a normal curve area table (table 3 in ref. C–3) for $R_\alpha = 0.1$ shows that $Z_\alpha = -1.28$. The value of K^2 when $k = 5$ and $\gamma = 0.80$ is obtained from figure 11–1 in reference C–3, where $K^2 = 1.05$. The equation for t is thus $t = mK^2 = (1000)(1.05) = 1050 \text{ hr} \cong 1000$ hr. The rejection number R for a time sample of 1000 hr and a confidence level $\gamma = 0.80$ is given by

$$R_{1000(0.80)} = K^2 + Z_\alpha K + 0.5$$
$$= 1.05 + (1.28)1.025 + 0.5 = 2.86 \approx 3$$

(2) Recalculate the subsample for $\gamma = 0.50$ and $k = 5$: From figure 11–1 in reference C–3, $K^2 = 0.29$. Therefore,

$$t = mK^2 = (1000)(0.29) = 290 \text{ hr} \approx 250 \text{ hr}$$

Looking up Z_α in table 3 in reference C–3 for

$$R_\alpha = \frac{1-\gamma}{2} = \frac{0.5}{2} = 0.25$$

shows that $Z_\alpha = -0.68$. Recalculate the rejection number as

$$R_{250(0.50)} = K^2 + Z_\alpha K + 0.5 = 0.29 + (0.68)0.54 + 0.5$$
$$= 1.16 \approx 1 \text{ failure}$$

(3) Calculate K^2 for each value of t shown in table C–10 as

$$K^2 = \frac{t}{m} = \frac{250}{1000} = 0.25 \qquad \text{for } k = 5;\ m = 1000 \text{ hr}$$

Look up in figure 11–1 of reference C–3 the confidence level γ values shown in table C–10. Calculate R_α for each confidence level. (The calculated values are shown in table C–10.)

$$R_\alpha = \frac{1-\gamma}{2} = \frac{1-0.46}{2} = 0.27$$

Look up Z_α for each confidence level in table 3 of reference C–3 (the values are tabulated in table C–10). Recalculate the

SUBSAMPLE *f* CHART

General identification	Subsample No.	1	2	3	4		5	6	7	8		9	10	11	12		
	Sub-Time Sample, Hrs.	250	250	250	250		250	250	250	250		250	250	250	250		
Chart Number 1 ARL = 0.001 fph	Cumulative Rejection No.	1	2	2	3		1	2	2	3		1	2	2	3		
	Cumulative Failures	1	2	4	4		0	1	2	2		1	2	2	2		
LTRFD = 0.005 fph k = 5	Sample unit S. N.	1	1	1	1		2	2	3	3		4	4	5	5		
	R ( ) _____																
Time Sample 1,000 hrs.																	
Subsample size 150 hrs.	R ( ) _____																
MTTR = MAP =	R_{1000} (0.80) _____																
Item Name	R_{500} (0.63) _____																
Item Drawing Number	R_{values} _____																

10 9 8 7 6 5 4 3 2 1 0

Fail. No.	Reason for the Failure	1	2	3	4	5	6	7	8	9	10	11	12		
1	A1A2-VR3 zener shorted	1		1											
2	Ground wire broke		2												
3	A1A2-Q2 transistor shorted			3											
4	A3A1-C3 capacitor leaked						4	4			4				
5	A7A1-VR1 unsoldered joint									5					
6															
7															
8															
9															
10															

Maintenance Time Analysis

UCL = 6.8 hrs (•)

Specified MTTR = 3.0 hrs. (C)

LCL = 1.8 hrs (•)

10 9 8 7 6 5 4 3 2 1 0

•4 •1 •3 •2 •5 •6 •8 •7 Δ Δ Δ

Figure C–13.—Completed subsample f chart for problem 22.

TABLE C-10.—SUBSAMPLE DATA

t	K^2	γ, percent	R_α	Z_α	$R_t(\gamma)$
250	0.25	0.46	0.27	0.61	1
500	.50	.63	.185	.89	2
750	.75	.73	.133	1.11	2
1000	1.0	.78	.11	1.22	3

rejection numbers $R_t(\gamma)$ for each subsample (the values are listed in table C–10):

$$R_{t(\gamma)} = K^2 + Z_\alpha K + 0.5_Z$$

$$R_{250(0.46)} = 0.25 + (0.61)0.5 + 0.5 = 1.05 \approx 1$$

$$R_{500(0.63)} = 0.50 + (0.89)0.71 + 0.5 = 1.63 \approx 2$$

$$R_{750(0.73)} = 0.75 + (1.11)0.87 + 0.5 = 2.21 \approx 2$$

$$R_{1000(0.78)} = 1.00 + (1.22)1 + 0.5 = 2.72 \approx 3$$

(4) Find the control limits on the mean time to repair for the data given in table C–9:

$$UCL_\phi = \frac{2f\phi}{\chi^2_{8(0.90)}} = \frac{2\times4\times3}{3.49} = 6.88 \text{ hr}$$

$$LCL_\phi = \frac{2f\phi}{\chi^2_{8(0.10)}} = \frac{2\times4\times3}{13.4} = 1.79 \text{ hr}$$

where f is the average number of failures and ϕ denotes mean time to repair. These control limits are shown in figure C–13 for the repair time process. The lower control limit in this case has no importance other than statistical completeness because any value less than 1.79 hr is an indication of a better maintenance activity than what has been specified—a desirable condition.

The completed subsample f chart is shown in figure C–13. Table C–11 shows the tabulated data calculated to solve this problem. During the various subsample intervals, some useful conclusions can be drawn:

(1) During subsample interval 1 to 4 failures

$$\sum_{i=1}^{4} f_i \geq R$$

reject serial number 1, request an engineering investigation, and repair and retest serial number 1 later.

(2) During subsample interval 5 to 8 failures

TABLE C-11.—POWER SUPPLY ANALYZED DATA
[Sample size, 250 hr.]

Time sample	Sample serial number	Subsample number	Reason for failure	Number of failures	Repair time, hr	Mean time to repair, hr
1	1	1	A1A2-VR3 zener shorted	1	1.2	---
		2	Ground wire broke	1	1.4	---
		3	A1A2-VR3 zener shorted; AHV2-Q2 transistor shorted	2	5.5, 7.3	5.1
		4	No failures occurred	0	-------	---
2	2	5	No failures occurred	0	-------	---
		6	A3A1-C3 capacitor leaked	1	9.5	---
	3	7	A3A1-C3 capacitor leaked	1	9.0	4.6
		8	No failures occurred	0	-------	---
3	4	9	A7A1-VR1 unsoldered joint	1	0.5	---
		10	A3A1-C3 capacitor leaked	1	9.5	---
	5	11	No failures occurred	0	-------	---
		12	No failures occurred	0	-------	---
			Totals	8	48.9	---

$$\sum_{i=5}^{8} f_i \leq R$$

ship serial numbers 2 and 3 after all failures have been reviewed, the cause identified, and appropriate corrective action worked out and approved by an engineering review board.

(3) During subsample interval 9 to 12 failures

$$\sum_{i=9}^{12} f_i \leq R$$

ship serial numbers 4 and 5 after all failures have been reviewed, properly closed out, and approved by the engineering review board.

References

C–1. Bazovsky, I.: Reliability Theory and Practice. Prentice-Hall, 1961.

C–2. Earles, D.R.; and Eddins, M.F.: Reliability Physics. AVCO Corp., Wilmington, MA, 1962.

C–3. Calabro, S.R.: Reliability Principles and Practices. McGraw-Hill, 1962.

C–4. Berrettoni, J.N.: Practical Applications of the Weibull Distribution. American Society for Quality Control, Annual Technical Conference Transactions, vol. 16, 1962, p. 303.

C–5. Failure Distribution Analyses Study. Vols. I, II, and III. Computer Applications Inc., NY, Aug. 1964.

C–6. Hoel, P.G.: Elementary Statistics. John Wiley & Sons, 1960.

Bibliography

Arsenault, I.E.; and Roberts, J.A.: Reliability and Maintainability of Electronic Systems. Computer Science Press, 1980.

Balaban, H.S.; and Retterer, B.L.: Guidelines for Application of Warranties to Air Force Electronics Systems. RADC–TR–76–32, Mar. 1976. (Avail. NTIS, AD–A023956.)

Balaban, H.S.; and Retterer, B.L.: Use of Warranties for Defense Avionics Procurements. Report RADC–TR–73–249, Feb. 1974. (Avail. NTIS, AD–769399/7.)

Bauer, I.A., et al.: Dormancy and Power On-Off Cycling Effects on Electronic Equipment and Part Reliability. Report RADC–TR–73–248, Aug. 1973. (Avail. NTIS, AD–768619.)

Bertin, A.P.: Development of Microcircuit Bond-Pull Screening Techniques. Report RADC–TR–73–123, Apr. 1973. (Avail. NTIS, AD–762333.)

Bevington, J.R., et al.: Reliability Evaluation of Plastic Integrated Circuits. Report RADC–TR–71–8, Jan. 1971. (Avail. NTIS, AD–722043.)

Butler, T.W.; Cottrell, D.F.; and Maynard, W.M.: Failure Rate Mathematical Models for Discrete Semiconductors. Report RADC–TR–78–3, Jan. 1978. (Avail. NTIS, AD–A050181.)

Citrin, D.A.: Electrical Characterization of Complex Microcircuits. Report RADC–TR–72–145, June 1972. (Avail. NTIS, AD–748242.)

Citrin, D.A.: Electrical Characterization of Complex Microcircuits. Report RADC–TR–73–373, Jan. 1974. (Avail. NTIS, AD–775740.)

Clarke, R.N.; and Stallard, B.: Reliability Study of Microwave Power Transistors. Report RADC–TR–75–18, Jan. 1975. (Avail. NTIS, AD–A007788.)

Coit, D.W.: Printed Wiring Assembly and Interconnection Reliability. Report RADC–TR–81–318, Nov. 1981. (Avail. NTIS, AD–A111214.)

Coit, D.W.; and Steinkirchner, J.J.: Reliability Modeling of Critical Electronic Devices. Report RADC–TR–83–108, May 1983. (Avail. NTIS, AD–A135705.)

Coppola, A.; and Sukert, A.: Reliability and Maintainability Management Manual. Report RADC–TR–79–200, July 1979. (Avail. NTIS, AD–A073299.)

Cottrell, D.F.; and Kirejczyk, T.E.: Crimp Connection Reliability—Failure Rate Mathematical Model for Electric Terminals and Connectors. Report RADC–TR–78–15, Jan. 1978. (Avail. NTIS, AD–A050505.)

Crum, F.B., et al.: Warranty-Guarantee Application Guidelines for Air Force Ground Electronic Equipment. RADC–TR–79–287, Aug. 1979. (Avail. NTIS, AD–A082318.)

Department of Defense Acquisition Management Systems and Data Requirements Control List (AMSDL). DOD 5030.19–L, vol. II, July 1981.

Department of Defense Directive 5000.28, Design to Cost, May 23, 1975.

Descriptive Statistics. IEEE Statistics Course at Case Western Reserve University, Spring 1963.

Design Requirements for Rigid Printed Wiring Boards and Assemblies. NHB 5300.4 (3K), Jan. 7, 1986.

Devine, J.: Ultrasonic Beam Lead Bonding Equipment. Report RADC–TR–73–27, Feb. 1973. (Avail. NTIS, AD–757561.)

Domingos, H.: Electro-Thermal Overstress Failure in Microelectronics. Report RADC–TR–73–87, Apr. 1973. (Avail. NTIS, AD–761792.)

Electrical, Electronic, and Electromechanical (EEE) Parts Management and Control Requirements for NASA Space Flight Programs. NHB 5300.4 (1F), July 11, 1989.

Ellingham, D.B., Jr.; Schreyer, W.M.; and Gaertner, W.W.: Development of Failure Rate Models for Semiconductor Optoelectronic Devices. Report FAA–RD–76–134, July 1976. (Avail. NTIS, AD–A029163/3.)

Engleman, J.H.; Kennedy, J.; and Wood, S.R.: Traveling Wave Tube Failure Rates. Report RADC–TR–80–288, Nov. 1980. (Avail. NTIS, AD–A096055.)

Flint, S.: Failure Rates for Fiber Optic Assemblies. Report RADC–TR–80–322, Oct. 1980, (Avail. NTIS, AD–A092315.)

Fulton, D.W.: Nonelectronic Parts Reliability Notebook. Report NPRD–1, 1978. (Avail. NTIS, AD–A059901.)

Gagier, T.R.; Kimball, E.W.; and Selleck, R.R.: Laser Reliability Prediction. Report RADC–TR–75–210, Aug. 1975. (Avail. NTIS, AD–A016437.)

Ghate, P.B.: Failure Mechanisms Studies on Multilevel Metallization Systems for LSI. Report RADC–TR–71–186, Sept. 1971. (Avail. NTIS, AD–731796.)

Guth, G.F.: Development of Nonelectronic Part Cyclic Failure Rates. Report RADC–TR–77–417, Dec. 1977. (Avail. NTIS, AD–A050678.)

Guth, G.F.: Quantification of Printed Circuit Board Connector Reliability. Report RADC–TR–77–33, Jan. 1978. (Avail. NTIS, AD–049980.)

Guth, G.F.: Reliability Prediction Models for Microwave Solid State Devices. Report RADC–TR–79–50, Apr. 1979. (Avail. NTIS, AD–A069386.)

Haberer, J.R.: Stress Induced Intermittent Failures in Encapsulated Microcircuits. Report RADC–TR–70–213, Oct. 1970. (Avail. NTIS, AD–715984.)

Hasty, T.E., et al.: Reliability Physics Study of Microwave Solid State Devices. Report RADC–TR–71–184, Sept. 1971. (Avail. NTIS, AD–731794.)

Hierholzer, E.L.: Passive Device Failure Rate Models for MIL–HDBK–217B. Report RADC–TR–77–432, Nov. 1977. (Avail. NTIS, AD–A050180.)

Hurley, H.C.; Strong, T.M; and Young, M.A.: Reliability Investigation of Thermal Stress/Fatigue Failure in Multilayer Interconnection Boards. Report RADC–TR–70–192, Oct. 1970. (Avail. NTIS, AD–714702.)

Inspection System Provisions for Aeronautical and Space System Materials, Parts, Components and Services. NHB 5330.4 (1C), July 1, 1971.

Joint Design-to-Cost Guide. Departments of the Army, the Navy, and the Air Force, DARCOM–P700–6, NAVMAT–P5242, AFLCP/AFSCP–1–800–19, Oct. 1977. (Avail. NTIS, AD–A048254.)

Klion, J.: A Redundancy Notebook. Report RADC–TR–77–287, Dec. 1977. (Avail. NTIS, AD–A050837.)

Lacombe, D.J.: Reliability Prediction of Microwave Transistor. Report RADC–TR–74–313, Dec. 1974. (Avail. NTIS, AD–A003643.)

Lane, C.H.: Reliability Problems With SiO_2 Passivation and Glassivation. Report RADC–TR–72–55, Mar. 1972. (Avail. NTIS, AD–741765.)

Lane, C.H.: Nichrome Resistor Properties and Reliability. Report RADC–TR–73–181, June 1973. (Avail. NTIS, AD–765534.)

Lauttenburger, H.; and Fuchs, J.: A System for Effective Transferral of Microelectronic Reliability Experience. Annals of Assurance Sciences, Proceedings of the Eighth Reliability and Maintainability Conference, AIAA/SAE, 1969, pp. 503–521.

Leone, F.C., et al.: Percentiles of the Binomial Distribution. Case Institute of Technology, 1967.

Lipow, M.: Airborne Systems Software Acquisition Engineering Guidebook for Quality Assurance. ASD–TR–78–8, Aug. 1977. (Avail. NTIS, AD–A059068.)

Lochner, R.H.: Estimation and Prediction Using the Binomial Distribution. Reliability Res. and Ed. Dept., General Motors Corp., Milwaukee, WI, 1963.

Lochner, R.H.: Reliability Calculations for Exponential Population. Reliability Res. and Ed. Dept., General Motors Corp., Milwaukee, WI, 1963.

Lochner, R.H.: When and How to Use the Weibull Distribution. Reliability Res. and Ed. Dept., General Motors Corp., Milwaukee, WI, 1963.

Long, R.G. Reliability Prediction Modeling of New Devices. Report RADCTR–80–273, July 1980. (Avail. NTIS, AD–A090029.)

Lloyd, D.K.; and Lipow, M.: Reliability: Management, Methods and Mathematics. Prentice-Hall, 1962.

Lyne, G.W.: Implementation of Operational Procedures for Optimized Reliability and Component Life Estimator (ORACLE). Report RADCTR–77–49, Mar. 1977. (Avail. NTIS, AD–A039344.)

Maintainability Engineering Handbook. Naval Air Systems Command, NAVAIR 01–1A–33, July 1977.

Maintainability Program Requirements for Space Systems. NHB 5300.4 (1E), Mar. 10, 1987.

Moore, J.R,; Furnival, C.; and Burt, J.: Reliability of Ceramic Multilayer Boards. Report RADC–TR–71–299, Dec. 1971. (Avail. NTIS, AD–737373.)

Morrison, G.N., et al.: RADC Thermal Guide for Reliability Engineers. Report RADC–TR–82–172, June 1982. (Avail. NTIS, AD–A118839.)

NAVAIR–01–1A–31, Reliability and Maintainability Management Handbook. Naval Air System Command. July 1977.

NAVAIR–01–1A–33, Maintainability Engineering Notebook. Naval Air System Command. July 1977.

Neil, G.R.; and Gold. H.I.: Software Acquisition Management Guidebook: Software Quality Assurance. ESD–TR–77–255, Aug. 1977. (Avail. NTIS, AD–A047318.)

Pieruschka, E.: Principles of Reliability. Prentice-Hall, 1963.

Plein, K.M.; Funk, J.R.; and James, L.E.: Reliability Study of Circular Electrical Connectors. Report RADC–TR–73–171, June 1973. (Avail. NTIS, AD–765609.)

Product Performance Agreement Guide. Joint AFSC/AFLC Publication, Aug. 1980.

Quality Assurance Program. AFSCR–74–1, Nov. 1978.

Quality Program Provisions for Aeronautical and Space System Contractors. NHB 5300.4 (1B), Apr. 1, 1969.

Reliability and Maintainability Management Handbook. NAVAIR 01–1A–31, July 1977.

Reliability and Maintainability Planning Guide for Army Aviation Systems and Components. U.S. Army Aviation Research and Development Command, St. Louis, MO, 1974.

Reliability and Maintainability Planning Notebook. Federal Aviation Administration, Washington, DC, 1980.

Reliability by Design. General Electric Co., Defense Elect. Div., Waynesboro, VA, 1964.

Reliability Modeling and Prediction. MIL–STD–756, Aug. 31, 1982.

Reliability Program Requirements for Aeronautical and Space System Contractors. NHB 5300.4 (1A–1), Jan. 21, 1987.

Reliability Theory and Practice. ARINC Res. Corp., Washington, DC, 1962.

Requirements for Conformal Coating and Stacking of Printed Wiring Boards and Electronic Assemblies. NHB 5300.4 (3J), Apr. 1, 1985.

Requirements for Crimping and Wire Wrap. NHB 5300.4(3H), May 1, 1984.

Requirements for Electrostatic Discharge Control. NHB 5300.4 (3X), Draft Copy, Dec. 1990.

Requirements for Interconnecting Cables, Harnesses, and Wiring. NHB 5300.4(3G), Apr. 1, 1989

Requirements for Printed Wiring Boards. NHB 5300.4 (3I), May 1, 1984.

Requirements for Soldered Electrical Connections. NHB 5300.4 (3A–1), Dec. 1, 1976.

Rickers, H.C.: LSI/Microprocessor Reliability Prediction Model Development. Report RADC–TR–79–97, Mar. 1979. (Avail. NTIS, AD–A068911.)

Rigling, W.S.: Reliability Study of Polyimide/Glass Multilayer Boards. Report RADC–TR–73–400, Jan. 1974. (Avail. NTIS, AD–771994.)

Safety, Reliability, Maintainability and Quality Provisions for the Space Shuttle Program. NHB 5300.4 (1D–2), Oct. 1, 1979.

Sandler, G.H.: System Reliability Engineering. Prentice-Hall, 1963.

Schafer, R.E., et al.: Contact Reliability Screening. Report RADC–TR–72–326, Dec. 1972. (Avail. NTIS, AD–755923.)

Schafer, R.E.; and Sheffield, T.S.: Bayesian Reliability Demonstration: Phase II, Development of A priori Distribution. Report RADC–TR–71–209, Oct. 1971. (Avail. NTIS, AD–732283.)

Schafer, R. E.; Sheffield, T.S.; and Collins, T.R.: Bayesian Reliability Demonstration: Phase III, Development of Test Plans. Report RADC–TR–73–139, June 1973. (Avail. NTIS, AD–765172.)

Smith, J.S.; Kapfer, V.C.; and Doyle, E.A., Jr.: Reliability Evaluation of 54L20 Radiation Hardened Dual NAND Gates. Report RADC–TR–73–180, June 1973. (Avail. NTIS, AD–765173.)

Tees, W.G.: Predicting Failure Rates of Yield Enhanced LSI. Comput. Des., vol. 10, no. 2, Feb. 1971, pp. 65–71.

Toohey, E.F., and Calvo, A.B.: Cost Analyses for Avionics Acquisition. 1980 Annual Reliability and Maintainability Symposium, IEEE, 1980, pp. 85–90.

Turner, T.E.: Hybrid Microcircuit Failure Rate Prediction. Report RADC–TR–78–97, Apr. 1978. (Avail. NTIS, AD–A055756.)

Wilcox, R.H.: Redundancy Techniques for Computer Systems. Spartan Books, Washington, DC, 1962.

Wilson, D.S.; and Smith, R.: Electric Motor Reliability Model. Report RADC–TR–77–408, Dec. 1977. (Avail. NTIS, AD–A050179.)

Wilson, D.S.; and Wilkinson, J.W.: Reliability Model for Miniature Blower Motors per MIL–B–23071B. Report RADC–TR–75–178, July 1975. (Avail. NTIS, AD–A013735.)

Woodgate, R.W.: Infrared Testing of Multilayer Boards. Report RADC–TR–74–88, Apr. 1974. (Avail. NTIS, AD–780550.)

Zimmer, R.P., et al.: High Power Microwave Tube Reliability Study. Report FAA–RD–76–172, Aug. 1976. (Avail. NTIS, AD–A033612/3.)

Reliability Training Answers

Chapter	Answers
1	(B), 2 (D), 3 (C), 4 (C)
2	1a (C), 1b (B), 2a (C), 2b (B), 3a (C), 3b (A), 4a (B), 4b (C), 5ai (B), 5aii (C), 5aiii (B), 5b (C), 6a (C), 6b (B), 7a (B), 7b (C), 8a (C), 8b (C), 9 (D), 10 (A), 11 (B), 12 (C), 13 (C), 14 (C), 15 (D), 16 (E), 17 (D), 18 (F)
3	1a (B), 1b (B), 1c (C), 2a (A), 2b (C), 2c (A), 3a (B), 3b (A), 3c (B), 4 (C), 5a (B), 5b (B), 6 (C), 7a (A), 7b (B), 7c (B), 7d (C), 7e (A), 8 (B), 9a (B), 9b (C), 10a (C), 10b (C), 10c (A)
4	1a (B), 1b (B), 2a (A), 2b (A), 3 (C), 4a (B), 4b (B)
5	1 (C), 2 (B), 3a (C), 3b (A), 3c (C), 4a (C), 4b (B), 4c (A), 5a (C), 5b (A), 6a (C), 6b (C), 6c (A), 7a (B), 7b (C), 7c (C), 7d (C), 8a (A), 8b (C), 8c (B), 8d (C), 8e (B), 8f (B)
6	1a (B), 1b (C), 1c (A), 2a (C), 2b (B), 2c (A), 2d (C), 3a (B), 3b (C), 3ci (B), 3cii (A)
7	1 (C), 2 (B), 3 (D), 4 (A), 5 (B), 6 (C), 7 (B), 8 (C)
8	1. Item 4, squawk, major, wrong, reliability, subsystem
9	1 (B), 2 (A), 3 (C), 4a (C), 4b (B), 4c (F), 5 (A), 6a (C), 6b (B), 7 (A), 8a (B), 8b (A)
10	1 (D), 2 (D), 3 (G), 4 (B), 5 (A), 6 (E), 7 (B), 8 (D), 9 (A), 10 (C), 11 (B), 12 (F), 13 (E), 14a (C), 14b (C), 15 (C), 16 (B), 17 (E), 18 (A), 19a (C), 19b (B), 19c (A)
11	1a (C), 1b (B), 1c (C), 1d (C), 2a (C), 2b (A), 2c (B), 2d (C)

Appendix D
Training Manual for Elements of Interface Definition and Control

As part of this reliability and maintainability training manual, the authors have included in this appendix the published document *Training Manual for Elements of Interface Definition and Control*. Their desire was to provide the reader the complete texts for reliability training. This manual was published in 1997 and appears here exactly as it does in print. To avoid confusion, the reader should note that the original page numbers and content have been retained.

