NASA-TM-108702

NASA SYSTEMS ENGINEERING HANDBOOK

(NASA-TM-108702) NASA SYSTEMS ENGINEERING HANDBOOK. DRAFT (NASA) 129 p

N93-21188

Unclas

G3/81 0153684

NASA

Systems Engineering

Handbook

Draft

September 1992

by

Robert Shishko, Ph.D.
Robert G. Chamberlain, P.E.

with contributions by

Robert Aster
Vincent Bilardo, Ph.D.
Kevin Forsberg, Ph.D.
Walter E. Hammond, Ph.D.
Harold Mooz
Lou Polaski
Ron Wade

edited by

Randy Cassingham

Foreword
By Francis T. Hoban
Program Manager, NASA Headquarters

When NASA began to sponsor agency-wide classes in systems engineering, it was to a doubting audience. Top management was quick to express concern. As a former Deputy Administrator stated: "How can you teach an agency-wide systems engineering class when we cannot even agree on how to define it?" Good question, and one I must admit caused us considerable concern at that time. The same doubt continued up until the publication of this handbook.

The initial systems engineering education conference was held in January 1989 at the Johnson Space Center. A number of representatives from other Centers attended this meeting and it was decided then that we needed to form a working group to support the development of appropriate and tailored systems engineering courses. At this meeting the representatives from Marshall Space Flight Center (MSFC) expressed a strong desire to document their own historic systems engineering process before any more of the key players left the Center. Other Centers also expressed a desire, if not as urgent as MSFC, to document their process.

It was thought that the best way to reflect the totality of the NASA systems engineering process and to aid in developing the needed training was to prepare a top level (Level O) document that would contain a broad definition of systems engineering, a broad process outline, and typical tools and procedures. In general, we wanted a top level overview of NASA systems engineering. To this document would be appended each Center's unique systems engineering manual. The group was well aware of the diversity each Center may have, but agreed that this approach would be quite acceptable.

The next step and the most difficult in this arduous process was to find someone to head this yet-to-be-formed working group. Fortunately for NASA, Donna Pivirotto of the Jet Propulsion Laboratory stepped up to the challange. Today, through her efforts, those of the working group, and the skilled and dedicated authors, we have a unique and possibly a historic document.

During the development of the manual we decided to put in much more than may be appropriate for a Level O document with the idea that we could always refine the document later. It was more important to capture the knowledge when we could in order to better position ourselves for later dissemination. If there is any criticism, it may be the level of detail contained in the manual, but this detail is necessary for young engineers. The present document does appear to serve as a good instructional guide, although it does go well beyond its original intent.

As such, this present document is to be considered a next-to-final draft. Your comments, corrections and suggestions are welcomed, valued and appreciated. Please send your remarks directly to Robert Shishko, NASA Systems Engineering Working Group, NASA/Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Drive, Pasadena, CA 91109-8099.

Contents

Introduction ... 1
Purpose ... 1
Scope and Depth ... 1

Fundamentals of Systems Engineering ... 3
Systems, Supersystems, and Subsystems ... 3
Definition of Systems Engineering ... 4
Objective of Systems Engineering ... 4
Disciplines Related to Systems Engineering ... 6
The Doctrine of Successive Refinement ... 6

The Project Cycle for Major NASA Systems ... 13
Pre-Phase A — Advanced Studies ... 13
Phase A — Conceptual Design Studies ... 14
Phase B — Concept Definition ... 14
Phase C — Design and Development ... 18
Phase D — Fabrication, Integration, Test and Certification ... 18
Phase E — Pre-Operations ... 19
Phase F — Operations and Disposal ... 19
Funding: The Budget Cycle ... 20
The Role of Systems Engineering in the Product Development Process ... 20

Management Issues in Systems Engineering ... 27
Harmony of Goals, Work Products and Organizations ... 27
Managing the Systems Engineering Process: The Systems Engineering Management Plan ... 28
Role of the SEMP ... 28
Contents of the SEMP ... 28
Development of the SEMP ... 29
Managing the Systems Engineering Process: Summary ... 30
The Work Breakdown Structure ... 30
Role of the WBS ... 31
Techniques for Developing the WBS ... 31
Common Errors in Developing a WBS ... 32
Scheduling ... 33
Role of Scheduling ... 33
Network Schedule Data and Graphical Formats ... 33
Establishing a Network Schedule ... 34
Reporting Techniques ... 35
Resource Leveling ... 35
Budgeting and Resource Planning ... 37
Risk Management ... 37
Types of Risks ... 39
Risk Identification and Characterization Techniques ... 40
Risk Analysis Techniques ... 41
Risk Mitigation and Tracking Techniques ... 42
Risk Management: Summary ... 44
Baseline Management ... 44

Baseline Evolution 44
Configuration Management 45
Change Control and Version Control 47
Data Management and Requirements Traceability 48
Reviews, Audits and Control Gates 48
Purpose and Definitions 48
General Principles for Reviews 49
Specific Types of Reviews 50
Status Reporting and Assessment 55
Cost and Schedule Control Measures 56
Technical Performance Measures 57
Systems Engineering Process Metrics 60

Systems Analysis and Modeling Issues 63
The Trade Study Process 63
Controlling the Trade Study Process 66
Using Models 67
Selecting the Selection Rule 69
Trade Study Process: Summary 72
Cost Definition and Modeling 73
Life-Cycle Cost (LCC) and Other Cost Measures 73
Controlling Life-Cycle Costs 75
Cost Estimating 76
Effectiveness Definition and Modeling 79
Strategies for Measuring System Effectiveness 79
NASA System Effectiveness Measures 80
Availability and Logistics Supportability Modeling 81
Probabilistic Treatment of Cost and Effectiveness 83
Sources of Uncertainty in Models 83
Modeling Techniques for Handling Uncertainty 84

Appendix A — Acronyms 87

Appendix B — Systems Engineering Templates and Examples 89
Appendix B.1 — A "Tailored" Project Cycle for R&D Projects 89
Appendix B.2 — A Sample SEMP Outline 92
Appendix B.3 — A "Tailored' WBS for an Airborne Telescope 93
Appendix B.4 — A Sample Configuration Management Plan Outline 96
Appendix B.5 — Characterization, Mission Success and SRM&QA Cost Guidelines for Class A–D Payloads 97
Appendix B.6 — An Example of a Critical Items List 98
Appendix B.7 — Techniques of Functional Analysis 99
B.7.1 Functional Flow Block Diagrams 99
B.7.2 N^2 Diagrams 99
B.7.3 Time Line Analysis 101
Appendix B.8 — The Effect of Changes in ORU MTBF on Space Station *Freedom* Operations 104

Appendix C — Use of the Metric System 107
C.1 NASA Policy 107
C.2 Definitions of Units 107
C.2.1 SI Prefixes 107
C.2.2 Base SI Units 107
C.2.3 Supplementary SI Units 108

C.2.4 Derived SI Units with Special Names 108
C.2.5 Units in Use with SI 109
C.3 Conversion Factors 110

Bibliography 113

Index 117

List of Figures

Figure 1 — The Enveloping Surface of Non-dominated Designs. 5
Figure 2 — Estimates of Outcomes to be Obtained from Several Design Concepts Including Uncertainty. 5
Figure 3 — The *Doctrine of Successive Refinement.* 7
Figure 4 — A Quantitative *Objective Function*, Dependent on Life-Cycle Cost and All Aspects of Effectiveness. 10
Figure 5 — The NASA Project Cycle. 15
Figure 6 — Overruns are Very Likely if Phases A and B are Underfunded. 17
Figure 7 — Typical NASA Budget Cycle. 20
Figure 8 — Overview of the Technical Aspect of the NASA Project Cycle. 21
Figure 9 — Technical Aspect of the NASA Project Cycle. 23
Figure 10 — The Relationship Between a System, a Product Breakdown Structure, and a Work Breakdown Structure. 31
Figure 11 — Examples of WBS Development Errors. 32
Figure 12 — Activity-on-Arrow and Precedence Diagrams for Network Schedules. 34
Figure 13 — An Example of a Gantt chart. 36
Figure 14 — An Example of an Unleveled Resource Profile. 37
Figure 15 — Risk Management Structure Diagram. 38
Figure 16 — Configuration Management Structure Diagram. 45
Figure 17 — Contract Change Control Process. 47
Figure 18 — Planning and Status Reporting Feedback Loop. 55
Figure 19 — Cost and Schedule Variances. 56
Figure 20 — Three TPM Assessment Methods. 58
Figure 21 — The Trade Study Process. 63
Figure 22 — Results of Design Concepts with Different Risk Patterns. 70
Figure 23 — Life-Cycle Cost Components. 74
Figure 24 — System Effectiveness Components (Generic). 80
Figure 25 — Roles of Availability and Logistics Supportability Models. 83
Figure 26 — A Monte Carlo Simulation with Three Uncertain Inputs. 85
Figure B-1 — Tailored Project Life Cycle, Advanced Technology Testbed Project. 89
Figure B-2 — Major Products of Generic System Analysis, Advanced Technology Testbed Project. 90
Figure B-3 — Major Products of Generic System Management Activities, Advanced Technology Testbed Project. 91
Figure B-4 — Stratospheric Observatory for Infrared Astronomy (SOFIA) Product Breakdown Structure. 93
Figure B-5 — SOFIA Project WBS (*Level 3*). 94
Figure B-6 — SOFIA Observatory System WBS (*Level 4*). 94
Figure B-7 — SOFIA Airborne Facility WBS (*Level 5*). 95
Figure B-8 — SOFIA Telescope Element WBS (*Level 6*). 95
Figure B-9 — Development of Functional Flow Block Diagrams. 100
Figure B-10 — N^2 Chart Definition. 101
Figure B-11 — N^2 Chart Key Features. 102
Figure B-12 — Flight Mission Time Lines. 103
Figure B-13 — Effect of MTBF on Operations Cost. 104
Figure B-14 — Effect of MTBF on Crew Time. 104
Figure B-15 — Effect of MTBF on Upmass. 105
Figure B-16 — Effect of MTBF on Number of Crew (Available Crew Time Maintained). 105
Figure B-17 — Effect of MTBF on Number of STS Flights (Available Upmass Maintained). 105
Figure B-18 — Effect of MTBF on Five-year Operations Cost (Maintaining vs. Not Maintaining Available Upmass and Crew Time). 106

PAGE iv INTENTIONALLY BLANK

PRECEDING PAGE BLANK NOT FILMED

List of Sidebars

Recommended Reading 1
A Hierarchical System Terminology 3
The Technical Sophistication Required to do Systems Engineering Depends on the Project 3
Systems Engineering per MIL-STD-499B 4
Cost, Effectiveness, and Cost-Effectiveness 4
The System Engineer's Dilemma 6
As an Example of the Process of Successive Refinement, Consider the Choice of Altitude for a Space Station such as *Freedom* 8
Simple Interfaces are Preferred 10
Pre-Phase A — Advanced Studies 14
Phase A — Conceptual Design 14
Phase B — Concept Definition 17
A Credible, Feasible Design 18
Phase C — Design and Development 18
Phase D — Fabrication, Integration, Test and Certification 19
Phase E — Pre-Operations 19
Phase F — Operations and Disposal 20
Multi-Disciplinary Product Development Teams 22
SEMP Lessons Learned from DoD Experience 30
Critical Path and Float Calculation 33
Desirable Features in Gantt Charts 36
Assessing the Effect of Schedule Slippage 37
Risk 38
Probabilistic Risk Assessment Pitfalls 42
An Example of a Decision Tree for Robotic Precursor Missions to Mars 43
Change Control Board Conduct 45
Project Termination 48
Computing the Estimate at Completion 57
Examples of High-Level TPMs for Planetary Spacecraft and Launch Vehicles 57
An Example of the Risk Management Method for Tracking Spacecraft Mass 59
Systems Analysis 63
Functional Analysis Techniques 64
An Example of a Trade Tree for a Mars Rover 66
Trade Study Reports 67
The Analytic Hierarchy Process 71
Multi-Attribute Utility Theory 72
Calculating Present Discounted Value 75
Statistical Cost Estimating Relationships: Example and Pitfalls 77
Learning Curve Theory 78
An Example of a Cost Spreader Function: The Beta Curve 78
Practical Pitfalls in Using Effectiveness Measures in Trade Studies 79
Logistics Supportability Models: Two Examples 81
Measures of Availability 82
The Cost S-Curve 84
Prefixes for SI Units 107

Preface

This handbook was written to bring the fundamental concepts and techniques of systems engineering to NASA personnel in a way that recognizes the nature of NASA systems and the NASA environment. The authors readily acknowledge that this goal will not be easily realized. One reason is that not everyone agrees on what systems engineering is, nor on how to do it. There are legitimate differences of opinion on basic definitions, content, and techniques. Systems engineering itself is a broad subject, with many different aspects. This initial handbook does not (and cannot) cover all of them. The authors fully recognize that perhaps no topic will be covered to the satisfaction of all.

The content and style of this handbook show a teaching orientation. This handbook was meant to accompany formal NASA training courses on systems engineering, not to be a stand-alone, comprehensive view of NASA systems engineering. Systems engineering, in the authors' opinions, cannot be learned simply by starting at a well-defined beginning and proceeding seamlessly from one topic to another. Rather, it is a discipline that draws from many traditional disciplines and intellectual domains. The boundaries are not always clear, and there are many interesting intellectual offshoots. Consequently, this handbook was designed to be a *top-level overview* of systems engineering as a discipline; brevity of exposition and the provision of pointers to other books and documents for details were considered important guidelines.

The material for this handbook was drawn from many different sources, including Center systems engineering handbooks, NASA Management Instructions, Center briefings on systems engineering processes, non-NASA systems engineering textbooks and guides, and three independent systems engineering courses taught to NASA audiences. The handbook uses this material to provide only top-level information and suggestions for good systems engineering practices; it is not intended in any way to be a directive.

By design, the handbook covers some topics that are also taught in Project Management/Program Control (PM/PC) courses, reflecting the unavoidable connectedness of these three domains. The material on the NASA Project Cycle is drawn from the work of the Inter-Center Systems Engineering Working Group (ICSEWG), which met periodically during 1991 to construct a strawman project cycle. Inclusion of this material does not imply that closure was reached on the project cycle; it reflects only the state of that work at the end of 1991.

This handbook consists of four core chapters: (1) systems engineering's intellectual process, (2) the NASA Project Cycle, (3) management issues in systems engineering, and (4) systems analysis and modeling issues. These core chapters are supplemented by appendices, which can be expanded to accommodate any number of templates and examples to illustrate topics in the core chapters. The handbook makes extensive use of sidebars to define, refine, illustrate, and extend concepts in the core chapters without diverting the reader from the main argument. There are no footnotes; sidebars are used instead. The structure of the handbook also allows for additional sections and chapters to be added at a later date. The authors in fact are planning an additional core chapter on the techniques used in specialty engineering disciplines.

Acknowledgements

The principal authors would like to thank Mr. Frank Hoban, NASA Headquarters/Code FT, for his steadfast financial and intellectual support for this effort, and Dr. Shahid Habib, NASA Headquarters/Code QE, for the additional financial support to the NASA Inter-Center Systems Engineering Working Group's project cycle work. The principal authors would also like to acknowledge the participation of many other individuals who served as contributing authors and reviewers or who shared their systems engineering ideas and material with us during this effort. The principal authors, however, accept all responsibility for the content of this handbook.

As contributing authors:
Mr. Robert Aster, NASA/Jet Propulsion Laboratory
Dr. Vincent Bilardo, NASA/Ames Research Center
Dr. Kevin Forsberg, Center for Systems Management
Dr. Walter E. Hammond, Sverdrup Technology, Inc.
Mr. Harold Mooz, Center for Systems Management
Mr. Lou Polaski, Center for Systems Management
Mr. Ron Wade, Center for Systems Management

As reviewers and commenters:
Mr. Robert C. Baumann, NASA/Goddard Space Flight Center
Mr. Chris Carl, NASA/Jet Propulsion Laboratory
Dr. David S.C. Chu, Assistant Secretary of Defense/Program Analysis and Evaluation
Dr. Frank Fogle, NASA/Marshall Space Flight Center
Mr. John L. Gasery, Jr., NASA/Stennis Space Center
Mr. Don Hedgepeth, NASA/Langley Research Center
Dr. Jerry Lake, Defense Systems Management College
Dr. Brian Mar, Department of Civil Engineering, University of Washington
Mr. Bernard G. Morais, Synergistic Applications, Inc.
Mr. Raymond L. Nieder, NASA/Johnson Space Center
Mr. David Pine, NASA Headquarters/Code B
Ms. Donna Shirley Pivirotto, NASA/Jet Propulsion Laboratory
Mr. Glen D. Ritter, NASA/Marshall Space Flight Center
Mr. Gerald Sadler, NASA/Lewis Research Center
Mr. Dick Smart, Sverdrup Technology, Inc.
Mr. Lanny Taliaferro, NASA/Marshall Space Flight Center
Dr. Arnold Ruskin, NASA/Jet Propulsion Laboratory and University of California at Los Angeles
Mr. L. Don Wodruff, NASA/Marshall Space Flight Center

As general contributors:
Mr. William Edmiston, NASA/Jet Propulsion Laboratory
Dr. Jairus Hihn, NASA/Jet Propulsion Laboratory
Dr. Ed Jorgenson, NASA/Jet Propulsion Laboratory
Mr. William C. Morgan, NASA/Johnson Space Center
Mr. Richard V. Morris, NASA/Jet Propulsion Laboratory

For editorial and graphics support:
Mr. Randy Cassingham, NASA/Jet Propulsion Laboratory
Mr. John Matlock, NASA/Jet Propulsion Laboratory

1 Introduction

1.1 Purpose

This handbook is intended to provide information on systems engineering that will be useful to NASA system engineers, especially new ones. Its primary objective is to provide a generic description of systems engineering as it *should be* applied throughout NASA. Field Centers' handbooks are encouraged to provide Center-specific details of implementation.

For NASA system engineers to choose to keep a copy of this handbook at their elbows, it must provide answers that cannot be easily found elsewhere. Consequently, it provides NASA-relevant perspectives and NASA-particular data. NASA management instructions (NMIs) are referenced when applicable.

This handbook's secondary objective is to serve as a useful companion to all of the various courses in systems engineering that are being offered under NASA's auspices.

Recommended Reading

See the Bibliography for full reference data and further reading suggestions.

Fundamentals of Systems Engineering
Systems Engineering and Analysis (2nd ed.), B.S. Blanchard and W.J. Fabrycky.

Management Issues in Systems Engineering
Systems Engineering, MIL-STD-499B.
Systems Engineering Management Guide, Defense Systems Management College.
Systems Engineering Management, B.S. Blanchard.
Systems Engineering Methods, Harold Chestnut.
Systems Concepts, Ralph Miles, Jr. (editor).

Systems Analysis and Modeling
Systems Engineering Tools, Harold Chestnut.
Systems Analysis for Engineers and Managers, R. de Neufville and J.H. Stafford.

Space Systems Design
Space Vehicle Design, Michael D. Griffin and James R. French.
Space Mission Analysis and Design, James R. Wertz and Wiley J. Larson (editors).
Design of Geosynchronous Spacecraft, Brij N. Agrawal.

1.2 Scope and Depth

The subject matter of systems engineering is very broad. The coverage in this handbook is limited to general concepts and generic descriptions of processes, tools, and techniques. It provides information on good systems engineering practices, and pitfalls to avoid. There are many textbooks that can be consulted for in-depth tutorials.

This handbook describes systems engineering as it should be applied to the development of major NASA systems. Systems engineering applies both to the system being developed *(the product system)* and to the system that does the developing *(the producing system)*. Consequently, the handbook's scope properly includes systems engineering functions regardless of whether they are performed by an in-house systems engineering organization, a program/project office, or a system contractor.

While many of the producing system's design features may be implied by the nature of the tools and techniques of systems engineering, it does not follow that institutional procedures for their application must be uniform from one NASA Center to another.

2 Fundamentals of Systems Engineering

2.1 Systems, Supersystems, and Subsystems

A *system* is a set of interrelated components which interact with one another in an organized fashion toward a common purpose. The components of a system may be quite diverse, consisting of persons, organizations, procedures, software, equipment and/or facilities. The purpose of a system may be as humble as distributing electrical power within a spacecraft or as grand as exploring the surface of Mars.

A Hierarchical System Terminology

The following hierarchical sequence of terms for successively finer resolution was agreed upon by the NASA Inter-Center Systems Engineering Working Group:

Program
Project
System
Segment
Element
Subsystem
Assembly
Subassembly
Part

The word *system* is used within NASA both generically, as defined in the text, and rather specifically, as the level of resolution below *project*. Which use is intended is generally clear from context.

Every system exists in the context of a *supersystem*, which has a broader scope. It is in that context that the system must be judged. Thus, managers in the supersystem set system policies, establish system objectives, determine system constraints, and define what costs are relevant. They often have oversight authority over system design and operations decisions.

Most NASA systems are sufficiently complex that their components are *subsystems*, which must function in a coordinated way for the system to accomplish its goals. From the point of view of systems engineering, each subsystem is a system in its own right — that is, policies, requirements, objectives, and which costs are relevant are established at the next level up in the hierarchy. Spacecraft systems often have such subsystems as *propulsion*, *attitude control*, *telecommunications*, and *power*. In a large project, the subsystems are likely to be called "systems".

The word *system* is used within NASA both generically, as defined in the first paragraph above, and rather specifically, as the level of resolution below *project*. In this handbook, the word "system" is generally used in its generic form.

The NASA management instruction for the acquisition of "major" systems (NMI 7100.14B) defines a *program* as "an organized set of activities directed toward a common purpose, objective, or goal undertaken or proposed by an agency in order to carry out responsibilities which have been assigned to it." The similarity to the above definition of *system* is not accidental.

In the NASA context, a *project* encompasses the design, acquisition and operation of a major system, and is generally managed by a NASA field Center. A *program*, on the other hand, is what NASA Headquarters manages, and may encompass several projects. Headquarters' management concerns include not only the engineering of the system, but all of the other activities required to achieve the desired end. These other activities include explaining the value of the system to Congress and enlisting interna-

The Technical Sophistication Required to do Systems Engineering Depends on the Project

- The system's goals may be simple and easy to identify and measure — or they may be technically complicated, requiring a great deal of insight about the environment or technology within or with which the system must operate.
- The system may have a single goal — or multiple goals. There are techniques available for determining the relative values of multiple goals — but sometimes goals are truly incommensurate and unquantifiable.
- The system may have users representing factions with conflicting objectives. When there are conflicting objectives, negotiated compromises will be required.
- Alternative system design concepts may be abundant — or they may require creative genius to develop.
- A "back-of-the-envelope" computation may be satisfactory for prediction of how well the alternative design concepts would do in achievement of the goals — or credibility may depend upon construction and testing of hardware or software models.
- The desired ends usually include an optimization objective, such as "minimize life-cycle cost" or "maximize the value of returned data", so selection of the best design may not be an easy task.

PAGE 2 INTENTIONALLY BLANK
PRECEDING PAGE BLANK NOT FILMED

tional cooperation. The term *mission* is often used for the system's purpose; its connotations of fervor make it particularly suitable for such political activities, where the emotional content of the term is a desirable factor.

2.2 Definition of Systems Engineering

Systems engineering is a robust approach to the design, creation, and operation of systems. In simple terms, the approach consists of identification and quantification of system goals, creation of alternative system design concepts, performance of design trades, selection and implementation of the best design, verification that the design is actually built and properly integrated, and post-implementation assessment of how well the system meets (or met) the goals. The approach is usually applied repeatedly and recursively, with several increases in the resolution of the system baselines (which contain requirements, design details, verification procedures and standards, cost and performance estimates, and so on).

Systems Engineering per MIL-STD-499B

Systems engineering is "an interdisciplinary approach to evolve and verify an integrated and life-cycle balanced set of system product and process solutions that satisfy customer needs. Systems engineering: (a) encompasses the scientific and engineering efforts related to the development, manufacturing, verification, deployment, operations, support, and disposal of system products and processes, (b) develops needed user training equipments, procedures, and data, (c) establishes and maintains configuration management of the system, (d) develops work breakdown structures and statements of work, and (e) provides information for management decision making."

Systems engineering is performed in concert with system management. A major part of the system engineer's role is to provide information that the system manager can use to make the right decisions. This includes identification of alternative design concepts and characterization of those concepts in ways that will help the system managers first discover their preferences, then be able to apply them astutely. An important aspect of this role is the creation of system models that facilitate assessment of the alternatives in various dimensions like cost, performance, and risk.

Application of this approach includes performance of some delegated management duties, such as maintaining control of the developing configuration and overseeing the integration of subsystems.

2.3 Objective of Systems Engineering

The objective of systems engineering is to see to it that the system is designed, built, and operated so that it accomplishes its purpose in the most *cost-effective* way possible, considering performance, cost, schedule, and risk.

A cost-effective system must provide a particular kind of balance between effectiveness and cost: the system must provide the most effectiveness for the resources expended or, equivalently, it must be the least expensive for the effectiveness it provides. This condition is a weak one because there are usually many designs that meet the condition. Think of each possible design as a point in the tradeoff space between effectiveness and cost. A graph plotting the *maximum* achievable effectiveness of designs

Cost

The cost of a system is the foregone value of the resources needed to design, build, and operate it. Because resources come in many forms — work performed by NASA personnel and contractors, materials, energy, and the use of facilities and equipment such as wind tunnels, factories, offices, and computers — it is often convenient to express these values in common terms by using monetary units (such as dollars).

Effectiveness

The effectiveness of a system is a quantitative measure of the degree to which the system's purpose is achieved. Effectiveness measures are usually very dependent upon system performance. For example, launch vehicle effectiveness depends on the probability of successfully injecting a payload onto a usable trajectory. The associated system performance attributes include the mass that can be put into a specified nominal orbit, the trade between injected mass and launch velocity, and launch availability.

Cost-Effectiveness

The cost-effectiveness of a system combines both the cost and the effectiveness of the system in the context of its objectives. While it may be necessary to measure either or both of those in terms of several numbers, it is sometimes possible to combine the components into a meaningful, single-valued *objective function* for use in design optimization. Even without knowing how to trade effectiveness for cost, designs that have lower cost and higher effectiveness are preferred.

available with current technology as a function of cost would in general yield a curved line such as the one shown in Figure 1. (In the figure, all the dimensions of effectiveness are represented by the ordinate and all the dimensions of cost by the abscissa.) In other words, the curved line represents the envelope of the currently available technology in terms of cost-effectiveness.

Points above the line cannot be achieved with currently available technology — that is, they do not represent feasible designs. (Some of those points may be feasible *in the future* when further technological advances have been made.) Points inside the envelope are feasible, but are dominated by designs whose combined cost and effectiveness lie *on* the envelope. Designs represented by points on the envelope are called cost-effective (or efficient or non-dominated) solutions.

Design trade studies, an important part of the systems engineering process, often attempt to find designs that provide a better combination of the various dimensions of cost and effectiveness. When the starting point for a design trade study is inside the envelope, there are alternatives that reduce costs without decreasing any aspect of effectiveness, or increase some aspect of effectiveness without decreasing others and without increasing costs. Then, the system manager's or system engineer's decision is

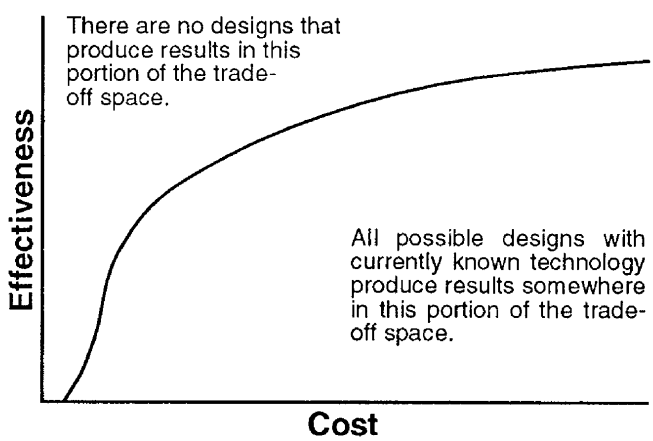


Figure 1 — The Enveloping Surface of Non-dominated Designs.

easy. Other than in the sizing of subsystems, such "win-win" design trades are uncommon, but by no means rare. When the alternatives in a design trade study, however, require trading cost for effectiveness, or even one dimension of effectiveness for another at the same cost, the decisions become harder.

The process of finding the most cost-effective design is further complicated by uncertainty, which is shown

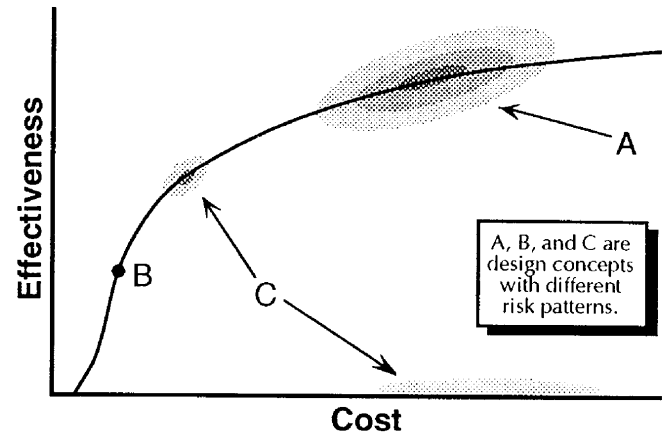


Figure 2 — Estimates of Outcomes to be Obtained from Several Design Concepts Including Uncertainty.

in Figure 2 as a modification of Figure 1. Exactly what outcomes will be realized by a particular system design cannot be known in advance with certainty, so the projected cost and effectiveness of a design are better described by a probability distribution than by a point. This distribution can be thought of as a cloud which is thickest at the most likely value and thinner farther away from the most likely point, as is shown for design concept A in the figure. Distributions resulting from designs which have little uncertainty are dense and highly compact, as is shown for concept B. Distributions associated with *risky* designs may have significant probabilities of producing highly undesirable outcomes, as is suggested by the presence of an additional low effectiveness/high cost cloud for concept C. (Of course, the envelope of such clouds cannot be a sharp line such as is shown in the figures, but must itself be rather fuzzy. The line can now be thought of as representing the envelope at some fixed confidence level — that is, a probability of x of achieving that effectiveness.)

Both *effectiveness* and *cost* may require several descriptors. Even the *Echo* balloons obtained scientific data on the electromagnetic environment and atmospheric drag, in addition to their primary mission as communications satellites. Furthermore, *Echo* was the first satellite visible to the naked eye, an unquantified — but not unrecognized — aspect of its effectiveness. Costs, the expenditure of limited resources, may be measured in the several dimensions of funding, personnel, use of facilities, and so on. *Schedule* may appear as an attribute of effectiveness or cost, or as a constraint. *Sputnik,* for example, drew much of its effectiveness from the fact that it was a "first", while a planetary launch to Venus that missed its launch

window used to have to wait two years for another opportunity. *Risk* results from uncertainties in realized effectiveness, costs, timeliness and budgets.

Sometimes, the systems that provide the highest ratio of effectiveness to cost are the most desirable. However, this ratio is likely to be meaningless or — worse — misleading. To be useful and meaningful, that ratio must be uniquely determined and independent of the system cost. Further, there must be but a single measure of effectiveness and a single measure of cost. If the numerical values of those metrics are obscured by probability distributions, the ratios become uncertain as well; then any usefulness the simple, single ratio of two numbers might have had disappears.

The System Engineer's Dilemma

- To reduce cost at constant risk, performance must be reduced.
- To reduce risk at constant cost, performance must be reduced.
- To reduce cost at constant performance, higher risks must be accepted.
- To reduce risk at constant performance, higher costs must be accepted.

In this context, time in the schedule is often a critical resource, so that *schedule* behaves like a kind of *cost*.

In some contexts, it is appropriate to seek the most effectiveness possible within a fixed budget; in other contexts, it is more appropriate to seek the least cost possible with specified effectiveness. In these cases, there is the question of what level of effectiveness to specify or of what level of costs to fix. In practice, these may be mandated in the form of performance or cost requirements; it then becomes appropriate to ask whether a slight relaxation of requirements could produce a significantly cheaper system or whether a few more resources could produce a significantly more effective system.

Usually, the system manager must choose among designs that differ in terms of numerous attributes. A variety of methods have been developed that can be used to help managers uncover their preferences between attributes and to quantify their subjective assessments of relative value. When this can be done, trades between attributes can be assessed quantitatively. Often, however, the attributes seem to be truly incommensurate; managers must make their decisions in spite of this multiplicity.

2.4 Disciplines Related to Systems Engineering

The definition of systems engineering given in Section 2.2 could apply to the design task facing a bridge designer, a radio engineer, or even a committee chair. The systems engineering process *can be* a part of all of these. It cannot be the whole of the job — the bridge designer must know the properties of concrete and steel, the radio engineer must apply Maxwell's equations, and a committee chair must understand the personalities of the members of the committee. In fact, the optimization of systems requires collaboration with experts in a variety of disciplines, some of which are compared to systems engineering in the remainder of this section.