NASA Reference Publication 1370

1997

Training Manual for Elements of Interface Definition and Control

Vincent R. Lalli
Lewis Research Center
Cleveland, Ohio

Robert E. Kastner
Vitro Corporation
Rockville, Maryland

Henry N. Hartt
Vitro Corporation
Washington, DC

lational Aeronautics and
pace Administration

ffice of Management
cientific and Technical
formation Program

Preface

This technical manual was developed under the Office of Safety and Mission Assurance continuous training initiative. The structured information contained in this manual will enable the reader to efficiently and effectively identify and control the technical detail needed to ensure that flight system elements mate properly during assembly operations (both on the ground and in space).

Techniques used throughout the Federal Government to define and control technical interfaces for both hardware and software were investigated. The proportion of technical information actually needed to effectively define and control the essential dimensions and tolerances of system interfaces rarely exceeded 50 percent of any interface control document. Also, the current Government process for interface control is very paper intensive. Streamlining this process can improve communication, provide significant cost savings, and improve overall mission safety and assurance.

The primary thrust of this manual is to ensure that the format, information, and control of interfaces between equipment are clear and understandable, containing only the information needed to guarantee interface compatibility. The emphasis is on controlling the engineering design of the interface and not on the functional performance requirements of the system or the internal workings of the interfacing equipment. Interface control should take place, with rare exception, at the interfacing elements and no further.

There are two essential sections of the manual. The first, Principles of Interface Control, discusses how interfaces are defined. It describes the types of interface to be considered and recommends a format for the documentation necessary for adequate interface control. The second, The Process: Through the Design Phases, provides tailored guidance for interface definition and control.

This manual can be used to improve planned or existing interface control processes during system design and development. It can also be used to refresh and update the corporate knowledge base. The information presented herein will reduce the amount of paper and data required in interface definition and control processes by as much as 50 percent and will shorten the time required to prepare an interface control document. It also highlights the essential technical parameters that ensure that flight subsystems will indeed fit together and function as intended after assembly and checkout.

Acknowledgments

In 1992 the NASA Reliability and Maintainability Steering Committee recognized the need to provide its engineers, especially its design engineers, with a practical understanding of the principles and applications of interface definition and control documentation. A working group was formed and met at Goddard Space Flight Center to discuss how some of the NASA centers were handling this topic. Four centers and NASA Headquarters participated in the meeting: Headquarters—NASA handbook methods; Johnson Space Center (JSC) and Marshall Space Flight Center (MSFC)—space station; Lewis Research Center—space station and launch vehicles; Jet Propulsion Laboratory (JPL)—Mars Observer; and Goddard Space Flight Center (GSFC)—space experiments.

To satisfy the need for a short, informative interface definition and control training manual, Robert E. Kastner and Henry N. Hartt of the Vitro Corporation prepared the manual using material from the working group meeting. Harvey L. Schabes and William J. Taylor of Lewis Research Center served as the final NASA project office reviewers. Their suggestions improved the usefulness of the text for flight projects. The dedication, time, and technical contributions of Jack Remez/GSFC; Donald Bush (retired)/MSFC; David Oberhettinger/SYSCON (JPL); Daniel Deans/LORAL (JSC); and Ronald Lisk/NASA Headquarters (Code Q) in the preparation of this manual are appreciated. Without the support of their individual centers and their enthusiastic personal support and willingness to serve on the NASA Reliability and Maintainability Steering Committee, this manual would not have been possible.

The following NASA members of the steering committee may be contacted for more information about the processes and products discussed in this manual:

James F. Clawson
Jet Propulsion Laboratory
California Institute of Technology
MS 301–456, SEC 505
4800 Oak Grove Drive
Pasadena, CA 91109
Office: (818) 354–7021
Facsimile: (818) 393–4699
E-mail: james.f.clawson@ccmail.jpl.nasa.gov

John Greco
NASA Langley Research Center
MS 421, Bldg. 1162A, Rm. 125
5A Hunsaker Loop
Hampton, VA 23681–0001
Office: (804) 864–3018
Facsimile: (804) 864–6327
E-mail: j.a.greco@larc.nasa.gov

Wilson Harkins
NASA Headquarters, Code QS
300 E. Street SW
Washington, DC 20546
Office: (202) 358–0584
Facsimile: (202) 358–3104
E-mail: wharkins@cc.hq.nasa.gov

Vincent R. Lalli
NASA Lewis Research Center
Code 0152, MS 501–4
Cleveland, OH 44135
Office: (216) 433–2354
Facsimile: (216) 433–5270
E-mail: rqlalli@lims01.lerc.nasa.gov

Michael E. Langley
NASA George C. Marshall Space Flight Center
CR–10, Bldg. 4203
Marshall Space Flight Center, AL 35812
Office: (205) 544–0056
Facsimile: (205) 544–4155
E-mail: michael.langley@msfc.nasa.gov

Dan Y. Lee
NASA Ames Research Center
MS 240A–3, P.O. Box 1000
Moffet Field, CA 94035–1000
Office: (415) 604–5962
Facsimile: (415) 604–0399
E-mail: dan_lee@qmgate.arc.nasa.gov

Leon R. Migdalski
NASA Kennedy Space Center
RT–SRD–2 OSB3309
Kennedy Space Center, FL 32899
Office: (407) 861–3284
Facsimile: (407) 861–4314
E-mail: lmigdals@srqa.ksc.nasa.gov

Jack W. Remez
NASA Goddard Space Flight Center
Code 302, Bldg. 6, Rm. S240
Greenbelt, MD 20771
Telephone: (301) 286–7113
Facsimile: (301) 286–1701
E-mail: remez@gsfc.nasa.gov

Donald L. Wiley
NASA Lyndon B. Johnson Space Center
Code NS3, Bldg. 45, Rm. 616B
Houston, TX 77058
Office: (713) 483–4084
Facsimile: (713) 483–3045
E-mail: dwiley@gp101.jsc.nasa.gov

Contents

Chapter 1
Introduction

This technical manual resulted from an investigation of techniques used throughout NASA and other Federal Government agencies to define and control technical interfaces for both hardware and software. The processes described herein distill the requirements for interface definition and control into a concise set of parameters that control the design of only the interface-related elements rather than providing extraneous design detail that must subsequently be configuration managed.

The purpose of this manual is to provide guidelines for establishing and conducting the interface control process so that items produced by different design agencies satisfactorily mate and operate in a way that meets mission requirements. These guidelines were drawn from the methodologies of a number of highly successful programs and therefore represent a compilation of "lessons learned."

The principles and processes of interface definition and control presented in this document apply to all projects and programs but may be tailored for program complexity. For example, the interface control process may be less formal for a project or program that requires only one or two end items and has few participants; however, the formal interface control document is still necessary. For a project or program that requires a number of end items and where several participants are involved, a carefully followed interface control process is imperative, with comments, decisions, agreements, and commitments fully documented and tracked. Individual managers should provide the implementation criteria for their interface control processes early in the project or program (ref. 1).

This manual covers the basic principles of interface definition and control: how to begin an interface control program during the development of a new project or program, how to develop and produce interface documentation, how to manage the interface control process, and how to transfer interface control requirements to hardware and software design.

Interface definition and control is an integral part of system engineering. It should enter the system engineering cycle at the end of the concept development phase. Depending on whether the system under development is designed for one-time or continuous use, the process may continue for the full life cycle of the system. Interface definition and control should not be equated to configuration management or configuration control. Rather it is a technical management tool that ensures that all equipment will mate properly the first time and will continue to operate together as changes are made during the life cycle of the system. Figure 1.1 depicts the elements of the system engineering cycle and is used in chapter 3 to describe the application of the interface discipline at different parts of the life cycle (ref. 2).

Establishing a system that ensures that all interface parameters are identified and controlled from the initial design activities of a program is essential. It is not necessary that the fine details of these parameters be known at that time, but it is very important that the parameters themselves are identified, that everything known about them at that time is recorded and controlled, and that voids[1] are identified and scheduled for elimination. The latter requirement is of primary importance to the proper design of any interface. Initial bounding of a void and scheduled tightening of those bounds until the precise dimensions or conditions are identified act as a catalyst to efficient design and development. An enforced schedule for eliminating voids is one of the strongest controls on schedule that can be applied (ref. 3).

The process of identifying, categorizing, defining, and documenting interfaces is discussed in the following chapter. Guidance for the analysis of interface compatibility is also provided.

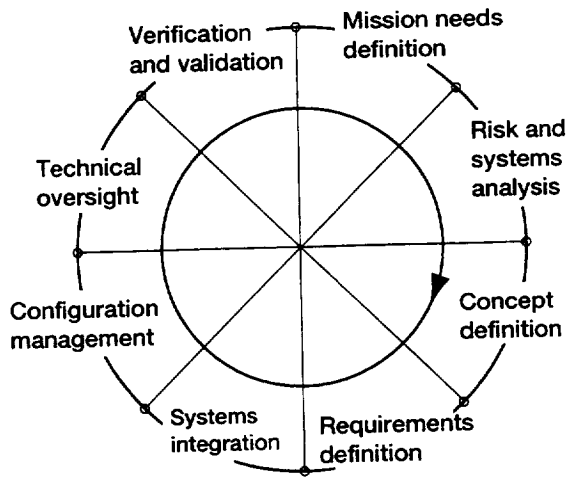


Figure 1.1—System engineering cycle. (The requirements definition phase must include the requirements for the interfaces as well as those which will eventually be reflected in the interface control document.)

[1] A "void" is a specific lack of information needed for control of an interface feature. Control and elimination of voids is fundamental to a strong interface definition and control program.

1.1 Training[2]

1. The processes explained in this manual for interface definition and control are
   A. A concise set of parameters that control the design of the interface-related elements
   B. A set of design details needed for configuration management

2. The process is very important for projects that require
   A. A number of end items
   B. Involvement of several participants
   C. Comments, decisions, agreements, and commitments that must be fully documented and tracked
   D. All of the above

3. What elements does the system engineering cycle contain?
   A. Mission needs, requirements, and integration
   B. Technical oversight, core design, and system configuration
   C. Mission needs definition, risk and systems analysis, concept and requirements definitions, system integration, configuration management, technical oversight, and verification and validation

4a. What is a void?
   A. Bracketed data
   B. Wrong data
   C. Lack of information needed

4b. How should voids be handled?
   A. Voids should be identified and their elimination scheduled.
   B. Data should be analyzed.
   C. Supplier should be guided.

4c. Name a strong control needed for voids.
   A. Precise dimensions
   B. Enforced schedule
   C. Identified catalysts

[2]Answers are given at the end of this manual.

Chapter 2
Principles of Interface Control

2.1 Purpose of Interface Control

An interface is that design feature of a piece of equipment[3] that affects the design feature of another piece of equipment. The purpose of interface control is to define interface requirements so as to ensure compatibility between interrelated pieces of equipment and to provide an authoritative means of controlling the design of interfaces. Interface design is controlled by an interface control document (ICD).

These documents

1. Control the interface design of the equipment to prevent any changes to characteristics that would affect compatibility with other equipment
2. Define and illustrate physical and functional characteristics of a piece of equipment in sufficient detail to ensure compatibility of the interface, so that this compatibility can be determined from the information in the ICD alone
3. Identify missing interface data and control the submission of these data
4. Communicate coordinated design decisions and design changes to program participants
5. Identify the source of the interface component

ICD's by nature are requirements documents: they define design requirements and allow integration. They can cause designs to be the way they are. They record the agreed-to design solution to interface requirements and provide a control mechanism to ensure that the agreed-to designs are not changed by one participant without negotiated agreement of the other participant.

To be effective, ICD's should track a schedule path compatible with design maturation of a project (i.e., initial ICD's should be at the 80% level of detail at preliminary design review, should mature as the design matures, and should reach the 99% mark near the critical design review).

2.2 Identifying Interfaces

Identifying where interfaces are going to occur is a part of systems engineering that translates a mission need into a configured system (a grouping of functional areas) to meet that need. Each functional area grouping is assigned certain performance requirements. These performance requirements are translated into design requirements as the result of parametric studies, tradeoff studies, and design analyses. The design requirements are the basis for developing the system specifications. The boundaries between the functional areas as defined in the system specifications become the interfaces. Early interface discussions often contribute to final subsystem specifications. Interface characteristics, however, can extend beyond the interface boundary, or interface plane, where the functional areas actually come together. The interface could be affected by, and therefore needs to be compatible with, areas that contribute to its function but may not physically attach. For example, it may be necessary to define the path of a piece of equipment as it traverses through another piece of equipment and rotates and articulates to carry out its function. Electrical characteristics of a transmitter and receiver separated by an interface plane may have to be defined for each to properly function. Similarly, the acoustic energy produced by one component and transmitted through the structure or onto another component may need a corresponding definition.

Identifying interfaces early in a program is essential to successful and timely development. Functional analyses are used for analyzing performance requirements and decomposing them into discrete tasks or activities (i.e., decomposing the primary system functions into subfunctions at ever increasing levels of detail). Functional block diagrams are used to define data flow throughout the system and interfaces within the system. Once the segments and elements within a system have been defined, a top-level functional block diagram is prepared. The block diagrams are then used in conjunction with *N*-squared diagrams to develop interface data flows. The *N*-squared diagram is a technique used extensively to develop data interfaces but can also be refined for use in defining hardware interfaces. However, use of this tool in this manual will be restricted to interface categorization. Additional description is provided in section 3.1.1.

In summary, identifying where interfaces are going to occur begins the systems integration component of systems engineering and must start early in design planning. The interface boundaries or planes vary from program to program depending on how design and development responsibilities are assigned. Interface control can occur within a functional area of other design and development agents. Therefore, interfaces can be identified at many levels, for example,

1. Center to center
2. Discipline to discipline (e.g., propulsion to guidance, sensor to structure, or power to users)
3. Contractor to contractor

[3]For purposes of this manual, a piece of equipment is a functional area assigned to a specific source. Thus, a piece of equipment can be an element of the space station, a system of a spacecraft, a work package assigned to a contractor, or a subsystem.

4. Center to contractor to discipline
5. Program to program (e.g., shuttle to National Launch System)

Once interface boundaries or planes are established, the interfaces must be categorized and defined.

2.3 Categorizing (Partitioning) and Defining Interfaces

Categorizing, or partitioning, interfaces separates the interface features by technical discipline and allows each category, in most cases, to proceed through the definition process independently.

The following basic interface categories (defined by the types of feature and data they encompass) are recommended for use in most programs:

1. Electrical/functional
2. Mechanical/physical
3. Software
4. Supplied services

During the early phases of systems engineering, interfaces may be assigned only the high-level designation of these categories. As the system becomes better defined, the details of the physical and functional interface characteristics become better defined and are documented.

An interface can be assigned to one of these categories by a number of processes of elimination. The one recommended for use is the *N*-squared diagram (ref. 4), which is currently being used by some NASA centers.

2.3.1 Electrical/Functional

Electrical/functional interfaces are used to define and control the interdependence of two or more pieces of equipment when the interdependence arises from the transmission of an electrical signal from one piece of equipment to another. All electrical and functional characteristics, parameters, and tolerances of one equipment design that affect another design are controlled by the electrical/functional ICD. The functional mechanizations of the source and receiver of the interface electrical signal are defined, as well as the transmission medium.

The interface definition includes the data and/or control functions and the way in which these functions are represented by electrical signals. Specific types of data to be defined are listed here:

1. Function name and symbol
2. Impedance characteristics
3. Shielding and grounding
4. Signal characteristics
5. Cable characteristics
6. Data definition
7. Data transmission format, coding, timing, and updating
8. Transfer characteristics
9. Circuit logic characteristics
10. Electromagnetic interference requirements
11. Data transmission losses
12. Circuit protective devices

Other data types may be needed. For example, an analog signal interface document would contain function name and symbol, cable characteristics, transfer characteristics, circuit protective devices, shielding, and grounding; whereas a digital data interface would contain function name and symbol, data format, coding, timing and updating, and data definition.

Additional data types under the electrical/functional heading are

1. Transmission and receipt of an electrical/electromagnetic signal
2. Use of an electrically conductive or electromagnetic medium

Appendix A shows recommended formats for electrical and functional interface control drawings.

2.3.2 Mechanical/Physical

Mechanical/physical interfaces are used to define and control the mechanical features, characteristics, dimensions, and tolerances of one equipment design that affect the design of another subsystem. They also define force transmission requirements where a static or dynamic force exists. The features of the equipment that influence or control force transmission are also defined in this ICD. Mechanical interfaces include those material properties of the equipment that can affect the functioning of mating equipment, such as thermal and galvanic characteristics. Specific types of data defined are

1. Optical characteristics
2. Parallelism and straightness
3. Orientation requirements
4. Space or provisions required to obtain access for performing maintenance and removing or replacing items, including space for the person performing the function
5. Size, shape, mass, mass distribution, and center of gravity
6. Service ports
7. Indexing provisions
8. Concentricity
9. Surface finish
10. Hard points for handling

11. Sealing, pressurization, attachment, and locking provisions
12. Location and alignment requirements with respect to other equipment
13. Thermal conductivity and expansion characteristics
14. Mechanical characteristics (spring rate, elastic properties, creep, set, etc.)
15. Load-carrying capability
16. Galvanic and corrosive properties of interfacing materials

Other data types may be needed. For example, an ICD controlling a form-and-fit interface would generally contain such characteristics as size and shape of the item, location of attachment features, location of indexing provisions, and weight and center of gravity of the item. However, an ICD controlling a structural load interface would contain weight and center of gravity, load-carrying capability, and elastic properties of the material if applicable to the loading conditions. Not all ICD's controlling a form-and-fit interface would have to contain all types of data given in this example, but some form-and-fit interface definitions contain more than the 16 types of data listed. Indexing definitions may require angularity, waviness, and contour definitions and tolerances.

Additional data types under the mechanical/physical heading would be

1. Dimensional relationships between mating equipment
2. Force transmission across an interface
3. Use of mechanically conductive media
4. Placing, retaining, positioning, or physically transporting a component by another component
5. Shock mitigation to protect another component

Appendix B (from ref. 5) shows a mechanical/physical drawing.

This extensive variety of possibilities and combinations prevents assigning a standard set of data types or level of detail to a form-and-fit interface. Each interface must be analyzed and the necessary controlling data identified before the proper level of interface definition and control can be achieved. This holds true for all examples given in this chapter.

2.3.3 Software

A software interface defines the actions required when interfacing components that result from an interchange of information. A software interface may exist where there is no direct electrical interface or mechanical interface between two elements. For example, whereas an electrical ICD might define the characteristics of a digital data bus and the protocols used to transmit data, a software interface would define the actions taken to process the data and return the results of the process. Software interfaces include operational sequences that involve multiple components, such as data-processing interactions between components, timing, priority interrupts, and watchdog timers. Controversy generally arises in determining whether these relationships are best documented in an electrical/functional ICD, a software ICD, or a performance requirements document. Generally, software interface definitions include

1. Interface communication protocol
2. Digital signal characteristics
3. Data transmission format, coding, timing, and updating requirements
4. Data and data element definition
5. Message structure and flow
6. Operational sequence of events
7. Error detection and recovery procedures

Other data types may be needed. Appendix C provides an example of a software interface signal.

2.3.4 Supplied Services

Supplied services are those support requirements that a piece of equipment needs to function. Supplied services are provided by an external separate source. This category of interface can be subdivided further into electrical power, communication, fluid, and environmental requirements. The types of data defined for these subcategories are

1. Electrical power interface:
   a. Phase
   b. Frequency
   c. Voltage
   d. Continuity
   e. Interrupt time
   f. Load current
   g. Demand factors for significant variations during operations
   h. Power factor
   i. Regulation
   j. Ripple
   h. Harmonics
   l. Spikes or transients
   m. Ground isolation
   n. Switching, standby, and casualty provisions
2. Communication interface:
   a. Types of communication required between equipment
   b. Number of communication stations per communication circuit
   c. Location of communication stations
3. Fluid interface:
   a. Type of fluid required
      i. Gaseous
      ii. Liquid

b. Fluid properties
   i. Pressure
   ii. Temperature
   iii. Flow rate
   iv. Purity
   v. Duty cycle
   vi. Thermal control required (e.g., fluid heat lost or gained)

4. Environmental characteristic interface:
   a. Ambient temperature
   b. Atmospheric pressure
   c. Humidity
   d. Gaseous composition required
   e. Allowable foreign particle contents

Other data types may be needed. Appendix D shows an example of a supplied services interface for air-conditioning and cooling water.

2.4 Documenting Interfaces

Once an interface has been categorized and its initial contents defined, that interface definition must be recorded in a document that is technically approved by the parties (designer and manager) and the owners of both sides of the interface. The document then is approved by the next higher level in the project management structure and becomes the official control for interface design.

The program manager must ensure that compliance with the approved interface control document is mandatory. Each level of program management must ensure that the appropriate contractors and Government agencies comply with the documentation. Therefore, technical approval of the interface control document indicates that the designated approving organization is ready to invoke the interface control document contractually on the approving organization's contractor or supporting organization.

The interface categories can be grouped together in one document, or each category can be presented in a separate document (i.e., electrical ICD's, mechanical ICD's, etc). The format for interface control documents is flexible. In most cases a drawing format is the easiest to understand and is adaptable to the full range of interface data.

The specification format (ref. 6) can also be used. The use of this type of format enables simple changes through the removal and insertion of pages; however, the format is often difficult to use when presenting complex interface definitions that require drawings, and normally requires many more pages to convey the same level of information.

In either case there must be agreement on a standard for data presentation and interpretation. ANSI standard Y14.5 (ref. 7) can be used for dimensions, along with DOD–STD–100 (ref. 8), for general guidance of a drawing format. The specification format should use MIL–STD–490 (ref. 6) for paragraph numbering and general content.

Some large programs require large, detailed ICD's. Maintaining a large, overly detailed document among multiple parties may be more difficult than maintaining a number of smaller, more focused documents. Grouping small documents by major category of interface and common participants is one of the most effective and efficient strategies. It minimizes the number of parties involved and focuses the technical disciplines, greatly streamlining the decision process and permitting much shorter preparation time. However, interfaces can be multidisciplinary and separate documents can result in miscommunications.

2.5 Identifying Steady-State and Non-Steady-State Interfaces

Interfaces can vary from a single set that remains constant for the life of a program to a multiple set of documents that reconfigures during specific events in the life of a system. The first category would be used for an interplanetary probe. The interfaces of its instruments with the basic spacecraft structure would remain the same from assembly for launch throughout the life of the experiment. However, a continually evolving platform, such as a lunar base, would perhaps be controlled in a series of interface documents based on the assembly sequence of the base. An initial base would be established and later made more complex with additional structures and equipment delivered by subsequent lunar flights. Pressurized elements, logistic elements, power-generating sources, habitats, laboratories, and mining and manufacturing facilities might be added and reconfigured over time. Each configuration would require a set of interface control documents to ensure compatibility at the construction site as well as with the transportation medium from Earth to Moon. Interfaces that remained constant during this process might be termed steady state and require no further consideration once the interface was verified and delivered; whereas interfaces that would evolve from the initial configuration through multiple iterations would require multicoordination of interface parameters and schedules. The selection of interface categories should identify the steady-state or non-steady-state nature of interfaces as well as their initial designations (ref. 9).

2.6 Selecting a Custodian

Selecting an ICD custodian can depend on several factors (e.g., percentage of interface ownership, relative mission importance of interface sides, and relative investment of interface sides). However, it is generally most effective if the custodian

selected has an objective point of view. An example of this would be someone who is independent of either side of the interface (i.e., without any "vested interest" in the interface hardware or software). Objectivity permits unbiased control of the interface, involvement of the custodian as an objective mediator, and documentation of the interface on a noninterference basis with program/project internal design. Selecting an independent interface custodian should be the first step in establishing an interface control organization. A set of criteria should be used to select the custodian by weighting the content and interests of the interface with the needs of interface control. One set of criteria is as follows:

1. Integration center: Is one center accountable for integrating the interfaces controlled by this ICD? This criterion is considered the most important because the integration center will have the final responsibility for certifying flight readiness of the interfaces controlled in the ICD.
2. U.S. center: Is the participant a U.S. center? This criterion is considered the next most important because of agency experience and projected responsibility.
3. Flight hardware or software: Is the interfacing article flight hardware or software (as opposed to support hardware or software)? Flight hardware or software takes precedence.
4. Flight sequence: Does one side of the interfacing equipment fly on an earlier manifest than the other? An earlier flight sequence takes precedence over follow-on interfacing hardware.
5. Host or user: Is the interfacing article a facility (as opposed to the user of the facility)? Procedure in this criterion is guided by the relative priority of the interfacing articles.
6. Complexity: How complex is the interfacing equipment (relative to each side)? The more complex side of the interface normally takes precedence.
7. Behavior: How active is the interfacing equipment? The active side normally takes precedence over the passive side.
8. Partitions: How are the partitions (categories) used by the interfacing equipment? The relative importance of the partitions to the interface is acknowledged, and selection of the custodian is sensitive to the most important partition developers.

Scores are assigned to each piece of interfacing equipment for each criterion. These scores can be determined by many different methods. Discrete values can be assigned to the first four criteria. A score of 1.0 is assigned if the interfacing piece of equipment is unique in meeting the criterion, the other piece of equipment then receives a score of 0.0. Scores of 0.5 are assigned to both sides if both (or neither) of them meet the criterion. There is no definitive way of assigning scores to the last four criteria; however, verbal consensus or an unbiased survey can be used to assign scores. Also, the partition criteria can be scored by partition evaluation analysis (ref. 4).

2.7 Analyzing for Interface Compatibility

The interface definitions to be documented on the ICD's must be analyzed for compatibility before the ICD is authenticated. Appendix E provides guidance on how compatibility analyses may be performed. They vary in their complexity from a simple inspection of the interface definitions to complex mathematical analyses where many variables are involved.

Regardless of complexity, the compatibility analysis should be documented and maintained as backup information for the ICD. It can be used to expedite any changes to the interface definition by providing a ready means for evaluating the compatibility of the proposed change. The compatibility analysis also can be used to document how the interface definition was arrived at and why the definition is presented as it is on an ICD.

2.8 Verifying Design Compliance With Interface Control Requirement

The ICD can only fulfill its purpose if the contractors' detailed design drawings and construction practices adhere to the limits imposed by the ICD. Verifying compliance of the design as well as of the construction process is an integral part of interface control.

Each contractor should be assigned the responsibility of denoting on their manufacturing and inspection drawings or documents those features and characteristics that, if altered, would affect interfaces controlled by the ICD's. To ensure that all ICD requirements are covered, the contractor should select, at the highest assembly level at which the equipment is inspected, the features and characteristics to be denoted. Any design change affecting an ICD-controlled feature or characteristic would be clearly identified even at the assembly level (ref. 10).

Entries identified as "to be resolved" (TBR) can be bracketed or shaded to indicate preliminary interface information or an interface problem. This information is subject to further review and discussion and is an interim value for use in evaluating effects. Entries identified as "to be supplied" (TBS) represent data or requirements to be furnished. Appendix F shows a typical bracket system.

2.9 Verifying Contract-Deliverable Item

Each contract-deliverable item that is a mating side to an ICD interface should also be tested or measured to verify that the item complies with the requirement as specified in the ICD. The

responsibility for administering and reporting on this verification program could be assigned to the design agent, the contractor, or an independent third party. If feasible, an independent third party should be selected for objectivity.

The verification methods should include analysis, measurement and inspection, demonstration, and functional testing. The specific methods employed at each interface will depend on the type of feature and the production sequence. Compliance should be verified at the highest practical assembly level. To preclude fabrication beyond the point where verification can be performed, an integrated inspection, measurement, and demonstration test outline of both hardware and software should be developed. This verification test outline will provide a schedule, tied to production, that allows all interface requirements to be verified. The resultant data and inspection sheets should become part of the verification data in the history jacket retained by the contractor for NASA.

2.10 Training[2]

1. What is the purpose of interface control?
   A. To define interfaces
   B. To ensure compatibility between interrelated equipment
   C. To provide an authority to control interface design
   D. All of the above

2. How is an interface identified?
   A. By boundaries between functional areas
   B. By functional analyses of performance requirements
   C. By design features of a component that can affect the design features of another component

3a. How can interfaces be categorized?
   A. Mechanical, electrical, software, and services
   B. Electrical/functional, mechanical/physical, software, and supplied services
   C. Electrical, physical, software, and supplies

3b. What is one method of assigning an interface to one of the four basic categories?
   A. Functional flow block diagram
   B. Timeline analysis
   C. *N*-squared diagram

4a. How can an interface be documented?
   A. By drawing format
   B. By specification format
   C. By both of the above

4b. Who approves the interface control document?
   A. Designer or manager
   B. Owners of both sides
   C. Both of the above

4c. Who ensures compliance with the approved ICD?
   A. Designer or manager
   B. Owners of both sides
   C. Project manager

5a. What is a steady-state interface?
   A. A single set that remains constant for the life of the project
   B. A multiple-set suite that reconfigures during specific events in the life of the system

5b. Give an example of a steady-state interface.
   A. An interplanetary probe
   B. A lunar base

5c. What features make this a good example of a steady-state interface?
   A. The basic structure of the spacecraft would remain the same from assembly for launch throughout the life of the experiment.
   B. An initial base would be established and subsequently made more complex with additional structures and equipment delivered by subsequent lunar flights.

6a. How should an ICD custodian be selected?
   A. Percentage of ownership of the interface
   B. Relative investment of interface sides
   C. An objective point of view

6b. What criteria should be used to select a custodian?
   A. Integration or U.S. center, flight hardware or software, flight sequence, host or user, complexity, behavior, and partitions
   B. Integration hardware, sequence user, and partitions

6c. What scoring system can be used for these criteria?
   A. Zero to 1.0, verbal consensus, unbiased survey, and partition evaluation analysis
   B. One to 100, priority ranking, and voting

7a. What is the purpose of an ICD compatibility analysis?
   A. Demonstrates definitions and provides mathematical analysis
   B. Demonstrates completeness of an interface definition and provides a record that the interface has been examined and found to be compatible

[2]Answers are given at the end of this manual.

7b. What are the four categories that require interface analysis?
   A. Electrical/functional, mechanical/physical, supplied/services, and hydraulic/pneumatic
   B. Electrical/functional, mechanical/physical, software, and supplied services

7c. The hardware for mounting the satellite vehicle (SV) adapter to the Titan IV Centaur is shown in figures 2.1 to 2.3.
   A. Is there sufficient data to perform a compatibility analysis?
      i. Yes ii. No
   B. Can the Jet Propulsion Laboratory specify the SV adapter ring?
      i. Yes ii. No
   C. What items need to be bracketed?
      i. Shear pin material and SV attachment view
      ii. SV panel and view C–C

8a. What does a bracket on an ICD represent?
   A. Verification of design compliance
   B. An interface problem

8b. What interface deficiency rating does a bracket discrepancy have?
   A. S & MA impact A > 1 or understanding of risk B > 2
   B. S & MA impact A < 1 or understanding of risk B < 2

9a. How are mating sides of an ICD interface verified?
   A. Testing or measurement to meet requirements
   B. Analysis, measurement or inspection, demonstration, and functional testing

9b. What does the verification test outline provide?
   A. Schedule, tied to production, that allows interface requirements to be verified
   B. Process controls, tied to manufacturing, for meeting schedules

9c. Where is the resultant test and inspection data stored?
   A. Contractor files for use by an independent third party
   B. History jackets for use by NASA

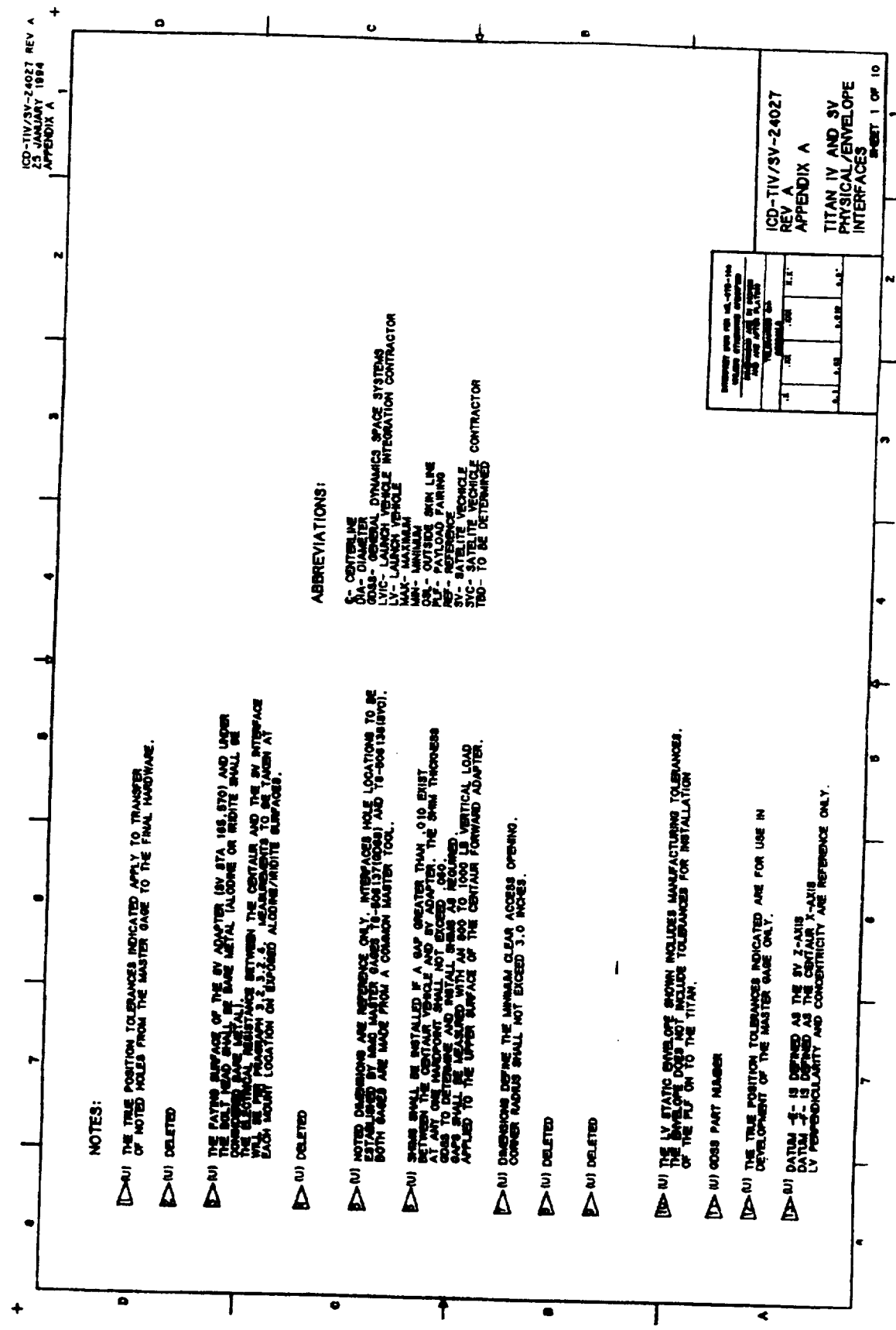


Figure 2.1.—Titan IV and satellite vehicle physical/envelope interfaces.

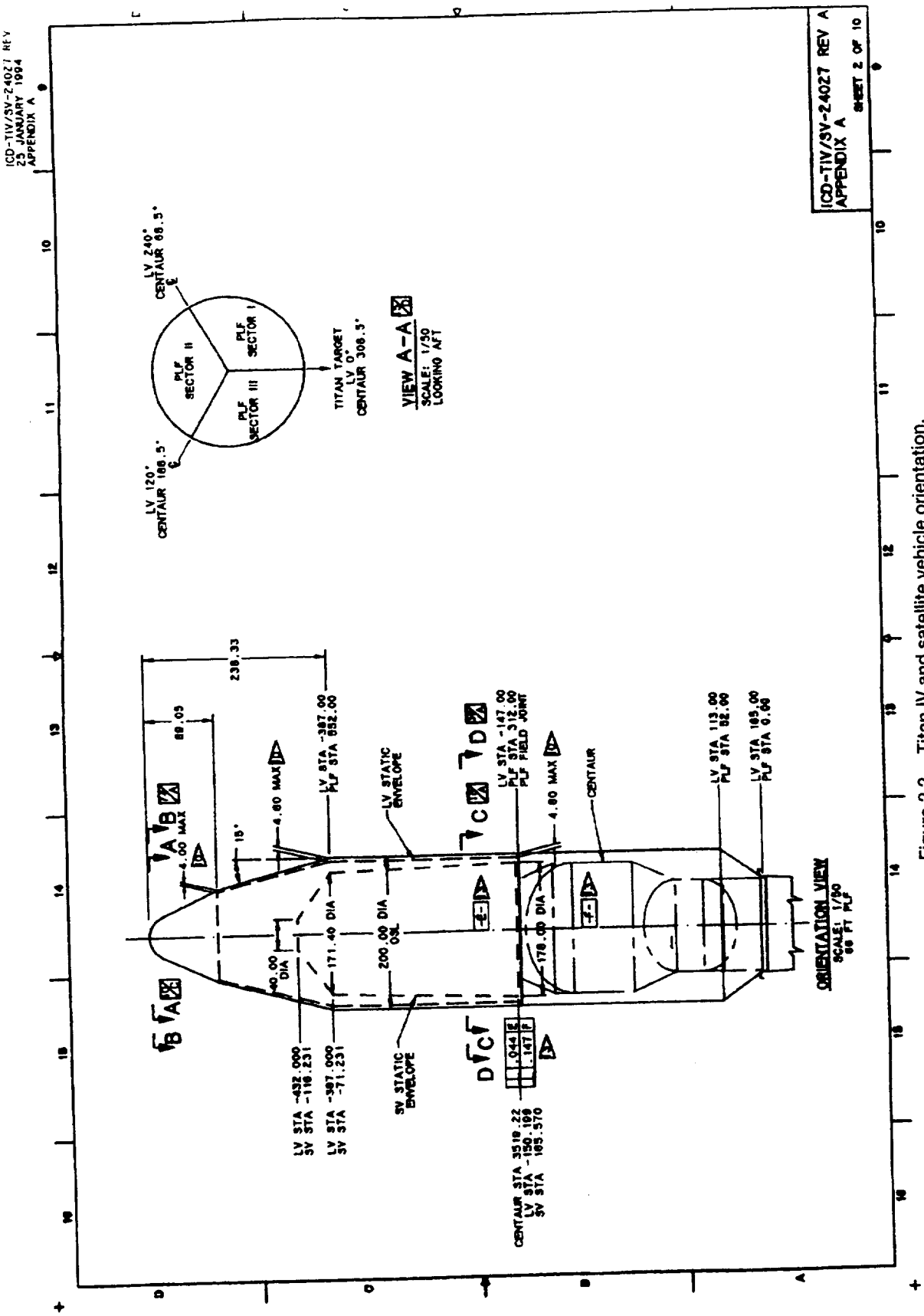


Figure 2.2.—Titan IV and satellite vehicle orientation.

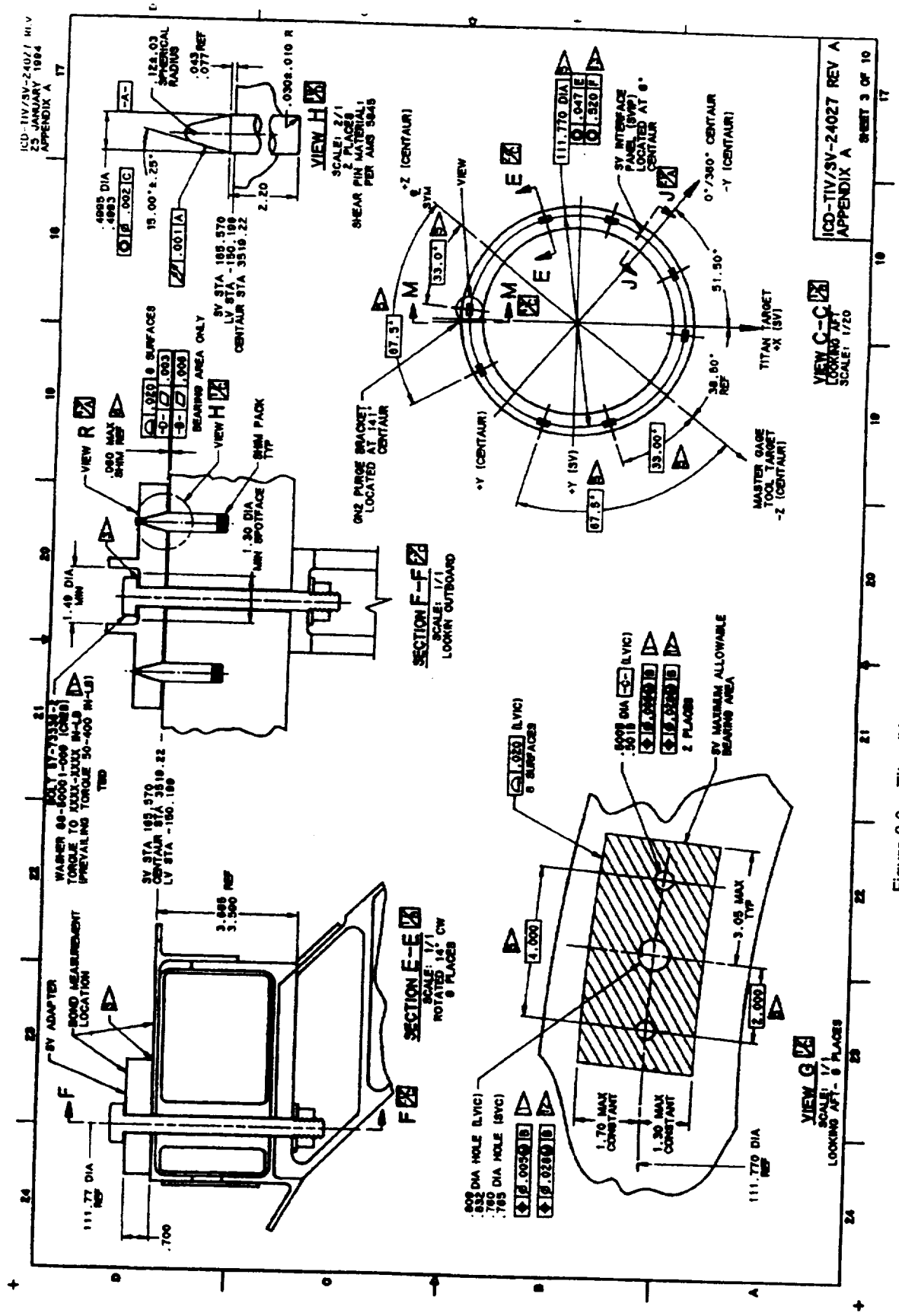


Figure 2.3.—Titan IV and satellite vehicle adapter ring.

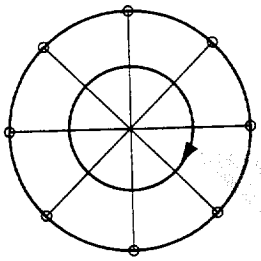
Chapter 3
The Process: Through the Design Phases

Interface control should be started when a program begins. This process eventually will define all interface design and documentation responsibilities throughout the life cycle of the program. Each program phase from concept development to project construction is directly related to the maturity level of interface control.

3.1 Program Phases

3.1.1 Concept Definition

During the system engineering concept definition phase (from fig. 1.1), basic functional areas of responsibility are assigned for the various pieces of equipment that will be employed by the project (electrical power, environment control, propulsion, etc.); see figure 3.1. At this point the design responsibilities of the responsible organization and related contractor (if chosen) should be defined to establish a set of tiered, traceable requirements. From these requirements the interfaces to be designed are identified by category (electrical/functional, mechanical/physical, software, and supplied services) and by type of data that must be defined. This categorization will include a detailed review of each requirement to determine which requirements or features will be controlled by the interface control process. (What is important for this item to fulfill its intended function? On what *interfacing* equipment is this function dependent?) Once the interfaces to be controlled are selected, the formal procedures for interface control need to be established. These procedures include identifying the participants responsible for the interface control documentation, the approval or signoff loop for documentation, and the degree to which all participants have to adhere to interface control parameters and establishing a missing design data matrix, change procedures, etc. (See section 3.2.)



Concept definition

- Assign basic functional areas of responsibility.
- Define design responsibilities.
- Categorize interfaces.
- Define interfaces to be controlled.
- Establish formal interface control procedures.
- Disseminate scheme, framework, traceability.

Figure 3.1.—Establishment of interface control process during concept definition.

Early development of the interface process, products, and participants provides a firm foundation for the design engineer to use the correct information in designing his or her portion of an interface. It minimizes the amount of paper to be reviewed, shortens the schedule, and concentrates the efforts of the designer on his or her area of responsibility.

Initial selection of interfaces generally begins with listing of all pieces of equipment in a system and then identifying the extent of interrelation among them. One tool used to help in this process is the *N*-squared diagram. Details of this process can be found in reference 4. The *N*-squared diagram was initially used for software data interfacing; however, some centers are using it for hardware interfaces. If the diagram is not polarized initially (input/output characteristics not labeled), it is a convenient format for identifying equipment interfaces and for categorizing them. An example of this form is shown in figure 3.2. This diagram can be further stratified to identify the interfaces for each of the categories; however, detailed stratification is best applied to electrical/functional, software, and supplied services interfaces. Using the *N*-squared diagram permits an orderly identification and categorization of interfaces that can be easily shown graphically and managed by computer.

By the end of this phase the basic responsibilities and management scheme, the framework for the interface control documentation, and the process for tracking missing interface design data (see section 3.2.2) should be established and disseminated.

3.1.2 Requirements Definition

During the requirements definition phase (fig. 3.3; from fig. 1.1), the definitions of the mission objectives are completed so that each subsystem design can progress to development. Here, the technology to be used in the project will be defined to limit the risk associated with the use of new, potentially unproven technologies. Defining requirements and baselining interface documents early in the design process provides information to the designer needed to ensure that interface design is done correctly the first time. Such proactive attention to interfaces will decrease review time, reduce unnecessary paperwork, and shorten schedule times. By the end of requirements definition all interface control documents should be prepared, interfaces defined to the most detailed extent possible, and ICD's presented for baselining.