The role of systems *engineering* differs from that of system *management* in that engineering is an analytical, advisory and planning function, while management is the decision-making function. Very often, the distinction is irrelevant, as the same individuals may perform both roles. When no factors enter the decision-making process other than those that are covered by the analyses, system management may delegate some of the management responsibility to the systems engineering function.

Systems engineering differs from what might be called *design* engineering in that systems engineering deals with the relationships of the thing being designed to its supersystem and subsystems, rather than with the internal details of how it is to accomplish its objectives. The systems viewpoint is broad, rather than deep: it encompasses the system from architect to user, from mission objectives to design details, and from cradle to grave.

System engineers must also rely on contributions from the *specialty engineering* disciplines, in addition to the traditional design disciplines, for functional expertise and specialized analytic methods. These specialty engineering areas typically include reliability, maintainability, logistics, test, production, transportation, human factors, quality assurance, and safety engineering. Specialty engineers contribute throughout the systems engineering process; part of the system engineer's job is to see that these functions are coherently integrated into the project at the right times and that they address the relevant issues.

In both systems *analysis* and systems *engineering*, the amounts and kinds of resources to be made available for the creation of the system are assumed to be among the decisions to be made. Systems engineering concentrates on the creation of hardware and software architectures and on the development and management of the interfaces between subsystems, relying on systems analysis to construct the mathematical models and analyze the data to evaluate alternative designs and to perform the actual design trade-off studies. Systems analysis often requires the use of

tools from operations research, economics, or other so-called *decision sciences*, and systems analysis curricula generally include extensive study of such topics as probability, statistics, decision theory, queueing theory, game theory, linear and non-linear programming, and so on. In practice, many system engineers' academic background is richer in the engineering disciplines than in the decision sciences. As a consequence, the system engineer is often a consumer of systems analysis products, rather than a producer of them. One of the major objectives for Chapter 5 is to develop an understanding and appreciation of the state of that art.

Operations research and *operations engineering* confine their attention to systems whose components are assumed to be more or less immutable. That is, it is assumed that the resources with which the system operates cannot be changed, but that the way in which they are *used* is amenable to optimization. Operations research techniques often provide powerful tools for the optimization of system designs.

Within NASA, terms such as *mission analysis and engineering* are often used to describe all study and design efforts that relate to determination of what the project's mission should be and how it should be carried out. Sometimes the scope is limited to the study of future projects. Sometimes the charters of organizations with such names include monitoring the capabilities of systems, ensuring that important considerations have not been overlooked, and overseeing tradeoffs between major systems — thereby encompassing operations research, systems analysis, and systems engineering activities.

Total quality management (TQM) is the application of systems engineering to the work environment. That is, part of the total quality management paradigm is the realization that an operating organization is a particular kind of system and should be engineered as one. A variety of specialized tools have been developed for this application area; many of them can be recognized as established systems engineering tools, but with different names. The injunction to focus on the satisfaction of customer needs, for example, is even expressed in similar terms. The use of statistical process control is akin to the use of technical performance and earned value measurements. *Quality function deployment* is a technique of requirements analysis.

The *systems approach* is common to all of these related fields. Essential to the systems approach is the recognition that a system exists, that it is embedded in a supersystem on which it has an impact, that it may contain subsystems, and that the system's objectives must be understood — preferably explicitly identified.

2.5 The Doctrine of Successive Refinement

The realization of a system over its life cycle results from a succession of decisions among alternative courses of action. If the alternatives are precisely enough defined and thoroughly enough understood to be well differentiated in the cost-effectiveness space, then the system manager can make choices among them with confidence.

The systems engineering process can be thought of as the pursuit of definition and understanding of design alternatives to support those decisions, coupled with the overseeing of their implementation. To obtain assessments that are crisp enough to facilitate good decisions, it is often necessary to delve more deeply into the space of possible designs than has yet been done, as is illustrated in Figure 3.

It should be realized, however, that this spiral represents neither the project cycle, which encompasses the system from inception through disposal, nor the product development process by which the system design is developed and implemented, which occurs in Phases C and D (see Chapter 3) of the project cycle. Rather, as the intellectual process of systems engineering, it is inevitably reflected in both of them.

Figure 3 is really a double helix — each *create concepts* step at the level of design engineering initiates a ca-

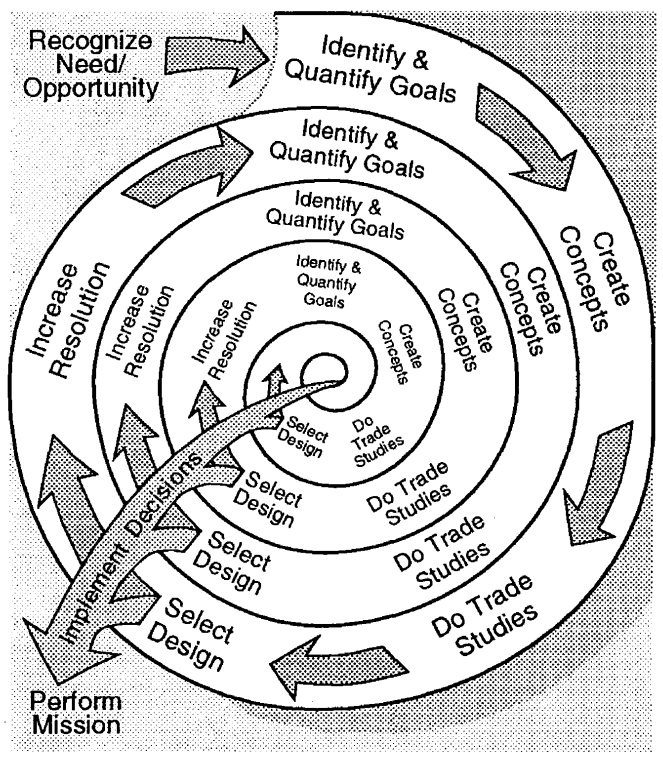


Figure 3 — The *Doctrine of Successive Refinement.*

pabilities definition spiral moving in the opposite direction. The concepts can never be created from whole cloth. Rather, they result from the synthesis of potential capabilities offered by the continually changing state of technology. This process of design concept development by the integration of lower-level elements is a part of the systems engineering process. In fact, there is always a danger that the top-down process cannot keep up with the bottoms-up process.

As an Example of the Process of Successive Refinement, Consider the Choice of Altitude for a Space Station such as *Freedom*

- The first issue is selection of the general location. Alternatives include Earth orbit, one of the Earth-Moon Lagrange points, or a solar orbit. At the current state of technology, cost and risk considerations made selection of Earth orbit an easy choice for *Freedom*.
- Having chosen Earth orbit, it is necessary to select an orbit region. Alternatives include low Earth orbit (LEO), high Earth orbit and geosynchronous orbit; orbital inclination and eccentricity must also be chosen. One of many criteria considered in choosing LEO for *Freedom* was the design complexity associated with passage through the Van Allen radiation belts.
- System design choices proceed to the selection of an altitude maintenance strategy — rules that implicitly determine when, where and why to reboost, such as "maintain altitude such that there are always at least *TBD* days to reentry", "collision avoidance maneuvers shall always increase the altitude", "reboost only after resupply flights that have brought fuel", "rotate the crew every *TBD* days".
- A next step is to write altitude specifications. These choices might consist of replacing the *TBD*s (values *to be determined*) in the altitude strategy with explicit numbers.
- Monthly operations plans are eventually part of the complete system design. These would include scheduled reboost burns based on predictions of the accumulated effect of drag and the details of on-board microgravity experiments.
- Actual firing decisions are based on determinations of the orbit which results from the momentum actually added by previous firings, the atmospheric density variations actually encountered, and so on.

Note that decisions at every step require that the capabilities offered by available technology be considered — often at levels of design that are more detailed than seems necessary at first.

There is often an early need to resolve the issues (such as the system architecture) enough so that the system can be modeled with sufficient realism to do reliable trade studies.

When resources are expended toward the implementation of one of several design options, the resources required to complete the implementation of that design decrease (of course), while there is usually little or no change in the resources that would be required by unselected alternatives. Selected alternatives thereby become relatively even more attractive than those that were not selected.

Consequently, it is reasonable to expect the system to be defined with increasingly better resolution as time passes. This tendency is formalized at some point (in Phase B) by defining a *baseline* system definition. Usually, the goals, objectives, and constraints are baselined as the *requirements* portion of the baseline. The entire baseline is then subjected to configuration control in an attempt to ensure that successive changes are indeed improvements.

As the system is realized, its particulars become clearer — but also harder to change. As stated above, the purpose of systems engineering is to make sure that the development process happens in a way that leads to the most cost-effective final system. The basic idea is that before those decisions that are hard to undo are made, the alternatives are carefully assessed.

The systems engineering process is applied again and again as the system is developed. As the system is realized, the issues addressed evolve and the particulars of the activity change.

Most of the major system decisions (goals, architecture, acceptable life-cycle cost, etc.) are made during the early phases of the project, so the turns of the spiral (that is, the *successive refinements*) do not correspond precisely to the phases of the system life cycle. Much of the system architecture can be "seen" even at the outset, so the turns of the spiral do not correspond exactly to development of the architectural hierarchy, either. Rather, they correspond to the successively greater resolution by which the system is defined.

Each of the steps in the systems engineering process is discussed below.

Recognize Need/Opportunity. This step is shown in Figure 3 only once, as it is not really part of the spiral but its first cause. It could be argued that recognition of the need or opportunity for a new system is an entrepreneurial activity, rather than an engineering one.

The end result of this step is the discovery and delineation of the system's goals, which generally express the desires and requirements of the eventual users of the system. In the NASA context, the system's goals should also represent the long term interests of the taxpaying public.

Identify and Quantify Goals. Before it is possible to compare the cost-effectiveness of alternative system design concepts, the *mission* to be performed by the system must be delineated. The goals that are developed should cover all relevant aspects of effectiveness, cost, schedule and risk, and should be traceable to the goals of the supersystem. To make it easier to choose among alternatives, the goals should be stated in quantifiable, verifiable terms, insofar as that is possible and meaningful to do.

It is also desirable to assess the constraints that may apply. Some constraints are imposed by the state of technology at the time of creating or modifying system design concepts. Others may appear to be inviolate, but can be changed by higher levels of management. The assumptions and other relevant information that underlie constraints should always be recorded so that it is possible to estimate the benefits that could be obtained from their relaxation.

At each turn of the spiral, the goals should be documented in a way that makes them traceable to the next higher level. As the systems engineering process continues, the system's goals become documented as *functional* requirements (what must be done to achieve those goals) and as *performance* requirements (quantitative descriptions of how well the functional requirements must be done). In later turns of the spiral, further elaborations may become documented as detailed specifications or design requirements.

Create Alternative Design Concepts. Once it is understood what the system is to accomplish, it is possible to devise a variety of ways that those goals can be met. Sometimes, that comes about as a consequence of integrating available subsystem design options. Ideally, as wide a range of plausible alternatives as is consistent with the design organization's charter should be defined, keeping in mind the current stage in the process of successive refinement. When the bottoms-up process is operating, a problem for the system engineer is that the designers tend to become fond of the designs they create, so they lose their objectivity; the system engineer often must stay an "outsider" so that there is more objectivity.

On the first turn of the spiral in Figure 3, the subject is often general approaches or strategies, sometimes architectural concepts. On the next, it is likely to be functional design, then detailed design, and so on.

The reason for avoiding a premature focus on a single design is to permit discovery of the truly best design. Part of the system engineer's job is to ensure that the design concepts to be compared take into account all interface requirements. "Did you include the cabling?" is a characteristic question. When possible, each design concept should be described in terms of controllable *design parameters* so that each represents as wide a class of designs as is reasonable. In doing so, the system engineer should keep in mind that the potentials for change may include organizational structure, schedules, procedures, and any of the other things that make up a *system*. When possible, constraints should also be described by parameters.

Owen Morris, former Manager of the *Apollo* Spacecraft Program and Manager of Space Shuttle Systems and Engineering, has pointed out that it is often useful to define *design reference missions* which stress all of the system's capabilities to a significant extent and which all designs will have to be able to accomplish. The purpose of such missions is to keep the design space open. Consequently, it can be very dangerous to write them into the system specifications, as they can have just the opposite effect.

Do Trade Studies. Trade studies begin with an assessment of how well each of the design alternatives meets the system goals (effectiveness, cost, schedule, and risk, both quantified and otherwise). The ability to perform these studies is enhanced by the development of system models that relate the design parameters to those assessments — but it does not depend upon them.

Controlled modification and development of design concepts, together with such system models, often permits the use of formal optimization techniques to find regions of the design space that warrant further investigation — those that are closer to the optimum surface indicated in Figure 1.

Whether system models are used or not, the design concepts are developed, modified, reassessed and compared against competing alternatives in a closed-loop process that seeks the best choices for further development. System and subsystem sizes are often determined during the trade studies. The end result is the determination of bounds on the relative cost-effectivenesses of the design alternatives, measured in terms of the quantified system goals. (Only bounds, rather than final values, are possible because determination of the final details of the design is intentionally deferred. The bounds, in turn, may be derived from the probability density functions.) Increasing detail associated with the continually improving resolution reduces the spread between upper and lower bounds as the process proceeds.

Select Concept. Selection among the alternative design concepts is a task for the system manager, who must take into account the subjective factors that the system engineer was unable to quantify, in addition to the estimates of how well the alternatives meet the quantified goals (and any effectiveness, cost, schedule, risk or other constraints).

When it is possible, it is usually well worth the trouble to develop a mathematical expression, called an *objective function*, that expresses the values of combinations of possible outcomes as a single measure of cost-effectiveness, as is illustrated in Figure 4, even if both cost and effectiveness must be described by more than one measure. When achievement of the goals can be quantitatively expressed by such an objective function, designs can be compared in terms of its value. Risks associated with design concepts can cause these evaluations to be somewhat nebulous (because they are uncertain and are best described by probability distributions). In this illustration, the risks are relatively high for design concept A. There is little risk in either effectiveness or cost for concept B, while the risk of an expensive failure is high for concept C, as is shown by the cloud of probability near the x axis with a high cost and essentially no effectiveness. Schedule factors may affect the effectiveness values, the cost values and the risk distributions.

The mission success criteria for systems differ significantly. In some cases, effectiveness goals may be much more important than all others. Other projects may demand low costs, have an immutable schedule, or require minimization of some kinds of risks. Rarely (if ever) is it possible to produce a combined quantitative measure that relates *all* of the important factors, even if it is expressed as a vector with several components. Even when that can be done, it is essential that the underlying factors and relationships be thoroughly revealed to and understood by the system manager. The system manager must weigh the importance of the unquantifiable factors along with the quantitative data provided by the system engineer.

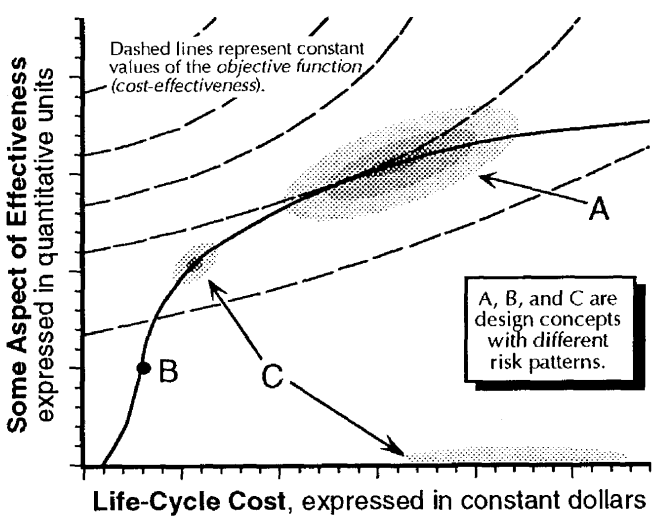


Figure 4 — A Quantitative *Objective Function*, Dependent on Life-Cycle Cost and All Aspects of Effectiveness.

Technical reviews of the data and analyses are an important part of the decision support packages prepared for the system manager. The decisions that are made are generally entered into the configuration management system as changes to (or elaborations of) the system baseline. The supporting trade studies are archived for future use. An essential feature of the systems engineering process is that trade studies are performed *before* decisions are made. They can then be baselined with much more confidence.

At this point in the systems engineering process, there is a logical branch point. For those issues for which the process of successive refinement has proceeded far enough, the next step is to implement the decisions at that level of resolution (that is, unwind the recursive process). For those issues that are still insufficiently resolved, the next step is to refine the development further.

Increase the Resolution of the Design. One of the first issues to be addressed is how the system should be subdivided into subsystems. (Once that has been done, the focus changes and the *subsystems* become *systems* — from the point of view of a system engineer. The partitioning process stops when the subsystems are simple enough to be managed holistically.) As noted by Morris, "the division of program activities to minimize the number and complexity of interfaces has a strong influence on the overall program cost and the ability of the program to meet schedules."

> **Simple Interfaces are Preferred**
>
> According to Morris, NASA's former Acting Administrator George Low, in a 1971 paper titled "What Made *Apollo* a Success", noted that only 100 wires were needed to link the *Apollo* spacecraft to the Saturn launch vehicle. He emphasized the point that a single person could fully understand the interface and cope with all the effects of a change on either side of the interface.

Charles Leising and Arnold Ruskin have (separately) pointed out that partitioning is more art than science, but that there are guidelines available: To make interfaces clean and simple, similar functions, designs and technologies should be grouped. Each portion of work should

be verifiable. Pieces should map conveniently onto the organizational structure. Some of the functions that are needed throughout the design (such as *electrical power*) or throughout the organization (such as *purchasing*) can be centralized. Standardization — of such things as parts lists or reporting formats — is often desirable. The accounting system should follow (not lead) the system architecture. In terms of breadth, partitioning should be done essentially all at once. As with system design choices, alternative partitioning plans should be considered and compared before implementation.

If a requirements-driven design paradigm is used for the development of the system architecture, it must be applied with care, for the use of "shalls" creates a tendency for the requirements to be treated as inviolable constraints rather than as agents of the objectives. A goal, objective or desire should never be made a *requirement* until its costs are understood and the buyer is willing to pay for it. The capability to compute the effects of lower-level decisions on the *quantified goals* should be maintained throughout the partitioning process. That is, there should be a *goals flowdown* embedded in the requirements allocation process.

The process continues with creation of a variety of alternative design concepts at the next level of resolution, construction of models that permit prediction of how well those alternatives will satisfy the quantified goals, and so on. It is imperative that plans for subsequent integration be laid throughout the partitioning. Integration plans include verification and validation activities as a matter of course.

Implement the Selected Design Decisions. When the process of successive refinement has proceeded far enough, the next step is to reverse the partitioning process. When applied to the system architecture, this "unwinding" of the process is called *system integration*. *Conceptual* system integration takes place in all phases of the project cycle. That is, when a design approach has been selected, the approach is verified by "unwinding the process" to test whether the concept at each physical level meets the expectations and requirements. *Physical* integration is accomplished during Phase D. At the finer levels of resolution, pieces must be tested, assembled and/or integrated, and tested again. The system engineer's role includes the performance of the delegated management duties, such as configuration control and overseeing the integration, verification, and validation process.

The purpose of *verification* of subsystem integration is to ensure that the subsystems conform to what was designed and interface with each other as expected in all respects that are important: mechanical connections, effects on center of mass and products of inertia, electromagnetic interference and connector impedance and voltage, power consumption, data flow, and so on. *Validation* consists of ensuring that the interfaced subsystems achieve their intended results. While validation is even more important than verification, it is usually much more difficult to accomplish.

Perform the Mission. Eventually, the system is called upon to meet the need or seize the opportunity for which it was designed and built.

The system engineer continues to perform a variety of supporting functions, depending on the nature and duration of the mission. On a large project such as Space Station *Freedom*, some of these continuing functions include the validation of system effectiveness at the operational site, overseeing the maintenance of configuration and logistics documentation, overseeing sustaining engineering activities, compiling development and operations "lessons learned" documents and, with the help of the specialty engineering disciplines, identifying product improvement opportunities. On smaller systems, such as a *Spacelab* payload, only the last two may be needed.

3 The Project Cycle for Major NASA Systems

One of the fundamental concepts used within NASA for the management of major systems is the *project cycle*, which consists of a categorization of everything that should be done to accomplish a project into distinct *phases*, separated by *control gates*. Phase boundaries are defined so that they provide more-or-less natural points for go/no-go decisions. Decisions to proceed may be qualified by *liens* that must be removed within a reasonable time. A project that fails to pass a control gate and has enough resources may be allowed to "go back to the drawing board" — or it may be terminated.

NASA management instructions (NMI 7100.14B) define the phases of a major system acquisition as:

- Phase A — Preliminary Analysis
- Phase B — Definition
- Phase C/D — Design, Full-Scale Development, Operation.

When considered in the context of phased project planning designed to encompass the entire life-cycle of a system, this list is rather truncated. One reason is that *acquisition* activities (which bound the scope of the NMI) do not include the pre-proposal part of the process, and tend to emphasize the remaining early phases to the exclusion of the later portions of the life-cycle. In the NASA context, operations are often treated as a new beginning — sometimes even the name of the project is changed (e.g., the *Mariner-Jupiter-Saturn 77* project became *Voyager* after the spacecraft were on their way).

Another reason the above list differs from the description which is about to follow is that the *product development process* consists of both the decomposition and definition of Phase C and the fabrication, integration and verification of Phase D. Barry W. Boehm described how several contemporary software development processes work; in some of these processes, the development and construction activities proceed in parallel, so that attempting to separate the associated phases on a time line is undesirable. Boehm describes a spiral which reflects the doctrine of successive refinement depicted in Figure 3, but Boehm's spiral describes the software product development process in particular. His discussion applies as well to the development of hardware products as it does to software.

All systems start with the recognition of a need or the discovery of an opportunity and proceed through various stages of development to a final disposition. While the most dramatic impacts of the analysis and optimization activities associated with systems engineering are obtained in the early stages, decisions that affect millions of dollars of value or cost continue to be amenable to the systems approach even as the end of the system lifetime approaches.

Generically, the phases can be categorized as:

- Pre-Phase A — Find a suitable project
- Phase A — Make sure the project is worthwhile
- Phase B — Define the project
- Phase C — Develop the system design
- Phase D — Build, integrate, test and certify the system
- Phase E — Prepare for operations
- Phase F — Operate the system and dispose of it properly

Sections 3.1 to 3.7 contain narrative descriptions of the purposes, major activities and products, and control gates that characterize the phases, and are based on workshops conducted by the NASA Inter-Center Systems Engineering Working Group. Figure 5 (foldout, next page) details the activities, products and control gates resulting from the workshops. Section 3.9 provides a more concentrated discussion of the role of systems engineering in the process.

The particular categorization of project phases described here need not be adhered to slavishly — that is, project phases can be tailored (see Appendix B.1). In particular, it is sometimes appropriate to perform some long-lead-time activities ahead of the time they would normally be done to stabilize project staffing levels. Long-lead-time activities might consist of analyses, prototype construction and testing, or even fabrication of difficult components. Doing things out of their usual sequence increases risk in that those activities could wind up having been either unnecessary or improperly specified. On the other hand, overall risk can sometimes be reduced by removal of such activities from the critical path.

NASA sometimes chooses to employ contractors for Phase A and/or Phase B, usually does so for Phase C/D, and often does so for Phase E.

3.1 Pre-Phase A — Advanced Studies

The purpose of this activity, which is usually performed more or less continually by "Advanced Projects" groups, is to uncover, invent, create, concoct and/or devise a broad spectrum of ideas and alternatives for missions from which new projects (programs) can be selected.

PAGE 12 PRECEDING PAGE BLANK NOT FILMED
INTENTIONALLY BLANK

Pre-Phase A — Advanced Studies

Purpose: To produce a broad spectrum of ideas and alternatives for missions from which new projects/programs can be selected.
Major Activities *and their Products*:
Identify *missions consistent with charter*
Identify and involve users
Perform *preliminary evaluations of possible missions*
Prepare *program/project proposals*, which include
- Mission justification and objectives
- Possible operations concepts
- Possible system architectures
- Cost, schedule and risk estimates.

Develop *master plans* for existing program areas
Information Baselined:
Program master plans (baselined in existing programs)
Control Gates:
Informal proposal reviews

Typically, this activity consists of loosely structured examinations of new ideas, usually without central control and mostly oriented toward small studies. Its major product is a stream of suggested projects, based on the identification of needs and the discovery of opportunities that are potentially consistent with NASA's mission, capabilities, priorities and resources.

In the NASA environment, demands for new systems derive from several sources. A major one is the opportunity to solve terrestrial problems that may be addressed by putting instruments and other devices into space. Two examples are weather prediction and communications by satellite. General improvements in technology for use in space will continue to open new possibilities. Such *opportunities* are rapidly perceived as *needs* once the immediacy of their value is understood.

Technological progress makes possible missions that were previously possible. Manned trips to the moon and the taking of high resolution pictures of planets and other objects in the universe illustrate past responses to this kind of opportunity. New opportunities will continue to become available as our technological capabilities grow.

Scientific progress also generates needs for NASA systems. As our understanding of the universe around us continues to grow, we are able to ask new and more precise questions. The ability to answer these questions often depends upon the changing state of technology.

Descriptions of suggested projects generally include initial system design and operational concepts, preliminary project organization, schedule, testing and review structure, documentation requirements, etc.

3.2 Phase A — Conceptual Design Studies

The purpose of this activity is to determine the feasibility and desirability of suggested new major systems in preparation for the seeking of funding. According to NMI 7100.14B, the major products of this phase are a formal *Mission Needs Statement* (MNS) and one or more credible, feasible designs.

John Hodge describes this phase as "a structured version of the previous phase", which is accurate from the point of view of the particular system being studied. Pre-Phase A screening is intended to pass possible projects that

Phase A — Conceptual Design Studies

Purpose: To determine the feasibility and desirability of a suggested new major system in preparation for the seeking of funding.
Major Activities *and their Products*:
Prepare *Mission Needs Statement*
Develop preliminary *system requirements*
Identify alternative *operations and logistics concepts*
Identify *project constraints and system boundaries*
Consider *alternative design concepts*; include
Feasibility and risk studies
Cost and schedule estimates
Advanced technology requirements
Demonstrate that *credible, feasible design(s)* exist
Initiate *system validation plans*
Acquire *systems engineering tools and models*
Initiate *environmental impact studies*
Prepare *program implementation plan*
Information Baselined:
(nothing)
Control Gates:
Conceptual design review
Pre-Phase B non-advocate review

are worthwhile in terms of the resources they require. In Phase A, larger teams, often associated with an ad hoc Program or Project Office, readdress the project concept to ensure that the project justification and practicality are sufficient to warrant a place in NASA's budget. The *Mission Needs Statement* is not shown in the sidebar as being baselined, as it is not under configuration control by the project. (It may be under configuration control at the program level, as may the program requirements documents and the *Project Initiation Agreement*.)

3.3 Phase B — Concept Definition

The purpose of this phase is to establish an initial baseline. Its primary products are a reaffirmation of the

FOLDOUT FRAME 1 — FOLDOUT FRAME 2 — FOLDOUT FRAME 3 — FOLDOUT FRAME 4

NASA PROJECT CYCLE (GENERIC)

Draft 921116

New Start Budget Cycle

NSY = New Start Year

FY NSY – 3: Mission Needs Statement (MNS) Approval; Phase A Studies; Phase B RFP

FY NSY – 2: First Pop for NSY; Second Pop for NSY; NASA Budget for NSY to OMB; Phase B Studies; Project Plan Approval; Phase C/D RFP

FY NSY – 1: MNS Reaffirmation; Congressional Hearings

FY NSY: FY NSY Appropriation; Annual Budget Request/Renewal; Phase C/D

System Acquisition

Phases

Phase A Mission Needs & Conceptual Studies — Source Selection (If required)

Phase B Concept Definition — Source Selection (If required)

Source Selection Process

Phase C Design & Development

Phase D Fabrication, Integration, Test, & Certification

Multi Unit Procurement

Phase E Pre-Operations — Source Selection (If required)

Phase F Operations/Disposal

Major Activities

Phase A
- Prepare Program Objectives, i.e., Define User & Mission Needs
- Identify Initial Operations & Logistics Concepts
- Develop Plan for Validating User Needs
- Write User Validation Plan
- Identify Programmatic Constraints & System Boundries
- Generate Broad Spectrum of Alternative System Concepts
- Define Concept Evaluation Criteria
- Mission Analysis
- Preliminary Evaluation of Alternative Concepts & Technologies (Killer Trades)
- Develop Preliminary Program Requirements
- Acquire System Engineering & Management Tools/Models
- Initiate Environmental Impact Studies
- Evaluate System Concept(s)
- Identify Risk & Assess Program Feasibility
- Initiate System Acquisition Planning
- Develop Operations & Logistics Validation Approach
- Generate Draft SRD
- Develop Business, Management, Technical Plans
- Develop Program Implementation Strategy (Business, Management, Technical)
- Identify Advanced Technology Development Requirements

Phase B
- Revalidate User Requirements
- Refine Trade-offs & Analysis
- Prepare Preliminary Project Plans
- Define Verification Approach & Requirements
- Perform Lifecycle Analysis
- Identify Government Resources Required
- Conduct Technology Assessments
- Define Internal & External Interface Requirements
- Develop SRD & System Specification
- Demonstrate Proof of Concept Models
- Identify Major Ground Test & Facilities Requirements
- Define Operations Requirements
- Initiate Advanced Technology Development
- Develop Statement of Work & WBS
- Refine Assessments
- Evaluate Changes/ Updates to SRD
- Verify Specifications Meet System Requirements
- Select Acquisition Approach (In/Out, When)
- Complete System Acquisition Planning
- Prepare Project Plan
- Review Project Justification
- Define Project Documentation Requirements List (DRL/DID)

Source Selection Process
- Option: Prepare Draft RFP; Send Draft RFP to Bidders; Update Draft RFP
- Final System Acquisition Plan
- Develop Source Selection Plan
- SEB, Selection Criteria, Schedule, Bidders List
- Prepare RFP
- Statement of Work
- Confirm Funds Appropriated/ Budgeted/ Programmed
- Brief Concept to Potential Bidders
- Implement Source Selection Plan
- Release RFP
- Evaluate Proposals
- Determine Projected Cost to Complete
- Select Contractor
- Audit Selection Process
- Negotiate Contract
- Award Contract

Phase C
- Review/Update Phase B Products & SOW
- Review Make-Buy Decision
- Review Contract Negotiations
- Update Project Plan
- Identify Resources
- Commit Resources
- Decompose Functional Requirements into CI (Configuration Item) Specifications
- Evaluate Risk Reduction Approaches
- Demonstrate Designs are Achievable
- Evaluate & Reduce Design Risk
- Develop Preliminary Design
- Develop Risk Reduction Models
- Prepare CI "Design-to" Specifications & Verification Plans
- Science Payload Confirmation
- Update Operations Plan
- Update Payload Integration Plan
- Generate PDR Submittal
- Develop Detailed Design
- Develop System Integration Approach
- Prepare Interface Documents
- Prepare CI Test Plans
- Trade Studies
- Prepare CI Development Specifications
- Prepare CI & System "As-built" Documents & Inspection Plans
- Develop System Deployment Approach
- Identify P³i (Pre-Planned Product Improvement) Opportunities
- Complete Risk Reduction Models
- Generate CDR Submittal

Phase D
- Program Code
- Code SW Components
- Fabricate HW Assemblies
- Prepare CI & System Verification Procedures for Review
- Perform Component Life Testing
- Assemble Configuration Items
- Finalize Operational CI Verification Procedures
- Finalize Qualification CI Certification Procedures
- Provide GFE
- Testing of HW Assemblies & SW Components
- Test SW Components
- Conduct CI Verification
- Test HW Assemblies
- Testing & Qualification
- Assemble System
- Verify Design Meets Interface Specifications
- Finalize System Verification Procedures
- Conduct Qualification CI Certification
- Preliminary Operational Demonstration Procedures
- Subsystem Integration & Test
- Conduct System Testing
- Prepare CI & System Operations & Maintenance Procedures for Review
- Prepare CI Maintenance Manual
- Conduct Qualification and Operational System Certification at Seller's Facility (Government Project Office Acceptance)
- Document CI "As-built" Configuration
- Document System "As-built" Configuration
- Develop Lessons Learned

Multi Unit Procurement
- Update Production Risk Analysis
- Prepare for Full-rate Production Acquisition (If required)

Phase E
- Documentation Audit
- Conduct Training
- Physically Integrate with Launch Vehicle (If required)
- Conduct Operational Demonstration (Operational Management Acceptance)
- Validation of System Performance at Operational Site (e.g., On Orbit Verification)

Phase F
- Operate the System
- On-going Training
- Perform System Maintenance
- Sustaining Engineering
- Assess Performance/ Supportability Trends
- Compile Operations Lessons Learned
- Implement P³i Opportunities
- NOC (Next Operational Concept) Decision
- Update Disposal/ Decommission Plan

Products

Phase A
- Draft Mission Needs Statement
- Initial System Concept(s)
- Concept Evaluation Criteria
- Initial Operations & Logistics Concept(s)
- Preliminary Program Requirements Document
- User Needs Validation Plan
- System Engineering & Management Tools/Models
- Program Implementation Plan
- Initial System Acquisition Plan
- Announcement of Opportunity
- Preliminary Operations & Logistics Validation Plan(s)
- Trade-off Study Results
- Draft MOUs
- Baseline PRD
- Feasible System Concepts
- Advanced Technology Development Plan
- Draft SRD
- Pre Phase B NAR Package

Phase B
- System Verification Plan
- Flight & Ground Operations Plan
- Technology Readiness Report
- System Engineering Management Plan
- Concept Robustness Assessment
- Program Initiation Agreement
- Cost Commitment
- Preliminary Project Plan
- Initial Requirements Traceability Matrix
- Baseline SRD
- Baseline System Specification
- Strawman Science Payloads
- NAR Package
- System Concept Document
- Risk Management Plan
- Center Commitments
- Project Plan
- Updated WBS
- Final System Acquisition Plan
- Source Selection Plan
- Statement of Work
- Preliminary Manufacturing Plan
- Final System Specification
- Final DRL/DID

Source Selection Process
- Option: Draft RFP; Bidders Comments to Draft RFP
- Source Selection Plan
- Bidders List
- Final RFP
- Government Resources Available
- Proposal Evaluation File
- Source Selection Decision Memo
- Contract

Phase C
- Subcontracts if Required
- Project Resources Available
- Revised Project Plan
- Hardware Specifications
- Software Specifications
- Risk Reduction Study Report
- PDR Submittal
- Risk Reduction Models
- CI Verification Plan
- CI "Design-to" Specifications
- Preliminary Drawings
- Trade Study & TPM Reports (Preliminary)
- Operations Plan
- Updated PIPs
- Assessment of Advanced Technical Development
- CDR Submittal
- System Integration Plan
- Trade Study & TPM Reports
- Interface Documents
- CI Development Specifications
- Requirements/Specification Traceability Matrix
- "Build-to" Drawings & Documentation
- P³i Study Results
- System Development Plan
- P³i Decision Document
- Final Environmental Impact Statement

Phase D
- CI & System "Build-to" Documentation
  - Engineering Drawings
  - Process Specifications
  - Material Specifications
  - SW Development Folders
  - Database Design Document
- HW Assemblies & SW Components Verification Procedures
- CI Verification Procedures
- Parts Procurement
- System Verification Procedures
- Verification Compliance Report
- HW Assemblies
- SW Components
- GFE
- HW & SW Engineering Test Data
- Qualification CI Certification Procedures
- Verified HW Assemblies
- Verified SW Components
- Test Discrepancy Resolution
- Functioning HWCI & CSCI
- Verified Operational HWCIs
- Verified Operational CSCIs
- CI Operations & Maintenance Manual
- Certified Qualification HWCIs
- Certified Qualification CSCIs
- Functioning System
- Verified Operational System
- Certified "As-built" Documentation
- System Verification Report (Qualification & Operation)
- CI "As-built"/ "As-coded" Documentation
- Final System Operations & Maintenance Methods
- Government Acceptance (DD-250)
- System Acceptance Report
- System Development Lessons Learned Document

Multi Unit Procurement
- Full-rate Production Acquisition Plan (If required)
- Manufacturing Plan (Update)
- Producible System
- Operational Test & Evaluation Plan

Phase E
- Trained Personnel
- GFE Available
- Final ILS Plan
- Final Operational Demonstration Procedures
- Operational System
- Operational Readiness Certificate

Phase F
- Activity Reports
- Trained Personnel
- Failure Reports
- Supportability Reports
- Operations Lessons Learned Documents
- Updated Disposal/ Decommission Plan
- System Modifications (As required)
- Updated Production Plan (As required)
- Updated Test Plan
- Updated Specifications & O&M Documentation

Control Gates

- Conceptual Design Review — CoDR
- Pre-Phase B Non-Advocate Review
- Non-Advocate Review — NAR
- Program/Project Requirements Review — PRR
- Safety Review
- PRR Findings & Recommendations
- Source Selection Initiation Review — SSIR
- Source Evaluation Board Review — SEB
- Competition Advocate Review — CAR
- Contract Award — AWARD
- Project Initiation Review — PIR
- NASA/Contractor Post-Contract Review — N.C. PCR
- Software Specification Review — SSR
- Hardware Specification Review — HSR
- Preliminary Design Review — PDR; Config Item — PDR
- Safety Review
- PDR Findings & Recommendations
- Critical Design Review — System — CDR; Config Item — CDR (*Build-to" Baseline)
- Safety Review
- CDR Findings & Recommendations
- CI Test Readiness Review — TRR (Qualification & Operation)
- CI Acceptance Review — CIAR
- System Test Readiness Review — TRR
- System Formal Qualification Review — FQR (Product Baseline)
- System Functional & Physical Configuration Audit — FCA, PCA
- System Acceptance Review — SAR
- Safety Review
- Production Readiness Review — ProdRR
- Launch Readiness Review — LRR (Space-Based System)
- Operational Readiness Review — ORR (Ground-Based System)
- Safety Review
- Operational Acceptance Review — OAR
- Annual System Operations Review — ASOR
- System Upgrade Review — SUR

Control Gate Legend: Government Only; Joint; Contractor Only; Source Selection Activity; Not to Scale; Not to Scale (Ref)

ASM ©1991 Center for Systems Management, 5333 Betsy Ross Drive, Santa Clara, CA 95054, (408) 988-8090 172 PC NASA 921116

Page intentionally left blank

Mission Needs Statement and that baseline, which consists (according to NMI 7100.14B) of "preliminary specifications, a preliminary schedule, and resource and management plans to support one of the alternative design concepts."

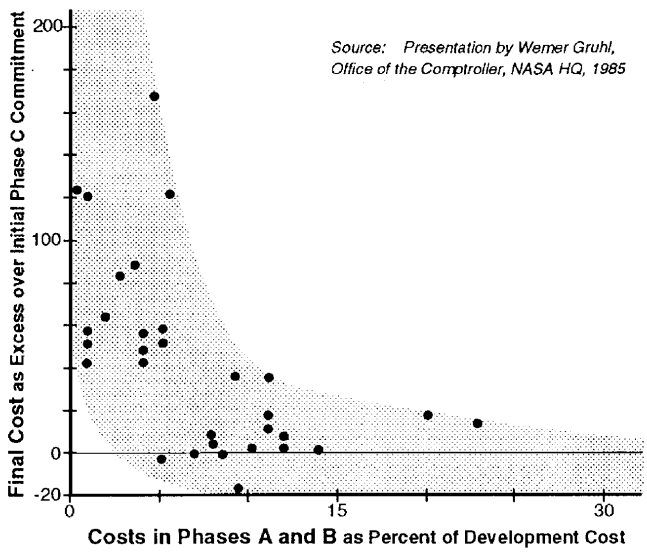


Figure 6 — Overruns are Very Likely if Phases A and B are Underfunded.

On the way to these products, projects are subjected to a *Project Definition and Cost Review* (PDCR, formerly known as the *Non-Advocate Review* or NAR). This activity seeks (according to NMI 7120.3) to assess the state of project definition in terms of its "clarity of objectives, thoroughness of technical and management plan, technical complexity, evaluation of technical, cost, and schedule risks, and contingency reserve allowances in schedule and cost." The timing of this review is often driven by the Federal budget cycle, which requires at least 16 months between NASA's budget preparation for submission to the President's Office of Management and Budget, and the Congressional funding for a new project start. (See Section 3.8.) There is thus a natural tension between the desire to have maturity in the project at the time of the PDCR and the desire to progress efficiently to full-scale design and development.

Eventually, "the" baseline will actually consist of a collection of baselines: system requirements and design; implementation, test and operations plans; and others. Establishment of baselines implies the implementation of configuration control procedures. At the end of this phase, the baseline normally contains project plans and requirements, with no design detail other than, perhaps, the system architecture. In any case, from this point on, almost all changes to the baseline are expected to represent successive refinement, not fundamental changes to the mission concept. Prior to baselining, the system architecture must have been validated by enough in-depth design work that there is high confidence that there is at least one way it can work. That is, the existence of a *credible, feasible design* must be ensured at a lower level of detail than was sufficient for Phase A.

Trade studies precede (rather than follow) system design decisions. Thus, Phase A's *credible, feasible designs* should *not* be baselined (though they should be archived, along with the rationale and trades that led to them). Generally, only true breakthroughs — or disasters — will lead to major design changes, though improved resolution sometimes brings recognition that a selected de-

Phase B — Concept Definition

Purpose: To define the project in enough detail to establish an initial baseline.

Major Activities *and their Products*:
Reaffirm the *Mission Needs Statement*
Prepare a *Program Initiation Agreement*
Prepare a *Systems Engineering Management Plan*
Prepare a *risk management plan*
Initiate *configuration management*
Develop *system-level cost-effectiveness model*
Restate mission needs as *system requirements*
Establish the initial *requirements traceability matrix*
Select a baseline *system architecture* (at some level of resolution) and *concept of operation*
Identify *strawman science payloads*
Define *internal and external interface requirements*
Define the *work breakdown structure*
Define *verification approach and policies*
Prepare preliminary *manufacturing plans*
Identify *government resource requirements*
Identify *ground test and facility requirements*
Develop *statement of work*
Revise and publish *project implementation plans*
Initiate *advanced technology developments*

Information Baselined:
System requirements and traceability matrix
System architecture and work breakdown structure
Concept of operation
Project implementation plans, including schedule, resource usage, and management

Control Gates:
Project Definition and Cost Review (formerly called the Non-Advocate Review)
Program/project requirements review
Safety review

A *Credible, Feasible* Design

A feasible system design is one that can be implemented as designed and can then accomplish the system's goals within the constraints imposed by the fiscal and operating environment. To be credible, a design must not depend on the occurrence of unforeseen breakthroughs in the state of the art. While a credible design may assume *likely* improvements in the state of the art, it is nonetheless *riskier* than one that does not.

sign concept is infeasible, so that a design change is required.

3.4 Phase C — Design and Development

The purpose of this phase is to unfold system requirements into system and subsystem designs. The concentration of effort is on the design of subsystems that integrate properly with the system. System trades and subsystem trades iterate back and forth; Chamberlain, Fox and Duquette described a decentralized technical process for ensuring that such trades lead efficiently to an optimum design.

Boehm described several popular approaches to the unfolding process. The "code and fix" paradigm works well for simple, well-understood systems: simply do the job and fix any problems. The "waterfall" or "requirements-driven design" paradigm works well for complex, well-understood systems, particularly when requirements can be thoroughly determined early in the process and there is little chance that discoveries will be made during the detailed design or integration steps that will make requirements changes desirable. These conditions are often met for major NASA systems. The "evolutionary development" paradigm works well when the product can be developed more or less automatically from requirements specifications — which is rarely the case with systems with large amounts of hardware. Boehm's "spiral model" encompasses most of the other paradigms as special cases. Selection of a product development process paradigm must be a case-dependent decision, based on the system engineer's judgment and experience.

This phase involves the full-scale development of the system and subsystem architecture, containing the system-level *preliminary design review* (PDR), then subsystem-level PDRs, then lower-level PDRs, and so on. PDRs reflect the successive refinement of requirements into designs. At each step in the unfolding process, corresponding integration and verification tests (and related activities) are planned. After the lowest level designs have passed

Phase C — Design and Development

Purpose: To design a system (and its associated subsystems, including its operations systems) so that it will be able to meet its requirements.

Major Activities *and their Products*:

Add *subsystem design specifications* to the system architecture
Publish *subsystem requirements documents*
Prepare *subsystem verification plans*
Prepare *interface documents*
(Repeat the process of successive refinement to get "design-to" and "build-to" *specifications and drawings, verification plans,* and *interface documents* at all levels)
Augment baselined documents to reflect the growing maturity of the system: *system architecture, requirements traceability matrix, work breakdown structure, project implementation plans*
Monitor project progress against project plans
Develop the *system integration plan* and the *system operations plans*
Archive documentation for *trade studies* performed
Develop the *end-to-end information system design* and the *system deployment approach*
Identify opportunities for pre-planned product improvement
Confirm *science payload selection*

Information Baselined:

Subsystem (and lower level) requirements and designs, including traceability to higher levels
"Design-to" specifications at all levels
"Build-to" specifications at all levels

Control Gates:

System-level preliminary design review
Subsystem (and lower level) preliminary design reviews
Subsystem (and lower level) critical design reviews
System-level critical design review

their PDRs and the design issues that were uncovered have been resolved, a sequence of *critical design reviews* (CDRs) begins. The sequence reflects the integration process that will occur in the next phase. It begins at the lowest level of design and culminates in the system-level CDR. The final products of the phase are baseline designs (drawings, pseudo-code, documentation, etc.) in sufficient detail that actual production can proceed.

3.5 Phase D — Fabrication, Integration, Test and Certification

The purpose of this phase is to build the system designed in the previous phase. Activities include fabrication

Phase D — Fabrication, Integration, Test and Certification

Purpose: To build the subsystems (including the operations system) and integrate them to create the system, meanwhile developing confidence that it will be able to meet the system requirements.

Major Activities *and their Products*:
Fabricate (or code) the *parts* (i.e.: the lowest-level items in the system architecture)
Integrate those items according to the integration plan and perform verification tests, yielding *verified subassemblies*
(Repeat the process of successive integration to get a *certified system*, with *verified system components* at all levels)
Perform *system qualification test(s)*
Perform *system acceptance test(s)*
Monitor project progress against project plans
Archive documentation for *verification tests* performed
Audit "as-built" configurations
Document *Lessons Learned*
Prepare *operator's manuals*
Prepare *maintenance manuals*

Information Baselined:
"As-built" configuration data
Operator's manuals
Maintenance manuals

Control Gates:
Test readiness reviews (at all levels)
Acceptance reviews (at all levels)
System qualification review(s)
System acceptance review
System functional and physical configuration audit

of hardware and coding of software, integration, verification and validation, and certified acceptance of the system.

3.6 Phase E — Pre-Operations

The purpose of this phase is to prepare the certified system for operations. Activities include the initial training of operating personnel and finalization of the Integrated Logistics Support Plan. For flight projects, the focus of activities then shifts to pre-launch integration and launch. For large flight projects, there may be an extended period of orbit insertion, assembly, and initial shake-down operations. In some projects, these activities may be minor, so that this phase is combined with either its predecessor or its successor.

Phase E — Pre-Operations

Purpose: To ensure that the certified system is ready for operations.

Major Activities *and their Products*:
Audit all *operations documentation*
Train *initial system operators*
Finalize *Integrated Logistics Support Plan*
Integrate with launch vehicles
Launch, orbit insertion, etc.
In-orbit assembly and check-out
Certify *operational readiness*

Information Baselined:
Integrated Logistics Support Plan
Command sequences for end-to-end command and telemetry validation and ground data processing

Control Gates:
Launch readiness reviews
Operational readiness reviews
Safety reviews

In any case, the major product is a system that has been shown to be capable of accomplishing the purpose for which it was created.

3.7 Phase F — Operations and Disposal

The purpose of this phase is to meet the initially identified need or to grasp the initially identified opportunity. The products of the phase are the results of the mission. This phase encompasses evolution of the system only insofar as that evolution does not involve major changes to the system architecture; changes of that scope constitute new "needs", and the project cycle starts over.

Phase F encompasses the problem of dealing with the system when it has completed its mission; the time at which this occurs depends on many factors. For a flight system with a short mission duration, such as a *Spacelab* payload, disposal may require little more than de-integration of the hardware and its return to its owner. On large flight projects of long duration, disposal may proceed according to long-established plans, or may begin as a result of unplanned events, such as accidents. Alternatively, technological advances may make it uneconomic to continue operating the system either in its current configuration or an improved one.

In addition to uncertainty as to when this part of the phase begins, the activities associated with safely deactivating and disposing of a system may be long and complex. Consequently, the costs and risks associated with different designs should be considered during the planning process.

Phase F — Operations and Disposal

Purpose: To actually meet the initially identified need or to grasp the opportunity, then to dispose of the system in a responsible manner.

Major Activities *and their Products*:
Train replacement *operators*
Conduct *the mission(s)*
Maintain the *operating system*
Dispose of the system

Information Baselined:
Mission outcomes, such as:

- Engineering data on system, subsystem and materials performance
- Science data returned
- High resolution photos from orbit
- Accomplishment records ("firsts")
- Discovery of the Van Allen belts
- Discovery of volcanoes on Io.

Operations and maintenance logs
Problem/failure reports

Control Gates:
Operational acceptance review
Regular system operations reviews
System upgrade reviews

3.8 Funding: The Budget Cycle

NASA operates with annual funding from Congress. This funding results, however, from a three-year rolling process of budget formulation, budget enactment, and finally, budget execution. A highly simplified representation of the typical budget cycle is shown in Figure 7.

NASA starts developing its budget each January with economic forecasts and general guidelines being provided by the Executive Branch's Office of Management and Budget (OMB). In early May, NASA conducts its Program Operating Plan (POP) and Institutional Operating Plan (IOP) exercises in preparation for submittal of a preliminary NASA budget to the OMB. A final NASA budget is submitted to the OMB in September for incorporation into the President's budget transmittal to Congress, which generally occurs in January. This proposed budget is then subjected to Congressional review and approval, culminating in the passage of bills authorizing NASA to obligate funds in accordance with Congressional stipulations and appropriating those funds. The Congressional process generally lasts through the summer. In recent years, however, final bills have often been delayed past the start of the fiscal year on October 1. In those years, NASA has operated on continuing resolutions by Congress.

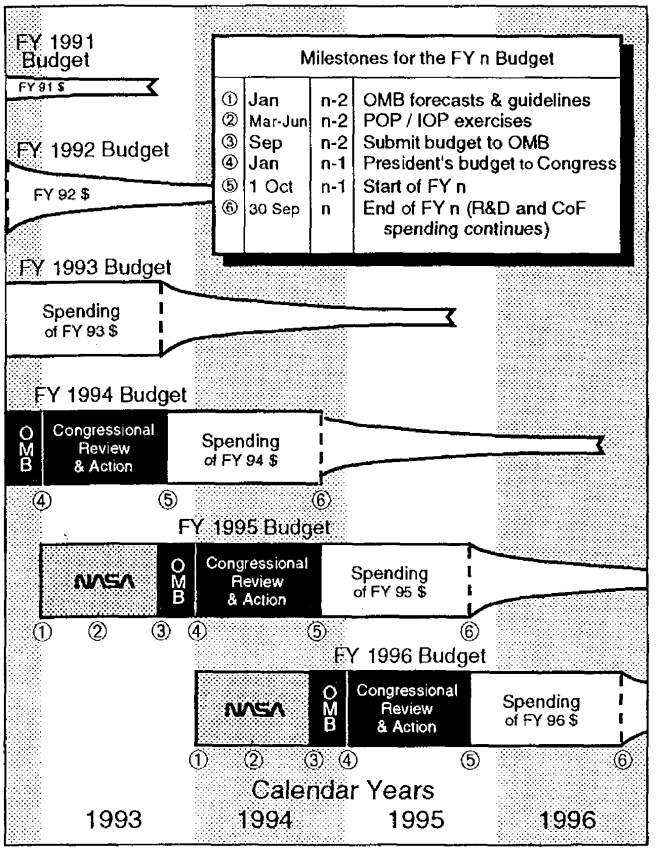


Figure 7 — Typical NASA Budget Cycle.

With annual funding, there is an implicit funding control gate at the beginning of every fiscal year. While these gates place planning requirements on the project and can make significant replanning necessary, they are not part of an orderly systems engineering process. Rather, they constitute one of the sources of uncertainty that affect project risks and should be included in project risk considerations.

3.9 The Role of Systems Engineering in the Product Development Process

Forsberg and Mooz describe what they call "the technical aspect of the project cycle" by a vee-shaped chart, starting with user needs on the upper left and ending with a user-validated system on the upper right. Figure 8 provides a summary level overview of those activities. On the left side of the chart, decomposition and definition activities resolve the system architecture, creating the details of the design. Integration and verification flows up and to

the right as successively higher levels of subsystems are verified, culminating at the system level. This summary chart follows the basic outline of the vee chart developed by NASA as part of the Software Management and Assurance Program. ("CIs" in the figure refer to the hardware and software *configuration items* which are controlled by the configuration management system.)

Decomposition and Definition. Figure 9 (a fold-out, next page) is one of the products developed by CSM as a result of the NASA Inter-Center Systems Engineering Working Group's workshops. It provides a three-dimensional view of the technical aspect of the project cycle. At each level,

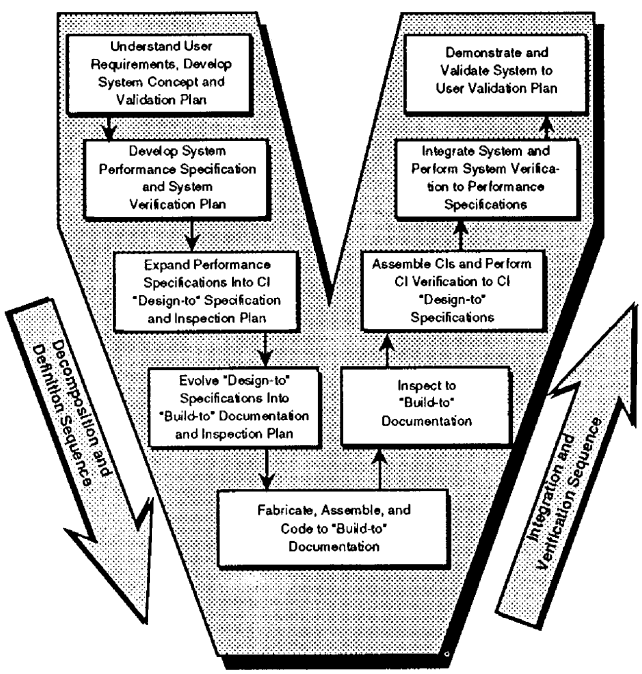


Figure 8 — Overview of the Technical Aspect of the NASA Project Cycle.

moving into the depth of the paper (perpendicular to the surface), there are a number of parallel boxes suggesting that there may be many subsystems that make up the system at that level of decomposition. Also, at the top level, on the left of the chart, the multiplicity of parallel boxes illustrates that alternative design concepts are evaluated. At the conclusion of Phase B, a baseline (which, in rare cases, might contain more than one design concept) is established for further definition.

As product development progresses, the baseline evolves under control of a formal configuration management system. Among the fundamental purposes of configuration management are ensuring that changes are real improvements, either to the resolution or to the cost-effectiveness of the final system. Another is to prevent requirements from "creeping".

The left side of the core of the vee (the shaded area in Figure 9) is similar to the so-called "waterfall" or "requirements-driven design" model of the product development process. The control gates define significant decision points in the process. Work should not progress beyond a decision point until the project manager is ready to publish and control the documents containing the decisions that have been agreed upon at that point.

There is no prohibition against doing detailed work early in the process. In fact, detailed hardware and/or software models may be required at the very earliest stages to clarify user needs or to establish credibility for the claim of feasibility. Early application of involved technical and support disciplines is an essential part of this process; this is in fact implementation of *concurrent engineering*.

As the process progresses, system modeling and tradeoff studies continue. This is shown on the chart by the ascending and descending vertical off-core activities.

While many kinds of studies and decisions are associated with the off-core activities, only decisions at the core level are put under configuration management at the various control gates. Off-core activities, analyses and models are used to substantiate the core decisions and to ensure that the risks have been mitigated or determined to be acceptable. The off-core work is not formally controlled, but the analyses, data and results should be archived to facilitate replication at the appropriate times and levels of detail to support introduction into the baseline.

The multiple arrows descending from the bottom of the left side of the core of the vee indicate that there can, and should, be sufficient iteration downward to establish feasibility and to identify and quantify risks. Upward iteration with the requirements statements (and with the intermediate products as well) is permitted, but should be kept to a minimum, or cost and schedule determinants of the system's final cost-effectiveness are likely to suffer. That is, only allow the requirements to change if you must — and if the project can afford the inevitable impact on cost and schedule.

In software projects, upward confirmation of solutions with the users is often necessary because user requirements cannot be adequately defined at the inception of the project. Even for software projects, however, iteration with user requirements should be stopped at the preliminary design review (PDR), or cost and schedule are likely to get out of control.

Modification of user requirements after PDR should be held for the next model or release of the product. If significant changes to user requirements are made after PDR, the project should be stopped and restarted with a new vee, reinitiating the entire process. The repeat of the process may be quicker because of the lessons learned the first time through, but all of the steps must be redone.

Time and project maturity flow from left to right on the vee. Once a control gate is passed, backward iteration is not possible. Iteration with the user requirements, for example, is possible only vertically, as is illustrated on Figure 9.

Incremental Development. If the user requirements are too vague to permit final definition at PDR, one approach is to develop the project in predetermined incremental releases. The first release is focused on meeting a minimum set of user requirements, with subsequent releases providing added functionality and performance. This is a common approach in software development.

The incremental development approach is easy to describe in terms of the vee chart: all increments have a common heritage down to the first PDR. The balance of the product development process has a series of displaced and overlapping vees, one for each release.

Concurrent Engineering. If the project passes early control gates prematurely, it is unlikely to have been adequately defined. This, in turn, is likely to result in a need for significant iterations of requirements and designs late in the development process. One way this can happen is by failing to involve the appropriate technical experts at early stages, thereby resulting in the acceptance of requirements that cannot be met and the selection of design concepts that cannot be built, tested, maintained, and/or operated.

Concurrent engineering is the simultaneous consideration of product and process downstream requirements by multifunctional teams. As suggested by the vertical lines in Figure 9, specialists from all disciplines whose expertise will eventually be represented in the product can be expected to have important contributions throughout the development process. The system engineer is responsible for ensuring that key personnel are involved at each step, starting with the system requirements and feasibility studies in Phase A. The specialty engineering issues of human factors, safety, reliability, maintainability, logistics, etc., are always in danger of being overlooked until too late in the process. In large projects, a large, dedicated team may be required. In small projects, it is often sufficient for the system engineer to have access to independent expert advice and detailed assistance.

Role of Systems Engineering. The interface between the roles of the system engineer and design engineers is indicated at the right side of Figure 9. System engineers are responsible for the accomplishment of the activities above the line, while design engineers provide technical assistance. Design engineers are responsible for the accomplishment of the activities below the line, while the system engineer performs technical audit.

At the lower levels of the chart, the tasks are shown as parallel efforts for different kinds of system components. Operations, hardware and software are illustrated; organizations, procedures and even facilities must sometimes be considered as well. The system engineer must often conduct trade studies between these areas, as well as within them, for many system functions can be performed by subsystems in several of the areas.

Technology Insertion. Projects are sometimes initiated with known technology shortfalls, or with areas for which new technology will result in substantial product improvement. Technology development can be done in parallel with the project evolution and inserted as late as the preliminary design review. A parallel approach that is *not* dependent on the development of new technology must be carried unless high risk is acceptable. The technology development activity would be represented by a horizontal bar off the core, generally below the dividing line between the roles of system and design engineering, and would be managed by the project manager and system engineer as a critical activity.

Multi-Disciplinary Product Development Teams

The detailed evaluation of product and process feasibility and the identification of significant uncertainties (system risks) must be done by experts from a variety of disciplines. An approach that has been found effective is to establish teams for the development of the product with representatives from all of the disciplines and processes that will eventually be involved. These multi-disciplinary product development teams (PDTs) often have multifunctional (technical and business) members. Technical personnel are needed to ensure that issues such as producibility, verifiability, deployability, supportability, trainability, operability, and disposability are all considered in the design. In addition, business (e.g., procurement) representatives are added to the team as the need arises. Continuity of support from the discipline organizations throughout the system life-cycle is highly desirable, though team composition and leadership can be expected to change as the system progresses from phase to phase.

Page intentionally left blank

FOLDOUT FRAME 1

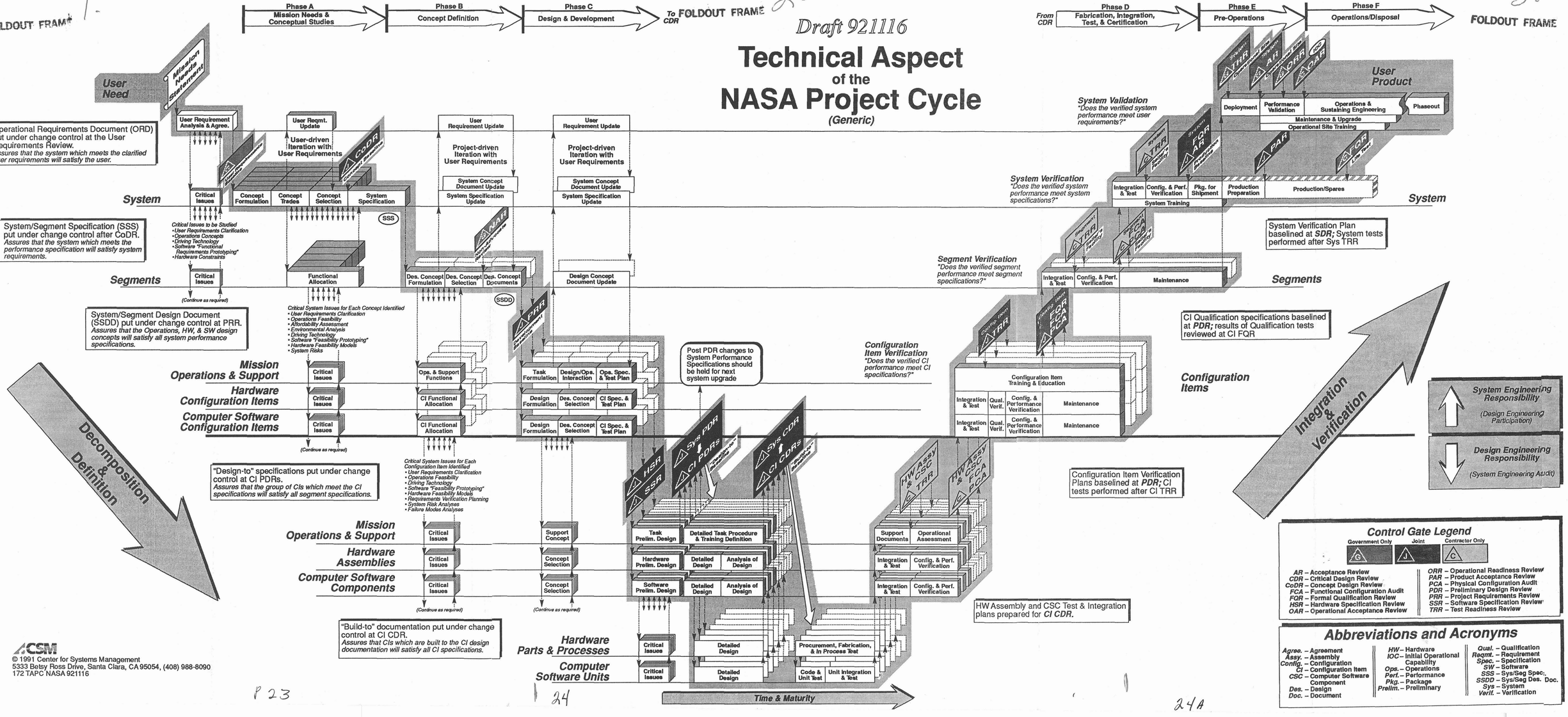
FOLDOUT FRAME 2

FOLDOUT FRAME 3

Draft 921116

Technical Aspect of the NASA Project Cycle

(Generic)



CSM
© 1991 Center for Systems Management
5333 Betsy Ross Drive, Santa Clara, CA 95054, (408) 988-8090
172 TAPC NASA 921116

Integration and Verification. Descending down the left side of the vee represents *decomposition and definition*. Ascending the right side is the process of *integration and verification*.

At each level, there is a direct correspondence between activities on the left and right sides of the chart. This is deliberate. The method of verification must be determined as the requirements are *developed and documented* at each level. This minimizes the chances that requirements are specified in a way which cannot be measured or verified.

Even at the highest levels, as user requirements are translated into system requirements, the system verification approach, which will prove that the system does what is required, must be determined. The demands of the verification process can drive cost and schedule, and may in fact be a discriminator between alternative concepts. For example, if engineering models are to be used for verification or validation, they must be specified and costed, their characteristics must be defined, and their development time must be incorporated into the schedule from the beginning.