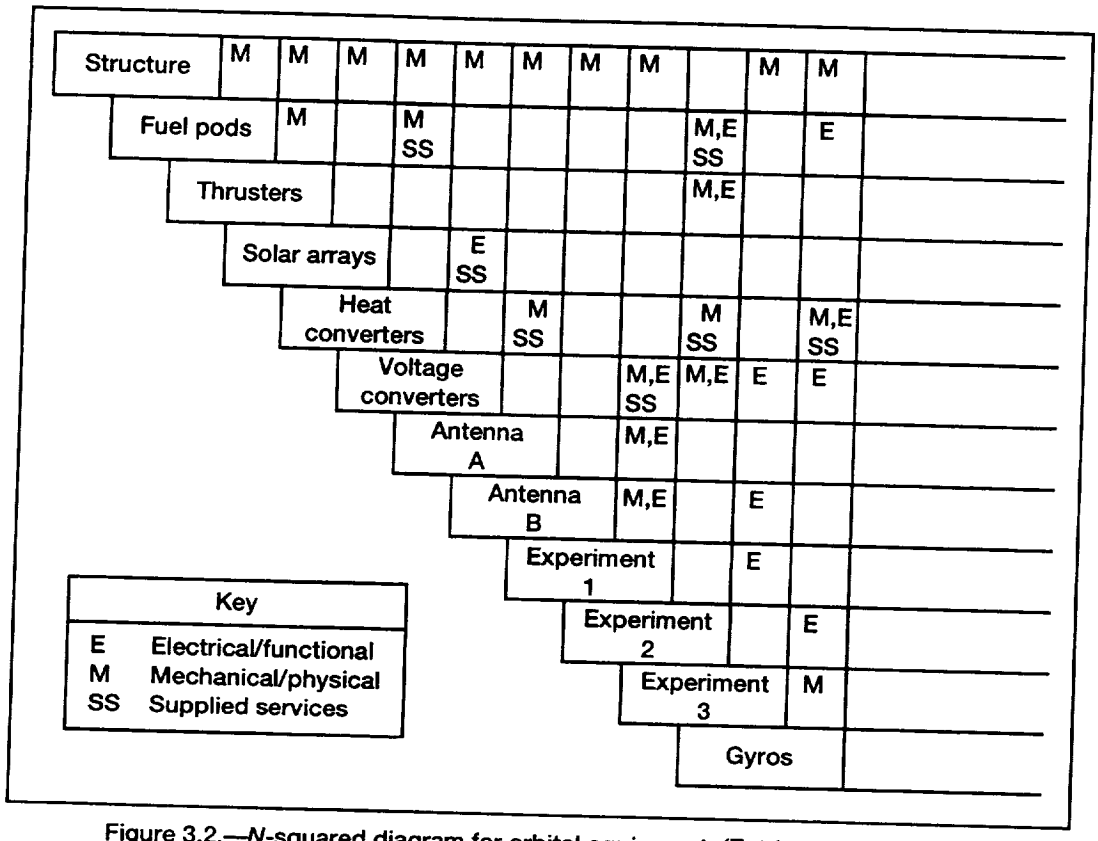
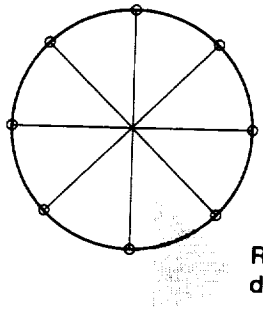


Figure 3.2.—*N*-squared diagram for orbital equipment. (Entries not polarized.)



Requirements definition

- Define technologies to be used.
- Define and categorize all interfaces.
- Prepare all interface control documents.
- Identify all voids and assign both responsibilities and due dates.
- Bound voids when possible.
- Baseline interface documents.

Figure 3.3.—Development and control of interfaces during requirements definition.

Baselining is the act by which the program manager or designated authority signs an ICD. That signature establishes the ICD as an official document defining interface design requirements. The term "baselining" is used to convey that the ICD is *the only* official definition and that this officiality comes from the technical management level. Not only is the initial version of the ICD baselined, but each subsequent change or update to an ICD is also baselined.

The baselined version of the ICD will identify (by a "void") any missing design data that cannot be included at that time. Agreed-to due dates will be noted on the ICD for each data element required. Each void will define the data required and specify when and by whom such data will be supplied. Where possible, the data to be supplied should be bounded initially on the ICD. These bounds will be replaced by detailed data when the void is filled. The initial bounds give the data user (the other side of the interface) a range that can be used without risk, until the detailed data are supplied. Establishing these voids on ICD's provides a means of ensuring that interface design data are supplied when they are required by the data user. Yet it allows design freedom to the data supplier until the data are needed. A recommended form for use in identifying the data needed is shown in figure 3.4. The criteria for choosing due dates are discussed in section 3.2.

Interface Design Data Required (IDDR)

(Drawing/document number + Void number)

Data required: Brief description of information needed to define interface element currently lacking details

Data supplier: (Project center/code/contractor)

Data user(s): (Project center/code/contractor)

Date due: (Date design data are needed, either actual date or a period of time related to a specific milestone.

Figure 3.4.—Format for interface design data required (IDDR).

Interface Design Data Required (IDDR)
_____ Program Status Report

Drawing/doc # ____ IDDR #	Sheet/page _____ /Zone	Short title _____ Data required	Supplier(s) _____ Center/code/ contractor	User(s) _____ Center/code/ contractor	Due date _____ Yr/Mo/Day	Remarks

Figure 3.5.—Format for monthly report on IDDR status.

Documents should be baselined as early as possible, as soon as the drawings contain 10% of the needed information. The significance of early baselining is that both sides of the interface have the latest, most complete, official, single package of information pertaining to the design of the interface.

The package includes all agreed-to design data plus a list of all data needed, its current level of maturity, and when it is to be supplied by whom to whom.

Technical information voids in interface documents must be accounted for and tracked. Otherwise, there is no assurance that the needed information is being provided in time to keep the design on schedule. The status of these voids must be reported, and the owners of the interface-design-data-required forms (IDDR's) must be held responsible for providing the needed information. It is recommended that the status be reported monthly to all parties having responsibility for the interfaces.

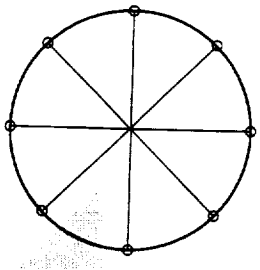
A consolidated report is the most efficient, consumes the least paper and mail services, and allows the program manager to track areas important to the integration of the system components. The basic form shown in figure 3.5 is recommended for reporting and tracking IDDR's.

3.1.3 Systems Integration

The interface control program continues to be active during the systems integration phase (fig. 3.6; from fig. 1.1). Design changes that establish a need for a new interface will follow the interface control change procedures as defined in section 3.2.

Proposed design changes that affect existing interfaces are not given final approval until all participants' and the cognizant center's baselinings have been received through the ICD change notice system.

During the various design reviews that occur in the full-scale engineering development phase, special attention should be given to design parameters that if altered, would affect interfaces controlled by the ICD. It is strongly recommended that each design activity denote, on design and manufacturing documentation at the preliminary design review, through a bracket or some highlighting system, those features and characteristics that would affect an interface (see section 2.8). At the critical design review all voids should be resolved and all detailed design drawings should comply with interface control documentation (see section 2.9).



Systems integration

- Manage and satisfy voids.
- Invoke use of brackets on design drawings.
- Ensure resolution of voids by the time of critical design review.
- Verify compliance of design documentation with ICD's.

Figure 3.6.—Development and control of interfaces during systems integration.

3.2 Preparing and Administering Interface Control Document

3.2.1 Selecting Type of Interface Control Document

A drawing, a specification, or some combination format should be selected for the ICD on a case-by-case basis. The drawing format generally is preferred when the ICD has features related to physical dimensions and shapes. The specification format is preferred when the ICD needs tables and text to describe system performance. Combinations are used when both dimensions and tables are needed. Members of the coordinating activity responsible for preparing the ICD determine the format, which is approved by the appropriate project authority. Examples of drawing formats are given in appendixes A and B.

The *level of detail* shown on the ICD varies according to the type and degree of design dependency at the interface being controlled. The ICD should clearly identify and control interfaces between designs and enable compatibility to be demonstrated between the design areas. The key to a useful ICD is limiting the detail shown to what is required to provide compatibility. Any unnecessary detail becomes burdensome and may confuse the contractors responsible for designing the mating interface. *Again, the ICD should, at a minimum define and illustrate physical and functional interface characteristics in sufficient detail that compatibility, under worst-case tolerances, can be determined from the ICD alone; or it should reference applicable revisions of detailed design drawings or documents that define and bracket or identify features, characteristics, dimensions, etc., under worst-case tolerances, such that compatibility can be determined from the bracketed features alone.*

3.2.2 Tracking and Resolving Missing Interface Design Data

Missing interface data should be identified on the ICD, and the ICD should control the date for its submission. The notation identifying the missing data should indicate the specific data required, how the data are being tracked for resolution, when the data are needed by the interfacing design agent, and by what date the required data will be supplied. Establishing data-required notations (or voids) on ICD's helps ensure that interface design data will be supplied when needed; yet it allows design freedom to the data supplier until the due date. Every attempt should be made to establish realistic due dates and to meet that schedule unless there is a valid and urgent need to change a due date.

These criteria and procedures should be followed in establishing, reporting, and managing data due dates:

1. Choose the due date as the date when the data user will start to be affected if agreed-upon or baselined data have not been received.

2. When establishing a due date, allow time to process and authenticate a change notice to the ICD (i.e., once the due date has been established, include a period of time to establish that due date for the data supplier).

3. The custodian responsible for the ICD should periodically, as determined by the appropriate project authority, prepare and distribute a report on the status of all missing design information for all project activities. The report should contain the following information:

   a. Identification of the data element needed, consisting of the ICD number, the date, and a two- or three-digit number that provides a unique identifier for the data element
   b. A short title for the ICD
   c. The activity that requires the data
   d. The date when the missing data are to be supplied or the period of time after the completion of a program event or milestone when the data are to be supplied
   e. The activity from which the data are due
   f. The status of the data required (i.e., late data, data in preparation, or change notice number)
   g. A description of the data required

3.3 Initial Issuance of ICD

The first issue of an ICD should be a comment issue. The comment issue is distributed to participating centers and contractors for review and comment as designated in the interface responsibility matrix (fig. 3.7).

The interface custodian generates the responsibility matrix for ICD's. The matrix specifies the center and contractor responsibilities for baselining, review and comment, and technical approval. The matrix lists all ICD's applicable to a particular program. Distribution of the ICD's can then be controlled through this matrix as well.

The review and comment process is iterative and leads to agreement on system interface definitions and eventual approval and baselining of the ICD. See figure 3.8 for a flow diagram of the issuance, review and comment, and baselining procedures for ICD's. Concurrent distribution of the comment issue to all participants minimizes the time needed for review and subsequent resolution of differences of opinion.

3.4 Document Review and Comment

As designated in the ICD responsibility matrix, all centers and contractors should submit technical comments through the appropriate authority to all other activities with review and comment responsibilities for the particular ICD and to the ICD custodian.

Technical comments by all activities should be transmitted to the custodian as soon as possible but not later than 30 working days[4] from receipt of the comment issue. If the comment issue is technically unacceptable to the Government authority or the interfacing contractor, the rationale for unacceptability should be explained, including technical and cost effects if the interface definition is pursued as presented.

3.4.1 Resolving Comments

The ICD custodian collects review comments and works in conjunction with project management for comment resolution until approval is attained, the comment is withdrawn, or the ICD is cancelled. Information on comments and their disposition and associated resolution should be documented and transmitted to all participants after all comments have been received and dispositioned. Allow two weeks[4] for participants to respond to the proposed resolution. Nonresponses can be considered concurrence with the resolutions if proper prenotification is given to all participants and is made part of the review and comment policy.

When comments on the initial comment issue require major changes and resolution is not achieved through informal communications, an additional comment issue may be required and/or interface control working group (ICWG) meetings may need to be arranged.

3.4.2 Interface Control Working Group

The ICWG is the forum for discussing interface issues. ICWG meetings serve two primary purposes: to ensure effective, detailed definition of interfaces by all cognizant parties, and to expedite baselining of initial ICD's and subsequent drawing changes by encouraging resolution of interface issues in prebaselining meetings. A major goal of interface control should be that baselining immediately follow a prebaselining ICWG meeting.

All ICWG meetings must be convened and chaired by the cognizant project organization. The project can choose a contractor to act as the chair of an ICWG when Government commitments are not required. In all cases the ICWG members must be empowered to commit the Government or contractor to specific interface actions and/or agreements. In cases where a contractor is ICWG chair, the contractor must report to the Government any interface problems or issues that surface during an ICWG meeting.

[4]The times assigned for commenting activities to respond are arbitrary and should be assigned on the basis of the schedule needs of the individual programs.

(1) Major project applicability			NASA data interface list Program __ (2) _____	(4) Review, comment, information responsibilities										Technical approval authority			AU	Notes
				(5)		(5)		(5)		(5)		(5)						
				(6)	(6)	(6)	(6)	(6)	(6)	(6)	(6)	(6)	(6)	(7)	(7)	(7)	(8)	
			(3) Interface category:															
			(9)															

Key:
(1) Major project/program applicability (e.g., NSTS).
(2) Primary project/program ownership of interfaces.
(3) Category of interface covered by list (e.g., electrical/functional).
(4) Review, comment, and information (RC&I) responsibilities by NASA center (listed in (5)).
(5) Center responsible for RC&I of the interface document.
(6) Center code or contractor doing RC&I.
(7) Center having technical approval signature authority (one column for each applicable center).
(8) NASA organization having authentication responsibility (single organization for authentication of a baselined document following technical approval by the organizations in columns marked by (7)).
(9) Drawing/document number and short title.

Figure 3.7.—Interface responsibility matrix.

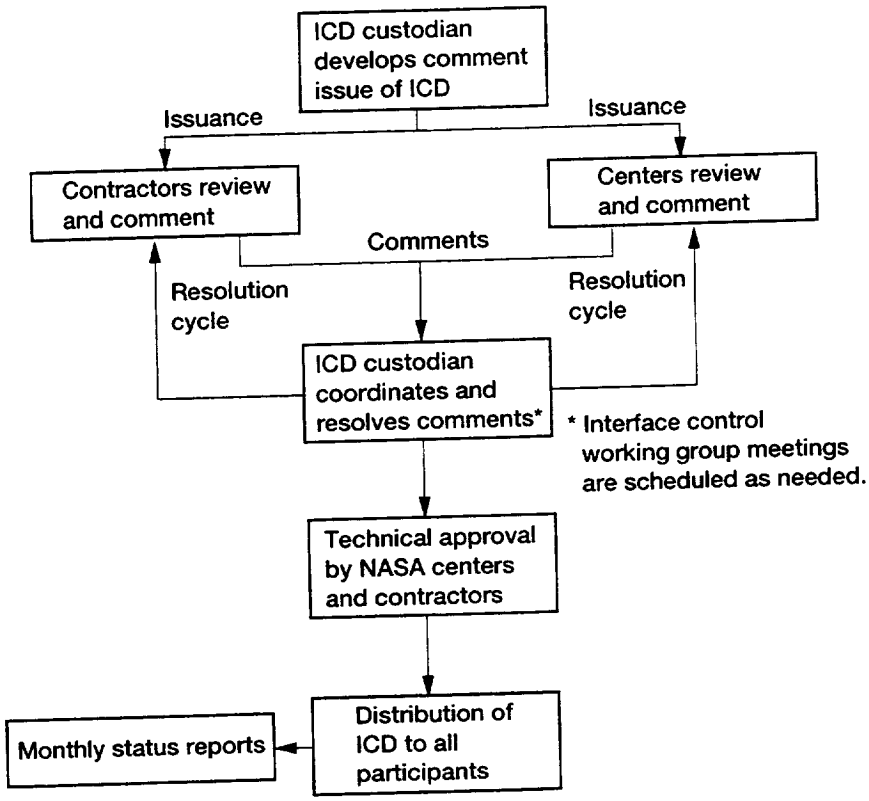


Figure 3.8.—Flow of interface control document production.

The ICWG chair prepares the ICWG meeting minutes or designates one of the meeting participants for this task. The minutes should include discussions of problems, agreements reached, decisions made, and action items. The ICWG chair also ensures that any updated interface control documentation reflecting the ICWG discussions is distributed within the timeframe agreed to by the affected participants.

3.4.3 Approval/Signoff Cycle

The management plan for the project assigns responsibility for each piece of equipment to a specific project authority and its contractor. The signoff loop for each ICD reflects this plan and can be related to the project and the origin of each design requirement. For each ICD, then, the signoff loop follows the sequence of technical approval by the contractors first and then by the appropriate project authority.

3.4.4 Technical Approval

The appropriate project authority and the primary and associate organizations with an interest in a particular ICD are listed in the responsibility matrix. They each sign the ICD to signify technical agreement and a readiness to contractually invoke its requirements.

3.4.5 Baselining

Interface control documents are baselined when the owners of both sides of the interface at the next level up in the program structure come to technical agreement and sign the document.

3.5 Change Notices

The procedure for initiation, review, technical approval, baselining, and distribution of changes to project ICD's (fig. 3.9) should conform to the following guidelines.

3.5.1 Initiating Changes

Any project activity should request a change to an ICD when

1. Data are available to fill a void.
2. Information contained in a data-required note needs to be modified.
3. Additional data are needed (i.e., a new data requirement has been established).
4. A technical error is discovered on the ICD.

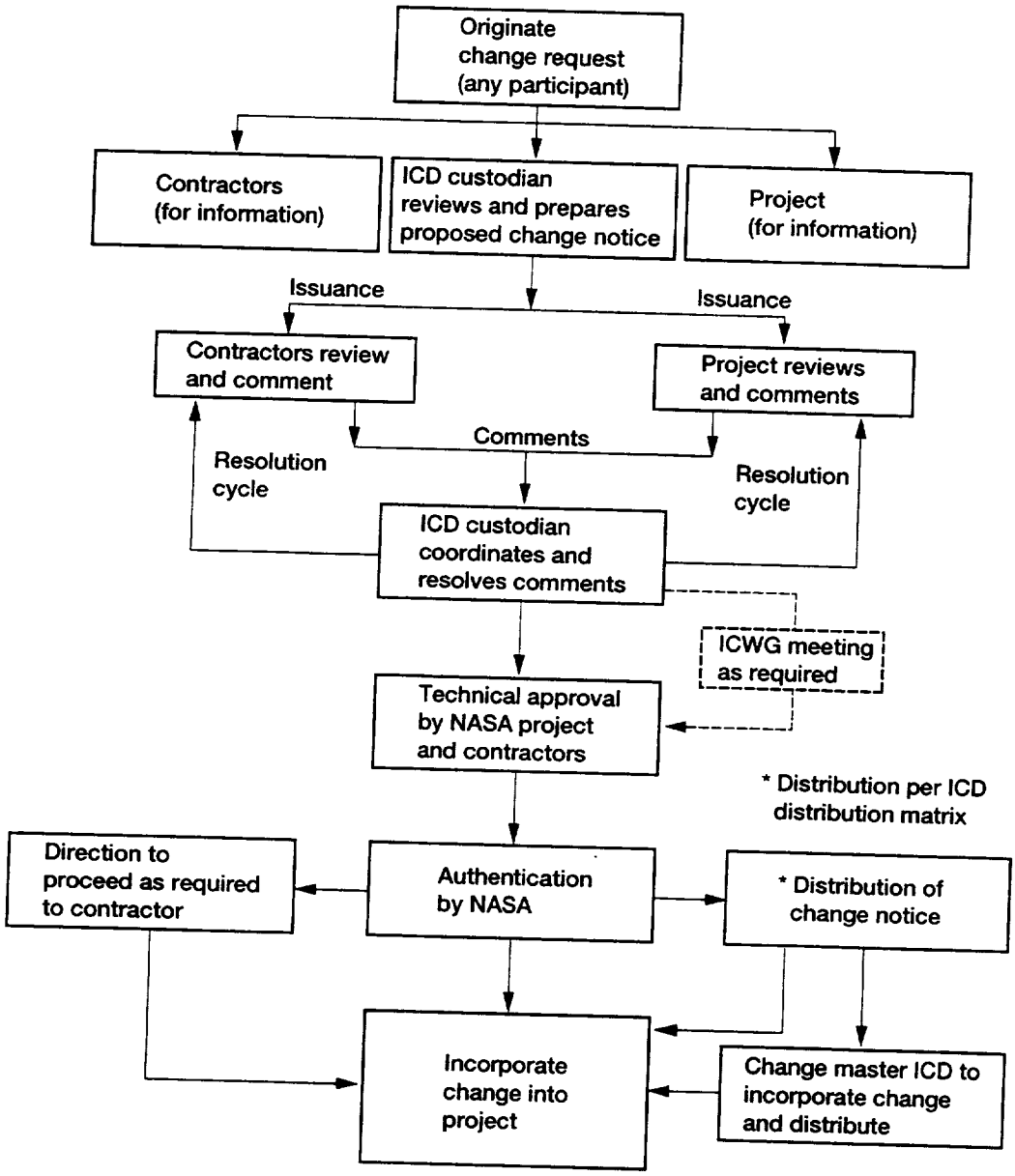


Figure 3.9.—Development and flow of change notices in the ICD revision process.

5. An equipment design change and a system or equipment rearrangement are proposed to improve performance, reduce cost, or expedite scheduled deliveries that would require changes to an interface or creation of new interfaces.

3.5.2 Requesting Changes

All requests for changes should be submitted to the organization responsible for maintaining the ICD, with copies to all activities that will review the resultant change notices and to the appropriate project authority. If baselining is needed in less than 30 days, a *critical* change should be requested. All requests for changes should be submitted in a standard format that includes the following items:

1. Originator's identification number—It is used as a reference in communications regarding the request and should appear on resulting change notices
2. Originating activity—originating project and code or originating contractor
3. Point of contact—name, area code, telephone number, facsimile number, and e-mail address of the person at the originating activity to be contacted regarding the request
4. Document affected—number, revision letter, and short title of each ICD that would be affected by the change
5. Number of data voids (if applicable)—number of data requirements for which data are being provided
6. Urgency—indication of whether this change is critical or routine (project decides whether to use critical route)
7. Detailed description of change—a graphic or textual description of the change in sufficient detail to permit a clear portrayal and evaluation of the request. Separate descriptions should be provided when more than one ICD is affected.
8. Justification—concise, comprehensive description of the need and benefit from the change
9. Impact—concise, comprehensive description of the effect in terms of required redesign, testing, approximate cost, and schedule effects if the requested change is not approved; also the latest date on which approval can occur and not affect cost or schedule
10. Authorizing signature of the organization requiring the change

Upon receipt of a change request to an ICD, the ICD custodian coordinates the issuance of a proposed change notice. First, the ICD custodian evaluates the technical effect of the proposed change on the operation of the system and mating subsystem. If the effect of the change is justified, the ICD custodian generates and issues a change notice. If the justification does not reflect the significance of the change, the ICD custodian rejects the request, giving the reason or asking for further justification from the originating organization. The ICD custodian evaluates an acceptable change request to determine whether it provides data adequate to generate a change notice.

The proposed change notice describes the specific changes (technical or otherwise) to the ICD in detail by "from-to" delineations and the reasons for the changes, as well as who requested the changes and how the change request was transmitted (i.e., by letter, facsimile, ICWG action item, etc.).

3.5.3 Proposed Change Notice Review and Comment Cycle

The review and comment cycle for proposed changes to ICD's should follow the same system as that used for the initial issuance of the ICD (see sections 3.3 and 3.4).

3.5.4 Processing Approved Changes

The baselined change notice should be distributed to all cognizant contractors and project parties expeditiously to promulgate the revised interface definition. The master ICD is revised in accordance with the change notice, and copies of the revised sheets of the ICD are distributed (see sections 3.3 and 3.4). Approval of the change by the project constitutes authority for the cognizant organization to implement the related changes on the detailed design.

3.5.5 Distributing Approved Changes

The custodian distributes the baselined change notice to all cognizant centers and contractors to expeditiously promulgate the revised interface definition. The master ICD is then revised in accordance with the change notice, and copies of the revised ICD sheets are distributed as was the change notice. The responsibility matrix (fig. 3.7) can be used to identify the distribution of change notices as it was used for the distribution of the ICD's.

3.5.6 Configuration Control Board

During development the project's configuration control board is responsible for reviewing and issuing changes to the configuration baseline. The board reviews all class I engineering change proposals to determine if a change is needed and to evaluate the total effect of the change. The configuration control board typically consists of a representative from the chairman, the project management office, customers, engineering, safety assurance, configuration management (secretary), fabrication, and others as required.