Verification vs. Validation. The distinction between verification and validation is significant: *verification* consists of proof of compliance with specifications, and may be determined by test, analysis, inspection, or demonstration. *Validation* consists of proof that the system accomplishes (or, more weakly, *can* accomplish) its purpose. It is usually much more difficult (and much more important) to validate a system than to verify it. Strictly speaking, validation can be accomplished only at the system level, while verification must be accomplished throughout the entire system architectural hierarchy.

4 Management Issues in Systems Engineering

This chapter provides more specific information on the systems engineering products and approaches used in the project cycle just described. These products and approaches are the system engineer's contribution to project management, and are designed to foster structured ways of managing a complex set of activities.

4.1 Harmony of Goals, Work Products and Organizations

When applied to a system, the doctrine of successive refinement is a "divide-and-conquer" strategy. Complex systems are successively divided into pieces that are less complex, until they are simple enough to be conquered. This decomposition results in several structures for describing the *product system* and the *producing system* ("the system that produces the system"). These structures play important roles in systems engineering and project management. Many of the remaining sections in this chapter are devoted to describing some of these key structures.

Structures that describe the product system include, but are not limited to, the requirements tree, system architecture and certain symbolic information such as system drawings, schematics, and data bases. The structures that describe the producing system include the project's work breakdown, schedules, cost accounts, and organization. These structures provide different perspectives on their common *raison d'être*: the desired product system. Creating a fundamental harmony among these structures is essential for successful systems engineering and project management; this harmony needs to be established in some cases by one-to-one correspondence between two structures, and in other cases, by traceable links across several structures. It is useful, at this point, to give some illustrations of this key principle.

System requirements serve two purposes in the systems engineering process: first, they represent a hierarchical description of the buyer's desired product system as understood by the system engineer. The interaction between the buyer and system engineer to develop these requirements is one way the "voice of the buyer" is heard. Determining the right requirements — that is, only those that the informed buyer is willing to pay for — is an important part of the system engineer's job. Second, system requirements also communicate to the design engineers what to design and build (or code). As these requirements are allocated, they become inexorably linked to the system architecture and product breakdown, which consists of the hierarchy of project, systems, segments, elements, subsystems, etc. (See the sidebar on system terminology on page 3.)

The Work Breakdown Structure (WBS) is also a tree-like structure that contains the pieces of work necessary to complete the project. Each task in the WBS should be traceable to one or more of the system requirements. Schedules, which are structured as networks, describe the time-phased activities that result in the product system in the WBS. The cost account structure needs to be directly linked to the work in the WBS and the schedules by which that work is done. (See Sections 4.3 through 4.5.)

The project's organization structure describes the clusters of personnel assigned to perform the work. These organizational structures are usually trees. Sometimes they are represented as a matrix of two interlaced trees, one for line responsibilities, the other for project responsibilities. In any case, the organizational structure should allow identification of responsibility for each WBS task.

Project documentation is the product of particular WBS tasks. There are two fundamental categories of project documentation: baselines and archives. Each category contains information about both the product system and the producing system. The baseline, once established, contains information describing the current state of the product system and producing system resulting from all decisions that have been made. It is usually organized as a collection of hierarchical tree structures, and should exhibit a significant amount of cross-reference linking. The archives contain all of the rest of the project's information that is worth remembering, even if only temporarily. The archives should contain all assumptions, data, and supporting analyses that are relevant to past, present, and future decisions. Inevitably, the structure (and control) of the archives is much looser than that of the baseline, though cross references should be maintained where feasible. (See Section 4.7.)

The structure of reviews (and their associated control gates) reflect the time-phased activities associated with the realization of the product system from its product breakdown. The status reporting and assessment structure provides information on the progress of those same activities. On the financial side, the status reporting and assessment structure should be directly linked to the WBS, schedules, and cost accounts. On the technical side, it should be linked to the product breakdown and/or requirements tree. (See Sections 4.8 and 4.9.)

PAGE 26 INTENTIONALLY BLANK PRECEDING PAGE BLANK NOT FILMED

4.2 Managing the Systems Engineering Process: The Systems Engineering Management Plan

Systems engineering management is a technical function and discipline that ensures that systems engineering and all other technical functions are properly applied.

Each project should be managed in accordance with a project cycle that is carefully tailored to the project's risks. While the project manager concentrates on managing the overall project cycle, the project-level or lead system engineer concentrates on managing its technical aspect (see Figure 9). This requires that the system engineer perform or cause to be performed the necessary multiple layers of decomposition, definition, integration, verification and validation of the system, while orchestrating and incorporating the appropriate concurrent engineering. Each one of these systems engineering functions requires application of technical analysis skills and tools to achieve the optimum system solution.

The techniques used in systems engineering management include baseline management, requirements traceability, change control, design reviews, audits, document control, failure review boards, control gates, and performance certification.

The Project Plan defines how the overall project will be managed to achieve the pre-established requirements within defined programmatic constraints. The Systems Engineering Management Plan (SEMP) is the subordinate document that defines to all project participants how the project will be technically managed within the constraints established by the Project Plan. The SEMP communicates to all participants how they must respond to pre-established management practices. For instance, the SEMP should describe the means for both internal and external (to the project) interface control.

4.2.1 Role of the SEMP

The SEMP is the rule book that describes to all participants how the project will be technically managed. The responsible NASA Center should have a SEMP to describe how it will conduct its technical management, and each contractor should have a SEMP to describe how it will manage in accordance with both its contract and NASA's technical management practices. Since the SEMP is project- and contract-unique, it must be updated for each significant programmatic change or it will become outmoded and unused, and the project could slide into an uncontrolled state. The NASA Center should have its SEMP developed before attempting to prepare a "should-cost" estimate, since activities that incur cost, such as technical risk reduction, need to be identified and described beforehand. The contractor should have its SEMP developed during the proposal process (prior to costing and pricing) because the SEMP describes the technical content of the project, the potentially costly risk management activities, and the verification and validation techniques to be used, all of which must be included in the preparation of project cost estimates.

The project SEMP is the senior technical management document for the project; all other technical control documents, such as the Interface Control Plan, Change Control Plan, Make-or-Buy Control Plan, Design Review Plan, Technical Audit Plan, etc., depend on the SEMP and must comply with it. The SEMP should be comprehensive and describe how a fully integrated engineering effort will be managed and conducted.

4.2.2 Contents of the SEMP

Since the SEMP describes the project's technical management approach, which is driven by the type of project, the phase in the project cycle, and the technical development risks, it must be specifically written for each project to address these situations and issues. While the specific content of the SEMP is tailored to the project, the recommended content is listed below.

Part I — Technical Program Planning and Control. This section should identify organizational responsibilities and authority for systems engineering management, include control of contracted engineering; levels of control established for performance and design requirements, and the control method used; technical progress assurance methods; plans and schedules for design and technical program reviews; and control of documentation.

This section should describe:

- The role of the project office
- The role of the user
- The role of the Contracting Office Technical Representative (COTR)
- The role of systems engineering
- The role of design engineering
- The role of specialty engineering
- Applicable standards
- Applicable procedures and training
- Baseline control process
- Change control process
- Interface control process
- Control of contracted (or subcontracted) engineering
- Data control process

- Make-or-buy control process
- Parts, materials, and process control
- Quality control
- Safety control
- Contamination control
- Electromagnetic interference and electromagnetic compatibility (EMI/EMC)
- Technical performance measurement
- Control gates
- Internal technical reviews
- Integration control
- Verification control
- Validation control.

Part II — Systems Engineering Process. This section should contain a detailed description of the process to be used, including the specific tailoring of the process to the requirements of the system and project; the procedures to be used in implementing the process; in-house documentation; the trade study methodology; the types of mathematical and/or simulation models to be used for system cost-effectiveness evaluations; and the generation of specifications.

This section should describe the:

- System decomposition process
- System decomposition format
- System definition process
- System analysis and design process
- Trade study process
- System integration process
- System verification process
- System qualification process
- System acceptance process
- System validation process
- Risk management process
- Life-cycle cost management process
- Use of mathematical models
- Use of simulations
- Specification and drawing structure
- Baseline management process
- Baseline communication process
- Change control process
- Tools to be used.

Part III — Engineering Specialty Integration. This section of the SEMP should describe the integration and coordination of the efforts of the specialty engineering disciplines into the systems engineering process during each iteration of that process. Where there is potential for overlap of specialty efforts, the SEMP should define the relative responsibilities and authorities of each.

This section should contain the project's approach to:

- Concurrent engineering
- The activity phasing of specialty disciplines
- The participation of specialty disciplines
- The involvement of specialty disciplines
- The role and responsibility of specialty disciplines
- The participation of specialty disciplines in system decomposition and definition
- The role of specialty disciplines in verification and validation
- Reliability
- Producibility
- Maintainability
- Human engineering
- Safety
- Quality assurance
- Survivability/vulnerability
- Integrated logistics.

4.2.3 Development of the SEMP

The SEMP must be developed concurrently with the Project Plan. In developing the SEMP, the technical approach to the project, and hence the technical aspect of the project cycle, are developed. This becomes the keel of the project that ultimately determines the length and cost of the project. The development of the programmatic and technical management approaches of the project requires that the key project personnel develop an understanding of the work to be performed and the relationships among the various parts of that work. (See Sections 4.3 and 4.4 on Work Breakdown Structures and network schedules, respectively.)

The SEMP's development requires contributions from knowledgeable programmatic and technical experts from all areas of the project that can significantly influence the project's outcome. The involvement of recognized experts is needed to establish a SEMP that is credible to the project manager and to secure the full commitment of the project team.

4.2.4 Managing the Systems Engineering Process: Summary

The systems engineering organization, and specifically the project-level system engineer, is responsible for managing the project through the technical aspect of the project cycle. This responsibility includes management of the decomposition and definition sequence, and management of the integration, verification and validation sequence. Attendant with this management is the requirement to control the technical baselines of the project. Typically, these baselines are the: functional, "design-to", "build-to" (or "code-to"), "as-built" (or "as-coded"), and "as-deployed". Systems engineering must ensure efficient and logical progression through these baselines.

Systems engineering is responsible for system decomposition and design until the "design-to" specifications of all lower level configuration items have been produced. Design engineering is then responsible for developing the "build-to" and "code-to" documentation that complies with the approved "design-to" baseline. Systems engineering audits the design and coding process and the design engineering solutions for compliance to all higher level baselines. In performing this responsibility, systems engineering must ensure requirements traceability and document the results in a requirements traceability/verification matrix.

Systems engineering is also responsible for the overall management of the integration, verification, and validation process. In this role, systems engineering conducts Test Readiness Reviews and ensures that only verified configuration items are integrated into the next higher assembly for further verification. Verification is continued to the system level, after which system validation is conducted to prove compliance with user requirements.

Systems engineering also ensures that concurrent engineering is properly applied through the project cycle by involving the required specialty engineering. The SEMP is the guiding document for these activities.

SEMP Lessons Learned from DoD Experience

- A well-managed project requires a coordinated Systems Engineering Management Plan that is used through the project cycle.
- A SEMP is a living document that must be updated as the project changes and kept consistent with the Project Plan.
- A meaningful SEMP must be the product of experts from all areas of the project.
- Projects with little or insufficient systems engineering discipline generally have major problems.
- Weak systems engineering, or systems engineering placed too low in the organization, cannot perform the functions as required.
- The systems engineering effort must be skillfully managed and well communicated to all the individuals.
- The systems engineering effort must be responsive to both the customer and the contractor interests.

4.3 The Work Breakdown Structure

A Work Breakdown Structure (WBS) is a hierarchical breakdown of the work necessary to complete a project. The WBS should be a product-based, hierarchical division of deliverable items and associated services. As such, it should contain the project's Product Breakdown Structure (PBS), with the specified prime product(s) at the top, and the systems, segments, subsystems, etc. at successive lower levels. At the lowest level are products such as hardware items, software items, and information items (e.g., documents, databases, etc.) for which there is a cognizant engineer or manager. Branch points in the hierarchy should show how the PBS elements are to be integrated. The WBS is built from the PBS by adding, at each branch point of the PBS, any necessary service elements such as management, systems engineering, integration and verification (I&V), and integrated logistics support (ILS). If several WBS elements require similar equipment or software, then a higher level WBS element might be defined to perform a block buy or a development activity (e.g., "System Support Equipment"). Figure 10 shows the relationship between a system, a PBS and a WBS.

A project WBS should be carried down to the cost account level appropriate to the risks to be managed. The appropriate level of detail for a cost account is determined by management's desire to have visibility into costs, balanced against the cost of planning and reporting. Contractors may have a Contract WBS (CWBS), which is appropriate to the contractor's needs to control costs. A summary CWBS, consisting of the upper levels of the full CWBS, is usually included in the project WBS to report costs to the contracting agency.

WBS elements should be identified by title and by a numbering system that performs the following functions:

- Identifies the level of the WBS element
- Identifies the higher level element into which the WBS element will be integrated
- Shows the cost account number of the element.

A WBS should also have a companion WBS dictionary that contains each element's title, identification number, objective, description, and any dependencies (e.g., receivables) on other WBS elements. This dictionary provides a structured project description that is valuable for orienting project members and other interested parties. It fully describes the products and/or services expected from each WBS element.

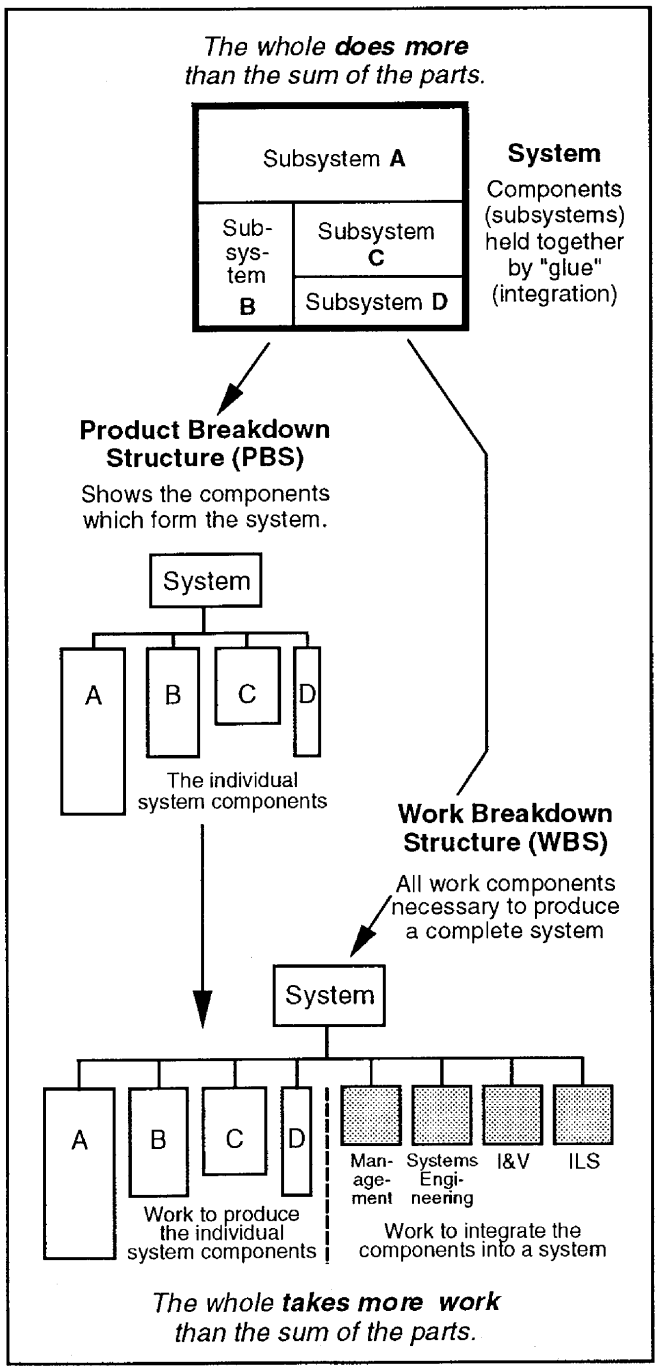


Figure 10 — The Relationship Between a System, a Product Breakdown Structure, and a Work Breakdown Structure.

This section provides some techniques for developing a WBS, and points out some mistakes to avoid. Appendix B.3 provides an example of a WBS for an airborne telescope that follows the principles of product-based WBS development.

4.3.1 Role of the WBS

A product-based WBS is the organizing structure for:

- Project and technical planning and scheduling
- Cost estimation and budget formulation. (In particular, costs collected in a product-based WBS can be compared to historical data. This is identified as a primary objective by DoD standards for WBSs.)
- Defining the scope of statements of work and specifications for contract efforts
- Project status reporting, including schedule, cost and workforce, technical performance, integrated cost/schedule data (such as earned value and estimated cost at completion)
- Plans, such as the SEMP, and other documentation products, such as specifications and drawings.

It provides a logical outline and vocabulary that describes the entire project, and integrates information in a consistent way. If there is a schedule slip in one element of a WBS, an observer can determine which other WBS elements are most likely to be affected. Cost impacts are more accurately estimated. If there is a design change in one element of the WBS, an observer can determine which other WBS elements will most likely be affected, and these elements can be consulted for potential adverse impacts.

4.3.2 Techniques for Developing the WBS

Developing a successful project WBS is likely to require several iterations through the project cycle since it is not always obvious at the outset what the full extent of the work may be. Prior to developing a preliminary WBS, there should be some development of the system architecture to the point where a preliminary PBS can be created.

The PBS and associated WBS can then be developed level by level from the top down. In this approach, a project-level system engineer finalizes the PBS at the project level, and provides a draft PBS for the next lower level. The WBS is then derived by adding appropriate

services such as management and systems engineering to that lower level. This process is repeated recursively until a WBS exists down to the desired cost account level.

An alternative approach is to define all levels of a complete PBS in one design activity, and then develop the complete WBS. When this approach is taken, it is necessary to take great care to develop the PBS so that all products are included, and all assembly/integration and verification branches are correct. The involvement of people who will be responsible for the lower level WBS elements is recommended.

A WBS for a Multiple Delivery Project. There are several terms for projects that provide multiple deliveries, such as: rapid development, rapid prototyping, and incremental delivery. Such projects should also have a product-based WBS, but there will be one extra level in the WBS hierarchy, immediately under the final prime product(s), which identifies each delivery. At any one point in time there will be both active and inactive elements in the WBS.

A WBS for an Operational Facility. A WBS for managing an operational facility such as a flight operations center is analogous to a WBS for developing a system. The difference is that the products in the PBS are not necessarily completed once and then integrated, but are produced on a routine basis. A PBS for an operational facility might consist largely of information products or service products provided to external customers. However, the general concept of a hierarchical breakdown of products and/or services would still apply.

The rules that apply to a development WBS also apply to a WBS for an operational facility. The techniques for developing a WBS for an operational facility are the same, except that services such as maintenance and user support are added to the PBS, and services such as systems engineering, integration and verification may not be needed.

4.3.3 Common Errors in Developing a WBS

There are three common errors found in WBSs:

- *Error 1:* The WBS describes functions, not products. This makes the project manager the only one formally responsible for products.

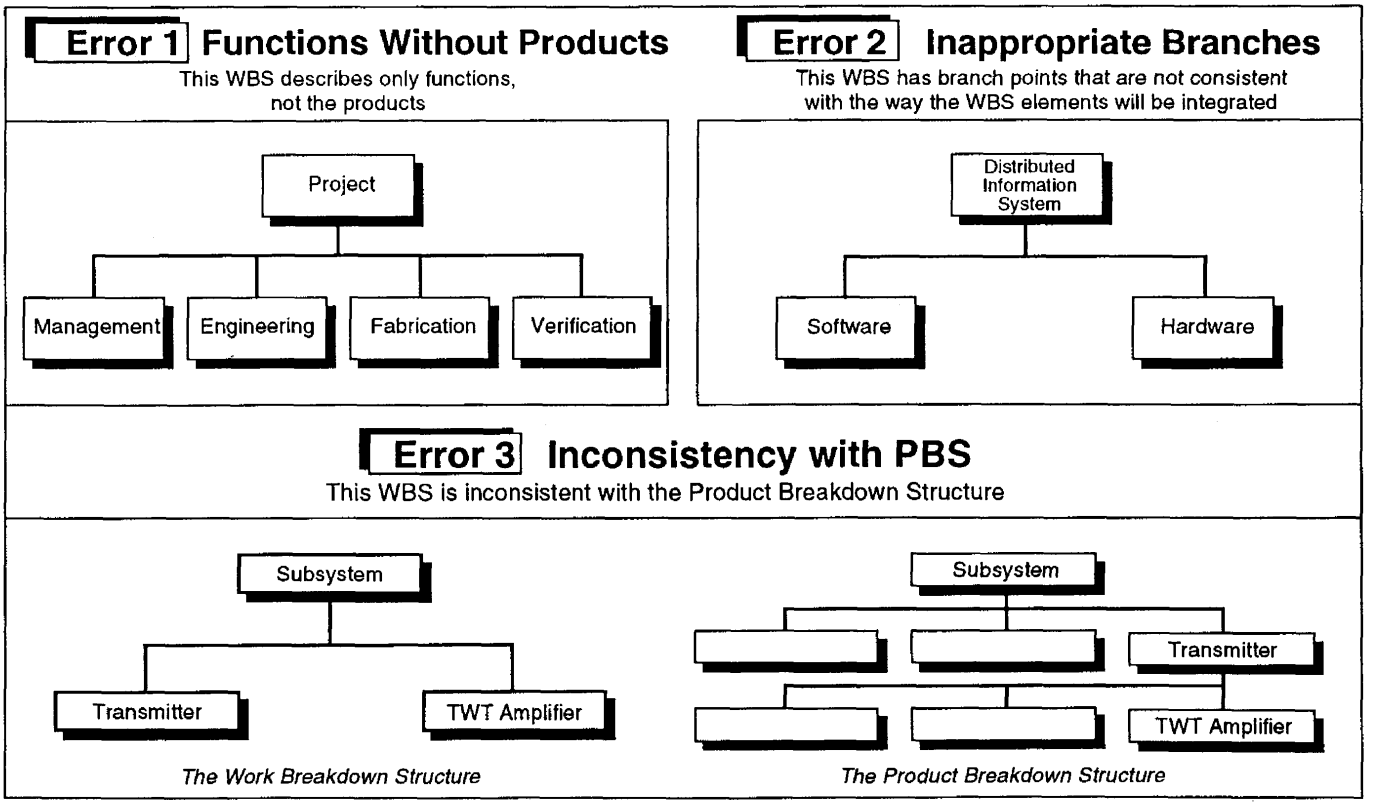


Figure 11 — Examples of WBS Development Errors.

- *Error 2:* The WBS has branch points that are not consistent with how the WBS elements will be integrated. For instance, in a flight operations system with a distributed architecture, there is typically software associated with hardware items that will be integrated and verified at lower levels of a WBS. It would then be inappropriate to separate hardware and software as if they were separate systems to be integrated at the system level. This would make it difficult to assign accountability for integration and to identify the costs of integrating and testing components of a system.
- *Error 3:* The WBS is inconsistent with the PBS. This makes it possible that the PBS will not be fully implemented, and generally complicates the management process.

Some examples of these errors are shown in Figure 11. Each one prevents the WBS from successfully performing its roles in project planning and organizing. These errors are avoided by using the WBS development techniques described above.

4.4 Scheduling

Products described in the WBS are the result of activities that take time to complete. An orderly and efficient systems engineering process requires that these activities take place in a way that respects the underlying time-precedence relationships among them. This is accomplished by creating a *network schedule*, which explicitly takes into account the dependencies of each activity on other activities and receivables from outside sources. This section discusses the role of scheduling and the techniques for building a complete network schedule.

4.4.1 Role of Scheduling

Scheduling is an essential component of planning and managing the activities of a project. The process of creating a network schedule can lead to a much better understanding of what needs to be done, how long it will take, and how each element of the project WBS might affect other elements. A complete network schedule can be used to calculate how long it will take to complete a project, which activities determine that duration (i.e., critical path activities), and how much spare time (i.e., float) exists for all the other activities of the project. (See sidebar on critical path and float calculation.) An understanding of the project's schedule is a prerequisite for accurate project budgeting.

Critical Path and Float Calculation

The *critical path* is the sequence of activities that will take the longest to accomplish. Activities that are not on the critical path have a certain amount of time that they can be delayed until they, too are on a critical path. This time is called *float*. There are two types of float, path float and free float. Path float is where a sequence of activities collectively have float. If there is a delay in an activity in this sequence, then the path float for all subsequent activities is reduced by that amount. Free float exists when a delay in an activity will have no effect on any other activity. For example, if activity A can be finished in 2 days, and activity B requires 5 days, and activity C requires completion of both A and B, then A would have 3 days of free float.

Float is valuable. Path float should be conserved where possible, so that a reserve exists for future activities. Conservation is much less important for free float.

To determine the critical path, there is first a "forward pass" where the earliest start time of each activity is calculated. The time when the last activity can be completed becomes the end point for that schedule. Then there is a "backward pass", where the latest possible start point of each activity is calculated, assuming that the last activity ends at the end point previously calculated. Float is the time difference between the earliest start time and the latest start time of an activity. Whenever this is zero, that activity is on a critical path.

Keeping track of schedule progress is an essential part of controlling the project, because cost and technical problems often show up first as schedule problems. Because network schedules show how each activity affects other activities, they are essential for predicting the consequences of schedule slips or accelerations of an activity on the entire project. Network scheduling systems also help managers accurately assess the impact of both technical and resource changes on the cost and schedule of a project.

4.4.2 Network Schedule Data and Graphical Formats

Network schedule data consist of:

- Activities
- Dependencies between activities (e.g., where an activity depends upon another activity for a receivable)

- Products or milestones that occur as a result of one or more activities
- Duration of each activity.

A *work flow diagram* (WFD) is a graphical display of the first three data items above. A network schedule contains all four data items. When creating a network schedule, graphical formats of these data are very useful. Two general types of graphical formats, shown in Figure 12, are used. One has *activities-on-arrows*, with products and dependencies at the beginning and end of the arrow. This is the typical format of the Program Evaluation and Review Technique (PERT) chart. The second, called *precedence diagrams*, has boxes that represent activities; dependencies are then shown by arrows. Due to its simpler visual format and reduced requirements on computer resources, the precedence diagram has become more common in recent years.

The precedence diagram format allows for simple depiction of the following logical relationships:

- Activity B begins when Activity A begins (Start-Start, or SS)
- Activity B begins only after Activity A ends (Finish-Start, or FS)
- Activity B ends when Activity A ends (Finish-Finish, or FF).

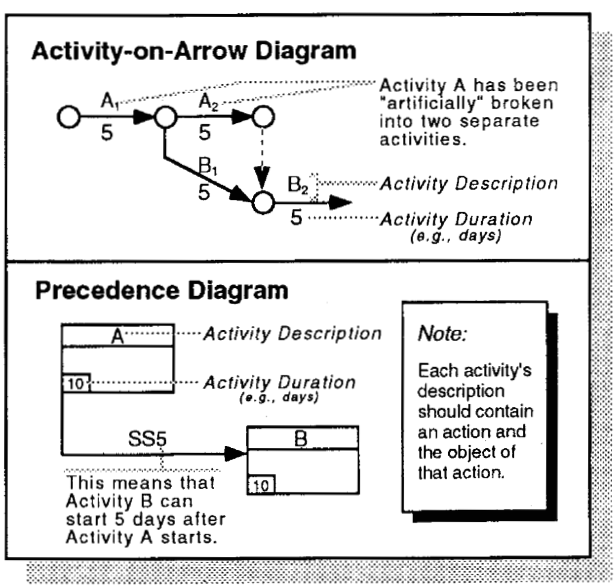


Figure 12 — Activity-on-Arrow and Precedence Diagrams for Network Schedules.

Each of these three activity relationships may be modified by attaching a lag (+ or –) to the relationship, as shown in Figure 12.

It is possible to summarize a number of low-level activities in a precedence diagram with a single activity. This is commonly referred to as hammocking. One takes the initial low-level activity, and attaches a summary activity to it using the first relationship described above. The summary activity is then attached to the final low-level activity using the third relationship described above. Unless one is hammocking, the most common relationship used in precedence diagrams is the second one mentioned above. The activity-on-arrow format can represent the identical time-precedence logic as a precedence diagram by creating artificial events and activities as needed.

4.4.3 Establishing a Network Schedule

Scheduling begins with project-level schedule objectives for delivering the products described in the upper levels of the WBS. To develop network schedules that are consistent with the project's objectives, the following six steps are applied to each cost account at the lowest available level of the WBS.

Step 1: Identify activities and dependencies needed to complete each WBS element. Enough activities should be identified to show exact schedule dependencies between activities and other WBS elements. It is not uncommon to have about 100 activities identified for the first year of a WBS element that will require 10 work-years per year. Typically, there is more schedule detail for the current year, and much less detail for subsequent years. Each year, schedules are updated with additional detail for the current year. This first step is most easily accomplished by:

- Ensuring that the cost account WBS is extended downward to describe all significant products, including documents, reports, hardware and software items
- For each product, listing the steps required for its generation and drawing the process as a work flow diagram
- Indicating the dependencies among the products, and any integration and verification steps within the work package.

Step 2: Identify and negotiate external dependencies. External dependencies are any receivables from outside of the cost account, and any deliverables that go outside of the cost account. Informal negotiations should

occur to ensure that there is agreement with respect to the content, format, and labeling of products that move across cost account boundaries. This step is designed to ensure that lower level schedules can be integrated.

Step 3: Estimate durations of all activities. Assumptions behind these estimates (workforce, availability of facilities, etc.) should be written down for future reference.

Step 4: Enter the schedule data for the WBS element into a suitable computer program to obtain a network schedule and an estimate of the critical path for that element. (There are many commercially available software packages for this function.) This step enables the cognizant engineer, team leader, and/or system engineer to review the schedule logic. It is not unusual at this point for some iteration of steps 1 to 4 to be required in order to obtain a satisfactory schedule. Often too, reserve will be added to critical path activities, often in the form of a dummy activity, to ensure that schedule commitments can be met for this WBS element.

Step 5: Integrate schedules of lower level WBS elements, using suitable software, so that all dependencies between WBS elements are correctly included in a project network. It is important to include the impacts of holidays, weekends, etc. by this point. The critical path for the project is discovered at this step in the process.

Step 6: Review the workforce level and funding profile over time, and make a final set of adjustments to logic and durations so that workforce levels and funding levels are reasonable. Adjustments to the logic and the durations of activities may be needed to converge to the schedule targets established at the project level. This may include adding more activities to some WBS element, deleting redundant activities, increasing the workforce for some activities that are on the critical path, or finding ways to do more activities in parallel, rather than in series. If necessary, the project level targets may need to be adjusted, or the scope of the project may need to be reviewed. Again, it is good practice to have some schedule reserve, or float, as part of a risk mitigation strategy.

The product of these last steps is a feasible baseline schedule for each WBS element that is consistent with the activities of all other WBS elements, and the sum of all these schedules is consistent with both the technical scope and the schedule goals for the project. There should be enough float in this integrated master schedule so that schedule and associated cost risk are acceptable to the project and to the project's customer. Even when this is done, time estimates for many WBS elements will have been underestimated, or work on some WBS elements will not start as early as had been originally assumed due to late arrival of receivables. Consequently, replanning is almost always needed to meet the project's goals.

4.4.4 Reporting Techniques

Summary data about a schedule is usually described in Gantt charts. A good example of a Gantt chart is shown in Figure 13. (See sidebar on Gantt chart features.) Another type of output format is a table that shows the float and recent changes in float of key activities. For example, a project manager may wish to know precisely how much schedule reserve has been consumed by critical path activities, and whether reserves are being consumed or are being preserved in the latest reporting period. This table provides information on the rate of change of schedule reserve.

4.4.5 Resource Leveling

Good scheduling systems provide capabilities to show resource requirements over time, and to make adjustments so that the schedule is feasible with respect to resource constraints over time. Resources may include workforce level, funding profiles, important facilities, etc. Figure 14 shows an example of an unleveled resource profile. The objective is to move the start dates of tasks that have float to points where the resource profile is feasible. If that is not sufficient, then the assumed task durations for resource-intensive activities should be reexamined and, accordingly, the resource levels changed.

4.5 Budgeting and Resource Planning

Budgeting and resource planning involves the establishment of a reasonable project baseline budget, and the capability to analyze changes to that baseline resulting from technical and/or schedule changes. The project's WBS, baseline schedule and budget should be viewed by the system engineer as mutually dependent, reflecting the technical content, time, and cost of meeting the project's goals and objectives.