Changes to configuration items can only be effected by the duly constituted configuration control board. The board first defines a baseline comprising the specifications that govern development of the configuration item design. Proposed changes to this design are classified as either class I or class II changes. Class I changes affect form, fit, or function. However, other factors, such as cost or schedule, can cause a class I change. Class I changes must be approved by the project before being implemented by the contractor.

All other changes are class II changes. Examples of class II changes are editorial changes in documentation or hardware changes (such as material substitution) that do not qualify as class I changes. Project concurrence, generally, is required for the contractor to implement class II changes. Government plant representatives (Defense Contracts Administration Services (DCAS), Navy Programs Resident Office (NAVPRO), and Air Force Programs Resident Office (AFPRO) usually accomplish these tasks.

3.5.7 Closing the Loop

A wide range of methods are available for verifying by test that the design meets the technical requirements. During the definition phase analysis may be the only way of assessing what is largely a paper design. Typical methods are testing by similarity, analysis, modeling, and use of flight-proven components; forecasting; and comparison, mathematical modeling, simulation modeling, and using flight-proven experience and decisions. The actual methods to be used are determined by the project office. Each method has associated costs, requires development time, and provides a specific level of performance verification. The Government and industry managers must carefully trade off program needs for performance verification with the related costs.

If any demonstrated or forecast parameter falls outside the planned tolerance band, corrective action plans are prepared by the contractor and reviewed by the Government project office. Each deviation is analyzed to determine its cause and to assess the effect on higher level parameters, interface requirements, and system cost effectiveness. Alternative recovery plans are developed showing fully explored cost, schedule, and technical performance implications. Where performance exceeds requirements, opportunities for reallocation of requirements and resources are assessed.

Although functional and performance requirements are contained in the appropriate configuration item specification, the definition, control, and verification of interface compatibility must be handled separately. Otherwise, the volume of detail will overwhelm both the designers and managers responsible for meeting the functional and performance requirements of the system. Early establishment of the interface definition and control process will provide extensive savings in schedule, manpower, money, and paper. This process will convey precise, timely information to the interface designers as to what the designer of the opposing side is committed to provide or needs and will subsequently identify the requirements for verifying compliance.

Whether the interface is defined in a drawing format or in a narrative format is at the discretion of the program. What is of primary importance is that only the information necessary to define and control the interface should be on these contractural documents to focus the technical users and minimize the need for updating information.

Appendix G provides seven ICD guidelines that have been used by many successful flight projects and programs to provide such a focus on the interface definition and control process.

3.6 Training[2]

1a. When should the ICD process be started?
A. Concept definition B. Requirements definition
C. Systems integration

1b. What are the benefits of early development of the ICD process?
A. Assigns basic areas of responsibility
B. Provides firm foundation for design, minimizes paper, shortens schedule, and concentrates efforts

1c. What tool can be used to list equipment and identify their interrelations in a system?
A. Prechart B. *N*-squared diagram

2a. What should be done in the ICD process during requirements definition?
A. Define mission objectives
B. Define technology and interfaces and present for baselining

2b. What is baselining?
A. The designated authority signing an ICD
B. The only official definition

2c. How are voids in an ICD accounted for and tracked?
A. Procedure or administration report
B. Monthly program status report on interface design data required

3a. What should be done in the ICD process during development?
A. Manage voids, invoke brackets, resolve voids, and verify compliance
B. Control interface developments

3b. How should proposed design changes be handled?
A. Discussed at critical design review
B. Discussed and approved by all participants

3c. What should be given special attention?
A. Design parameters that affect controlled ICD
B. Manufacturing documentation

[2]Answers are given at the end of this manual.

4a. When is the drawing format used for an ICD?
   A. To describe the type and nature of the component
   B. To describe physical dimensions and shapes

4b. When should a specification be used?
   A. To describe performance with tables and text
   B. To describe a software function

4c. What is the key to providing a useful ICD?
   A. Give as much detail as possible
   B. Limit the detail to what is necessary to demonstrate compatibility

5a. What is the purpose of the initial issue of an ICD?
   A. Issuance, review, comment, and baselining
   B. Review and resolution of differences of opinion

5b. Who is responsible for controlling the flow of an ICD?
   A. Contractor
   B. Custodian

6a. Who should review ICD's?
   A. Organizations designated in the responsibility matrix
   B. ICD custodian

6b. How are comments resolved?
   A. By the project office
   B. By project management and custodian working for resolution and approval or the comment being withdrawn

6c. Where are interface issues discussed?
   A. Project office
   B. Interface control working group

6d. Who approves and baselines an ICD?
   A. Projects at the next level up in program structure
   B. The project office

7a. When should a project activity request a change to an ICD?
   A. At the custodian's request
   B. When data are available, requirements need change, an error is discovered, or the design changes

7b. What items should be included in a change notice request?
   A. Identification number, activity, contact, document affected, number of data voids, urgency, description, justification, impact, and authorizing signature
   B. Those established by the ICWG

7c. Who evaluates and issues a proposed change notice?
   A. ICD custodian
   B. Project office

7d. What does a proposed change notice describe?
   A. Specific changes (from-to), reasons, and the requestor
   B. Project notices

7e. How is a change notice approved and distributed?
   A. By the project authority to all cognizant parties
   B. By all cognizant parties to the contractors

National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio, 44135, July 1995.

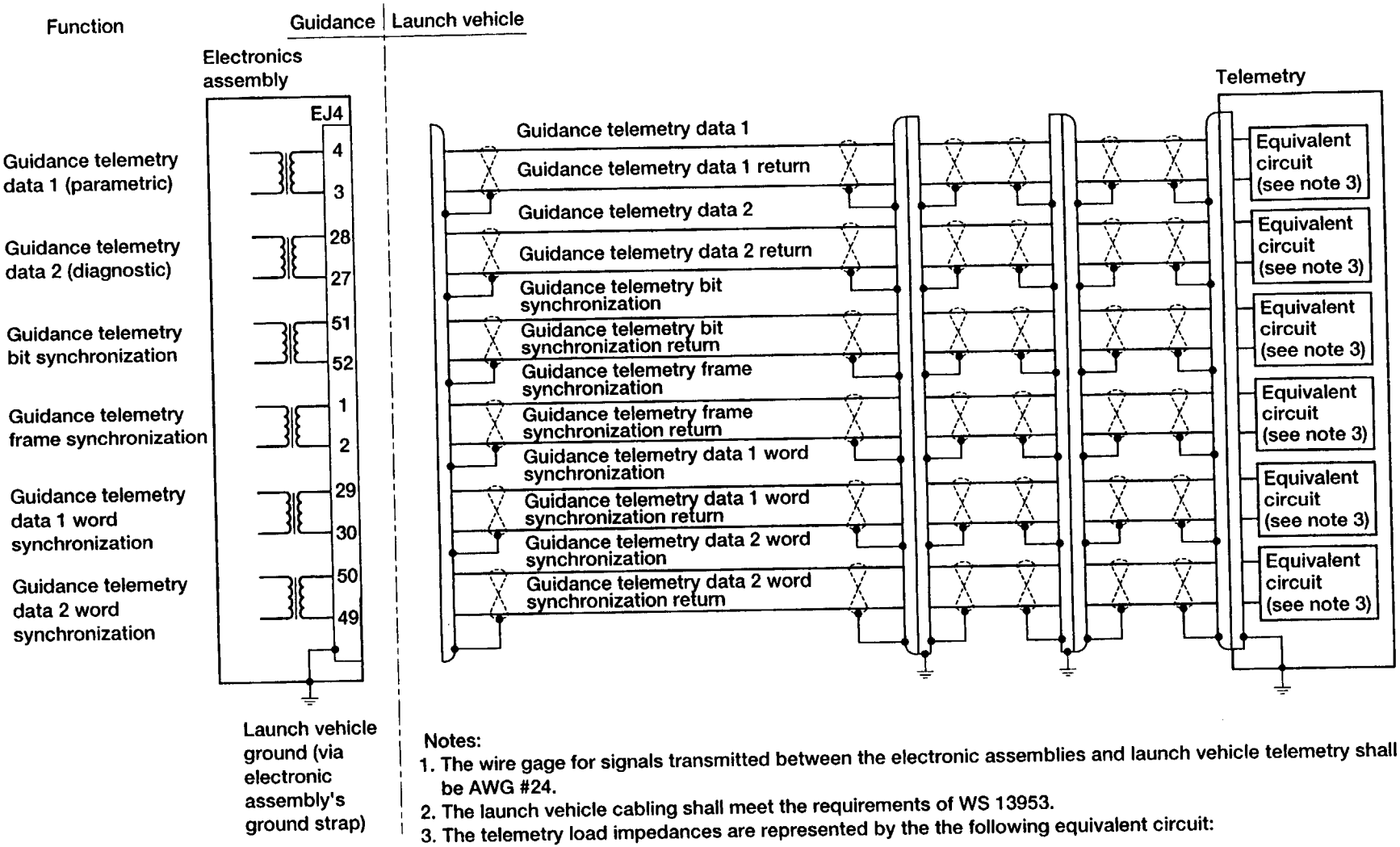
Appendix A
Electrical/Functional Interface Example

This appendix illustrates elements of a telemetry drawing interface control document showing control of waveform parameters and data rates. This interface example depicts data transfer between a guidance system electronics assembly and a launch vehicle telemetry system. The basic drawing (fig. A.1) covers the isolation elements of the guidance system, the jack and pins assigned, and shielding and grounding on the guidance side of the interface. Bus functions are named (e.g., guidance telemetry data 1(parametric)), and the shielding requirements through to the first isolating elements of the telemetry system are provided (see notes on fig. A.1).

Table A.1 contains the details to be controlled for each bus function. Signal source (electronics assembly) and destination (telemetry system) are identified. The waveform (fig. A.2) and its critical characteristics (table A.2) are provided, as well as data rates and sources and load impedances. Telemetry load impedance is further described by an equivalent circuit (see note 3 on fig. A.1).

The final value of pulse minimum amplitude is missing in this example. This is noted by the design-data-required (DDR) callout in table A.2 and the accompanying DDR block (fig. A.3). The DDR block notes that the responsible parties have agreed on an amplitude band with which they can work until the guidance design becomes firm. However, there is also a date called out that indicates when (45 days after preliminary design review) the telemetry contractor must have the data to be able to complete design and development and deliver the telemetry in time to support launch vehicle flight.

The parameters called out in this example are *only* those needed to control the design of either side of the interface through the first isolating element. Also note that only the shielding and wire gage of the launch vehicle cabling between the two systems are provided. Only pin numbers for the guidance side of the interface are called out and controlled. Connector types and other pertinent cable specifications are as per a referenced standard that applies to all launch vehicle cabling. In this case the same pulse characteristics apply to each of the functions covered; however, table A.2 is structured to permit variation for each function if the design should dictate different values for the characteristics of each function.



Notes:

1. The wire gage for signals transmitted between the electronic assemblies and launch vehicle telemetry shall be AWG #24.
2. The launch vehicle cabling shall meet the requirements of WS 13953.
3. The telemetry load impedances are represented by the the following equivalent circuit:

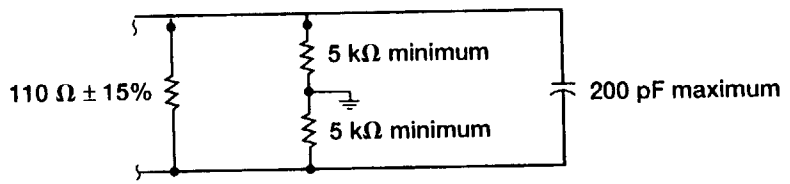
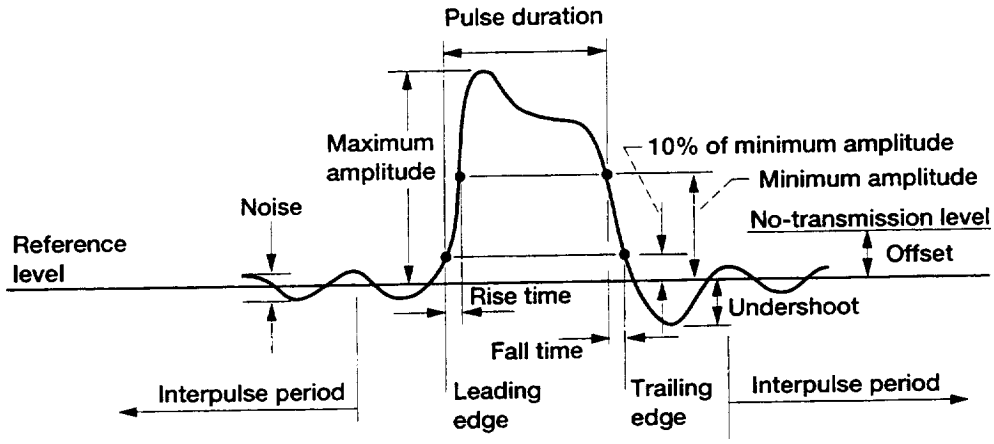


Figure A.1.—Guidance/launch vehicle telemetry interface.

Table A.1.—GUIDANCE/LAUNCH VEHICLE TELEMETRY DATA TRANSFER INTERFACE PARAMETERS

[Source, electronic assembly; destination, telemetry; waveform, see fig. A.2; source impedance, 50 Ω maximum during pulse time.]

Function	Data rate	Load impedance	Remarks
Guidance telemetry data 1	Bit rate: 979.2 kilobits per second Word rate: constrained to 153- x 16-bit words per 10-ms frame	110 Ω ± 15% balanced to ground by not less than 5000 Ω and shunted by 200 pF maximum (excluding cabling) (see note 3 on fig. A.1(c))	1. All timing is synchronous with the guidance to telemetry bit synchronization. 2. The guidance telemetry frame synchronization consists of 16 pulses at a rate of (void #3) and occurs at 100 frames per second. 3. The launch vehicle telemetry shall function properly with a guidance telemetry bit synchronization and frame synchronization tolerance of ± 0.1%. 4. All pulse characteristics are defined at the guidance electronic assembly interface connectors. 5. Guidance signal characteristics are defined for the minimum wire gauge shown.
Guidance telemetry data 2			
Guidance telemetry bit synchronization	Void #3		
Guidance telemetry frame synchronization	100 frames per second (see Remarks)		
Guidance telemetry data 1 word synchronization	Coincident with first bit of each guidance telemetry data 1 word		
Guidance telemetry data 2 word synchronization	Coincident with first bit of each guidance telemetry data 2 word		



Notes:

1. The interpulse period shall be the period from 150 ns after the trailing edge of a pulse until 100 ns prior to the leading edge of the subsequent pulse.
2. The reference level shall be the average voltage for the last 200 ns of the interpulse period.
3. The no-transmission level shall be 0 V differential at the guidance/launch vehicle interface using the test load specified in table A.2.
4. Shielding depicted represents the telemetry shielding requirements only. For cable routing see void #01. Telemetry shielding shall be carried through all connectors between the electronic assembly and the telemetry subsystem.
5. A radiofrequency cap shall be provided on electronic assemblies in all launch vehicles in lieu of this connector.

Figure A.2.—Guidance data pulse characteristics.

Table A.2.—REQUIRED PULSE CHARACTERISTICS AND TEST PARAMETERS

Pulse characteristics (see fig. A.2)	Guidance telemetry					
	Data 1	Data 2	Bit synchronization	Frame synchronization	Data 1 word synchronization	Data 2 word synchronization
Pulse duration	255 + 50 ns					
Minimum amplitude	9 ± 2 V (see V027)					
Maximum amplitude	15 V					
Rise time	75 ns maximum					
Fall time	125 ns maximum					
Undershoot	2.5 V maximum					
Reference level offset	0 to –4.5 V relative to no-transmission level					
Noise	1.4 V maximum peak to peak					
Receiver susceptibility	2.0 V minimum					
Test parameters:						
Test load	75 Ω±5% resistive					
Receiver susceptibility	2.0 V minimum					

DDR No. 3288399–V027

Data required:	Guidance subsystem waveform parameter data (minimum amplitude value to replace coordinated temporary amplitude band currently on ICD–3288399)
Data supplier:	SP–2012/guidance telemetry steering committee
Data user(s):	SP–2732/launch vehicle telemetry contractor/interface coordinator
Date due:	45 days following guidance preliminary design review

Figure A.3.—Typical design data required for table A.2.

Appendix B
Mechanical/Physical Interface Examples

B.1 Mechanical Interface for Distributed Electrical Box

Figure B.1 is an example of an interface development document (IDD) that, from initial inspection, appears to be fairly complete. This figure contains a great amount of detail and just about everything appears to be dimensioned. However, closer examination will reveal serious shortcomings.

First, the basic function of the interface must be defined. The box depicted must be capable of being removed and replaced on orbit, in many cases outside the crew habitat. In some cases it is to be removed and replaced robotically. The box slides along the L-shaped bracket held to the support structure by three mounting bolts labeled "bolt 1," " bolt 2," and "bolt 3." As the box slides along the L-shaped bracket from left to right in the figure, some piloting feature on the box connectors engages the connectors mounted to the support structure by the spring-mounted assembly, and the connector engages fully when the lead screw is completely engaged.

1. The initial interface area to be examined is that of the L-shaped bracket to the support structure (i.e., the interface of the three mounting bolts). The interface is being examined from the perspective of the designer of the support structure. Does figure B.1 contain enough information for a mating interface to be designed? (The area of interest has been enlarged and is presented as figure B.2.)

   a. The dimensions circled in figure B.2 and lettered a, b, c, and d locate the position of the mounting bolts relative to the box data. The following pertinent differences are noted concerning this dimensioning:

      i. Dimension a locates the holes relative to a "reference datum for coldplate support structure," but the datum is not defined on the drawing. Is it a line or a plane? What are the features that identify/locate the datum? What is the relationship of this datum to other data identified on the IDD (data A, B, and D)? **This information is required so that the designer of the support structure can relate his or her interface features easily to those of the box IDD.**

      ii. The IDD states that the tolerances on three-place decimals is ±0.010. Dimensions a, b, c, and d are three-place decimal dimensions and would, therefore, fall under this requirement. Elsewhere on the IDD a true position tolerance for bolt locations is indicated. A feature cannot be controlled by both bilateral and true positioning tolerancing. It must be one or the other. Considering the function of the mounting bolts—to locate the box relative to the electrical connectors, it has to be assumed that dimensions a, b, c, and d are basic dimensions. **Interface control drawings cannot require the designer of the mating interface to assume anything. IDD's must stand by themselves.**

   b. Figure B.3 depicts initial details of mounting bolts for the L-shaped bracket. On first inspection there appears to be a great amount of detail. However, further examination shows that much of the detail is not related to interface definition. The interface is the bolt. Where is it relative to other features of the box? What is the relationship of bolts 1 and 2 to bolt 3 (datum C)? What is the thread of the bolt? How long is the bolt? The following data on the IDD are not required:

      i. Counterbore for bolt head

      ii. Diameter of bolt hole in bracket for bolts 1, 2, and 3

      iii. Distance of bolt hole to first thread

      iv. The fact that there is a screw retaining ring

      Adding data not required for the interface, even if they are only pictorial, is expensive. It takes time for the organization to develop and present it, and it takes time for the designer of the mating interface to determine that the information is not necessary and discard it. If the extraneous information stays on the IDD, it must be maintained (i.e., changed if the design details change). **Only the features of a design that affect the features of the design of the mating interfaces need be placed on the IDD.**

   c. Once the unnecessary data are removed, what remains is shown in figure B.4. The data that remain are not complete and are unclear. The true position notations are indicated as being those for the "mounting interface for bolt," suggesting that the true position applies to the hole in the support structure. However, since the IDD is basically covering the features of the box, it is assumed that these locations apply to the bolts on the box. **It should not be necessary to have to make assumptions about data on an IDD or ICD. The document should stand by itself.**

The only other data left in figure B.4 are the callouts for the locking inserts. These callouts refer to the method used by the designer of the support structure for retaining the bolts. This IDD should not have this callout, since the

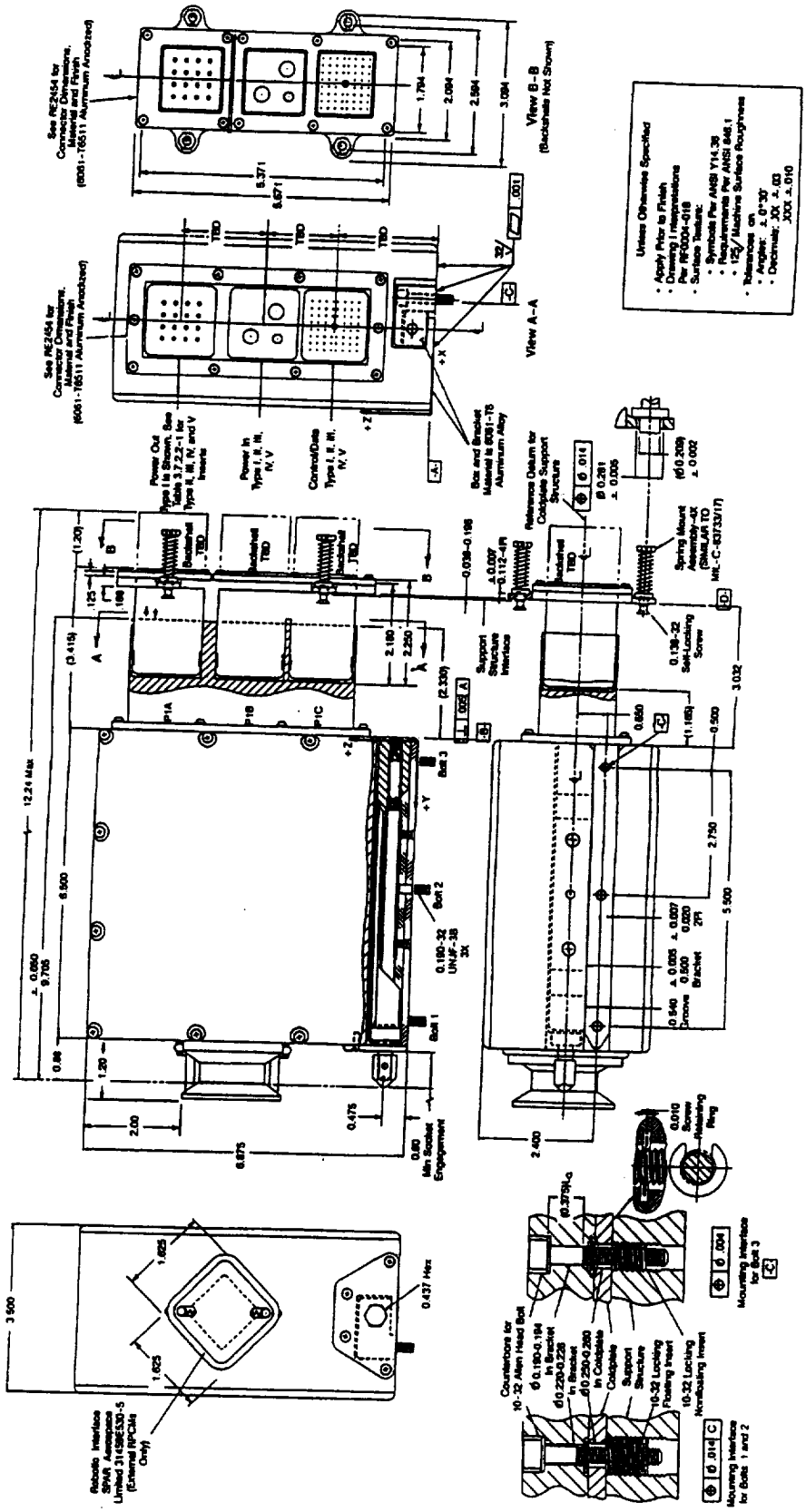


Figure B.1.—Partial interface development document for multiuse electrical power interface box showing bolt locations.

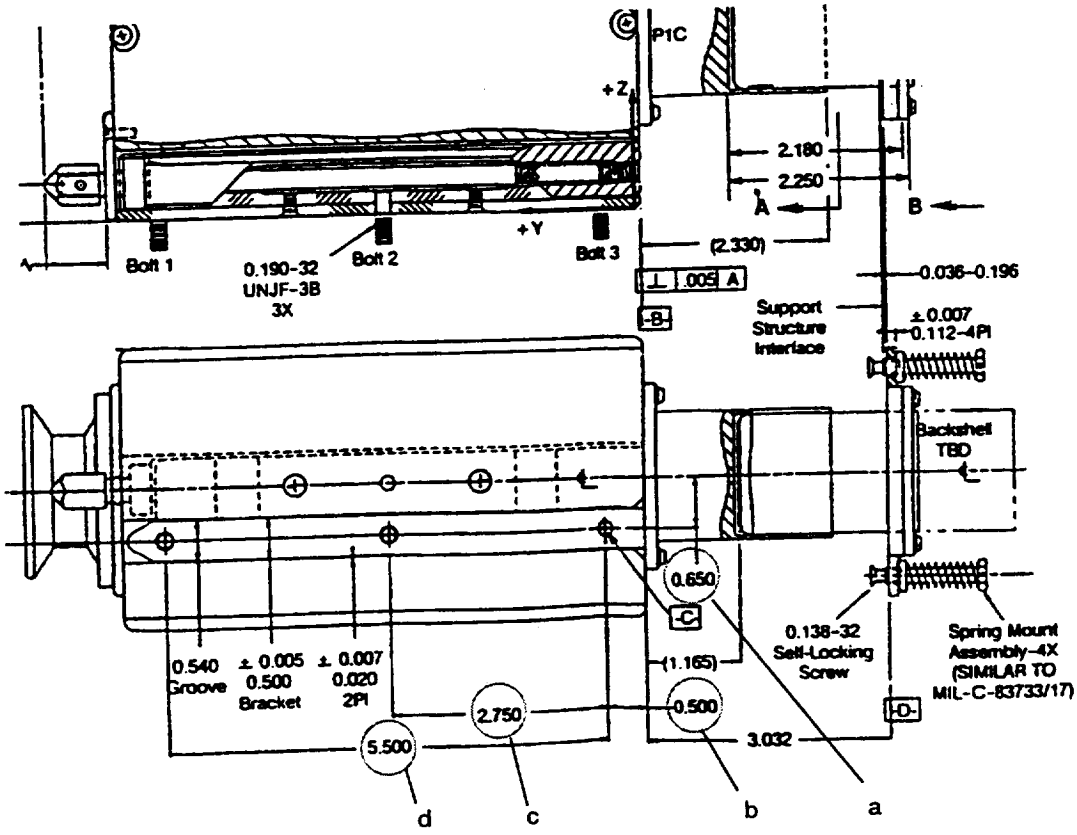


Figure B.2.—Detail of L-shaped bracket interface.

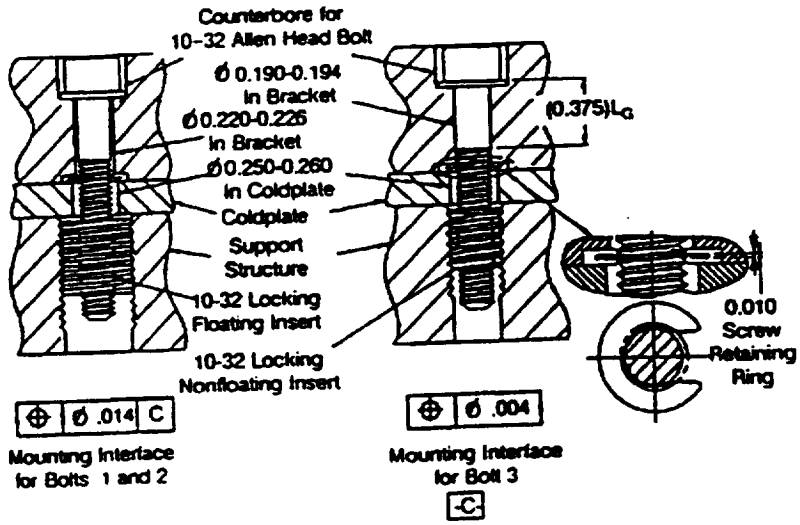


Figure B.3.—Initial details of mounting bolts.

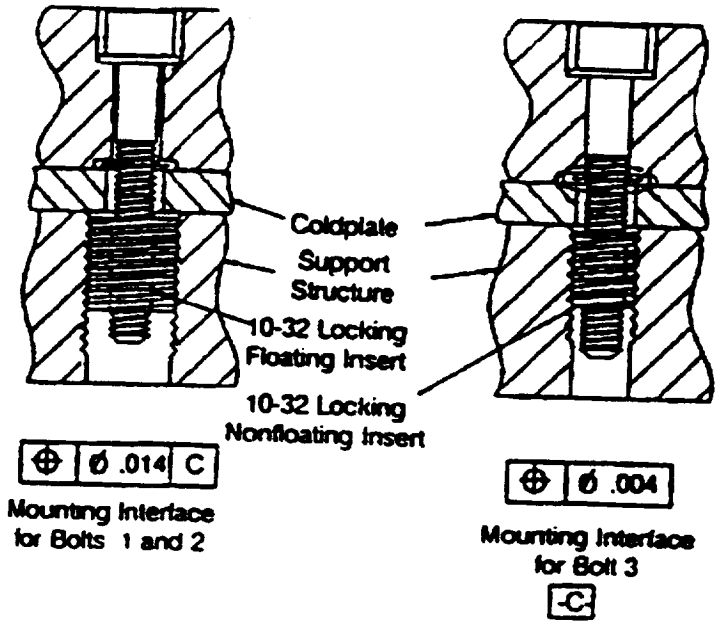


Figure B.4.—Necessary details of mounting bolts.

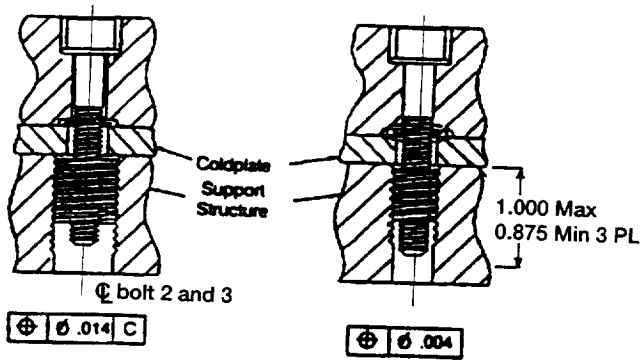


Figure B.5.—Minimal interface definition.

method used for retaining the bolts is not the responsibility of the box designer. **Generally IDD's and ICD's should not specify design solutions, especially when the design solutions are not the responsibility of the one specifying them.**

What is missing is how far the bolts protrude from the box. These data are required so that the designer of the support structure knows how deep to make the mating hole and how much of a mating thread must be supplied to grip the bolts on the box.

Considering all of the above, figure B.5 represents what is really required (along with the locations and thread types already defined in fig. B.1) to define the box side of the interface and for the designers of the support structure to design a compatible interface between the retaining bolts and the support structure.

2. The next area to be examined is that of the connector interface. Since both parts of the connector are being provided by the box designer, the interface is the plate on which the connectors are attached to the support structure. Again, the question is, Does figure B.1 contain enough information for a mating interface to be designed? The answer to that question is, Definitely not! The interface of the plate (holding the connectors) that mates with the support structure is identified as datum D. Again, there is no definition of this datum. Is it a plane passing through the three highest points of the plate or some other features of the connector plate?

If a compatible mating interface is to be designed, the relationship between the surface to which the connector plate is attached and the surface to which the L-shaped bracket is attached must be known. None of these data are supplied in figure B.1. The following are data needed to establish this relationship:

a. The required perpendicularity of D to A
b. The required parallelism of D to B
c. The required angular relationship of the vertical centerline shown in view B–B with the vertical centerline shown in view A–A
d. The pattern required for the four fasteners holding the connector plate to the support structure. View B–B does contain a dimension of 2.594 for a horizontal spacing of the lower two features but does not indicate that this dimension is applicable to the upper two fasteners. In addition, there is no dimension for the distance between the fasteners in the *Z* direction.
e. The required relationship of the hole pattern for the connector plate relative to the box, namely,
    i. The location of the hole pattern above A in the *Z* direction
    ii. The location of the hole pattern relative to C in the *X* direction
    iii. The distance of datum D from C in the *Y* direction when the box is fully installed

Since none of these data are identified as items to be determined (TBD's), it must be assumed either that the data are not required because the connectors can be mated properly with a great deal of misalignment or that the box designer did not recognize that this type of data is required. Designers never wish to freeze a design. The placement of design constraints in an ICD is basically freezing an area of a design or at least impeding the ability to change a design without that design being scrutinized at another level. Therefore, **the tendency of designers is to disclose the minimum that they feel is necessary in the interface for the control process. This is the primary reason for the ICD custodian not to be organizationally a part of the design process.** Yet the ICD custodian must have access to the design function of an agency or contractor organization to ensure the ready flow of the data required for proper interface definition. (Can interface compatibility be demonstrated from the ICD's alone?)

The ICD custodian must always test the data in interface documentation from the viewpoint of another design agent who must develop a compatible mating interface.

The preceding discussion simplifies specification of the L-shaped bracket and the mounting bolts. This redefinition of the interface tied up loose ends and provided needed dimensions and callouts absent from the original document. These portions of the document can now be controlled more easily and related to a 100% mate design.

B.2 Space Reservation and Attachment Features for Space Probe Onboard Titan IV Launch Vehicle

Figure B.6 is an example of an ICD that defines the space envelope available onboard the Titan IV launch vehicle for a payload and the attachment feature details for the launch vehicle side of the interface. The intended payload is the Cassini Mission spacecraft. The Titan payload fairing, as would be expected, is defined. The other side of this envelope (i.e., the spacecraft) must also be defined to show compatibility. When the spacecraft dimensions are established, compatibility should be shown by a comparison of the two envelopes. The Titan documentation defines the available space reserved for equipment (i.e., a stay-out zone for the Titan launch vehicle items). Ideally, this ICD should define a minimum space available for the spacecraft. Therefore, if the spacecraft dimensions are constrained to a maximum size equal to the launch vehicle's minimum, less a value for environmental effects, etc., then the two envelopes are compatible.

Since interface data have been provided for the attachment details for the launch vehicle side of the interface, the design of the Cassini adapter for mounting to the Centaur launch vehicle at station –150.199 can be explained by using the Titan design data.

The following key interface features have been established for this connection:

1. Sheet 1 (fig. B.6(a)), note 5: Location of holes is established by a common master gauge tool with reference dimensions provided.
2. Sheet 3 (fig. B.6(c)), section F–F: Bearing areas are to be flat within 0.006 (units), and per view G the maximum bearing area has been defined.
3. Sheet 3 (fig. B.6(c)), view H: Shape and dimensions of the shear alignment pins have been established.
4. Sheet 1 (fig. B.6(a)), note 4: How loads are to be transmitted is indicated.

The following data elements missing from figure B.6 are mostly related to the lack of spacecraft design data:

1. No apparent tracking of TBD's. **A tracking system should be in place at the beginning of ICD development. Each TBD should have a unique sequential identifier with due dates and suppliers established.**
2. No revision block for tracking the incorporation of changes. Some type of revision record should be placed on each sheet.

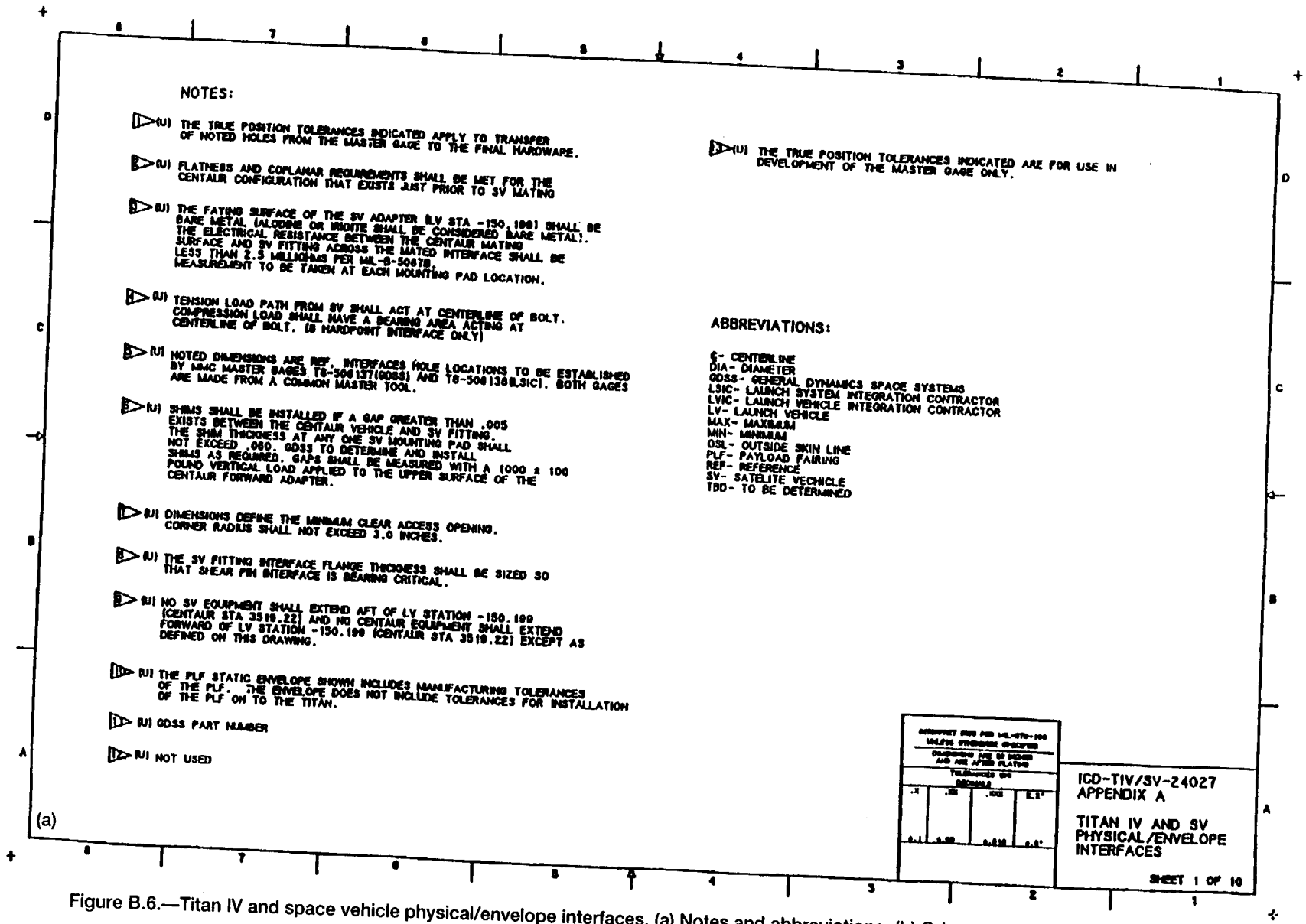


Figure B.6.—Titan IV and space vehicle physical/envelope interfaces. (a) Notes and abbreviations. (b) Orientation view. (c) Section views.

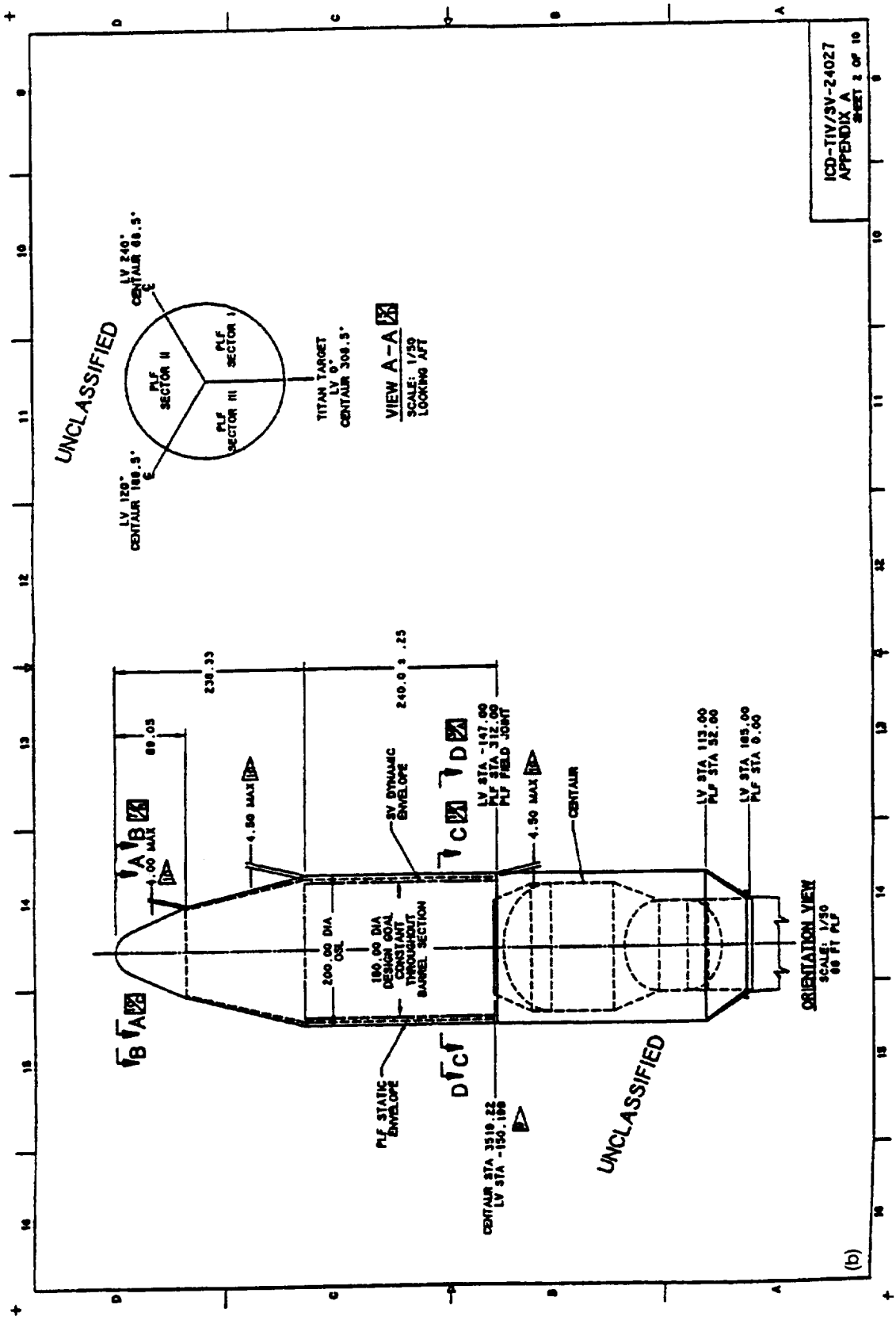


Figure B.6.—Continued. (b) Orientation view.

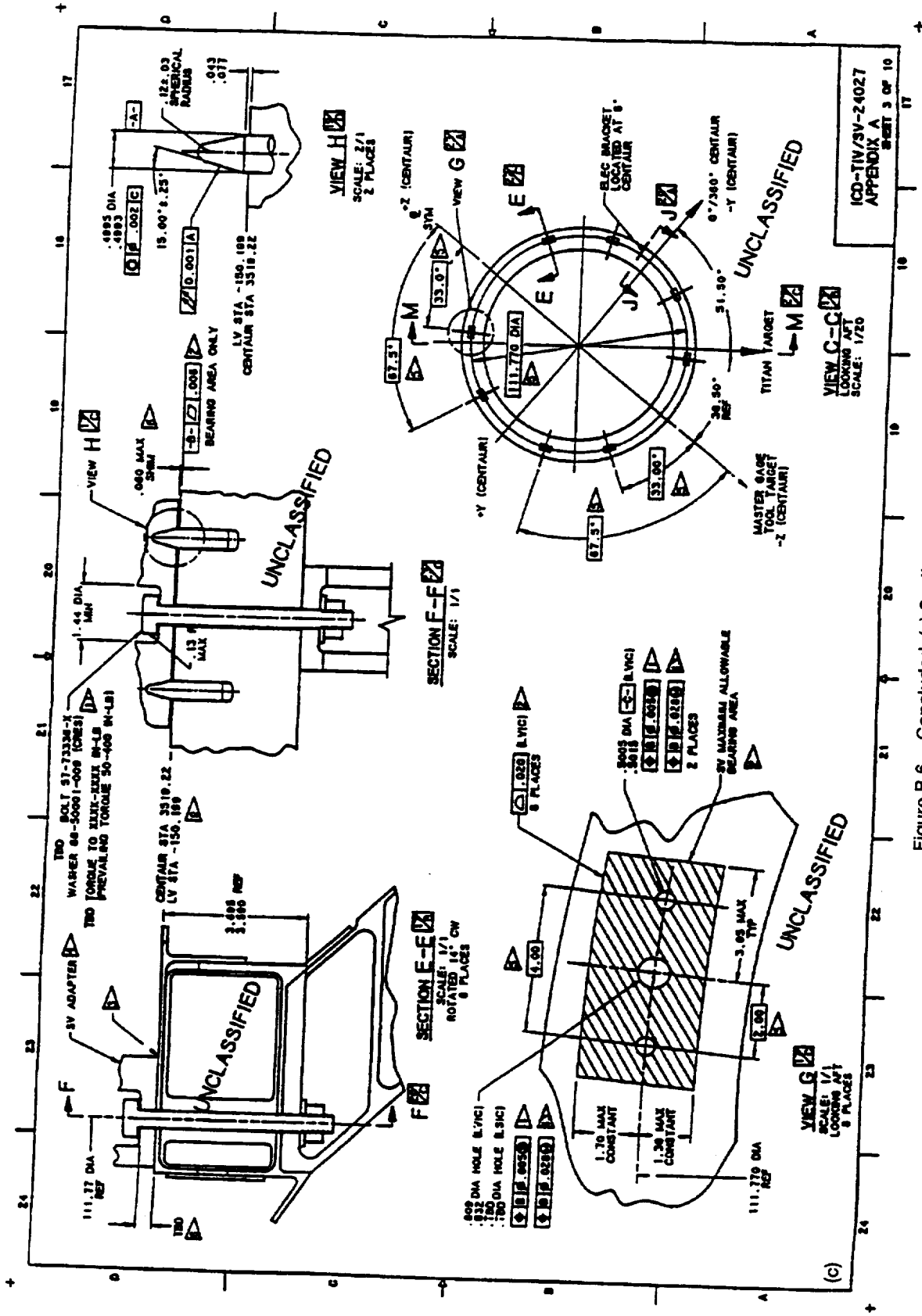


Figure B.6.—Concluded. (c) Section views.

Upon exchange of design data relating to the Cassini probe it would be expected that the probe's maximum envelope would be established and related to the data system of the Titan/Centaur launch vehicle.

This example is basically a one-sided interface. The Titan/Centaur side of the interface is well defined, which is to be expected considering the maturity of the design. The tendency should be resisted, in cases like this, to ignore or place less emphasis on the definition and documentation of the mating interface, given the completeness of the launch vehicle side of the interface. The mating interface, namely, the spacecraft side, should be completely defined. Otherwise, the spacecraft designer will be signing up to design a compatible interface by agreeing with what the interface on the launch vehicle side looks like. Although this approach allows freedom to go off and "do independent things," it lacks the degree of positive control needed for interface compatibility. The chances for an incompatibility are much less if the spacecraft side of the interface is defined. Space vehicle data, stations, and fasteners must be identified and controlled. The designer of the space vehicle is then able to commit to the design and production of an interface that is defined. The launch vehicle designers can then verify that the spacecraft interface will mate with the launch vehicle available for the spacecraft. Therefore, if the spacecraft dimensions are constrained to a maximum size equal to the launch vehicle's minimum, less a value for environmental effects, etc., then the two envelopes are compatible.

Since interface data have been provided for the attachment details for the launch vehicle side of the interface, the design of the Cassini adapter for mounting to the Centaur launch vehicle at station −150.199 can be explained by using the Titan design data.