The budgeting process needs to take into account whether a fixed cost cap or cost profile exists. When no such cap or profile exists, a baseline budget is developed from the WBS and network schedule. This specifically involves combining the project's workforce and other resource needs with the appropriate workforce rates and other financial and programmatic factors to obtain cost element estimates. These *elements of cost* include:

Desirable Features in Gantt Charts

The Gantt chart shown in Figure 13 (below) illustrates the following desirable features:

- A heading that describes the WBS element, the responsible manager, the date of the baseline used, and the date that status was reported.
- A milestone section in the main body (lines 1 and 2)
- An activity section in the main body. Activity data shown includes:
  a. WBS elements (lines 3, 5, 8, 12, 16, and 20)
  b. Activities (indented from WBS elements)
  c. Current plan (shown as thick bars)
  d. Baseline plan (same as current plan, or if different, represented by thin bars under the thick bars)
  e. Status line at the appropriate date
  f. Slack for each activity (dashed lines above the current plan bars)
  g. Schedule slips from the baseline (dashed lines below the milestone on line 12)
- A note section, where the symbols in the main body can be explained.

This Gantt chart shows only 23 lines, which is a summary of the activities currently being worked for this WBS element. It is appropriate to tailor the amount of detail reported to those items most pertinent at the time of status reporting.

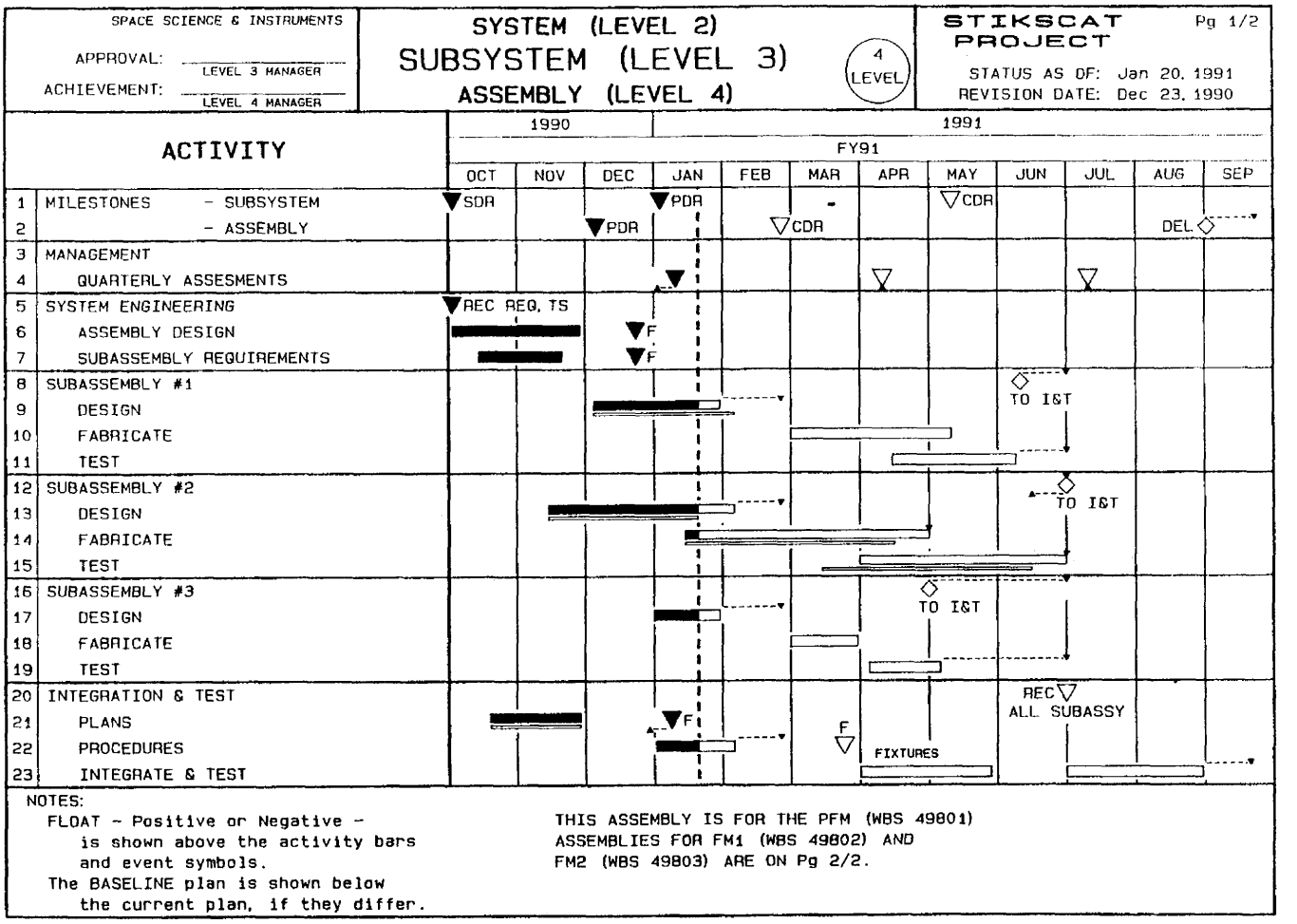


Figure 13 — An Example of a Gantt Chart.

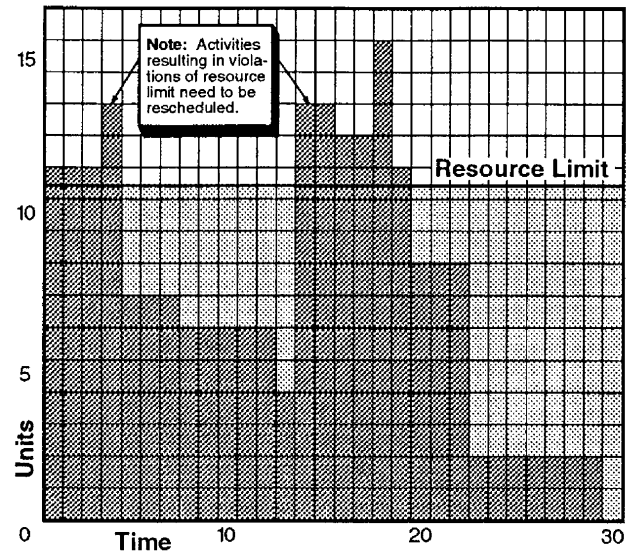


Figure 14 — An Example of an Unleveled Resource Profile.

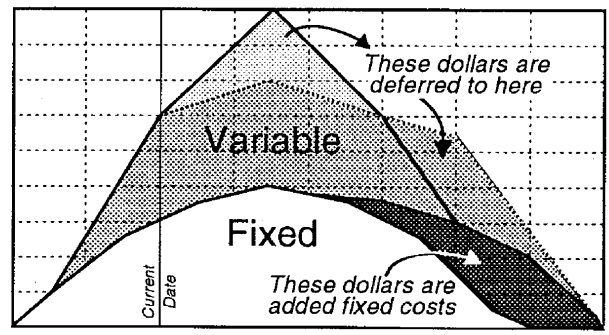
- Direct labor costs
- Overhead costs
- Other direct costs (travel, data processing, etc.)
- Subcontract costs
- Material costs
- General and administrative costs
- Cost of money (i.e., interest payments, if applicable)
- Fee (if applicable)
- Contingency.

When there is a cost cap or a fixed cost profile, there are additional logic gates that must be satisfied before the system engineer can complete the budgeting and planning process. A determination needs to be made whether the WBS and network schedule are feasible with respect to mandated cost caps and/or cost profiles. If not, the system engineer needs to recommend the best approaches for either stretching out a project (usually at an increase in the total cost), or descoping the project's goals and objectives, requirements, design, and/or implementation approach. (See sidebar on schedule adjustments.)

Whether a cost cap or fixed cost profile exists, it is important to control costs after they have been baselined. An important aspect of cost control is project cost and schedule status reporting and assessment, methods for which are discussed in Section 4.9.1 of this handbook. Another is cost and schedule risk planning, such as developing risk avoidance and work-around strategies. At the project level, budgeting and resource planning must also ensure that an adequate level of contingency funds are included to deal with unforeseen events. Some risk management methods are discussed in Section 4.6.

Assessing the Effect of Schedule Slippage

Certain elements of cost, called *fixed costs*, are mainly time related, while others, called *variable costs*, are mainly product related. If a project's schedule is slipped, then the fixed costs of completing it increase. The variable costs remain the same in total (excluding inflation adjustments), but are deferred downstream, as in the figure below.



To quickly assess the effect of a simple schedule slippage:

- Convert baseline budget plan from nominal (real-year) dollars to constant dollars
- Divide baseline budget plan into fixed and variable costs
- Enter schedule slip implementation
- Compute new variable costs including any workforce disruption costs
- Repeat last two steps until an acceptable implementation is achieved
- Compute new fixed costs
- Sum new fixed and variable costs
- Convert from constant dollars to nominal (real-year) dollars.

4.6 Risk Management

Risk management comprises purposeful thought to the sources, magnitude and mitigation of risk, and actions directed toward its balanced reduction. As such, risk management is an integral part of project management, and contributes directly to the objectives of systems engineering.

NASA policy objectives with regard to project risks are expressed in NMI 8070.4A, *Risk Management Policy*. These are to:

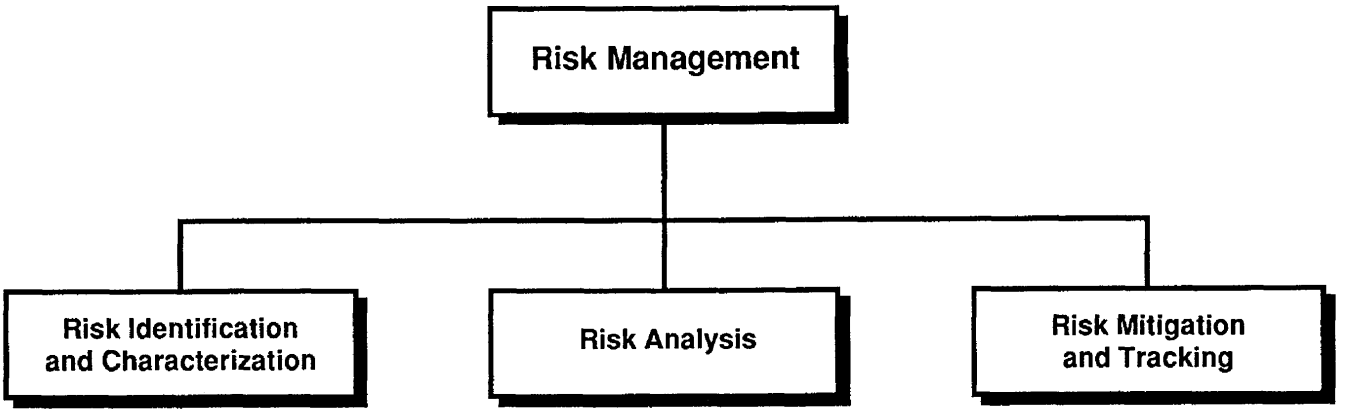


Figure 15 — Risk Management Structure Diagram.

- Provide a disciplined and documented approach to risk management throughout the project cycle
- Support management decision making by providing integrated risk assessments (i.e., taking into account cost, schedule, performance and safety concerns)
- Communicate to NASA management the significance of assessed risk levels and the decisions made with respect to them.

There are a number of actions the system engineer can take to effect these objectives. Principal among them is planning and completing a well-conceived *risk management program*. Such a program encompasses several related activities during the systems engineering process. The structure of these activities is shown in Figure 15.

Risk

The term *risk* has different meanings depending on the context. Sometimes it simply indicates the degree of variability in the outcome or result of a particular action. In the context of risk management during the systems engineering process, the term denotes a combination of both the likelihood of various outcomes *and* their distinct consequences. The focus, moreover, is generally on undesired or unfavorable outcomes such as the risk of a technical failure, or the risk of exceeding a cost target.

The first is the process of identifying and characterizing the project's risks. The objective of this step is to understand what uncertainties the project faces, and which among them should be given greater attention. This is accomplished by categorizing (in a consistent manner) uncertainties by the likelihood of occurrence (e.g., high, medium, or low), and separately, according to severity of consequences. This categorization forms the basis for ranking uncertainties by their relative riskiness. Uncertainties with both high likelihood and severely adverse consequences are ranked higher than those without these characteristics. The primary methods used in this process are qualitative; hence in systems engineering literature, this step is sometimes called qualitative risk assessment. The output of this step is a list of significant risks (by phase) to be given specific management attention.

In some projects, qualitative methods are adequate for making risk management decisions; in others, these methods are not precise enough to understand the magnitude of the problem, or to allocate scarce risk reduction resources. Risk analysis is the process of quantifying both the likelihood of occurrence and consequences of potential future events (or "states of nature" in some texts). The system engineer needs to decide whether risk identification and characterization are adequate, or whether the increased precision of risk analysis is needed for some uncertainties. In making that determination, the system engineer needs to balance the (usually) higher cost of risk analysis against the value of the additional information.

Risk mitigation is the formulation, selection, and execution of strategies designed to economically reduce risk. Tracking the effectivity of these strategies is also considered part of risk mitigation. Risk mitigation is often a challenge because efforts and expenditures to reduce one type of risk may increase another type. (Some have called

Table 1 — Techniques of Risk Management.

Risk Identification and Characterization	Risk Analysis	Risk Mitigation and Tracking
Expert interviews	Decision analysis	Watchlists/milestones
Independent assessment (cost, schedule and technical)	Probabalistic Risk Assessment (PRA)	Contingency planning
Risk templates (e.g., DoD 4245.7-M)	Probabilistic network schedules (e.g., PERT)	Critical items/issues lists
Lessons learned files from previous projects	Probabilistic cost and effectiveness models (e.g., Monte Carlo models)	Cost/schedule control systems and Technical Performance Measure (TPM) tracking
FMECAs/FMEAs/Digraphs		

this the systems engineering equivalent of the Heisenberg Uncertainty Principle in quantum mechanics.) The ability (or necessity) to trade one type of risk for another means that the project manager and the system engineer need to understand the system-wide effects of various strategies in order to make a rational allocation of resources.

Several techniques have been developed for each of these risk management activities. The principal ones, which are shown in Table 1, are discussed in Sections 4.6.2 through 4.6.4. The system engineer needs to choose the techniques that best fit the unique requirements of each project.

A risk management program is needed throughout the project cycle. In keeping with the doctrine of successive refinement, its focus, however, moves from the "big picture" in the early phases of the project cycle (Phases A and B) to more specific issues during product design and development (Phases C and D). During pre-operations and operations (Phases E and F), the focus changes again. A good risk management program is always forward-looking. In other words, a risk management program should address the project's on-going risk issues and future uncertainties. As such, it is a natural part of concurrent engineering.

Risk management activities for a project should be documented in a risk management program plan. That plan, which elaborates on the SEMP and should be updated at each phase of the project cycle, contains:

- The project's overall risk policy and objectives
- The programmatic aspects of the risk management activities (i.e., responsibilities, resources, schedules and milestones, etc.)
- A description of the tools and techniques to be used for risk identification and characterization, risk analysis, and risk mitigation
- A description of the role of risk management with respect to systems analysis, baseline change control, formal reviews, and status reporting and assessment
- Documentation requirements for each risk management product and action.

The level of risk management activities should be consistent with the project's overall risk policy established in conjunction with its NASA Headquarters program office. At present, formal guidelines for the classification of projects with respect to overall risk policy do not exist; such guidelines exist only for NASA payloads. These are promulgated in NMI 8010.1A, *Classification of NASA Payloads, Attachment A,* which is reproduced as Appendix B.5.

4.6.1 Types of Risks

There are several ways to describe the various types of risk a project manager/system engineer faces. Traditionally, project managers and system engineers have attempted to divide risks into three or four broad categories — namely, cost, schedule, technical, and sometimes, safety (and/or hazard) risks. More recently, others have entered the lexicon, including the categories of organizational, management, acquisition, supportability, political, and programmatic risks. These newer categories reflect the expanded set of concerns of project managers and system engineers who must operate in the current NASA environment. Some of these newer categories also represent supersets of other categories. For example, the Defense Systems Management College (DSMC) Systems Engineering Management Guide wraps "funding, schedule, contract relations, and political risks" into the broader category of programmatic risks. While these terms are useful in informal discussions, there appears to be no formal taxonomy free of ambiguities. One reason, mentioned above, is that often one type of risk can be exchanged for another. A second reason is that some of these categories move together, as for example, cost risk and political risk (e.g., the risk of project cancellation).

Another way some have categorized risk is by the degree of mathematical predictability in its underlying uncertainty. The distinction has been made between an uncertainty that has a known probability distribution, with known or estimated parameters, and one in which the underlying probability distribution is either not known, or its parameters cannot be objectively quantified.

An example of the first kind of uncertainty occurs in the unpredictability of the spares upmass requirement for alternative Space Station *Freedom* designs. While the requirement is stochastic in any particular logistics cycle, the probability distribution can be estimated for each design from reliability theory and empirical data. Examples

of the second kind of uncertainty occur in trying to predict whether a Shuttle accident will make resupply of *Freedom* impossible for a period of time greater than x months, or whether life on Mars exists.

Modern subjectivist (also known as *Bayesian*) probability theory holds that the probability of an event is the degree of belief that a person has that it will occur, given his/her state of information. As that information improves (e.g., through the acquisition of data or experience), the subjectivist's estimate of a probability should converge to that estimated as if the probability distribution were known. In the examples of the previous paragraph, the only difference, then, is the probability estimator's perceived state of information. Consequently, subjectivists find the distinction between the two kinds of uncertainty of little or no practical significance. The implication of the subjectivist's view for risk management is that, even with little or no data, the system engineer's subjective probability estimates form a valid basis for risk decision making.

4.6.2 Risk Identification and Characterization Techniques

A variety of techniques are available for risk identification and characterization. The thoroughness with which this step is accomplished is an important determinant of the risk management program's success.

Expert Interviews. When properly conducted, expert interviews can be a major source of insight and information on the project's risks in the expert's area of knowledge. One key to a successful interview is in identifying an expert who is close enough to a risk issue to understand it thoroughly, and at the same time, able (and willing) to step back and take an objective view of the probabilities and consequences. A second key to success is advanced preparation on the part of the interviewer. This means having a list of risk issues to be covered in the interview, developing a working knowledge of these issues as they apply to the project, and developing methods for capturing the information acquired during the interview.

Initial interviews may yield only qualitative information, which should be verified in follow-up rounds. Expert interviews are also used to solicit quantitative data and information for those risk issues that qualitatively rank high. These interviews are often the major source of inputs to risk analysis models built using the techniques described in Section 4.6.3.

Independent Assessment. This technique can take several forms. In one form, it can be a review of project documentation, such as Statements of Work, acquisition plans, verification plans, manufacturing plans, and the SEMP. In another form, it can be an evaluation of the WBS for completeness and consistency with the project's schedules. In a third form, an independent assessment can be an independent cost (and/or schedule) estimate from an outside agency and/or group.

Risk Templates. This technique consists of examining and then applying a series of previously developed risk templates to a current project. Each template generally covers a particular risk issue, and then describes methods for avoiding or reducing that risk. The most-widely recognized series of templates appears in DoD 4245.7-M, *Transition from Development to Production ...Solving the Risk Equation.* Many of the risks and risk responses described are based on lessons learned from DoD programs, but are general enough to be useful to NASA projects. As a general caution, risk templates cannot provide an exhaustive list of risk issues for every project, but they are a useful input to risk identification.

Lessons Learned. A review of the lessons learned files, data, and reports from previous similar projects can produce insights and information for risk identification on a new project. For technical risk identification, as an example, it makes sense to examine previous projects of similar function, architecture, or technological approach. The lessons learned from the *Infrared Astronomical Satellite* (IRAS) project might be useful to the *Space Infrared Telescope Facility* (SIRTF) project, even though the latter's degree of complexity is significantly greater. The key to applying this technique is in recognizing what aspects are analogous in two projects, and what data are relevant to the new project. Even if the the documented lessons learned from previous projects are not applicable at the system level, there may be valuable data applicable at the subsystem or component level.

FMECAs, FMEAs, and Digraphs. Failure Modes, Effects, and Criticality Analysis (FMECA), Failure Modes and Effects Analysis (FMEA), and digraphs are specialized techniques for safety (and/or hazard) risk identification and characterization. These techniques focus on the hardware components that make up the system. According to MIL-STD-1629A, FMECA is "an ongoing procedure by which each potential failure in a system is analyzed to determine the results or effects thereof on the system, and to classify each potential failure mode according to its severity."

Failures are generally classified into four severity categories:

- Category I — Catastrophic failure (possible death or system loss)
- Category II — Critical failure (possible major injury or system damage)
- Category III — Major failure (possible minor injury or mission effectiveness degradation)
- Category IV — Minor failure (requires system maintenance, but does not pose a hazard to personnel or mission effectiveness).

A complete FMECA also includes an estimate of the probability of each potential failure. These probabilities are usually based, at first, on subjective judgment or experience factors from similar kinds of hardware components, but may be refined from reliability data as the system development progresses. An FMEA is similar to an FMECA, but typically excludes the severity classification category.

Digraph analysis is an aid in determining fault tolerance, propagation, and reliability in large, interconnected systems. Digraphs exhibit a network structure and resemble a schematic diagram. The digraph technique permits the integration of data from a number of individual FMECAs/FMEAs, and can be translated into fault trees, described below, if quantitative probability estimates are needed.

4.6.3 Risk Analysis Techniques

The tools and techniques of risk analysis rely heavily on the concept and "laws" (actually, axioms and theorems) of probability. The system engineer needs to be familiar with these in order to appreciate the full power and limitations of these techniques. The products of risk analyses are generally quantitative probability and consequence estimates for various outcomes, more detailed understanding of the dominant risks, and improved capability for allocating risk reduction resources.

Decision Analysis. Decision analysis is one technique to help the individual decision maker deal with a complex set of uncertainties. Using the divide-and-conquer approach common to much of systems engineering, a complex uncertainty is decomposed into simpler ones, which are then treated separately. The decomposition continues until it reaches a level at which either hard information can be brought to bear, or intuition can function effectively. The decomposition can be graphically represented as a *decision tree*. The branch points, called nodes, in a decision tree represent either decision points or chance events. Endpoints of the tree are the potential outcomes. (See the sidebar on a decision tree example for Mars exploration.)

In most applications of decision analysis, these outcomes are generally assigned dollar values. From the probabilities assigned at each chance node, and the dollar value of each outcome, the distribution of dollar values (i.e., consequences) can be derived for each set of decisions. Even large complex decision trees can be represented in currently available decision analysis software. This software can also calculate a variety of risk measures.

In brief, decision analysis is a technique that allows:

- A systematic enumeration of uncertainties and encoding of their probabilities and outcomes
- An explicit characterization of the decision maker's attitude toward risk, expressed in terms of his/her *risk aversion*
- A calculation of the value of "perfect information", thus setting a normative upper bound on information-gathering expenditures
- Sensitivity testing on probability estimates and outcome dollar values.

Probabilistic Risk Assessment (PRA). A PRA seeks to measure the risk inherent in a system's design and operation by quantifying both the likelihood of various possible accident sequences and their consequences. A typical PRA application is to determine the risk associated with a specific nuclear power plant. Within NASA, PRAs are used to demonstrate, for example, the relative safety of launching spacecraft containing RTGs (Radioisotope Thermoelectric Generators).

The search for accident sequences is facilitated by *event trees*, which depict initiating events and combinations of system successes and failures, and *fault trees*, which depict ways in which the system failures represented in an event tree can occur. When integrated, an event tree and its associated fault tree(s) can be used to calculate the probability of each accident sequence. The structure and mathematics of these trees is similar to that for decision trees. The consequences of each accident sequence are generally measured both in terms of direct economic losses and in public health effects. (See sidebar on PRA pitfalls.)

Doing a PRA is itself a major effort, requiring a number of specialized skills other than those provided by reliability engineers and human factors engineers. PRAs also require large amounts of system design data at the component level, and operational procedures data. For additional information on PRAs, the system engineer can ref-

erence the *PRA Procedures Guide* (1983) by the American Nuclear Society and Institute of Electrical and Electronic Engineers (IEEE).

Probabilistic Network Schedules. Probabilistic network schedules, such as PERT (Program Evaluation and Review Technique), permit the duration of each activity to be treated as a random variable. By supplying PERT with the minimum, maximum, and most likely duration for each activity, a probability distribution can be computed for project completion time. This can then be used to determine, for example, the chances that a project (or any set of tasks in the network) will be completed by a given date. In this probabilistic setting, however, a unique critical path may not exist. Some practitioners have also cited difficulties in obtaining meaningful input data for probabilistic network schedules.

Probabilistic Cost and Effectiveness Models. These models offer a probabilistic view of a project's cost and effectiveness outcomes. (Recall Figure 2.) This approach explicitly recognizes that single point values for these variables do not adequately represent the risk conditions inherent in a project. These kinds of models are discussed more completely in Section 5.4.

Probabilistic Risk Assessment Pitfalls

Risk is generally defined in a probabilistic risk assessment (PRA) as the expected value of a consequence function — that is:

$$R = \sum_s P_s C_s$$

where P_s is the probability of outcome s, and C_s is the consequence of outcome s. To attach probabilities to outcomes, event trees and fault trees are developed. These techniques have been used since 1953, but by the late 1970s, they were under attack by PRA practitioners. The reasons include the following:

- Fault trees are limiting because a complete set of failures is not definable.
- Common cause failures could not be captured properly. An example of a common cause failure is one where all the valves in a system have a defect so that their failures are not truly independent.
- PRA results are sometimes sensitive to simple changes in event tree assumptions
- Stated criteria for accepting different kinds of risks are often inconsistent, and therefore not appropriate for allocating risk reduction resources.
- Many risk-related decisions are driven by perceptions, not necessarily objective risk as defined by the above equation. Perceptions of consequences tend to grow faster than the consequences themselves — that is, several small accidents are not perceived as strongly as one large one, even if fatalities are identical.
- There are difficulties in dealing with incommensurables, as for example, lives vs. dollars.

4.6.4 Risk Mitigation and Tracking Techniques

Risk identification and characterization and risk analysis provide a list of significant project risks that require further management attention and/or action. Because risk mitigation actions are generally not costless, the system engineer, in making recommendations to the project manager, must balance the cost (in resources and time) of such actions against their value to the project. Four responses to a specific risk are usually available: (1) deliberately do nothing, and accept the risk, (2) share the risk with a co-participant, (3) take preventive action to avoid or reduce the risk, and (4) plan for contingent action.

The first response is to accept a specific risk consciously. Sometimes, a risk can be shared with a co-participant — that is, with a foreign partner or a contractor. In this situation, the goal is to reduce NASA's risk independent of what happens to total risk, which may go up or down. There are many ways to share risks, particularly cost risks, with contractors. These include various incentive contracts and warranties. The third and fourth responses require that additional specific planning and actions be undertaken.

Typical technical risk mitigation actions include additional (and usually costly) testing of subsystems and systems, designing in redundancy, and building a full engineering model. Typical cost risk mitigation actions include using off-the-shelf hardware and, according to Figure 6, providing sufficient funding during Phases A and B. Major supportability risk mitigation actions include providing sufficient initial spares to meet the system's availability goal and a robust resupply capability (when transportation is a significant factor). For those risks that cannot be mitigated by a design or management approach, the system engineer should recommend the establishment of reasonable financial and schedule contingencies, and technical margins.

Whatever strategy is selected for a specific risk, it and its underlying rationale should be documented in a risk mitigation plan, and its effectivity should be tracked

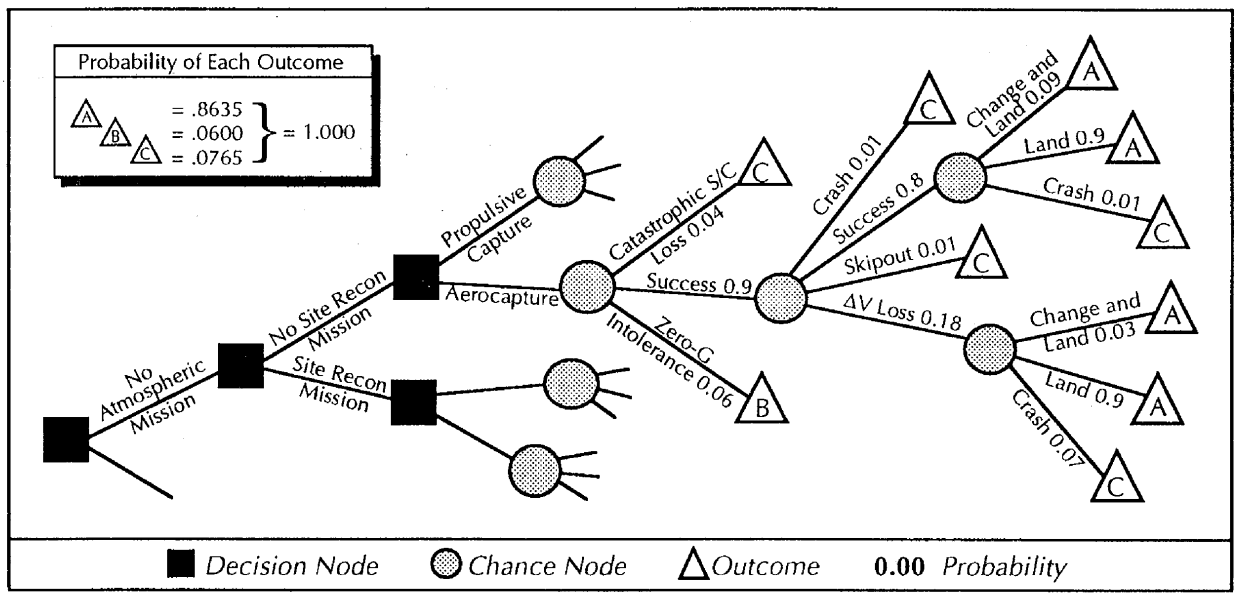
through the project cycle, as required by NMI 8070.4A. The techniques for choosing a (preferred) risk mitigation strategy are discussed in Chapter 5, which deals with the larger role of trade studies and system modeling in general. Some techniques for planning and tracking are briefly mentioned here.

Watchlists and Milestones. A *watchlist* is a compilation of specific risks, their projected consequences, and early indicators of the start of the problem. The risks on the watchlist are those that were selected for management attention as a result of completed risk management activities. A typical watchlist also shows for each specific risk a triggering event or missed milestone (for example, a delay in the delivery of long lead items), the related area of impact (production schedule), and the risk mitigation strategy, to be used in response. The watchlist is periodically reevaluated and items are added, modified, or deleted as appropriate. Should the triggering event occur, the projected consequences should be updated and the risk mitigation strategy revised as needed.

Contingency Planning. This technique is generally used in conjunction with a watchlist. The focus in contingency planning is on developing credible hedges and workarounds, which are activated upon a triggering event. To be credible, hedges often require that additional resources be expended, which provide a return only if the triggering event occurs. In this sense, contingency planning and resources act as a form of project insurance.

An Example of a Decision Tree for Robotic Precursor Missions to Mars

In 1990, the Lunar/Mars Exploration Program Office (LMEPO) at JSC wanted to know how robotic precursor missions might reduce the risk of a manned Mars mission. Structuring the problem as a decision tree allows the effects of different missions and chance events to be systematically and quantitatively evaluated. The portion of the decision tree shown here illustrates the calculation of the probabilities for three distinct outcomes: (A) a successful Mars landing, (B) a safe return without a landing, or (C) a disaster resulting in mission and crew loss, when no atmospheric or site reconnaissance robotic precursor missions were made and aerocapture at Mars was selected. As new information becomes available, the decision tree's data can be reviewed and updated.



Making the same calculations for every branch in the decision tree allows a determination of the best mix of robotic precursor missions as an explicit function of: (a) the contribution of each robotic precursor mission to manned mission risk reduction, (b) the cost, schedule and riskiness of each robotic mission, (c) the value of the manned mission, and (d) the science value of each robotic mission in the absence of a subsequent manned mission. Another benefit of this quantitative approach is that robotic precursors can be traded against other risk mitigation strategies in the manned mission architecture.

For more information on decision analysis, see de Neufville and Stafford, *Systems Analysis for Engineers and Managers*, 1971, and Barclay, et al., *Handbook for Decision Analysis*, 1977.

(The term *contingency* here should not be confused with use of the same term for project reserves.)

Critical Items/Issues Lists. A Critical Items/Issues List (CIL) is similar to a watchlist, and has been extensively on the Shuttle program, to track items with significant system safety consequences. An example is shown as Appendix B.6.

C/SCS and TPM Tracking. Two very important risk tracking techniques — cost and schedule control systems (C/SCS) and Technical Performance Measure (TPM) tracking — are discussed in Sections 4.9.1 and 4.9.2, respectively.

4.6.5 Risk Management: Summary

Uncertainty is a fact of life in systems engineering. To deal with it effectively, the risk manager needs a disciplined approach. In a project setting, a good-practice approach includes efforts to:

- Plan, document, and complete a risk management program
- Identify and characterize risks for each phase of the project; high risks, those for which the combined effects of likelihood and consequences are significant, should be given specific management attention. Reviews conducted throughout in the project cycle should help to force out risk issues.
- Apply qualitative and quantitative techniques to understand the dominant risks and to improve the allocation of risk reduction resources; this may include the development of project-specific risk analysis models such as decision trees and PRAs.
- Formulate and execute a strategy to handle each risk, including establishment, where appropriate, of reasonable financial and schedule contingencies and technical margins
- Track the effectivity of each risk mitigation strategy.

Good risk management requires a team effort — that is, system engineers and managers at all levels of the project need to be involved. However, risk management responsibilities must be assigned to specific individuals. Successful risk management practices often evolve into institutional policy.

4.7 Baseline Management

The *baseline* for a project contains all of the technical requirements and related cost and schedule requirements that are sufficiently mature to be accepted and placed under change control by the NASA project manager. The project baseline consists of two parts: the technical baseline and the business baseline. The system engineer is responsible for managing the technical baseline and ensuring that the technical baseline is consistent with the costs and schedules in the business baseline. Typically, the project control office manages the business baseline.

Baseline management requires the formal agreement of both the buyer and the seller to proceed according to the up-to-date, documented project requirements (as they exist at that phase in the project cycle), and to change the baseline requirements only by a formal change control process. The buyer might be an external funding agency. For example, the buyer for the GOES project is NOAA, and the seller is the NASA GOES project office. Baseline management must be enforced at all levels; in the next level for this same example, the NASA GOES project office is the buyer and the seller is the contractor, the Loral GOES project office.

The project-level system engineer is responsible for ensuring the completeness and technical integrity of the technical baseline. The technical baseline includes:

- Definition of (or specification of) the functional and performance requirements for hardware, software, and operations
- Interface definitions
- Specialty engineering requirements
- Verification plans
- Documentation trees.

Baseline management includes the following techniques:

- Baseline definition and approval
- Configuration control (and version control, if needed)
- Change control
- Traceability
- Data management
- Baseline communication.

4.7.1 Baseline Evolution

The project baseline evolves in discrete steps through the project life cycle. An initial baseline may be

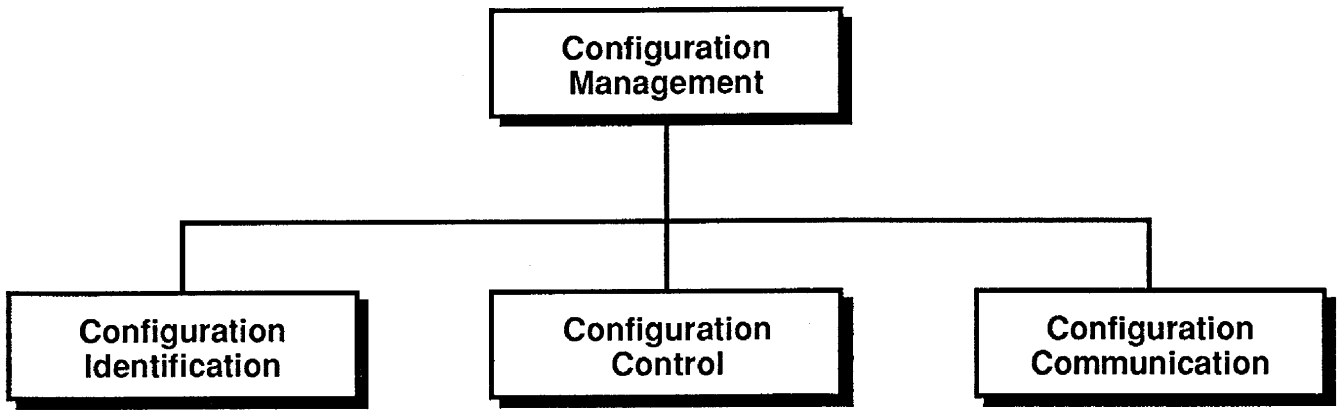


Figure 16 — Configuration Management Structure Diagram.

established when the top-level user requirements expressed in the *Mission Needs Statement* are placed under configuration control. At each interphase control gate, increased technical detail is added to the maturing baseline. For a typical project, there are five sequential technical baselines:

- Functional baseline at Program/Project Requirements Review (PRR, sometimes called development baseline)
- "Design-to" baseline at Preliminary Design Review (PDR)
- "Build-to" (or "code-to") baseline at the Critical Design Review (CDR)
- Production (or "as-built" or "as-coded") baseline at the System Acceptance Review (SAR)
- Operational (or "as-deployed") baseline at Operational Acceptance Review (OAR).

The positions of the five baselines are illustrated in Figure 8. As discussed in Section 3.9, only decisions made along the core of the "vee" in that figure are put under configuration control and included in the approved baseline. Risk management activity (off the core of the vee) must begin early and continue throughout the decomposition process of the project cycle to prove that the core-level decisions are sound. These early detailed studies and tests must be documented and retained in the project archives, but they are not part of the technical baseline.

4.7.2 Configuration Management

Configuration management is the discipline of identifying and formalizing the physical and functional characteristics of a configuration item at discrete points in the product evolution for the purpose of maintaining the integrity of the product and controlling changes to the baseline. As a functional discipline, configuration management manages the documentation of the approved evolution of a product's configuration. Configuration management includes configuration or baseline identification, configuration control, and configuration communication (see Figure 16).

Configuration management is essential to the execution an orderly development process, to enable the modification of an existing design, and to provide for later replication of an existing design. Configuration management often provides the information needed to track the technical progress of the project. (See Section 4.9.1 on Technical Performance Measures.)

Configuration identification of a baseline is evidenced by documentation such as requirements documents, specifications, drawings, code listings, process specifications, and material specifications. Configuration documen-

Change Control Board Conduct

Objective: To review evaluations, and then approve or disapprove proposed changes to the project's technical, operations, or business baselines.

Participants: Project manager (chair), project-level system engineer, managers of each affected organization, configuration manager (secretary), presenters.

Format: Presenter covers recommended change and discusses related system impact. The presentation is reviewed by the system engineer for completeness prior to presentation.

Decision: The CCB members discuss the Change Request (CR) and formulate a decision. Project manager agrees or overrides.

tation is not considered part of the technical baseline until approved by control gate action of the buyer.

Configuration control is the process of controlling changes to any approved baseline by formal action of a change board that is controlled by the same authority that previously approved the baseline. Typically, the change control board meets to consider change requests to the business or technical baselines of the project. The project manager is usually the board chair, and the configuration manager the secretary, who skillfully guides the process and records the official events of the process.

In a change control board forum, a number of issues should be addressed:

- What is the proposed change?
- What is the reason for the change?
- What is the design impact?
- What is the effectiveness or performance impact?
- What is the schedule impact?
- What is the project life-cycle cost impact?
- What is the impact of not making the change?
- What is the risk of making the change?
- What is the impact on operations?
- What is the impact to support equipment and services?
- What is the impact on spares requirements?
- What is the effectivity of the change?
- What documentation is affected by the change?
- Is the buyer supportive of the change?

A review of this information should lead to a well-informed decision. When this information is not available to the change control board, unfounded decisions are made, often with negative consequences to the project.

Configuration control always includes the management of approved baseline documentation, so configuration control is required on a no-change project as well as a frequently changing one. Configuration management and configuration control embrace the function of data management, which ensures that only up-to-date baseline information is available to the project staff. The data management function also encompasses managing and archiving supporting analyses and trade study data, and keeping it convenient for project use.

Configuration verification is part of configuration control. It ensures that the resulting products conform to the intentions of the designers and to the standards established by preceding approved baselines. Each control gate serves to review and challenge the data presented for conformance to the previously established baseline constraints. The Physical Configuration Audit control gate verifies that the physical configuration of the product corresponds to the "build-to" (or "code-to") documentation previously approved at the CDR. The Functional Configuration Audit control gate verifies that the acceptance test results are consistent with the test requirements previously approved at the PDR and CDR. The Formal Qualification Review control gate verifies that the "as-built" product is consistent with the "as-built" or "as-coded" documentation and describes the ultimate configuration of the product. This review follows all modifications needed to implement qualification-caused corrective actions.

For disciplined software development, additional configuration control methods are recommended:

- Computer Resources Working Group (CRWG) — ensures the development environment is adequate for the job
- Software Configuration Review Board — change board for software baseline changes
- Software Development Library — management-controlled repository for software development documentation and tools
- Software Development Folder (SDF) — developer-controlled repository for development documentation and tools.

The configuration manager performs the following functions:

- Conceives, documents and manages the configuration management system
- Acts as secretary of the change control board (controls the change approval process)
- Controls changes to baseline documentation
- Controls release of baseline documentation
- Initiates configuration verification audits.

Configuration communication is the process of conveying to all involved parties the approved baseline progression in a timely manner. This is essential to ensure that developers only pursue options that are compatible with the approved baseline. Communication also keeps developers knowledgeable of the approved baseline and the necessity of approaching the change control board for approval of any deviations considered necessary to further develop the system.

The project's approach to configuration management should be documented in the project's Configuration Management Plan. A sample outline for this plan is illustrated in Appendix B.4.

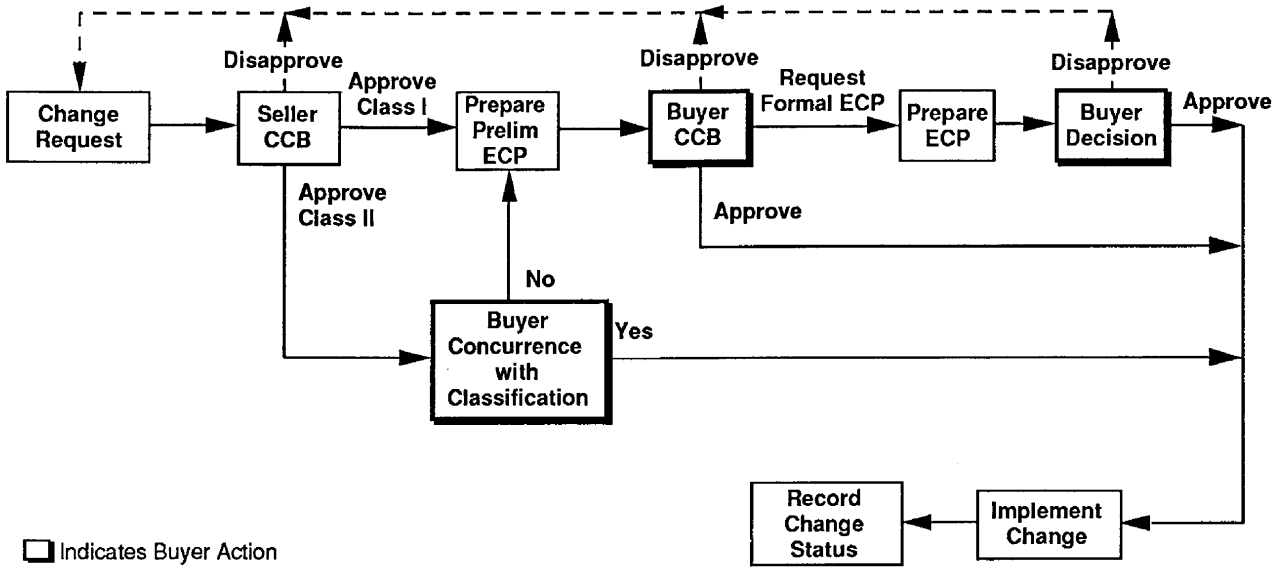


Figure 17 — Contract Change Control Process.

4.7.3 Change Control and Version Control

Once a baseline is placed under change control, any change requires the approval of the change control board. The project manager chairs the change control board, while the system engineer or configuration manager is responsible for reviewing all material for completeness before it is presented to the board, and for ensuring that all affected organizations are represented in the change control board forum.

Change control is essential at both the contractor and NASA Center levels. Changes determined to be Class 1 to the contractor must be referred to the NASA project manager for resolution. This process is described in Figure 17. The use of a preliminary Engineering Change Proposal (ECP) to forewarn of an impending change provides the project manager with sufficient preliminary information to determine whether the contractor should spend NASA contract funds on a formal ECP. This technique is designed to save significant contract dollars.

Class 1 changes affect the approved baseline and hence the product version identification. Class 2 changes are editorial changes or internal changes not "visible" to the external interfaces.

Overly formalized systems can become so burdensome that members of the project team may try to circumvent the process. It is essential that the formality of the change process be appropriately tailored to the needs of each project. However, there must always be an effective change control process on every project.

For software projects, it is routine to use version control for both pre-release and post-release deliverable systems. It is equally important to maintain version control for hardware-only systems.

Approved changes on a development project that has only one deliverable obviously are only applicable to that one deliverable item. However, for projects that have multiple deliverables of "identical" design, changes may become effective on the second or subsequent production articles. In such a situation, the change control board must decide the effectivity of the change, and the configuration control system must maintain version control and identification of the "as-built" configuration for each article. Incremental implementation of changes is common in projects that have a deliberate policy of introducing product or process improvements. As an example, the original 1972 plan held that each of the Space Shuttle orbiters would be identical. In reality, each of the orbiters is different, driven primarily by the desire to achieve the original payload requirement of 65,000 pounds. Proper version control documentation has been essential to the sparing, fielding, and maintenance of the operational fleet.

4.7.4 Data Management and Requirements Traceability

Data management is an essential and associated function to configuration management. Data management ensures that official baseline data is retained, available, and distribution-controlled for all official project use. Data management is essentially the official project library and reference desk.

The data manager performs the following functions:

- Conceives, documents and manages the documentation management system
- Manages changes to baseline documentation
- Manages the release of baseline documentation
- Manages the project library.

Before the project team can produce a tangible product, engineering must produce descriptions of the system using words, icons (drawings), and numbers (i.e., symbolic information). The project team *must* have a common understanding of the words and icons in order to be able to go from an idea to a properly functioning system.

Since the system engineer spends time working with information about the system rather than the system itself, there are several vital characteristics the symbolic information must have. First the information must be *shareable*. Whether it is in electronic or paper form, the data must be readily available, in the most recently approved version, to all members of the team.

Second, symbolic information must be *durable*. This means that it must be recalled accurately every time and represent the most current version of the baseline. The baseline information cannot change or degrade with repeated access of the data base or paper files, and cannot degrade with time. This is a non-trivial statement, since poor data management practices (e.g., allowing someone to borrow the only copy of a document or drawing) can allow controlled information to become lost. Also, the material must be retained for the life of the program (and possibly beyond), and a complete set of documentation for each baseline change must be retained.

Third, the symbolic information must be *traceable* upward and downward. A data base must be developed and maintained to show the parentage of any requirement. The data base must also be able to display all children derived from a given requirement. Finally, traceability must be provided to engineering reports that document trade study results and other decisions that played a key role in the flowdown of requirements.

It is the responsibility of the system engineer to ensure the active approved baseline is communicated to all those relying on it. This technique keeps all participants apprised as to the distinction between what is frozen under formal change control and what can still be decided without change control board approval.

4.8 Reviews, Audits and Control Gates

The intent and policy for reviews, audits and control gates should be developed during Phase A and defined in the Project Implementation Plan. The specific implementation of these activities should be consistent with, though not limited to, the types of reviews and audits described in this section. The same tailoring applies to the timing of reviews, audits and control gates. See the NASA Project Cycle chart (Figure 5) and the Technical Aspect of the NASA Project Cycle chart (Figure 9) for guidance as to when these relationships should be formed.

4.8.1 Purpose and Definitions

The purpose of a *review* is to furnish the forum and process to provide NASA management and their contractors assurance that the most satisfactory approach, plan or design has been selected, that a configuration item has been produced to meet the specified requirements, or that a configuration item is ready. Reviews (technical or management) are scheduled to communicate an approach, demonstrate an ability to meet requirements, or establish status.

Project Termination

It should be noted that project termination, while usually disappointing to project personnel, may be a proper reaction to changes in external conditions or to an improved understanding of the system's projected cost-effectiveness.

Reviews help to develop a better understanding among task or project participants, open communication channels, alert participants and management of problems, and open avenues for solutions.

The purpose of an *audit* is to provide NASA management and its contractors a thorough examination of adherence to program or project policies, plans, requirements and specifications. Audits are the systematic examination of tangible evidence to determine adequacy, validity and effectiveness of the activity or documentation under review. An audit may examine documentation of policies and procedures, as well as verify adherence to them.

The purpose of a *control gate* is to provide a scheduled event (either a review or an audit) that NASA management will use to make program or project go/no-go decisions. A control gate is a management event in the project cycle that is of sufficient importance to be identified, defined and included in the project schedule. It requires formal examination to evaluate project status and to obtain approval to proceed to the next management event according to the Project Implementation Plan.

4.8.2 General Principles for Reviews

Review Boards. The convening authority, who supervises the manager of the activity being reviewed, normally appoints the review board chair. Unless there are compelling technical reasons to the contrary, the chair should not be directly associated with the project or task under review. The convening authority also names the review board members. The majority of the members should not be directly associated with the program or project under review.

Internal Reviews. During the course of a project or task, it is necessary to conduct internal reviews that present technical approaches, trade studies, analyses, and problem areas to a peer group for evaluation and comment. The timing, participants, and content of these reviews is normally defined by the project manager or the manager of the performing organization. Internal reviews are also held prior to participation in a formal, control gate review.

The internal reviews provide an excellent means for controlling the technical progress of the project. They also should be used to ensure that all interested parties are involved in the design/development process, early on, and throughout the process. Thus, representatives from areas such as manufacturing and quality assurance should attend the internal reviews as active participants. They can then, for example, ensure that the design is producible and that quality is managed through the project cycle.

In addition, some organizations utilize a *Red Team*. This is an internal, independent, peer-level review conducted to identify any deficiencies in requests for proposals, proposal responses, documentation, or presentation material prior to its release. The project or task manager is responsible for establishing the Red Team membership and for deciding which of their recommendations are to be implemented.

Review Presentation Material. Presentations using existing documentation such as specifications, drawings, analyses and reports may be adequate. Copies of any prepared materials (such as viewgraphs) should be provided to the review board and meeting attendees. Background information and review presentation material of use to board members should be distributed to the members early enough to enable them to examine it prior to the review. For major reviews, this time may be as long as 30 calendar days.

Review Conduct. All reviews should consist of oral presentations of the applicable project requirements and the approaches, plans or designs that satisfy those requirements. These presentations normally are given by the cognizant design engineer or his immediate supervisor.

It is highly recommended that in addition to the review board, the review audience include project personnel (NASA and contractor) not directly associated with the design being reviewed. This is required to utilize their cross-discipline expertise to identify any design shortfalls or recommend design improvements. The review audience should also include non-project specialists in the area under review, and specialists in manufacturing and fabrication, testing, quality assurance, reliability and safety. Some reviews may also require the presence of both the contractor's and NASA's contracting officers.

Prior to and during the review, board members and review attendees may submit requests for action or engineering change requests (ECRs) that document a concern, deficiency or recommended improvement in the presented approach, plan or design. Following the review, these are screened by the review board to consolidate them and to ensure that the chair and cognizant manager(s) understand the intent of the requests. It is the responsibility of the review board to ensure that adequate closure responses for each of the action requests are obtained.

Post Review Report. The review board chair has the responsibility to develop, where necessary, a consensus of the findings of the board, including an assessment of the risks associated with problem areas, and develop recommendations for action. The chair will submit, on a timely basis, a written report, including recommendations for action, to the convening authority with copies to the cognizant managers.

Standing Review Boards. Standing review boards are selected for projects or tasks that have a high level of activity, visibility and/or resource requirements. Selection of board members by the convening authority is generally made from senior Center technical and management staff. Supporting members or advisors may be added to the board as required by circumstances. If the review board is to function over the lifetime of a project, it is advisable to select extra board members and rotate active assignments to cover needs.

4.8.3 Specific Types of Reviews

This section describes the types, purpose, timing, and content of most of the reviews which may occur during the conduct of projects or tasks. Review material should be keyed to project documentation when available to minimize separate efforts.

Program/Project Requirements Review.

Purpose — The Program/Project Requirements Review (PRR) establishes the project development (i.e., functional) baseline. It ensures that:

- The project objectives (particularly the research and/or science objectives) have been properly translated into definite and unambiguous statements of requirements
- The impact of these requirements on the design of the major project elements and systems is sufficiently well understood that trades between requirements and constraints can be properly made
- The management techniques, procedures, agreements, and resources to be utilized by all project participants are evaluated.

Timing — At the completion of the Concept Definition Phase (Phase B) activities just prior to issuing the Source Selection Request for Proposal.

Agenda — The appropriate items from the following review items/data checklist should be addressed:

- Status of action items from the Conceptual Design Review (CoDR)
- Project Plan
- Mission objectives
- Research objectives
- Science objectives
- Design criteria and approach
- System trade analyses
- Design analyses and trade studies
- Final system specification
- Preliminary interface specifications
- Software system requirements
- Work breakdown structure
- Preliminary manufacturing plan
- Preliminary ground operations plan
- Preliminary payload integration plan
- Preliminary flight operations plan
- Preliminary data management plan
- Configuration management plan
- Reliability requirements and plan
- Quality assurance requirements and plan
- System safety requirements and plan
- Project policy and requirements
- Management structure
- Budget constraints
- Schedule
- Risk management activities.

Preliminary Design Review. The Preliminary Design Review (PDR) is not a single review but a number of reviews starting with the system PDR, followed by reviews conducted on specific Configuration Items (CIs).

Purpose — The PDR establishes the "design-to" baseline and ensures that it meets the program, project, system, subsystem or specific CI baseline requirements. The PDR process should:

- Establish the ability of the selected design approach to meet the technical requirements
- Establish the compatibility of the interface relationships between the specific configuration item and other interfacing items
- Establish the integrity of the selected design approach
- Establish the operability of the selected design
- Assess compliance with quality assurance, reliability and system safety requirements
- Address status, schedule and cost relationships
- Establish the feasibility of the approach.

Timing — After "design-to" specifications are developed and after risk reduction analyses are available.

Agenda — The appropriate items from the following review items/data checklist should be addressed:

- Status of action items from the applicable Hardware or Software Specification Review(s)
- Final functional requirements and specifications
- Technical justification for the performance specified
- Experiment performance analysis, including an analysis of instrument accuracy requirements
- Design parameters, restraints and constraints
- Environmental design requirements
- Interface design requirements
- Requirements traceability results
- Software standards to be applied
- Design and safety codes and standards to be applied
- Results of technical feasibility modeling and testing
- Design optimization analyses
- Discussion of block diagrams

- Compliance with functional requirements and specifications
- Suitability of inherited designs and hardware
- Lists of preliminary parts, materials and processes
- Spares requirements philosophy
- Preliminary data management flow and reduction plans
- Preliminary payload integration plan
- Preliminary ground operations plan
- Preliminary flight operations plan
- Requirements and plans for support equipment, including Ground Support Equipment (GSE)
- Preliminary reliability analyses, including single-point failure mode policy
- Preliminary system safety analyses
- Quality Assurance Plan
- Hardware and/or software verification plans
- Hardware and software development plans and schedules (including verification tests or analyses to be performed)
- Present status of item under review, including cost and technical developments
- Risk management activities.

Critical Design Review. The Critical Design Review (CDR) is not a single review but a number of reviews starting with specific CIs and ending with the system CDR.

Purpose — The CDR verifies the suitability of a CI design in meeting the specified requirements and establishes its "build-to" and/or "code-to" baseline. The CDR determines whether the design is compatible with the specified requirements, and verifies that the design conforms to the requirements established at the PDR and updated to the time of the CDR. During the CDR, the integrity of the design is verified through review of analytical and test data.

Following the CDR, the CI specifications and drawings are updated and placed under configuration control, and may be then released for fabrication and/or coding.

Timing — When the design of a CI is complete and after the completion of producibility demonstration. It should be held early enough to allow for corrective action and before total design freeze, the purchase of significant equipment, or fabrication of final hardware.

Agenda — The appropriate items from the following review items/data checklist should be addressed:

- Status of PDR action items
- Design requirements and specifications
- Interface requirements and specifications
- Design approach
- Assessment of hardware and software inheritance
- Test procedures
- Producibility demonstration results
- Scale model test results
- Design trades and alternatives considered
- Reliability, maintainability and operability considerations
- Spares list
- Conformance of the design to functional and user requirements
- Conformance to environmental design requirements
- Differences between the configuration item, system and subsystem performances in relation to the performances estimated at the PDR
- Final hardware and software design verification plans
- Detailed mechanical (including electronic packaging, thermal, hydraulic and pneumatic) design
- Detailed electronic/electrical circuit design
- Detailed software design
- Interface details and agreements
- Mechanical and electronic parts stress analysis results
- Final reliability analyses, including single-point failure analyses against the reliability policy
- System safety analyses
- Electronic parts classifications and screening specifications
- Non-electric parts, materials and processing list
- Materials and processing specifications
- Purchased devices list
- Manufacturing and fabrication plans
- Quality assurance plans and procedures
- Configuration control plans
- Qualification and acceptance test plans
- Calibration plan
- Data management flow and data reduction plan
- Support equipment and GSE requirements and plans
- Spares provisioning plan
- Ground operations plan
- Payload integration plan
- Flight operations plan
- Present status of item under review, including cost and technical developments
- Risk management activities.

Test Readiness Review. The Test Readiness Review (TRR) is not a single review but a series of reviews conducted prior to the start of verification testing of each test article, CI, subsystem and/or system.

Purpose — The TRR establishes the decision point to proceed with planned verification (qualification and/or acceptance) testing of test articles, CIs, subsystems and/or systems to acquire official sell-off verification data. The TRR assesses the adequacy of the test planning and compatibility with the verification requirements and specifications.

Timing — After completion of preliminary testing and prior to the start of official verification testing.

Agenda — The appropriate items from the following review items/data checklist should be addressed:

- Description of test article
- Test objectives
- Verification requirements and specifications
- Applicable test plans
- Applicable test procedures
- Test configuration and functional block diagrams
- Test equipment and circuitry
- Test equipment calibration
- Data to be collected, and collection and preservation methods
- Quality assurance plan
- Safety plan
- Test failure procedures
- Personnel responsibilities and qualifications
- Present status of item under review including cost and technical developments
- Risk management activities.

System Formal Qualification Review.

Purpose — The System Formal Qualification Review (SFQR) establishes the system production baseline by verifying that the system performance meets the system qualification specifications. The qualification testing demonstrates that the system meets its performance and operational requirements within the specified margins. The SFQR is the decision point for customer approval of the qualification certification of the design.

Timing — After the completion of all lower-level qualification testing.

Agenda — The appropriate items from the following review items/data checklist should be addressed:

- Status of action items from the applicable CDRs and TRRs
- Description of system tested, including all subsystems and functional block diagrams
- Qualification test objectives
- Qualification test requirements and specifications
- Description of test facilities
- Description of test configurations
- Subsystem qualification test results
- System qualification test results
- Qualification by similarity analysis
- Non-conformance reports/status
- Waivers and deviations
- Open work list
- Environmental retest following corrective action of any failures
- Strength and fracture mechanics for "as-built" hardware
- Software development documentation
- Summary of qualification status of all end items subjected to separate qualification tests
- Operational manuals
- Maintenance manuals
- Present status of system under review, including cost and technical developments
- Risk management activities.

Functional and Physical Configuration Audit.

Purpose — A Functional Configuration Audit (FCA) verifies that each "as-built" configuration item, test article, subsystem and/or system satisfies the functional and performance requirements specified in their respective "design-to" specifications.

A Physical Configuration Audit (PCA) verifies that each "as-built" test article, CI, subsystem and/or system:

- Satisfies the physical requirements (weight, center of gravity, moments of inertia, surface finish, cleanliness, etc.) specified in their respective design specifications
- Is correctly documented in "as-built" drawings, code listings, user manuals, etc.

Timing — Following the completion of the SFQR. Usually held in conjunction with the System Acceptance Review (SAR). For single unit projects, the FCA/PCA may be held prior to qualification testing.

Agenda — The appropriate items from the following project documentation should be addressed:

- CI, subsystem and system specifications
- Design drawings and engineering orders
- Subsystem and system schematics and block diagrams
- Design verification matrices for each configuration item, subsystem and system
- Inspection results
- Material and electronic parts certifications

- Materials process certifications
- Material Utilization List (MUL)
- Installed non-flight hardware list
- Test results
- Demonstration results
- Non-conformance reports/status
- Results of each Configuration Item Acceptance Review (CIAR)
- Results of the SFQR.

System Acceptance Review.

Purpose — The System Acceptance Review (SAR) provides the decision point to confirm that the design is ready for either integration, acceptance or replication.

Timing — Following the completion of the SFQR and prior to the Multi-Unit Procurement Phase and/or the Pre-Operations Phase (Phase E).

Agenda — The appropriate items from the following project documentation should be addressed:

- Brief description of system under review
- Verification requirements
- Results of the system FCA and PCA
- Results of the SFQR
- System verification report (qualification and operation)
- System acceptance report
- Final systems operations and maintenance methods
- System development lessons learned document
- Safety analyses status
- Present status of system under review, including cost and technical developments
- Risk management activities.

Safety Reviews. *System safety* is the application of engineering and management principles, criteria and techniques to optimize safety within the constraints of operational effectiveness, time and cost through all phases of the project cycle. A series of system and occupational safety reviews are held during the project cycle, many of which are held concurrently with other project reviews. Following are descriptions of these reviews and their relationship to the other project reviews.

Occupational Safety Reviews. The requirements for these reviews are not covered in this handbook. However, the system engineer should be aware that many occupational safety requirements can impose requirements on flight and/or ground equipment, such as the shipping and handling of pressure vessels, or toxic or explosive materials. Early reviews with Center occupational safety personnel should be held to identify and understand any problem areas and specify the requirements to control them.

Conceptual Design Safety Review.

Purpose — The Conceptual Design Safety Review (CoDSR) ensures that safety requirements have been included in the conceptual design and that a preliminary assessment of the potential hazards has been made. At several NASA Centers, the CoDSR is called the Phase 0 Safety Review.

Timing — At the completion of the Mission Needs and Conceptual Studies Phase (Phase A). It should be held concurrently with the Conceptual Design Review (CoDR).

Agenda — The appropriate items from the following list should be addressed:

- Purpose of the project, facility or equipment
- Design requirements
- Safety requirements
- Preliminary project safety plan
- Preliminary hazard analysis
- Safety staffing and management structure
- Safety budget
- Schedule
- Risk management activities.

Project Requirements Safety Review.

Purpose — The Project Requirements Safety Review (PRSR) establishes the project safety requirements baseline and ensures that:

- The project safety objectives have been properly translated into definite and unambiguous statements of requirements
- The impact of these requirements on the design of the major project elements and systems is sufficiently well understood that trades between requirements and constraints can be properly made
- The management techniques, procedures, agreements and resources to implement the safety program by all project participants are evaluated.

Timing — At the completion of the Concept Definition Phase (Phase B) activities just prior to issuing the Source Selection Request for Proposal. It should be held concurrently with the PRR.

Agenda — The appropriate subjects from the following list should be addressed:

- Purpose of the project, facility or equipment
- Status of action items from the CoDSR

- Design requirements
- Safety requirements
- Updated preliminary project safety plan
- Updated preliminary hazard analysis
- Safety staffing and management structure
- Safety budget
- Schedule
- Risk management activities.

Preliminary Design Safety Review. The Preliminary Design Safety Review (PDSR) is not a single review but a series of reviews conducted on specific configuration items, subsystems and the system.

Purpose — The PDSR ensures that the proposed CI, subsystem and/or system designs satisfy the project and Center safety requirements. At several NASA Centers, the PDSR is called the Phase I Safety Review.

Timing — At the completion of preliminary design and prior to the start of major detail design activities. It should be held concurrently with the PDRs.

Agenda — The appropriate subjects from the following list should be addressed:

- Description of design under review
- Status of safety-related action items from applicable hardware or software specification reviews
- Updated project safety plan
- Updated safety analysis reports
- Updated preliminary hazard analyses (sometimes called the Phase I Hazard Analyses)
- Preliminary Failure Modes and Effects Analysis (FMEA)
- Preliminary Critical Items List (CIL)
- List of limited-life items
- Accident or mishap investigation reports
- Waiver and deviation request dispositions
- Present status of safety activities, including cost and technical developments
- Risk management activities.

Critical Design Safety Review. The Critical Design Safety Review (CDSR) is not a single review but a series of reviews conducted on specific configuration items, subsystems and the system.

Purpose — The CDSR establishes the baseline for safety requirements, safety hazard controls and verification methods to be implemented in verifying those controls. At several NASA Centers, the CDSR is called the Phase II Safety Review.

Timing — When the design of a configuration item is essentially complete and prior to total design freeze, the purchase of significant equipment, or fabrication of final hardware. It should be held concurrently with the CDRs.

Agenda — The appropriate subjects from the following list should be addressed:

- Description of design under review
- Status of safety related action items from applicable hardware or software PDSRs
- Final project safety plan
- Updated safety analysis reports
- Updated preliminary hazard analyses (sometimes called the Phase II Hazard Analyses)
- Final Failure Modes and Effects Analysis
- Final Critical Items List
- List of limited-life items
- Accident or mishap investigation reports
- Waiver and deviation request dispositions
- Present status of safety activities including cost and technical developments
- Risk management activities.