Appendix C

Software Interface Example: Definitions and Timing Requirements for Safety Inhibit Arm Signals

Signal definition	Centaur sequence control unit switch number	Initiating event + time	Persistence	Function
Satellite vehicle (SV) pyro unshort (primary)	45	Main engine cutoff (MECO) 2 + 3±0.5 sec	3±0.5 sec	Unshorts SV pyro capacitor banks
SV latch valve arm (primary)	33	MECO2 + 10±0.5 sec	3±0.5 sec	Arms safety inhibit relay for SV main engines
SV pyro unshort (secondary)	89	MECO2 + 15±0.5 sec	3±0.5 sec	Provides redundant unshort of SV pyro capacitor banks
SV latch valve arm (secondary)	88	MECO2 + 17±0.5 sec	3±0.5 sec	Provides redundant arm of inhibit relay for SV main engines
Radiofrequency monopropellant driver backup enable	34	Titan IV/Centaur separation + 24±0.5 sec	3±0.5 sec	Services backup (redundant to SV ground support equipment command) enable of safety inhibit SV functions (radiofrequency sources and reaction control system thruster drivers)

Appendix D

Supplied Services Interface Example

This appendix provides a simplistic text-based example of a supplied services (air-conditioning and cooling water) interface control document with a typical design-data-required (DDR) block. This example contains elements condensed from a number of service documents originally used for a submarine weapons program; however, the principles contained herein are universally applicable to any complex system of interfaces. Page 1 of the ICD lists the DDR's (table D.1) showing DDR numbers, location on the drawing, brief description, and due date. The DDR block (fig. D.1) on the drawing expands on this information and identifies supplier, user, and time urgency of the data needed. The DDR numbering convention used here is "V09 = Void #09." Preceding the void number with the ICD number provides a program-unique DDR number that is easily related to its associated ICD and easily maintained in a data base.

TABLE D.1.—DESIGN-DATA-REQUIRED SUMMARY AND LOCATOR

Void number	Location	Description	Date due
V01			
⋮			
V09	Sheet 1, zone C–7	Main heating and cooling (MHC) water schedule	30 Days after authentication of data fulfilling DDR 5760242–V12
⋮			

DDR No. 1466134–V09

Data required: Heating and cooling (HC) system upper zone water schedule (supply water temperature versus environmental temperature)

Data supplier: HC working group

Data user: Launch vehicle design agent

Date due: 30 days after authentication of data fulfilling DDR No. 2543150–V12

Figure D.1.—Typical design-data-required block.

The following pages present the kinds of data required to fully define the air-conditioning requirements for suites of equipment located in a launch control center. Table D.2 details conditioned-air distribution; table D.3 presents typical interface data required to ensure that a cooling water service is provided to electrical equipment and indicates requirements for the equipment before and after the incorporation of an engineering change.

701. Launch vehicle control center services:

A. Air-conditioning shall be provided with a dedicated closed-circuit system capable of supplying a minimum total flow of 12 820 scfm with a 50% backup capability.

1. The conditioned air shall be distributed to each equipment flue as specified in table D.2. The distributed conditioned air at the inlet to the equipment shall satisfy the following parameters:
   a. Temperature: The minimum temperature shall be 65 °F and the maximum, 70 °F.
   b. Humidity: The maximum humidity shall be 75 grains per pound of dry air.
   c. Working pressure: The working pressure shall be enough to overcome equipment pressure drops and to maintain positive pressure at the equipment outlet with respect to compartment ambient pressure. A 10% minimum leakage rate in the compartment shall be assumed.
   d. Flow resistance: The system shall be able to overcome the pressure drop across the equipment (i.e., from exit of orifice plate to top of equipment) as shown in table D.2.
   e. Flow profile:
      (1) The flow distribution for each flue shall be such that the flow velocity between the flue centerline and 1.3 in. from the edge of the flue, and (where equipment permits) 6 in. above the flue gasket, shall not be less than 80% of the achieved average flow velocity. The achieved average flow velocity must equal or exceed velocity based on the minimum flow rate specified in table D.2.
      (2) Velocity profiling is not required for flues designated 301 through 310, 011 through 015, 446BC, 405–2A, 405–2B, 405–6A, and 405–6B.
   f. Adjustment capability: The system shall provide flow adjustment from 0 to 300 scfm at each of the equipment flues requiring velocity profiling.
   g. Air quality: Air at the inlet to the equipment shall be equivalent to or better than air filtered through a 0.3-μm filter with an efficiency of 95%.
2. The closed-loop system shall have the capacity of removing 52.8 kW (minimum) of heat dissipated by equipment using closed-circuit conditioned air. This heat load includes 1.3 kW reserved for launcher equipment in the launch vehicle control center (see note 702 below).

702. The system shall provide the capability of removing 1.65 kW minimum of heat dissipated by equipment by using compartment ambient air as a cooling medium while maintaining the compartment within specified limits.

A. The ship shall take no action that eliminates the option for launcher equipment to use compartment ambient air or closed-circuit conditioned air for dissipating launcher-generated heat of 1.3 kW.
B. Heat dissipated to ambient air by equipment using closed-circuit conditioned air is not included.

703. The system shall provide distribution trunks to equipment flues with total flow capacity as designated below for the conditions of table D.2:

Trunk	Minimum flow, scfm
A	2700
B	1620
C	2300
D	3400
E	1300
F	1500

704. Flow at reference designations marked with an asterisk in table D.2 are to be considered flow reserve capabilities. These designated flues do not require verification of flow per table D.2 nor profiling per note 701.A.1.e(1) until these flues are activated. The Government-furnished pipe assemblies and caps will be supplied for flues not activated.

705. The minimum flow for flues 446BC and 447BC is 100 scfm before change 30175 and 250 SCFM after change 30175.

TABLE D.2.—CONDITIONED-AIR DISTRIBUTION

Equipment	Trunk (see note 703)	Flue	Minimum flow, scfm	Flow resistance/ pressure drop at minimum flow (see note 701A.1.d), in. H_2O
Data cabinets	A	301B	225	0.54
		301C	260	---
		305B	80	.50
		305C	80	.50
		306B	290	.56
		306C	50	.50
Data console	A	308B	100	.50
		308C	50	.50
		309	0*	---
		310B	135	.50
		310C	50	.50
Control console	E	405–2A	100	1.0
		405–2B	100	
		405–6A	50	
		405–6B	50	
Power buffer and conversion	B	011	440	2.0
		012	440	
		013–1	150	
		013–2	150	
		015	440	
Control computer group	D	440BC	200	1.0
		440–441D	300	
		444BC	300	
		444–445D	250	
		446BC	See note	
		447BC	705	
	E	471	200	
Control group	E	450BC	200	
		450–451D	200	
		451BC	100	
	C	452BC	200	
		452–453D	200	
		458BC	200	
		458–459D	200	
		459BC	150*	
	E	472	150*	
Power distribution	F	002BC	150	
		003BC	150	
		004BC	150*	
		004D	150*	
Load	F	271BC	275	1.0
		271D	0*	0
		005BC	100*	1.0
		005D	0*	0

*Flow reserve capability.

TABLE D.3.—WATER FLOW RATE INTERFACE PARAMETERS

[Water inlet temperature: 54 °F max and 48 °F min; temperature alarm set at 56 °F ±1 °F (increasing) and 47 °F ± 1 °F (decreasing); see Remarks.
Working pressure: 85 psig max and 57 psig min. Test pressure, 125 psig max with binnacles to be isolated at vehicle hydrostatic test.
Pressure drop: nominal differential pressure range, 13 to 23 psid ref. Water quality: dual filters required;
filters to 10 μm with 98% efficiency by weight, 20 μm absolute.]

Function	Minimum cooling capability	Water flow rate	Remarks
Electrostatically supported gyro navigator (ESGN) and gravity sensor system (GSS) binnacle cooling	2.25-kW gain	[a]6.0-gal/min nominal total flow for two ESGN binnacles and one GSS binnacle. The supply shall maintain constant flow of 2.0 gal/min ±10% to each binnacle. [b]A remote, low-flow alarm shall be provided for the ESGN binnacles and the GSS binnacle.	Reliability of water supply shall support a navigation subsystem availability of 0.97. This service requirement shall be continuously available during patrol and refit. The water temperature shall not vary by more than 6 °F when changing at the rate of 0.25 °F/sec maximum. This change shall not occur more than once per 30-min period.
Reserve capability for future navigation development	3.25-kW gain	2.6-gal/min minimum	
ESGN binnacle cooling	1.5-kW gain	[a]4.0-gal/min nominal total flow for two ESGN binnacles. The supply shall maintain a constant flow of 2.0 gal/min ±10% to each binnacle. [b]A remote, low-flow alarm shall be provided for the ESGN binnacles.	
Reserve capability for future navigation development	4.0-kW gain	4.5-gal/min minimum	

[a]The system shall provide test connections at the inlet and outlet of each binnacle to permit periodic measurement of differential pressure.
[b]Local flow indication shall be provided for each binnacle.

Appendix E

Compatibility Analysis

E.1 Definition

Compatibility analysis of the interface definitions contained in an ICD is a major tool of interface control. It serves a twofold purpose:

1. Demonstrates completeness of interface definition. If any interface data are missing or presented in a manner that cannot be integrated by using the ICD alone as a data source, the ICD is considered deficient.
2. Provides a record (traceability) that the interface has been examined and found to have the right form and fit. This record can then be used in evaluating the acceptability of subsequent change proposals.

E.2 Kinds of Data

The following compilation identifies the kinds of data that must be obtained for a compatibility analysis and outlines the general steps that should be followed for three categories of interface: electrical/functional, mechanical/physical, software, and supplied services:

I. Interface category—electrical/functional
  A. Data required to perform analyses
    1. The following parameters are required, considering the specific function or signal involved:
      a. Cabling and connectors
      b. Power requirements
      c. Electromagnetic interference, electromagnetic compatability, electromagnetic radiation, and grounding requirements
      d. Functional flow and timing requirements
      e. Signal definition
      f. Digital data definition to the bit level
      g. Protocol levels
      h. Seven-layer International Standards Organization open systems instruction stack definition or its equivalent
      i. Error recovery procedures
      j. Startup and shutdown sequences
      k. Adequacy of standards used or referenced
    2. Unique requirements for an interface or a piece of equipment different from overall system requirements (i.e., the hierarchy of specifications required)
    3. Adequate definition of all signals crossing the interface. "Adequate" is difficult to define precisely but depends on the signal type (e.g., analog or digital) and the intended use. In general, the interface must show the characteristics of the isolating device (element) on each side of the interface and define the signal characteristics in engineering terms suitable for the particular type of signal.
    4. Timing and other functional interdependencies
    5. System handling of error conditions
    6. Full definition of any standards used. Most digital transmission standards have various options that must be selected; few, if any, standards define the data that are passed.
  B. Steps to be followed
    1. Verify interoperability of connectors.
    2. Size cables to loads.
    3. Determine cable compatibility with signal and environmental conditions.
    4. Define data in one document only.
    5. Determine adequency of circuit protection devices and completeness of signal definition.
II. Interface category—mechanical/physical
  A. Type of interface—form and fit
    1. Data required to perform analysis
      a. A datum (reference) that is common to both sides of the interface (e.g., a mounting hole in one part that will mate with a hole or fastener in the other mating parts or a common mating surface of the two mating parts)
      b. Dimensions and tolerances for all features of each part provided in a manner that gives the optimum interface fit and still provides the required design functions. Optimum interface means dimensioning so that the tolerance accumulation is kept to a minimum.
    2. Steps to be followed
      a. Start with the common datum and add and subtract dimensions (adding the tolerance accumulations for each dimension) for each feature of the part interface.
      b. Determine the dimensional location of the interface-unique features by adding and subtracting the tolerance accumulations from resulting dimensions to achieve the worst-case maximum and minimum feature definitions.
      c. Perform the same analysis for the mating features of the interfacing part.
      d. Compare and question the compatibility of the worse-case features of the two mating parts (Will

the maximum condition of one part fit within the minimum condition of the mating part?)

B. Type of interface—structural load

1. Data required to perform analysis

a. A description of the loading conditions (static or dynamic) and the duration of those conditions

b. Characteristics of the equipment involved: weight or mass; mass distribution; elastic properties; and sensitivity of elastic properties to temperature, moisture, atmospheric gas content, pressure, etc.

2. Steps to be followed. This analysis involves placing the interfacing items in a position that produces the maximum loads while the items are interfacing. A space experiment is primarily designed for flight loads, yet it must withstand the loads developed during the launch and deployment cycles and perhaps unique loads during launch processing. The complexity of the compatibility analysis will vary depending on the types of interfacing items and environments.

a. Attachment loads are the simplest, being a statement of the loads applied by the attaching feature (bolt) and the load capability of the component being retained (flange).

b. Hoisting and handling loads require the calculation of bending moments or shear for various loading scenarios. Dynamic and environmental loads must also be considered. (How quickly is the load applied? What are the wind loading factors?)

c. A more complex situation will be the loads developed during a dynamic interaction of interfacing items where different material characteristics must be considered along with the reaction characteristics of the materials (e.g., a flexible beam of varying moments of inertia supported by an elastomeric medium where the entire system is subjected to a high-velocity impulse of a few microseconds duration). Such a condition could produce loads that exceed those for which one of the interfacing items is designed. Another interfacing item may have to be redesigned so as not to jeopardize the mission of the primary item (i.e., increasing the strength of the item being supported could increase the weight).

III. Interface category—software

A. Type of interface—software. The ICD is required to specify the functional interface between the computer program and any equipment hardware with which it must operate. Often, the supplier documentation for standard computer peripherals and terminals is adequate for this purpose. Conversely, it has been found that performance specifications governing the design of new equipment are not satisfactory for use in a functional ICD. The purpose of an ICD is to communicate equipment interface requirements to programmers in terms that the programmers readily and accurately understand and to require equipment designers to consider the effect of their designs on computer programs.

B. Type of interface—hardware/software integration. The ICD provides an exact definition of every interface, by medium and by function, including input/output control codes, data format, polarity, range, units, *bit weighting*, frequency, minimum and maximum timing constraints, legal/illegal values, accuracy, resolution, and significance. Existing documentation may be referenced to further explain the effect of input/output operations on external equipment. Testing required to validate the interface designs is also specified.

IV. Interface category—supplied services

A. Type of interface—fluid service

1. Data required to perform analysis

a. Type of fluid required by the equipment and type of fluid the service supplier will provide. This may be in the form of a Federal or military specification or standard for both sides or for one side of the interface.

b. Location of the equipment/service interface (hose connection, pipe fitting, etc.)

c. Equipment requirements at the interface location in regard to characteristics (pressure, temperature, flow rate, duty cycle, etc.)

d. Capability of the service supplier at the interface location

e. Manner in which the equipment can affect the capability of the service supplier (e.g., having a large backpressure that the supplier fluid must push against or a combination of series and parallel paths that the supplier fluid must pass through)

2. Steps to be performed. Examine the supplier and equipment requirements to determine

a. If the supplier capability meets or exceeds the equipment requirements. This may require converting a Federal/military specification or standard requirement into what is specified for the equipment.

b. If the supplier capability meets the requirements, considering the effects resulting from the fluid passing through the mating equipment

B. Type of interface—environmental

1. Data required to perform analysis

a. Conditions required for equipment to function properly. Storage, standby, and operating scenarios need to be established and defined.

b. Supplier's capability to provide the environment specified in terms of time to reach steady

state from transients resulting from uncontrollable external environments; the limits of the steady-state conditions (maximum/minimum); and monitoring features

2. Steps to be performed. Perform analyses (e.g., thermal) under extreme and nominal environmental conditions to verify that supplier's equipment can maintain the environment required for the equipment. The complexity of the analysis may vary depending on the types of items involved.
   a. Simple inspection, which considers the environment required by an item versus the capability of the ambient in which the item resides
   b. Complex analysis, which must consider uncontrolled external environmental inputs, the thermal properties of intermediate systems that do not contribute to the end environment but act as conduits or resistors in the model, and the interaction of the item and the system that controls the desired environment

Appendix F
Bracket System for Interfaces

Brackets are used on hardware/engineering drawings to flag or identify details controlled by the ICD. Changes cannot be made to the drawings or designs without the effects on the interface being assessed and coordinated through the ICD process.

The process uses a rating similar to that used in the problem/failure reporting bracket system with the same controls and traceability. Once a bracket has been assigned to an interface void or problem, specific analyses and actions are required for the bracketed item to be removed. The bracketed item remains in open status with assignment to the responsible cognizant subsystem or design section until (1) the corrective action or coordinated information has been developed, (2) a proper risk assessment has been performed, (3) ICD change actions have been completed, (4) adequate verification of the interface is planned, and (5) the proper approval signatures have been obtained.

The following ratings are used to establish a category of "bracket" identifiers for interface deficiencies. Any discrepancy having an A rating greater than 1 or a B rating greater than 2 will be designated a bracketed discrepancy (see figure F.1).

I. Interface deficiency rating A (S&MA impact)
   A. Rating A1: Negligible effect on interface or mission performance
      1. No appreciable change in functional capability (form, fit, and function are adequate for the mission)
      2. Minor degradation of engineering or science data
      3. Support equipment or test equipment failure but not mission-critical element failure
      4. Support-equipment- or test-equipment-induced failures
      5. Drawing errors not affecting element construction
   B. Rating A2: Significant degradation to interface or mission performance
      1. Appreciable change in functional capability
      2. Appreciable degradation of engineering or science data
      3. Significant operational difficulties or constraints
      4. Decrease in life of interfacing equipment
      5. Significant effect on interface or system safety
   C. Rating A3: Major degradation to interface or mission performance or catastrophic effect on interface or system safety

II. Interface deficiency rating B (understanding of risk)
   A. Rating B1: Effect of interface deficiency is identified by analysis or test, and resolution or corrective action is assigned and scheduled or implemented and verified. There is no possibility of recurrence.
   B. Rating B2: Effect of interface deficiency is not fully determined. However, the corrective action proposed, scheduled, or implemented is considered effective in correcting the deficiency. There is minimal possibility of recurrence and little or no residual risk.
   C. Rating B3: Effect of interface deficiency is well understood. However, the corrective changes proposed do not completely satisfy all doubts or concerns regarding the correction, and the effectiveness of corrective action is questionable. There is some possibility of recurrence with residual risk.
   D. Rating B4: Effect of interface deficiency is not well understood. Corrections have not been proposed or those proposed have uncertain effectiveness. There is some possibility of recurrence with residual risk.

	Interface discrepancy red flag; project or task manager approval required

Rating A (S&MA impact)	Numerical rating		Rating B (understanding of risk)
Negligible impact	1	1	Known deficiency with corrective action assigned, scheduled, and implemented
Significant degradation	2	2	Deficiency poorly defined but acceptable corrective action proposed, scheduled, and implemented (low residual risk)
Major degradation	3	3	Known deficiency but effectiveness of corrective action is unclear and does not satisfy all doubts and concerns (residual risk)
		4	Impact not defined with confidence; corrective action with uncertain effectiveness (residual risk)

Figure F.1.—Interface deficiency rating system.

Appendix G

ICD Guidelines

1. Interface control documents should not require the designer of the mating interface to assume anything. ICD's should be compatible with each other and stand alone.

2. Only the definition that affects the design of the mating interfaces need be used.

3. ICD's should not specify design solutions.

4. The ICD custodian should be independent of the design organization.

5. The ICD custodian should verify that the data being controlled by an ICD are sufficient to allow other organizations to develop the interface described by the ICD.

6. An interface control system should be in place at the beginning of system (hardware or software) development.

7. Each void should have a unique sequential identifier establishing due dates, identifying exact data to be supplied, and identifying the data supplier.

Appendix H
Glossary

baseline—The act by which the program manager or a designated authority signs an interface control document (ICD) and by that signature establishes the genuineness of the ICD as an official document defining the interface design requirements. The term "baseline" conveys the idea that the ICD is *the only* official definition and that this officiality comes from the technical management level. Not only is the initial version of the ICD baselined, but each change to an ICD is likewise approved.

comment issue—An issue of an ICD distributed for review and comment before a meeting of the affected parties and before baselining

custodian—The contractor or project assigned the responsibility of preparing and processing an ICD through authentication and subsequently through the change process

data—Points, lines, planes, cylinders, and other geometric shapes assumed to be exact for the purpose of computation and from which the location or geometric relationship (form) of features of a piece of equipment can be established

interface responsibility matrix—A matrix of contractors, centers, and project organizations that specifies responsibilities for each ICD listed for a particular task. Responsibilities are designated as review and comment, technical approval, baselining, and information.

electrical/functional interface—An interface that defines the interdependence of two or more pieces of equipment when the interdependence arises from the transmission of an electrical signal from one piece of equipment to another. All electrical and functional characteristics, parameters, and tolerances of one equipment design that affect another equipment design are specified.

interface—That design feature of one piece of equipment that affects a design feature of another piece of equipment. An interface can extend beyond the physical boundary between two items. (For example, the weight and center of gravity of one item can affect the interfacing item; however, the center of gravity is rarely located at the physical boundary. An electrical interface generally extends to the first isolating element rather than terminating at a series of connector pins.)

interface control—The process of (1) defining interface requirements to ensure compatibility between interrelated pieces of equipment and (2) providing an authoritative means of controlling the interface design.

interface control document (ICD)—A drawing or other documentation that depicts physical and functional interfaces of related or cofunctioning items. (The drawing format is the most common means of controlling the interface.)

interface control working group—A group convened to control and expedite interface activity between the Government, contractors, and other organizations, including resolution of interface problems and documentation of interface agreements

interface definition—The specification of the features, characteristics, and properties of a particular area of an equipment design that affect the design of another piece of equipment

interoperability—The ability of two devices to exchange information effectively across an interface

mechanical/physical interface—An interface that defines the mechanical features, characteristics, dimensions, and tolerances of one equipment design that affect the design of another subsystem. Where a static or dynamic force exists, force transmission requirements and the features of the equipment that influence or control this force transmission are also defined. Mechanical interfaces include those material properties of the equipment that can affect the functioning of mating equipment or the system (e.g., thermal and galvanic characteristics).

software interface—The functional interface between the computer program and any equipment hardware with which it must operate. Tasking required to validate the interface designs is also specified.

supplied-services interface—Those support requirements that equipment needs to function and that are provided by an external separate source. This category of interface can be further subdivided into environmental, electrical power, and communication requirements.

technical approval—The act of certifying that the technical content in an interface document or change issue is acceptable and that the signing organization is committed to implementing the portion of the interface design under the signer's cognizance.

References

1. MIL–STD–499: Engineering Management. May 1974.
2. Blanchard, B.S.; and Fabrycky, W.J.: Systems Engineering and Analysis. Prentice-Hall Inc., 1981.
3. MIL–STD–1521B: Technical Reviews and Audits for Systems, Equipment, and Computer Software, Notice 1, Dec. 1985.
4. Kockler, F.; Withers, T.; Poodiack, J.; and Gierman, M.: Systems Engineering Management Guide. Defense Systems Management College, Fort Belvior, VA, Jan. 1990.
5. ICD–Titan IV/Satellite Vehicle–24027, Cassini Mission. Martin Marietta Technologies, Inc., Denver, CO, Jan. 1994.
6. MIL–STD–490A: Specification Practices. Military Standard, June 1985.
7. ANSI Standard Y14.5: Dimensioning and Tolerancing.
8. DOD–STD–100: Engineering Drawing Practices.
9. SAMSO–STD 77–4: Format and Requirements for Interface Documents. Space & Missile Systems Org., Jan. 1979.
10. MIL–STD–704: Aircraft Electrical Power Characteristics.

Bibliography

AFR 65–3, AR 70–37, NAVELEXINST 4130.1, and 4139.1A: Joint Services Regulation on Configuration Management. Air Force Regulation, Naval Electronics Systems Instructions, Marine Corp Order.

AFSCP 800–7: Configuration Management. Air Force Systems Command Pamphlet.

DOD 4120.3–M: Defense Standardization and Specification Program Policies, Procedures and Instructions, Aug. 1978.

DOD 4245.7–M: Transition From Development to Production. Military Specification, Sept. 1985.

DOD 5010.19: Configuration Management. Military Specification, July 1990.

DOD–D–1000B: Drawings, Engineering and Associated Lists. Military Specification, July 1990.

DOD–STD–480B: Configuration Control—Engineering Changes, Deviations, and Waivers, July 1988. (Cancelled July 1992.)

ESMC Req 160–1: Radiation Protection Program. Eastern Space & Missile Center, Patrick AFB, FL.

Fairley, R.E.: Software Engineering Concepts. McGraw-Hill, New York, 1985.

FED–STD–209E: Airborne Particulate Cleanliness Classes in Cleanrooms and Clean Zones. Federal Standard, Sept. 1992.

KHB 1860.1: Kennedy Space Center Ionizing Radiation Protection Program. Kennedy Space Center, FL, 1972.

MCR–86–2550: Titan IV System Contamination Control Plan. Martin Marietta Aerospace Corp., Denver, CO, or Bethesda, MD, 1987.

MIL–B–5087B: Bonding, Electrical and Lightning Protection for Aerospace Systems. Military Standard, Dec. 1984.

MIL–N–7513F: Nomenclature Assignment, Contractor's Method for Obtaining. Military Standard, Notice 2, July 1993.

MIL–HDBK–259: Life Cycle Cost in Navy Acquisitions. Military Handbook, Apr. 1983.

MIL–P–27401C: Propellant Pressurizing Agent, Nitrogen. Military Standard, Aug. 1988.

MIL–S–83490: Specifications, Types and Forms. Military Standard, Oct. 1968.

MIL–STD–100E: Engineering Drawing Practices. Military Standard, Sept. 1992.

MIL–STD–482A: Configuration Status Accounting Data Elements and Related Features. Military Standard, Sept. 1968.

MIL–STD–973: On Configuration Management Practices for Systems, Equipment, Munitions, and Computer Software. Military Standard, 1993.

MIL–STD–1246C: Product Cleanliness Levels and Contamination Control Program. Military Standard, Apr. 1994.

MIL–STD–1388–1A: Logistic Support Analysis Reviewer. Military Standard, Mar. 1991.

MIL–STD–1456: Contractor Configuration Management Plans. Sept. 1989. (Cancelled July 1992.)

MIL–STD–1528A: Manufacturing Management Program. Military Standard, Sept. 1986.

MIL–STD–1541: Electromagnetic Compatibility Requirements for Space Systems. Military Standard, Dec. 1987.

PD 699–120: Cassini Final Targeting Specification. Program Directive, NASA or Executive Office of the President.

SECNAVINST 4130: Navy Configuration Management Manual. Executive Office of the Secretary of the Navy.

Training Answers

Chapter	Answers
1	1(A); 2(D); 3(C); 4a(C), 4b(A), 4c(B)
2	1(D); 2(C); 3a(B), 3b(C); 4a(C), 4b(C), 4c(C); 5a(A), 5b(A), 5c(A); 6a(C), 6b(A), 6c(A); 7a(B), 7b(B), 7cA(i), 7cB(ii), 7cC(i); 8a(B), 8b(A); 9a(A), 9b(A), 9c(B)
3	1a(A), 1b(B), 1c(B); 2a(B), 2b(A), 2c(B); 3a(A), 3b(B), 3c(A); 4a(B), 4b(A), 4c(B); 5a(A), 5b(B); 6a(A), 6b(B), 6c(B), 6d(B); 7a(B), 7b(A), 7c(A), 7d(A), 7e(A)

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (*Leave blank*)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	January 1997	Reference Publication

4. TITLE AND SUBTITLE

Training Manual for Elements of Interface Definition and Control

5. FUNDING NUMBERS

WU-323-42-02

6. AUTHOR(S)

Vincent R. Lalli, Robert E. Kastner, and Henry N. Hartt

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135-3191

8. PERFORMING ORGANIZATION REPORT NUMBER

E-9790

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

NASA RP-1370

11. SUPPLEMENTARY NOTES

This manual was edited by Vincent R. Lalli, NASA Lewis Research Center; Robert E. Kastner, Vitro Corporation, Rockville, Maryland; and Henry N. Hartt, Vitro Corporation, Washington, DC. Responsible person, Vincent R. Lalli, (216) 433-2354.

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified - Unlimited
Subject Category 15

This publication is available from the NASA Center for Aerospace Information, (301) 621-0390. Multiple copies are for sale by the National Technical Information Service, Springfield, VA 22161, (703) 487-4822.

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 words)***

The primary thrust of this manual is to ensure that the format and information needed to control interfaces between equipment are clear and understandable. The emphasis is on controlling the engineering design of the interface and not on the functional performance requirements of the system or the internal workings of the interfacing equipment. Interface control should take place, with rare exception, at the interfacing elements and no further. There are two essential sections of the manual. Chapter 2, Principles of Interface Control, discusses how interfaces are defined. It describes different types of interfaces to be considered and recommends a format for the documentation necessary for adequate interface control. Chapter 3, The Process: Through the Design Phases, provides tailored guidance for interface definition and control. This manual can be used to improve planned or existing interface control processes during system design and development. It can also be used to refresh and update the corporate knowledge base. The information presented herein will reduce the amount of paper and data required in interface definition and control processes by as much as 50 percent and will shorten the time required to prepare an interface control document. It also highlights the essential technical parameters that ensure that flight subsystems will indeed fit together and function as intended after assembly and checkout.

14. SUBJECT TERMS

Systems engineering; Configuration control; Documentation; Change notices; Interface management

15. NUMBER OF PAGES

60

16. PRICE CODE

A04

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	

Slide Rule and Insert Assembly

Cut out slide rule outer covers on pages 3 and 7 and slide inserts on pages 5 and 9 along cut lines.

Slide Rule Outer Covers (pages 3 and 7):

Cut out the 14 rectangular shapes, marked "cutout" *(a razor blade knife is recommended).*

Fold spacers back.

Note: Outer covers are assembled upside down to each other.

Be certain to match the words "top" to "top"
Assemble outer covers using glue on folded spacers.

Cut out the black half-circle notches.

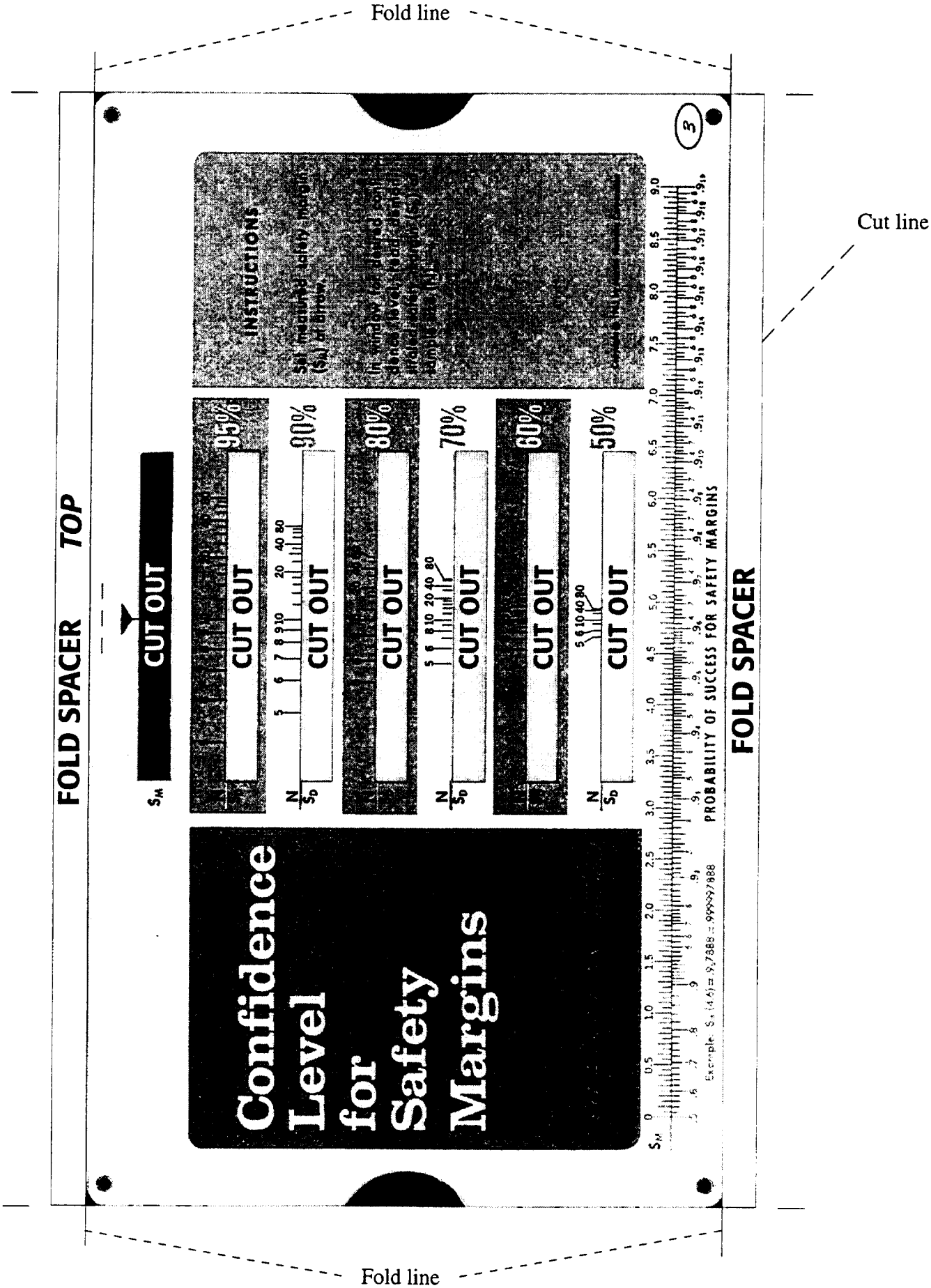
To reinforce the glue, staple outer covers where you see – – – (three places on top and bottom).

Slide Rule Insert (pages 5 and 9):

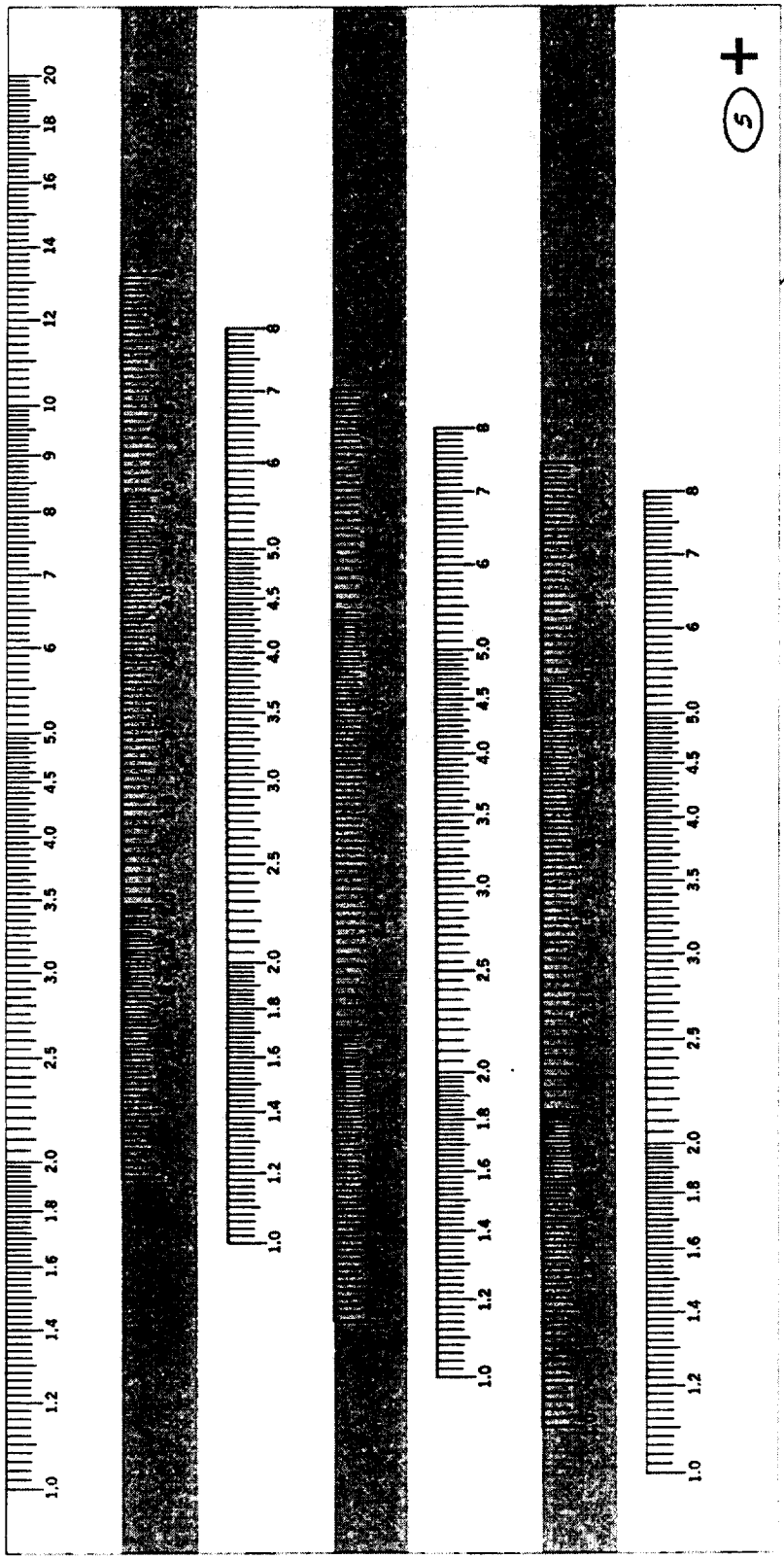
Make certain that plus signs are back to back.
Paste together (5) and (9) to make insert.

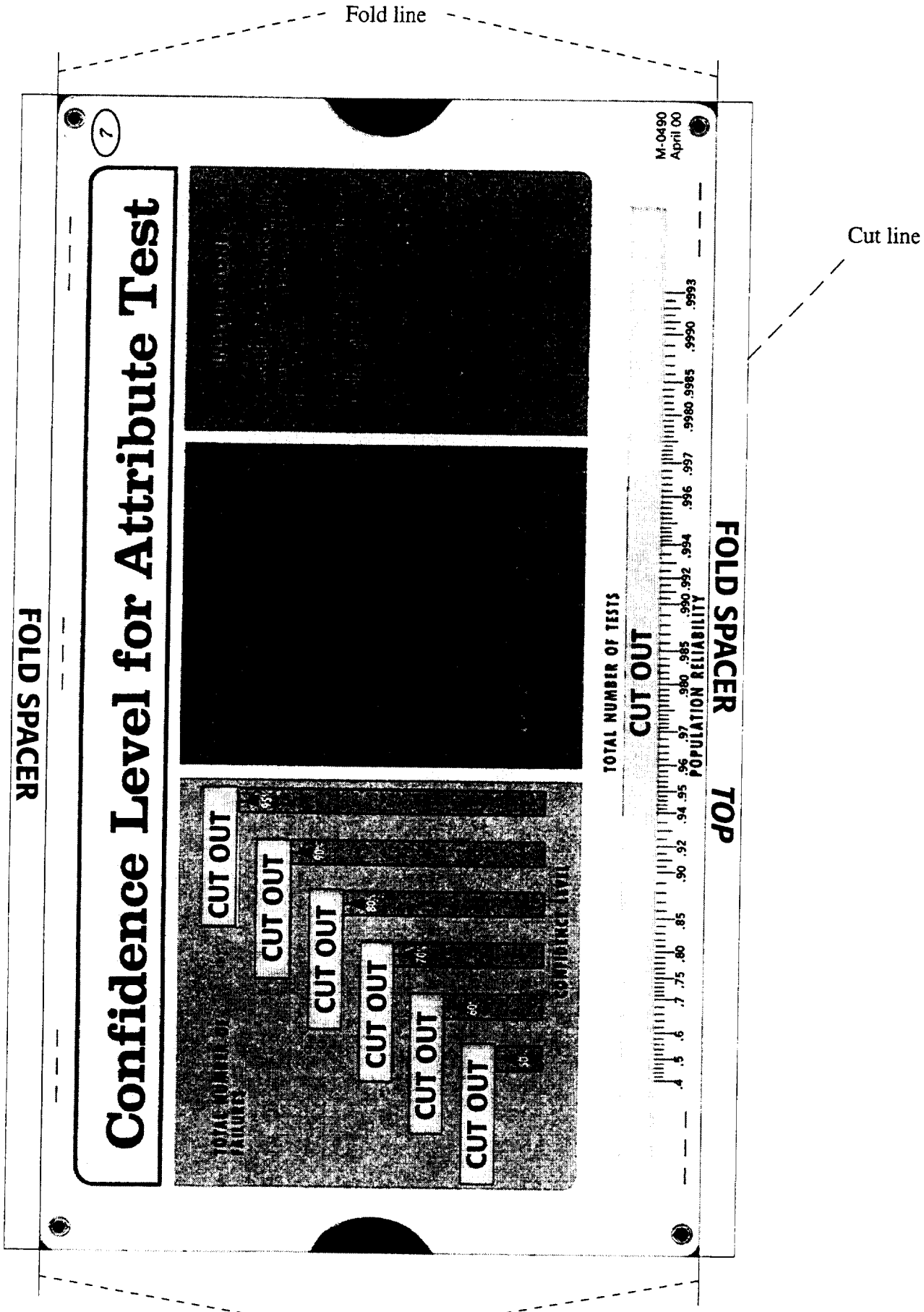
To assemble:

Place insert so that (3) on the cover and (5) on the insert are at the bottom right corner. When flipped, that will automatically align (7) on the cover and (9) on the insert on the top right corner.

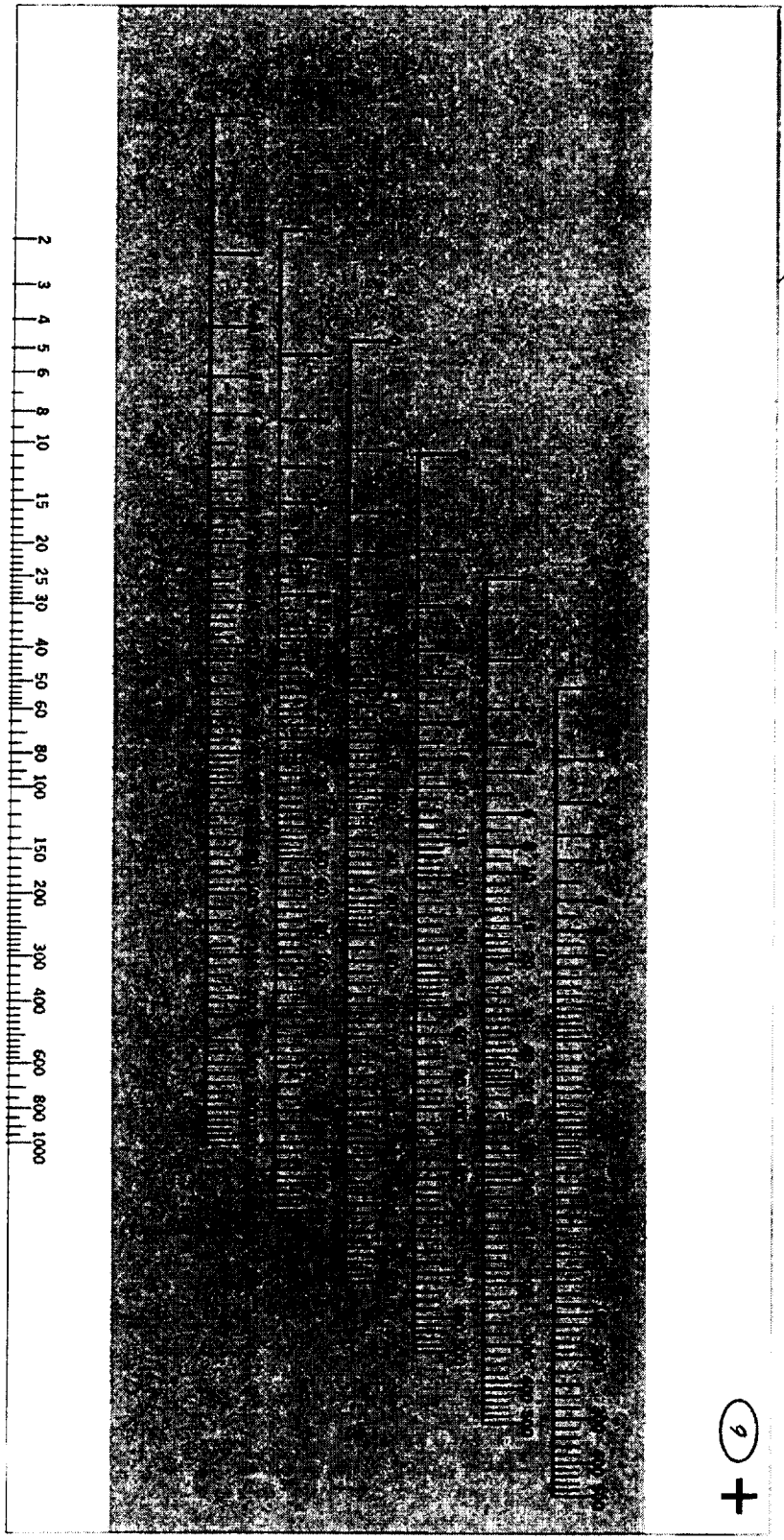

Fold line
Cut line
FOLD SPACER
TOP
FOLD SPACER
CUT OUT
95%
90%
80%
70%
60%
50%
INSTRUCTIONS
Confidence Level for Safety Margins
PROBABILITY OF SUCCESS FOR SAFETY MARGINS
Fold line

3


Cut line


Fold line
Cut line
Confidence Level for Attribute Test
CUT OUT
CUT OUT
CUT OUT
CUT OUT
CUT OUT
CUT OUT
TOTAL NUMBER OF TESTS
CUT OUT
POPULATION RELIABILITY
.4 .5 .6 .7 .75 .80 .85 .90 .92 .94 .95 .96 .97 .980 .985 .990 .992 .994 .996 .997 .9980 .9985 .9990 .9993
FOLD SPACER
FOLD SPACER
TOP
M-0490
April 00
7
Fold line

7


Cut line
2
3
4
5
6
8
10
15
20
25
30
40
50
60
80
100
150
200
300
400
600
800
1000

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (*Leave blank*)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	July 2000	Technical Paper

4. TITLE AND SUBTITLE

Reliability and Maintainability (RAM) Training

5. FUNDING NUMBERS

WU-323-93-00-00

6. AUTHOR(S)

Vincent R. Lalli, Henry A. Malec, and Michael H. Packard, editors

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
John H. Glenn Research Center at Lewis Field
Cleveland, Ohio 44135-3191

8. PERFORMING ORGANIZATION REPORT NUMBER

E-11144

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

NASA TP—2000-207428

11. SUPPLEMENTARY NOTES

This manual was edited by Vincent R. Lalli, NASA Glenn Research Center, Henry A. Malec, Siemens Stromberg-Carlson, Albuquerque, New Mexico 87123-2840, and Michael H. Packard, Ratheon Engineers and Constructors, Cleveland, Ohio 44135. Responsible person, Vincent R. Lalli, organization code 0510, (216) 433-2354.

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified - Unlimited
Subject Category: 15 Distribution: Standard

This publication is available from the NASA Center for AeroSpace Information, (301) 621-0390.

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 words)***

The theme of this manual is failure physics—the study of how products, hardware, software, and systems fail and what can be done about it. The intent is to impart useful information, to extend the limits of production capability, and to assist in achieving low-cost reliable products. In a broader sense the manual should do more. It should underscore the urgent need for mature attitudes toward reliability. Five of the chapters were originally presented as a classroom course to over 1000 Martin Marietta engineers and technicians. Another four chapters and three appendixes have been added. We begin with a view of reliability from the years 1940 to 2000. Chapter 2 starts the training material with a review of mathematics and a description of what elements contribute to product failures. The remaining chapters elucidate basic reliability theory and the disciplines that allow us to control and eliminate failures.

14. SUBJECT TERMS

Statistical concepts; Reliability; Maintainability; System safety; Quality assurance; Logistics; Human factors; Software performance; System effectiveness

15. NUMBER OF PAGES

363

16. PRICE CODE

A16

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102