System Acceptance Safety Review.

Purpose — The System Acceptance Safety Review (SASR) provides the decision point to confirm that all project safety requirements have been satisfied and confirms the satisfactory completion of all hazard control verification items and open safety items. At several NASA Centers, the SASR is called the Phase III Safety Review.

Timing — Following the completion of the SFQR and prior to the Multi-Unit Procurement Phase and the Pre-Operation Phase (Phase E). It should be held concurrently with the SAR.

Agenda — The appropriate subjects from the following list should be addressed:

- Description of design under review
- Status of safety-related action items from applicable hardware or software CDRs
- Updated safety analysis reports
- Updated preliminary hazard analyses (sometimes called the Phase III Hazard Analyses)
- Accident or mishap investigation reports
- Waiver and deviation request dispositions
- Present status of safety activities, including cost and technical developments
- Risk management activities.

Launch or Operational Safety Readiness Reviews.

Purpose — These reviews ensure the flight and/or ground operational safety of the item under review by certifying that:

- A CI, subsystem or system complies with all program and/or project safety requirements
- Approved controls for all identified safety hazards have been implemented
- All personnel involved in the handling and/or operation of the item under review have received the required training.

Timing — Following installation/integration, and prior to flight and/or start of ground operations.

Agenda — The appropriate subjects from the following list should be addressed:

- Brief description of item under review
- Safety requirements and specifications
- Safety compliance data package
- Hazard analyses/reports with supporting data
- Critical items list
- Limited-life item list
- Accident or mishap investigation reports
- Non-conformance reports/status
- Personnel training requirements
- Personnel training status
- Present status of safety activities, including cost and technical developments
- Risk management activities.

4.9 Status Reporting and Assessment

An important part of systems engineering planning is determining what is needed in time, resources and people to realize the system that meets the desired goals and objectives. Planning functions, such as WBS preparation, scheduling, and fiscal resource requirements planning, were discussed in Section 4.3 through 4.5. Project management, however, does not end with planning; project managers need visibility into the progress of those plans in order to exercise proper management control. This is the purpose of the status reporting and assessing processes. *Status reporting* is the process of determining where the project stands in dimensions of interest such as cost, schedule, and technical performance. *Assessing* is the analytical process that converts the output of the reporting process into a more useful form for the project manager — namely, what are the future implications of current trends? Lastly, the manager must decide whether that future is acceptable, and what changes, if any, in current plans are needed. Planning, status reporting, and assessing are systems engineering and/or program control functions; decision making is a management one.

These processes together form the feedback loop depicted in Figure 18. This loop takes place on a continual basis throughout the project cycle.

This loop is applicable at each level of the project hierarchy. Planning data, status reporting data, and assessments flow up the hierarchy with appropriate aggregation at each level; decisions cause actions to be taken down the hierarchy. Managers at each level determine (consistent with policies established at the next higher level of the project hierarchy) how often, and in what form, reporting data

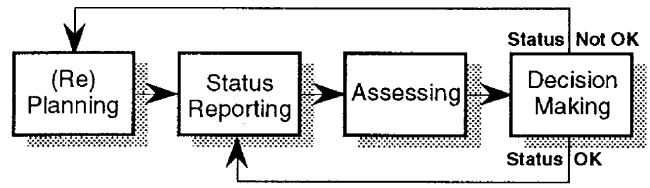


Figure 18 — Planning and Status Reporting Feedback Loop.

and assessments should be made. In establishing these status reporting and assessment requirements, some principles of good practice are:

- Use an agreed-upon set of well-defined status reporting variables
- Report these core variables in a consistent format at all project levels
- Maintain historical data for both trend identification and cross-project analyses
- Encourage a logical process of rolling up status reporting variables, (e.g., use the WBS for obligations/costs status reporting and PBS for mass status reporting)
- Support assessments with quantitative risk measures
- Summarize the condition of the project by using color-coded (red, yellow, and green) alert zones for all core reporting variables.

Regular, periodic (e.g., monthly) tracking of the core status reporting variables is recommended, through some status reporting variables should be tracked more often when there is rapid change or cause for concern. Key reviews, such as PDRs and CDRs, are points at which status reporting measures and their trends should be carefully scrutinized for early warning signs of potential problems. Should there be indications that existing trends, if allowed to continue, will yield an unfavorable outcome, replanning should begin as soon as practical.

This section provides additional information on status reporting and assessment techniques for costs and schedules, technical performance, and systems engineering process metrics.

4.9.1 Cost and Schedule Control Measures

Status reporting and assessment on costs and schedules provides the project manager and system engineer visibility into how well the project is tracking against its planned cost and schedule targets. From a management point of view, achieving these targets is on a par with meeting the technical performance requirements of the system. It is useful to think of cost and schedule status reporting and assessment as measuring the performance of the "system that produces the system."

NHB 9501.2B, *Procedures for Contractor Reporting of Correlated Cost and Performance Data*, provides specific requirements for cost and schedule status reporting and assessment based on a project's dollar value and period of performance. Generally, the NASA Form 533 series of reports is applicable to NASA cost-type (i.e., cost reimbursement and fixed-price incentive) contracts. However, on larger contracts (>$25M), which require Form 533P, NHB 9501.2B allows contractors to use their own reporting systems in lieu of 533P reporting. The project manager/system engineer may choose to evaluate the completeness and quality of these reporting systems against criteria established by the project manager/system engineer's own Center, or against the DoD's *Cost/Schedule Cost System Criteria (C/SCSC)*. The latter are widely accepted by industry and government, and a variety of tools exist for their implementation.

Assessment Methods. The traditional method of cost and schedule control is by comparing baselined cost and schedule plans against their actual values. In program control terminology, a difference between actual performance and planned costs or schedule status is called a *variance*.

Figure 19 illustrates two kinds of variances and some related concepts. A properly constructed Work Breakdown Structure (WBS) divides the project work into discrete tasks and products. Associated with each task and product (at any level in the WBS) is a schedule and a budgeted (i.e., planned) cost. The *Budgeted Cost of Work Scheduled* ($BCWS_t$) for any set of WBS elements is the budgeted cost of all work on tasks and products in those elements scheduled to be completed by time *t*. The *Budgeted Cost of Work Performed* ($BCWP_t$) is a statistic representing actual performance. $BCWP_t$, also called *Earned Value* (EV_t), is the budgeted cost for tasks and products

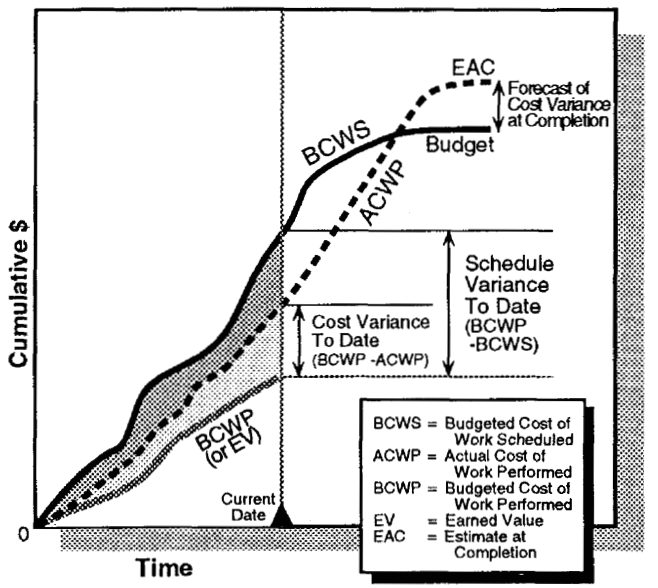


Figure 19 — Cost and Schedule Variances.

that have actually been produced (completed or in progress) at time t in the schedule for those WBS elements. The difference, $BCWP_t - BCWS_t$, is called the schedule variance at time *t*.

The Actual Cost of Work Performed ($ACWP_t$) is a third statistic representing the funds that have been expended up to time *t* on those WBS elements. The difference between the budgeted and actual costs, $BCWP_t - ACWP_t$, is called the cost variance at time *t*. Such variances may indicate that the cost *Estimate At Completion* (EAC_t) of the project is different from the budgeted cost. These types of variances enable a program analyst to estimate the EAC at any point in the project cycle. (See sidebar on computing EAC.)

If the cost and schedule baselines and the technical scope of the work are not fully integrated, then cost and schedule variances can still be calculated, but the incomplete linkage between cost data and schedule data makes it very difficult (or impossible) to estimate the current cost EAC of the project.

Control of Variances and the Role of the System Engineer. When negative variances are large enough to represent a significant erosion of reserves, then management attention is needed to either correct the variance, or to replan the project. It is important to establish levels of variance at which action is to be taken. These levels are generally lower when cost and schedule baselines do not support Earned Value calculations.

The first action taken to control an excessive negative variance is to have the cognizant manager or system

Computing the Estimate at Completion

EAC can be estimated at any point in the project. The appropriate formula depends upon the reasons associated for any variances that may exist. If a variance exists due to a one-time event, such as an accident, then EAC = BUDGET + ACWP – BCWP where BUDGET is the original planned cost at completion. If a variance exists for systemic reasons, such as a general underestimate of schedule durations, or a steady redefinition of requirements, then the variance is assumed to continue to grow over time, and the equation is: EAC = BUDGET x (ACWP / BCWP).

It is also possible that EAC will grow at a greater rate than estimated by the above equation if there are a growing number of liens, action items, or significant problems that will increase the difficulty of future work. Such factors could be addressed using risk management methods described in Section 4.6.

In a large project, a good EAC is the result of a variance analysis that may use of a combination of these estimation methods on different parts of the WBS. A rote formula should not be used as a substitute for understanding the underlying causes of variances.

engineer investigate the problem, determine its cause, and recommend a solution. There are a number of possible reasons why variance problems occur:

- A receivable was late or was unsatisfactory for some reason
- A task is technically very difficult and requires more resources than originally planned
- Unforeseeable (and unlikely to repeat) events occurred, such as illness, a labor strike, a fire, or some other calamity.

Although the identification of variances is largely a program control function, there is an important systems engineering role in their control. That role arises because the correct assessment of why a negative variance is occurring greatly increases the chances of successful control actions. This assessment often requires an understanding of the cost, schedule, and technical situation that can only be provided by the system engineer.

4.9.2 Technical Performance Measures

Status reporting and assessment of the system's technical performance measures (TPMs) complements cost and schedule control. By tracking the system's TPMs, the project manager gains visibility into whether the delivered system will actually meet its performance specifications (requirements). Beyond that, tracking TPMs ties together a number of basic systems engineering activities — that is, a TPM tracking program forges a relationship among systems analysis, functional and performance requirements definition, and verification and validation activities.

- Systems analysis activities identify the key performance or technical attributes that determine system effectiveness; trade studies performed in systems analysis help quantify the system's performance requirements.
- Functional and performance requirements definition activities help identify verification and validation requirements.
- Verification and validation activities result in quantitative evaluation of TPMs.

Examples of High-Level TPMs for Planetary Spacecraft and Launch Vehicles

High-level technical performance measures (TPMs) for planetary spacecraft include:

- End-of-mission (EOM) dry mass
- Injected mass (includes EOM dry mass, baseline mission plus reserve propellant, other consumables and upper stage adaptor mass)
- Consumables at EOM
- Power demand (relative to supply)
- Onboard data processing memory demand
- Onboard data processing throughput time
- Onboard data bus capacity
- Total pointing error.

Mass and power demands by spacecraft subsystems and science instruments may be tracked separately as well.

For launch vehicles, high-level TPMs include:

- Total vehicle mass at launch
- Payload mass (at nominal altitude or orbit)
- Payload volume
- Injection accuracy
- Launch reliability
- In-flight reliability
- For reusable vehicles, percent of value recovered
- For expendable vehicles, unit production cost at the nth unit. (See sidebar on Learning Curve Theory.)

- "Out-of-bounds" TPMs are signals to replan fiscal, schedule and people resources; sometimes new systems analysis activities need to be initiated.

Tracking TPMs can begin as soon as a baseline design has been established, which can occur as early as Phase B. A TPM tracking program should begin not later than the start of Phase C. Data to support the full set of selected TPMs may, however, not be available until later in the project cycle.

Selecting TPMs. In general, TPMs can be generic (attributes that are meaningful to each Product Breakdown Structure (PBS) element, like mass or reliability) or unique (attributes that are meaningful only to specific PBS elements). The system engineer needs to decide which generic and unique TPMs are worth tracking at each level of the PBS. The system engineer should track the measure of system effectiveness (when the project maintains such a measure) and the principal performance or technical attributes that determine it, as top-level TPMs. At lower levels of the PBS, TPMs worth tracking can be identified through the functional and performance requirements levied on each individual system, segment, etc. (See sidebar on high-level TPMs.)

In selecting TPMs, the system engineer should focus on those that can be objectively measured during the project cycle. This measurement can be done directly by testing or indirectly by a combination of testing and analysis. Analyses are often the only means available to determine some high-level TPMs such as system reliability, but the data used in such analyses should be based on demonstrated values to the maximum practical extent. These analyses can be performed using the same measurement methods or models used during trade studies. In TPM tracking, however, instead of using estimated (or desired) performance or technical attributes, the models are exercised using demonstrated values. As the project cycle proceeds through Phases C and D, the measurement of TPMs should become increasingly more accurate because of the availability of more "actual" data about the system.

Lastly, the system engineer should select those TPMs that must fall within well-defined (quantitative) limits for reasons of system effectiveness or mission feasibility. Usually these limits represent either a firm upper or lower bound constraint. A typical example of such a TPM for a spacecraft is its injected mass, which must not exceed the capability of the selected launch vehicle. Tracking injected mass as a high-level TPM is meant to ensure that this does not happen.

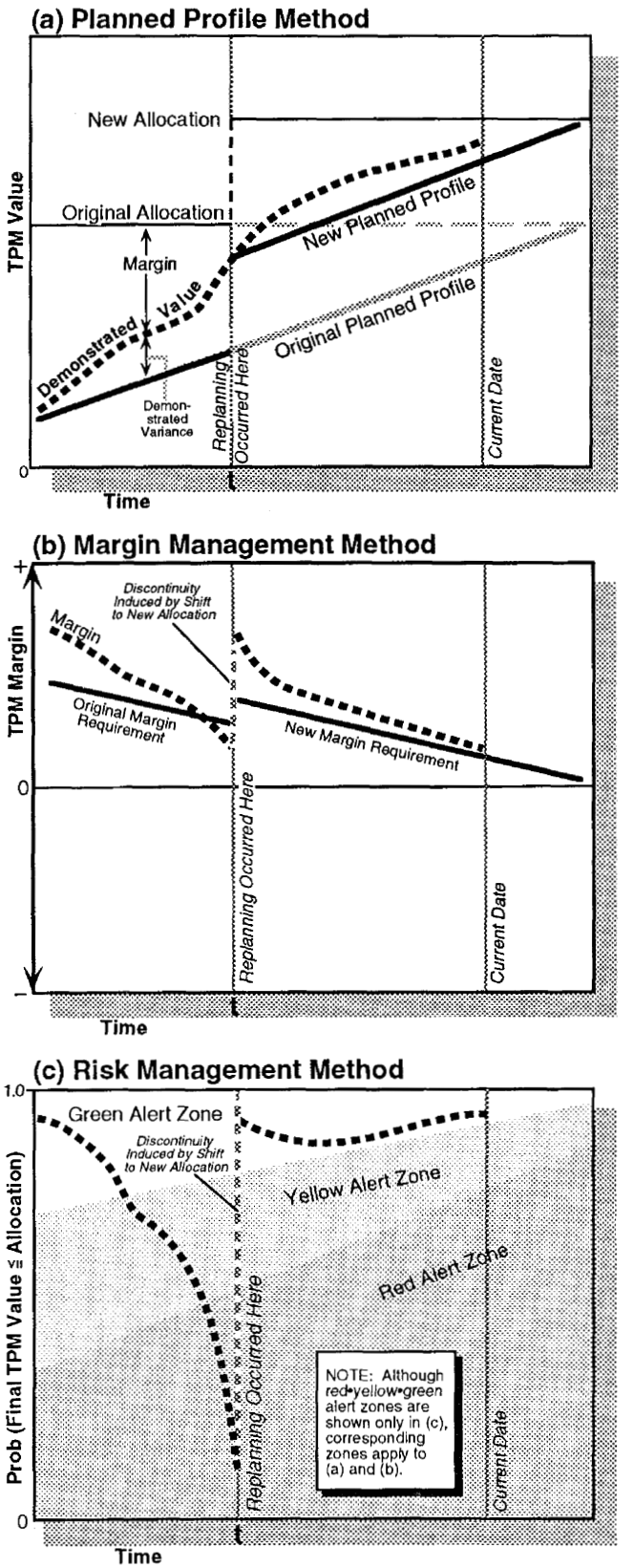


Figure 20 — Three TPM Assessment Methods.

Assessment Methods. The traditional method of assessing a TPM is by establishing a time-phased *planned profile* for it, and comparing the demonstrated value against that profile. The planned profile represents a nominal "trajectory" for that TPM taking into account a number of factors. These factors include the technological maturity of the system, the planned schedule of tests and demonstrations, and any historical experience with similar or related systems. As an example, spacecraft dry mass tends to grow during Phases C and D by as much as 25 to 30 percent. A planned profile for spacecraft dry mass may try to compensate for this growth with a lower initial value. The final value in the planned profile usually either intersects or is asymptotic to an allocated requirement (or contract specification). The planned profile method is the technical performance measurement counterpart to the Earned Value method for cost and schedule control described earlier.

A closely related method of assessing a TPM relies on establishing a time-phased *margin requirement* for it, and comparing the actual margin against that requirement. The margin is generally defined as the difference between a TPM's demonstrated value and its allocated requirement. The margin requirement may be expressed as a percent of the allocated requirement. The margin requirement generally declines through Phases C and D, reaching or approaching zero at their completion.

Depending on which method is chosen, the system engineer's role is to propose reasonable planned profiles or margin requirements for approval by the cognizant manager. The value of either of these methods is that they allow management by exception — that is, only deviations from planned profiles or margins below requirements signal potential future problems requiring replanning. If this occurs, then new cost, schedule and/or technical changes should be proposed. Technical changes may imply some new planned profiles. This is illustrated for a hypothetical TPM in Figure 20(a). In this example, a significant demonstrated variance (i.e., unanticipated growth) in the TPM during design and development of the system resulted in replanning at time *t*. The replanning took the form of an increase in the allowed final value of the TPM (the "allocation"). A new planned profile was then established to track the TPM over the remaining time of the TPM tracking program.

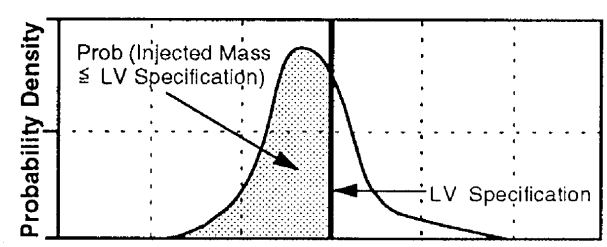
The margin management method of assessing is illustrated for the same example in Figure 20(b). The replanning at time *t* occurred when the TPM fell significantly below the margin requirement. The new higher allocation for the TPM resulted in a higher margin requirement, but it also immediately placed the margin in excess of that requirement.

Both of these methods recognize that the final value of the TPM being tracked is uncertain throughout most of Phases C and D. The margin management method attempts to deal with this implicitly by establishing a margin

An Example of the Risk Management Method for Tracking Spacecraft Mass

During Phases C and D, a spacecraft's injected mass can be considered an uncertain quantity. Estimates of each subsystem's and each instrument's mass are, however, made periodically by the design engineers. These estimates change and become more accurate as actual parts and components are built and integrated into subsystems and instruments. Injected mass can also change during Phases C and D as the quantity of propellant is fine-tuned to meet the mission design requirements. At each point during development then, the spacecraft's injected mass is better represented as a probability distribution rather than as a single point.

The mechanics of obtaining a probability distribution for injected mass typically involve making estimates of three points — the lower and upper bounds and the *most likely* injected mass value. These three values can be combined into parameters that completely define a probability distribution like the one shown in the figure below.



Spacecraft Injected Mass, Kg

The launch vehicle's "guaranteed" payload capability, designated the "LV Specification," is shown as a bold vertical line. The area under the probability curve to the left of the bold vertical line represents the probability that the spacecraft's injected mass will be less than or equal to the launch vehicle's payload capability. If injected mass is a TPM being tracked using the risk management method, this probability could be plotted in a display similar to Figure 20(c).

If this probability were nearly one, then the project manager might consider adding more objectives to the mission in order to take advantage of the "large margin" that appears to exist. In the above figure, however, the probability is significantly less than one. Here, the project manager might consider descoping the project, for example, by removing an instrument or otherwise changing mission objectives. The project manager could also solve the problem by requesting a larger launch vehicle!

requirement that reduces the chances of the final value exceeding its allocation to a low number, for example, five percent or less. A third method of reporting and assessing deals with this risk explicitly. The risk management method is illustrated for the same example in Figure 20(c). The replanning at time *t* occurred when the probability of the final TPM value being less than the allocation fell precipitously into the red alert zone. The new higher allocation for the TPM resulted in a substantial improvement in that probability.

The risk management method requires an estimate of the probability distribution for the final TPM value. (See sidebar on mass risk.) Early in the TPM tracking program, when the demonstrated value is based on indirect means of estimation, this distribution typically has a larger statistical variance than later, when it is based on measured data, e.g., a test result. When a TPM stays along its planned profile (or equivalently, when its margin remains above the corresponding margin requirement), the narrowing of the statistical distribution should allow the TPM to remain in the green alert zone (in Figure 20(c)) despite its growth. The three methods represent different ways to assess TPMs and communicate that information to management, but whichever is chosen, the pattern of success or failure should be the same for all three.

Relationship of TPM Tracking Program to the SEMP. The SEMP is the usual document for describing the project's TPM tracking program. This description should include a master list of those TPMs to be tracked, and the measurement and assessment methods to be employed. If analytical methods and models are used to measure certain high-level TPMs, then these need to be identified. The reporting frequency and timing of assessments should be specified as well. In determining these, the system engineer must balance the project's needs for accurate, timely, and effective TPM tracking against the cost of the TPM tracking program. The TPM tracking program plan, which elaborates on the SEMP, should specify each TPM's allocation, time-phased planned profile or margin requirement, and alert zones, as appropriate to the selected assessment method.

4.9.3 Systems Engineering Process Metrics

Status reporting and assessment of systems engineering process metrics provides additional visibility into the performance of the "system that produces the system." As such, these metrics supplement the cost and schedule control measures discussed in Section 4.9.1.

Systems engineering process metrics try to quantify the effectiveness and productivity of the systems engineering process and organization. Within a single project, tracking these metrics allows the system engineer to better understand the health and progress of that project. Across projects (and over time), the tracking of systems engineering process metrics allows for better estimation of the cost and time of performing systems engineering functions. It also allows the systems engineering organization to demonstrate its commitment to the TQM principle of continuous improvement.

Selecting Systems Engineering Process Metrics. Generally, systems engineering process metrics fall into three categories — those that measure the progress of the systems engineering effort, those that measure the quality of that process, and those that measure its productivity. Different levels of systems engineering management are generally interested in different metrics. For example, a project manager or lead system engineer may focus on metrics dealing with systems engineering staffing, project risk management progress, and major trade study progress. A subsystem system engineer may focus on subsystem requirements and interface definition progress and verification procedures progress. It is useful for each system engineer to focus on just a few process metrics. Which metrics should be tracked depends on the system engineer's role in the total systems engineering effort. The systems engineering process metrics worth tracking also change as the project moves through the project cycle.

Collecting and maintaining data on the systems engineering process is not without cost. Status reporting and assessment of systems engineering process metrics divert time and effort from the process itself. The system engineer must balance the value of each systems engineering process metric against its collection cost. The value of these metrics arises from the insights they provide into the process that cannot be obtained from cost and schedule control measures alone. Over time, these metrics can also be a source of hard productivity data, which are invaluable in demonstrating the potential returns from investment in systems engineering tools and training.

Examples and Assessment Methods. Table 2 lists some systems engineering process metrics to be considered. This list is not intended to be exhaustive. Because some of these metrics allow for different interpretations, each NASA Center needs to define them in a common-sense way that fits its own processes. For example, each Center needs to determine what it meant by a *completed* versus an *approved* requirement, or whether these terms are even relevant. As part of this definition, it is important to rec-

ognize that not all requirements, for example, need be lumped together. It may be more useful to track the same metric separately for each of several different types of requirements, for example.

Quality-related metrics should serve to indicate when a part of the systems engineering process is overloaded and/or breaking down. These metrics can be defined and tracked in several different ways. For example, requirements volatility can be quantified as the number of newly identified requirements, or as the number of changes to already-approved requirements. As another example, Engineering Change Request (ECR) processing could be tracked by comparing cumulative ECRs opened versus cumulative ECRs closed, or by plotting the age profile of open ECRs, or by examining the number of ECRs opened last month versus the total number open. The system engineer should apply his/her own judgment in picking the status reporting and assessment method.

Table 2 — Systems Engineering Process Metrics.

Function	Systems Engineering Process Metric	Category
Requirements development and management	Requirements identified vs. completed vs. approved	S
	Requirements volatility	Q
	Trade studies planned vs. completed	S
	Requirements approved per systems engineering hour	P
Design and Development	Specifications planned vs. completed	S
	Processing of ECRs/ECOs	Q
	Engineering drawings planned vs. released	S
Verification and validation (V&V)	V&V plans identified vs. approved	S
	V&V procedures planned vs. completed	S
	Functional requirements approved vs. verified	S
	V&V plans approved per systems engineering hour	P
	Processing of trouble reports	P Q
Reviews	Processing of Review Item Discrepancies (RIDs)	Q
	Processing of action items	Q

S = Progress, or schedule-related
Q = Quality-related
P = Productivity-related

Productivity-related metrics provide an indication of systems engineering output per unit of input. Although more sophisticated measures of input exist, the most common is the number of systems engineering hours dedicated to a particular function or activity. Because not all systems engineering hours cost the same, an appropriate weighing scheme should be developed to ensure comparability of hours across systems engineering personnel.

Displaying schedule-related metrics can be accomplished in a table or graph of planned quantities vs. actuals. With quality- and productivity-related metrics, trends are generally more important than isolated snapshots. The most useful kind of assessment method allows comparisons of the trend on a current project with that for a successful completed project of the same type. The latter provides a benchmark against which the system engineer can judge his/her own efforts.

5 Systems Analysis and Modeling Issues

The role of systems analysis and modeling is to produce rigorous and consistent evaluations so as to foster better decisions in the systems engineering process. By helping to progress the system design toward an optimum, systems analysis and modeling contribute to the objective of systems engineering. This is accomplished primarily by performing trade studies of plausible alternatives. The purpose of this chapter is to describe the trade study process, the methods used in trade studies to quantify system effectiveness and cost, and the pitfalls to avoid.

Systems Analysis

Gene Fisher defines *systems analysis* as "inquiry to assist decision makers in choosing preferred future courses of action by (1) systematically examining and reexamining the relevant objectives and alternative policies and strategies for achieving them; and (2) comparing quantitatively where possible the economic costs, effectiveness and risks of the alternatives."

5.1 The Trade Study Process

The trade study process is a critical part of the systems engineering spiral described in Chapter 2. This section discusses the steps of the process in greater detail. Trade studies help to define the emerging system at each level of resolution. One key message of this section is that to be effective, the process requires the participation of many skills and a unity of effort to move toward an optimum system design.

Figure 21 shows the trade study process in simplest terms, beginning with the step of *defining the system's goals and objectives, and identifying the constraints it must meet.* In the early phases of the project cycle, the goals,

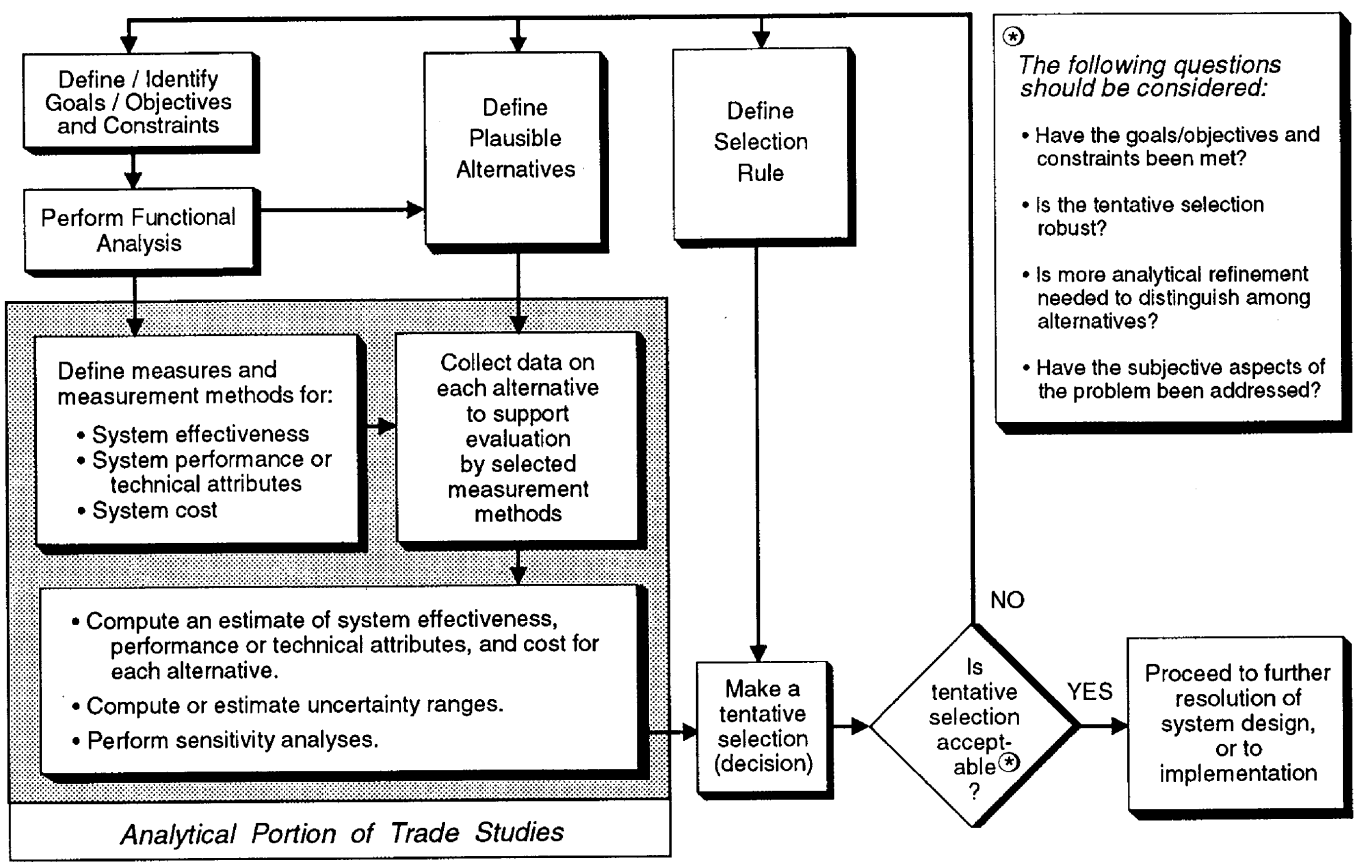


Figure 21 — The Trade Study Process.

PAGE 62 INTENTIONALLY BLANK

PRECEDING PAGE BLANK NOT FILMED

objectives and constraints are usually stated in general operational terms. In later phases of the project cycle, when the architecture and, perhaps, some aspects of the design have already been decided, the goals and objectives may be stated as performance requirements that a segment or subsystem must meet.

At each level of system resolution, the system engineer needs to understand the full implications of the goals, objectives, and constraints in order to formulate an appropriate system solution. This step is accomplished by *performing a functional analysis*. Functional analysis is the systematic process of identifying, describing, and relating the functions a system must perform in order to fulfill its goals and objectives. In the early phases of the project cycle, the functional analysis deals with the top-level functions that need to be performed by the system, where they need to be performed, how often, under what operational concept and environmental conditions, and so on. The functional analysis needs only to proceed to a level of decomposition that enables the trade study to define the system architecture. In later phases of the project cycle, the functional analysis proceeds to whatever level of decomposition is needed to fully define the system design and interfaces. (See sidebar on functional analysis techniques.)

Closely related to defining the goals and objectives, and performing a functional analysis, is the step of *defining the measures and measurement methods* for system effectiveness (when this is practical), system performance or technical attributes, and system cost. (These variables are collectively called *outcome variables*, in keeping with the discussion in Section 2.3. Some systems engineering books refer to these variables as *decision criteria*, but this term should not be confused with *selection rule*, described below. Sections 5.2 and 5.3 discuss the concepts of system cost and system effectiveness, respectively, in greater detail.) This step begins the analytical portion of the trade study process, since it suggests the involvement of those familiar with quantitative methods.

For each measure, it is important to address the question of how that quantitative measure will be computed — that is, which measurement method is to be used. One reason for doing this is that this step then explicitly identifies those variables that are important in meeting the system's goals and objectives.

Evaluating the likely outcomes of various alternatives in terms of system effectiveness, the underlying performance or technical attributes, and cost before actual fabrication and/or programming usually requires the use of a mathematical model or series of models of the system. So a second reason for specifying the measurement methods is that the necessary models can be identified.

Sometimes these models are already available from previous projects of a similar nature; other times, they need to be developed. In the latter case, defining the measurement methods should trigger the necessary system modeling activities. Since the development of new models can take a considerable amount of time and effort, early identification is needed to ensure they will be ready for formal use in trade studies.

Defining the selection rule is the step of explicitly determining how the outcome variables will be used to make a (tentative) selection of the preferred alternative. As an example, a selection rule may be to choose the alternative with the highest estimated system effectiveness that costs less than x dollars (with some given probability), meets safety requirements, and possibly meets other politi-

Functional Analysis Techniques

Functional analysis is the process of identifying, describing and relating the functions a system must perform in order to fulfill its goals and objectives. Functional analysis is logically structured as a top-down hierarchical decomposition of those functions, and serves several important roles in the systems engineering process:

- To draw out all the requirements the system must meet
- To help identify measures for system effectiveness and its underlying performance or technical attributes at all levels
- To weed out from further consideration in trade studies those alternatives that cannot meet the system's goals and objectives; and
- To provide insights to the system-level (and below) model builders, whose mathematical models will be used in trade studies to evaluate the alternatives.

Several techniques are available to do functional analysis. The primary functional analysis technique is the Functional Flow Block Diagram (FFBD). These diagrams show the network of actions that lead to the fulfillment of a function. Although the FFBD network shows the logical sequence of "what" must happen, it does not ascribe a time duration to functions or between functions. To understand *time-critical* requirements, a Time Line Analysis (TLA) is used. A TLA can be applied to such diverse operational functions as spacecraft command sequencing and launch vehicle processing. A third technique is the N^2 diagram, which is a matrix display of functional interactions, or data flows, at a particular hierarchical level. Appendix B.7 provides further discussion and examples of each of these techniques.

cal or schedule constraints. Defining the selection rule is essentially deciding how the selection is to be made. This step is independent from the actual measurement of system effectiveness, system performance or technical attributes, and system cost.

Many different selection rules are possible. The selection rule in a particular trade study may depend on the context in which the trade study is being conducted — in particular, what level of system design resolution is being addressed. At each level of the system design, the selection rule generally should be chosen only after some guidance from the next higher level. The selection rule for trade studies at lower levels of the system design should be in consonance with the higher level selection rule.

Defining plausible alternatives is the step of creating some alternatives that can potentially achieve the goals and objectives of the system. This step depends on understanding (to an appropriately detailed level) the system's functional requirements and operational concept. Running an alternative through an operational timeline or *reference mission* is a useful way of determining whether it can plausibly fulfill these requirements. (Sometimes it is necessary to create a separate behavioral model to determine whether it can plausibly fulfill time-critical and safety requirements.) Defining plausible alternatives also requires an understanding of the technologies available, or potentially available, at the time the system is needed. Each plausible alternative should be documented qualitatively in a description sheet. The format of the description sheet should, at a minimum, clarify the allocation of required system functions to that alternative's lower-level architectural or design components (e.g., subsystems).

One way to represent the trade study alternatives under consideration is by a trade tree. During Phase A trade studies, the trade tree should contain a number of alternative high-level system architectures to avoid a premature focus on a single one. As the systems engineering process proceeds, branches of the trade tree containing unattractive alternatives will be "pruned", and greater detail in terms of system design will be added to those branches that merit further attention. The process of pruning unattractive early alternatives is sometimes known as doing "killer trades". (See sidebar on trade trees.)

Given a set of plausible alternatives, the next step is to *collect data* on each to support the evaluation of the measures by the selected measurement methods. If models are to be used to calculate some of these measures, then obtaining the model inputs provides some impetus and direction to the data collection activity. By providing data, engineers in such disciplines as reliability, maintainability, producibility, integrated logistics, software, testing, operations and costing have an important supporting role in trade studies. The data collection activity, however, should be orchestrated by the system engineer. The results of this step should be a quantitative description of each alternative to accompany the qualitative.

Test results on each alternative can be especially useful. Early in the systems engineering process, performance and technical attributes are generally uncertain and must be estimated. Data from breadboard and brassboard testbeds can provide additional confidence that the range of values used as model inputs is correct. Such confidence is also enhanced by drawing on data collected on related previously developed systems.

The next step in the trade study process is to quantify the outcome variables by *computing estimates of system effectiveness, its underlying system performance or technical attributes, and system cost.* If the needed data have been collected, and the measurement methods (for example, models) are in place, then this step is, in theory, mechanical. In practice, considerable skill is often needed to get meaningful results.

Point estimates of the outcome variables for each alternative should be supplemented by computed or estimated uncertainty ranges. The uncertainty range should be estimated for each input to the measurement methods. Using this range of input values, the sensitivity of the outcome variables can be gauged, and their uncertainty ranges calculated. Ideally, all input values would be precisely known, and the measurement methods would perfectly predict the outcome variables. In reality, the system engineer may only be able to provide ranges and sensitivities for the outcome variables without probabilities. With more powerful measurement methods, ranges, sensitivities *and* probabilities result from this step of the trade study process.

This essentially completes the analytical portion of the trade study process. The next steps can be described as the judgmental portion. Combining the selection rule with the results of the analytical activity should enable the system engineer to array the alternatives from most preferred to least, in essence *making a tentative selection.*

This tentative selection should not be accepted blindly. In most trade studies, there is a need to subject the results to a "reality check" by considering a number of questions. Have the goals, objectives and constraints truly been met? Is the tentative selection heavily dependent on a particular set of input values to the measurement methods, or does it hold up under a range of reasonable input values? (In the latter case, the tentative selection is said to be robust.) Are there sufficient data to back up the tentative selection? Are the measurement methods sufficiently discriminating to be sure that the tentative selection is really better than other alternatives? Have the subjective aspects of the problem been fully addressed?

If the answers support the tentative selection, then the system engineer can have greater confidence in a recommendation to proceed to a further resolution of the system design, or to the implementation of that design. The estimates of system effectiveness, its underlying performance or technical attributes, and system cost generated during the trade study process serve as inputs to that further resolution. The analytical portion of the trade study process often provide the means to quantify the performance or technical (and cost) attributes that the system's lower levels must meet. These can be formalized as *performance requirements*.

If the reality check is not met, the trade study process returns to one or more earlier steps. This iteration may result in a change in the goals, objectives and constraints, a new alternative, or a change in the selection rule, based on the new information generated during the trade study. The reality check may, at times, lead instead to a decision to first improve the measures and measurement methods (e.g., models) used in evaluating the alternatives, and then to repeat the analytical portion of the trade study process.

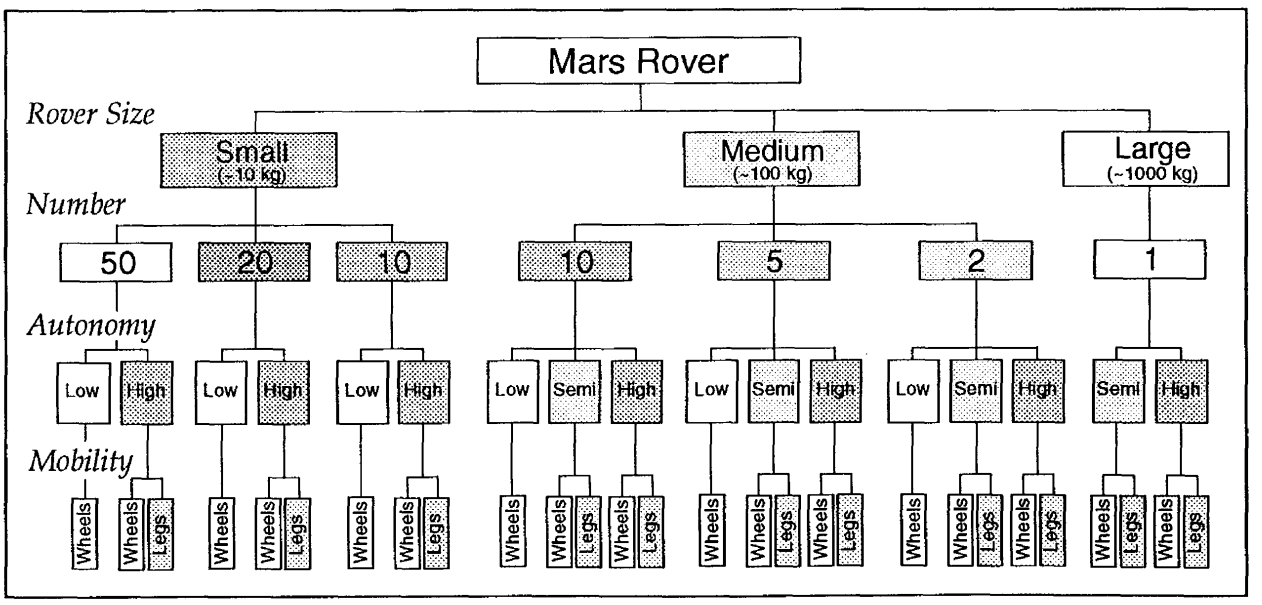
5.1.1 Controlling the Trade Study Process

There are a number of mechanisms for controlling the trade study process. The most important one is the Systems Engineering Management Plan (SEMP). The SEMP specifies the major trade studies that are to be performed during each phase of the project cycle. It should also spell out the general contents of trade study reports, which form part of the *decision support packages* (i.e., documentation submitted in conjunction with formal reviews and change requests).

A second mechanism for controlling the trade study process is the selection of the study team leaders and mem-

An Example of a Trade Tree for a Mars Rover

The figure below shows part of a trade tree for a robotic Mars rover system, whose goal is to find a suitable manned landing site. Each layer represents some aspect of the system that needs to be treated in a trade study to determine the best alternative. Some alternatives have been eliminated *a priori* because of technical feasibility, launch vehicle constraints, etc. The total number of alternatives is given by the number of end points of the tree. Even with just a few layers, the number of alternatives can increase quickly. (This tree has already been pruned to eliminate low-autonomy, large rovers.) As the systems engineering process proceeds, branches of the tree with unfavorable trade study outcomes are discarded. The remaining branches are further developed by identifying more detailed trade studies that need to be made. A whole family of (implicit) alternatives can be represented in a trade tree by a continuous variable. In this example, rover speed or range might be so represented. By treating a variable this way, mathematical optimization techniques can be applied. Note that a trade tree is, in essence, a decision tree without chance nodes. (See the sidebar on decision trees.)



Trade Study Reports

Trade study reports should be prepared for each trade study. At a minimum, each trade study report should identify:

- The system issue under analysis
- System goals and objectives (or requirements, as appropriate to the level of resolution) and constraints
- The measures and measurement methods (models) used
- All data sources used
- The alternatives chosen for analysis
- The computational results, including uncertainty ranges and sensitivity analyses performed
- The selection rule used
- The recommended alternative.

Trade study reports should be maintained as part of the system archives so as to ensure traceability of decisions made through the systems engineering process. Using a generally consistent format for these reports also makes it easier to review and assimilate them into the formal change control process.

bers. *Because doing trade studies is part art and part science, the composition and experience of the teams is an important determinant of the study's ultimate usefulness.* A useful technique to avoid premature focus on a specific technical designs is to include in the study team individuals with differing technology backgrounds.

Another mechanism is limiting the number of alternatives that are to be carried through the study. This number is usually determined by the time and resources available to do the study because the work required in defining additional alternatives and obtaining the necessary data on them can be considerable. Focusing on too few or too similar alternatives defeats the purpose of the trade study process.

A fourth mechanism for controlling the trade study process can be exercised through the use (and misuse) of models. Lastly, the choice of the selection rule exerts a considerable influence on the results of the trade study process. These last two issues are discussed in Sections 5.1.2 and 5.1.3, respectively.

5.1.2 Using Models

Models play important and diverse roles in systems engineering. A model can be defined in several ways, including:

- An abstraction of reality designed to answer certain questions about the real world that cannot be answered by direct experimentation
- An imitation or analogue of a real-world process or structure; or
- A tool to assist a decision maker.

Together, these definitions are broad enough to encompass physical engineering models used in the verification of a system design, as well as schematic models like a functional flow block diagram and mathematical (i.e., quantitative) models used in the trade study process. This section focuses on the last.

The main reason for using mathematical models in trade studies is to provide estimates of system effectiveness, performance or technical attributes, and cost from a set of known or estimable quantities. Typically, a collection of separate models is needed to provide all of these outcome variables. The heart of any mathematical model is a set of meaningful quantitative relationships among its inputs and outputs. These relationships can be as simple as adding up constituent quantities to obtain a total, or as complex as a set of differential equations describing the trajectory of a spacecraft in a gravitational field. Ideally, the relationships express causality, not just correlation.

Types of Models. There are a number of ways mathematical models can be usefully categorized. One way is according to its purpose in the trade study process — that is, what system issue the model addresses and with which outcome variable or variables the model primarily deals. Other commonly used ways of categorizing mathematical models focus on specific model attributes such as whether a model is:

- Static or dynamic
- Deterministic or probabilistic (also called *stochastic*)
- Descriptive or optimizing.

These terms allow model builders and model users to enter into a dialogue with each other about the type of model used in a particular analysis or trade study. No hierarchy is implied in the above list; none of the above dichotomous categorizations stands above the others.

Another taxonomy can be based on the degree of analytic tractability. At one extreme on this scale, an "analytic" model allows a closed-form solution for a outcome variable of interest as a function of the model inputs. At the other extreme, quantification of a outcome variable of interest is at best ordinal, while in the middle are many forms of mathematical simulation models.

Mathematical simulations are a particularly useful type of model in trade studies. These kinds of models have been successfully used in dealing quantitatively with large complex systems problems in manufacturing, transportation and logistics. Simulation models are used for these problems because it is not possible to "solve" the system's equations analytically to obtain a closed-form solution, yet it is relatively easy to obtain the desired results (usually the system's behavior under different assumptions) using the sheer computational power of current computers.

Linear, nonlinear, integer and dynamic programming models are another important class of models in trade studies because they can optimize an objective function representing an important outcome variable (for example, system effectiveness) for a whole class of implied alternatives. Their power is best applied in situations where the system's objective function and constraints are well understood, and these constraints can be written as a set of equalities and inequalities.

Pitfalls in Using Models. Models always embody assumptions about the real world they purport to represent, and they always leave something out. Moreover, they are usually capable of producing highly accurate results only when they are addressing rigorously quantifiable questions in which the "physics" is well understood as, for example, a load dynamics analysis or a circuit analysis.

In dealing with system issues at the top level, however, this is seldom the case. There is often a significant difference between the substantive system cost-effectiveness issues and questions, and the questions that are mathematically tractable from a modeling perspective. For example, the program/project manager may ask: "What's the best space station we can build in the current budgetary environment?" The system engineer may try to deal with that question by translating it into: "For a few plausible station designs, what does each provide its users, and how much does each cost?" When the system engineer then turns to a model (or models) for answers, the results may only be some approximate costs and some user resource measures based on a few engineering relationships. The model has failed to adequately address even the system engineer's more limited question, much less the program/project manager's. Compounding this sense of model incompleteness is the recognition that the model's relationships are often chosen for their mathematical convenience, rather than a demonstrated empirical validity. Under this situation, the model may produce insights, but it cannot provide definitive answers to the substantive questions on its own. Often too, the system engineer must make an engineering interpretation of model results and convey them to the project manager or other decision maker in a way that captures the essence of the original question.

As mentioned earlier, large complex problems often require multiple models to deal with different aspects of evaluating alternative system architectures (and designs). It is not unusual to have separate models to deal with costs and effectiveness, or to have a hierarchy of models — i.e., models to deal with lower level engineering issues that provide useful results to system-level mathematical models. This situation itself can have built-in pitfalls.

One such pitfall is that there is no guarantee that all of the models work together the way the system engineer intends or needs. One submodel's specialized assumptions may not be consistent with the larger model it feeds. Optimization at the subsystem level may not be consistent with system-level optimization. Another such pitfall occurs when a key effectiveness variable is not represented in the cost models. For example, if spacecraft reliability is a key variable in the system effectiveness equation, and if that reliability does not appear as a variable in the spacecraft cost model, then there is an important disconnect. This is because the models allow the spacecraft designer to believe it is possible to boost the effectiveness with increased reliability without paying any *apparent* cost penalty. When the models fail to treat such important interactions, the system engineer must ensure that others do not reach false conclusions regarding costs and effectiveness.

Characteristics of a Good Model. In choosing a model (or models) for a trade study, it is important to recognize those characteristics that a good model has. This list includes:

- Relevance to the trade study being performed
- Credibility in the eye of the decision maker
- Responsiveness
- Transparency
- User friendliness.

Both relevance and credibility are crucial to the acceptance of a model for use in trade studies. Relevance is determined by how well a model addresses the substantive cost-effectiveness issues in the trade study. A model's credibility results from the logical consistency of its mathematical relationships, and a history of successful (i.e., correct) predictions. A history of successful predictions lends credibility to a model, but full validation — proof that the model's prediction is in accord with reality — is very difficult to attain since observational evidence on those predictions is generally very scarce. While it is certainly advantageous to use tried-and-true models, this is not always possible. Systems that address new problems often require

that new models be developed for their trade studies. In that case, full validation is out of the question, and the system engineer must be content with models that have logical consistency and some limited form of outside, independent corroboration.

Responsiveness of a model is a measure of its power to distinguish among the different alternatives being considered in a trade study. A responsive lunar base cost model, for example, should give a different cost for different system architectures or designs, operations concepts, or logistics strategies.

Another desirable model characteristic is transparency, which occurs when the model's mathematical relationships, algorithms, parameters, supporting data, and inner workings are open to the user. The benefit of this visibility is in the traceability of the model's results. Not everyone may agree with the results, but at least they know how they were derived. Transparency also aids in the acceptance process. It is easier for a model to be accepted when its documentation is complete and open for comment. Proprietary models often suffer from a lack of acceptance because of a lack of transparency.

Upfront user friendliness is related to the ease with which the system engineer can learn to use the model and prepare the inputs to it. Backend user friendliness is related to the effort needed to interpret the model's results and to prepare trade study reports for the tentative selection using the selection rule.

5.1.3 Selecting the Selection Rule

The analytical portion of the trade study process serves to produce specific information on system effectiveness, its underlying performance or technical attributes, and cost (along with uncertainty ranges) for a few alternative system architectures (and later, system designs). These data need to be brought together so that one alternative may be selected. This step is accomplished by applying the selection rule to the data so that the alternatives may be ranked in order of preference.

The structure and complexity of real world decisions in systems engineering often make this ranking a difficult task. For one, securing higher effectiveness almost always means incurring higher costs and/or facing greater uncertainties. In order to choose among alternatives with different levels of effectiveness and costs, the system engineer must understand how much of one is worth in terms of the other. An explicit cost-effectiveness objective function is seldom available to help guide the selection decision, as any system engineer who has had to make a budget-induced system descope decision will attest.

A second, and major, problem is that an expression or measurement method for system effectiveness may not be possible to construct, even though its underlying performance and technical attributes are easily quantified. These underlying attributes are often the same as the technical performance measures (TPMs) that are tracked during the product development process to gauge whether the system design will meet its performance requirements. In this case, system effectiveness may, at best, have several irreducible dimensions.

What selection rule should be used has been the subject of many books and articles in the decision sciences — management science, operations research and economics. A number of selection rules are applicable to NASA trade studies. Which one should be used in a particular trade study depends on a number of factors:

- The level of resolution in the system design
- The phase of the project cycle
- Whether the project maintains an overall system effectiveness model
- How much less-quantifiable, subjective factors contribute to the selection
- Whether uncertainty is paramount, or can effectively be treated as a subordinate issue
- Whether the alternatives consist of a few qualitatively different architectures/designs, or many similar ones that differ only in some quantitative dimensions.

This handbook can only suggest some selection rule for NASA trade studies, and some general conditions under which each is applicable; definitive guidance on which to use in each and every case has not been attempted.

Table 3 first divides selection rules according to the importance of uncertainty in the trade study. This division is reflective of two different classes of decision problems — decisions to be made under conditions of certainty, and decisions to be made under conditions of uncertainty. Uncertainty is an inherent part of systems engineering, but the distinction may be best explained by reference to Figure 2, which is repeated here as Figure 22. In the former class, the measures of system effectiveness, performance or technical attributes, and system cost for the alternatives in the trade study look like those for alternative B. In the latter class, they look like those for alternative C. When they look like those for alternative A, conditions of uncertainty should apply, but often are not treated that way.

The table further divides each of the above classes of decision problems into two further categories: those that apply when cost and effectiveness measures are scalar quantities, and thus suffice to guide the system engineer to

Effectiveness and Cost	Importance of Uncertainty in Trade Study	
	Uncertainty Subordinate or Not Considered	Uncertainty Predominates
Can be represented as scalar quantities	Maximize net benefits Maximize effectiveness subject to a cost constraint Minimize cost subject to an effectiveness constraint Maximize cost-effectiveness objective function	Maximize expected utility Minimize maximum loss ("minimax")
Cannot be represented as scalar quantities	Maximize value function (i.e., figure of merit) Maximize value function subject to individual objective constraints Minimize cost subject to individual performance requirements constraints	Maximize expected utility

Table 3 — Some Selection Rules Applicable to NASA Trade Studies.

the best alternative, and those that apply when cost and effectiveness cannot be represented as scalar quantities.

Selection Rules When Uncertainty Is Subordinate, or Not Considered. Selecting the alternative that *maximizes net benefits* (benefits minus costs) is the rule used in most cost-benefit analyses. Cost-benefit analysis applies, however, only when the return on a project can be measured in the same units as the costs, as, for example, in its classical application of evaluating water resource projects.

Another selection rule is to choose the alternative that *maximizes effectiveness for a given level of cost.* This rule is applicable when system effectiveness and system cost can be unambiguously measured, and the appropriate level of cost is known. Since the purpose of the selection rule is to compare and rank the alternatives, practical application requires that each of the alternatives be placed on an equal cost basis. For certain types of trade studies, this does not present a problem. For example, changing system size or output, or the number of platforms or instruments, may suffice. In other types of trade studies, this may not be possible.

A related selection rule is to choose the alternative that *minimizes cost for a given level of effectiveness.* This rule presupposes that system effectiveness and system cost can be unambiguously measured, and the appropriate level of effectiveness is known. Again, practical application requires that each of the alternatives be put on an equal effectiveness basis. This rule is dual to the one above in the following sense: For a given level of cost, the same alternative would be chosen by both rules; similarly, for a given level of effectiveness, the same alternative would be chosen by both rules.

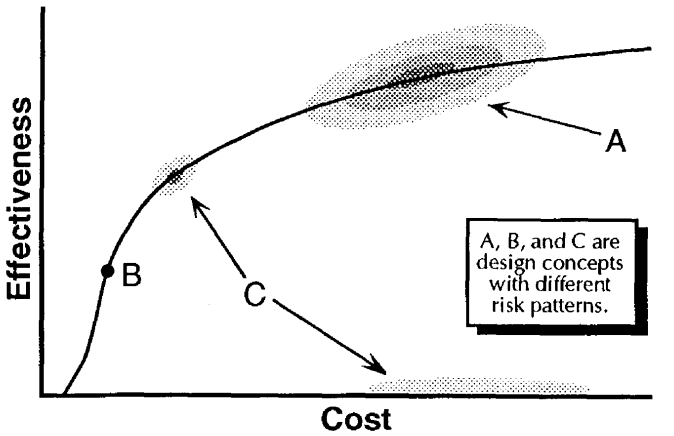


Figure 22 — Results of Design Concepts with Different Risk Patterns.

When it is not practical to equalize the cost or the effectiveness of competing alternatives, and cost caps or effectiveness floors do not rule out all alternatives save one, then it is necessary to form, either explicitly or implicitly, a cost-effectiveness objective function like the one shown in Figure 4 (Section 2.5). The cost-effectiveness objective function provides a single measure of worth for all combinations of cost and effectiveness. When this selection rule is applied, the alternative with the highest value of the cost-effectiveness objective function is chosen.

Another group of selection rules is needed when cost and/or effectiveness cannot be represented as scalar quantities. To choose the best alternative, a *multi-objective* selection rule is needed. A multi-objective rule seeks to select the alternative that, in some sense, represents the best balance among competing objectives. To accomplish this, each alternative is measured (by some quantitative method) in terms of how well it achieves each objective. For example, the objectives might be national prestige, upgrade or expansion potential, science data return, low cost, and potential for international partnerships. Each alternative's "scores" against the objectives are then combined in a value function to yield an overall figure of merit for the alternative. The way the scores are combined should reflect the decision maker's preference structure. The alternative that *maximizes the value function* (i.e., with the highest figure of merit) is then selected. *In essence, this*

selection rule recasts a multi-objective decision problem into one involving a single, measurable objective.

One way, but not the only way, of forming the figure of merit for each alternative is to linearly combine its scores computed for each of the objectives — that is, compute a weighted sum of the scores. *MSFC-HDBK-1912, Systems Engineering (Volume 2)* recommends this selection rule. The weights used in computing the figure of merit can be assigned *a priori* or determined using Multi-Attribute Utility Theory (MAUT). Another technique of forming a figure of merit is the Analytic Hierarchy Process (AHP). Several microcomputer-based commercial software packages are available to automate either MAUT or AHP. If the wrong weights, objectives, or attributes are chosen in either technique, the entire process may obscure the best alternative. Also, with either technique, the individual evaluators may tend to reflect the institutional biases and preferences of their respective organizations. The results, therefore, may depend on the mix of evaluators. (See sidebars on AHP and MAUT.)

Another multi-objective selection rule is to choose the alternative with the highest figure of merit from among those that meet specified individual objectives. This selection rule is used extensively by Source Evaluation Boards (SEBs) in the NASA procurement process. Each proposal, from among those meeting specific technical objectives (requirements), is scored on such attributes as technical design, price, systems engineering process quality, etc. In applying this rule, the attributes being scored by the SEB are known to the bidders, but their weighing may not be. (See NHB 5103.6B.)

In trade studies where no measure of system effectiveness can be constructed, but performance or technical attributes can be quantified, a possible selection rule is to choose the alternative that *minimizes cost for given levels of performance or technical attributes.* This rule presupposes that system cost can be unambiguously measured, and is related to the all of the quantified performance or technical attributes that are considered constraints. Practical application again requires that all of the alternatives be put on an equal basis with respect to the performance or technical attributes. This may not be practical for trade studies in which the alternatives cannot be described by a set of continuous mathematical relationships.

Selection Rule When Uncertainty Predominates. When the measures of system effectiveness, performance or technical attributes, and system cost for the alternatives in the trade study look like those for alternative C in Figure 22, the selection of the best alternative may need to be handled differently. This is because of the general propensity of decision makers to show risk-averse behavior when dealing

The Analytic Hierarchy Process

AHP is a decision technique in which a figure of merit is determined for each of several alternatives through a series of pair-wise comparisons. AHP is normally done in six steps:

(1) Describe in summary form the alternatives under consideration.
(2) Develop a set of high-level evaluation objectives; for example, science data return, national prestige, technology advancement, etc.
(3) Decompose each high-level evaluation objective into a hierarchy of evaluation attributes that clarify the meaning of the objective.
(4) Determine, generally by conducting structured interviews with selected individuals ("experts") or by having them fill out structured questionnaires, the relative importance of the evaluation objectives and attributes through pair-wise comparisons.
(5) Have each evaluator make separate pair-wise comparisons of the alternatives with respect to each evaluation attribute. These subjective evaluations are the raw data inputs to a separately developed AHP program, which produces a single figure of merit for each alternative. This figure of merit is based on relative weights determined by the evaluators themselves.
(6) Iterate the questionnaire and AHP evaluation process until a consensus ranking of the alternatives is achieved.

With AHP, sometimes consensus is achieved quickly; other times, several feedback rounds are required. The feedback consists of reporting the computed values (for each evaluator and for the group) for each option, reasons for differences in evaluation, and identified areas of contention and/or inconsistency. Individual evaluators may choose to change their subjective judgments on both attribute weights and preferences. At this point, inconsistent and divergent preferences can be targeted for more detailed study.

AHP assumes the existence of an underlying preference "vector" (with magnitudes and directions) that is revealed through the pair-wise comparisons. This is a powerful assumption, which may at best hold only for the participating evaluators. The figure of merit produced for each alternative is the result of the group's subjective judgments and is not necessarily a reproducible result. For more information on AHP, see Thomas L. Saaty, *The Analytic Hierarchy Process,* 1980.

Multi-Attribute Utility Theory

MAUT is a decision technique in which a figure of merit (or utility) is determined for each of several alternatives through a series of preference-revealing comparisons of simple lotteries. An abbreviated MAUT decision mechanism can be described in six steps:

(1) Choose a set of descriptive, *but quantifiable,* attributes designed to characterize each alternative.
(2) For each alternative under consideration, generate values for each attribute in the set; these may be point estimates, or probability distributions, if the uncertainty in attribute values warrants explicit treatment.
(3) Develop an attribute utility function for *each* attribute in the set. Attribute utility functions range from 0 to 1; the least desirable value, x_i^0, of an attribute (over its range of plausible values) is assigned a utility value of 0, and the most desirable, x_i^*, is assigned a utility value of 1. That is, $u_i(x_i^0) = 0$ and $u_i(x_i^*) = 1$. The utility value of an attribute value, x_i, intermediate between the least desirable and most desirable is assessed by finding the value x_i such that the decision maker is indifferent between receiving x_i for sure, or, a lottery that yields x_i^0 with probability p_i or x_i^* with probability $1-p_i$. From the mathematics of MAUT, $u_i(x_i) = p_i\, u_i(x_i^0) + (1-p_i)\, u_i(x_i^*)$.
(4) Repeat the process of indifference revealing until there are enough discrete points to approximate a continuous attribute utility function.
(5) Combine the individual attribute utility functions to form a multiattribute utility function. This is also done using simple lotteries to reveal indifference between receiving a particular set of attribute values with certainty, or, a lottery of attribute values. In its simplest form, the resultant multiattribute utility function is a weighted sum of the individual attribute utility functions.
(6) Evaluate each alternative using the multiattribute utility function.

The most difficult problem with MAUT is getting the decision makers or evaluators to think in terms of lotteries. This can often be overcome by an experienced interviewer. MAUT is based on a set of mathematical axioms about the way individuals should behave when confronted by uncertainty. Logical consistency in ranking alternatives is assured so long as evaluators adhere to the axioms; no guarantee can be made that this will always be the case. An extended discussion of MAUT is given in Keeney and Raiffa, *Decisions with Multiple Objectives: Preferences and Value Tradeoffs,* 1976. A textbook application of MAUT to a NASA problem can be found in Jeffrey H. Smith, et al., *An Application of Multiattribute Decision Analysis to the Space Station Freedom Program, Case Study: Automation and Robotics Technology Evaluation,* 1990.

with large variations in cost and/or effectiveness outcomes. In such cases, the *expected value* (i.e., the mean) of some stochastic outcome variable is not a satisfactory point measure of that variable.

To handle this class of decision problem, the system engineer may wish to invoke a von Neumann-Morgenstern selection rule. In this case, alternatives are treated as "gambles" (or lotteries). The probability of each outcome is also known or can be subjectively estimated, usually by creating a decision tree. The von Neumann-Morgenstern selection rule applies a separately developed utility function to each outcome, and chooses the alternative that *maximizes the expected utility.* This selection rule is easy to apply when the lottery outcomes can be measured in dollars. Although multi-attribute cases are more complex, the principle remains the same.

The basis for the von Neumann-Morgenstern selection rule is a set of mathematical axioms about how individuals should behave when confronted by uncertainty. Practical application of this rule requires an ability to enumerate each "state of nature" (hereafter, simply called "state"), knowledge of the outcome associated with each enumerated state for each alternative, the probabilities for the various states, and a mathematical expression for the decision maker's utility function. This selection rule has also found use in the evaluation of system procurement alternatives. See Section 4.6.2 for a discussion of some related topics, including decision analysis, decision trees and probabilistic risk assessment.

Another selection rule for this class of decision problem is called the *minimax rule.* To apply it, the system engineer computes a loss function for each enumerated state for each alternative. This rule chooses the alternative that *minimizes the maximum loss.* Practical application requires an ability to enumerate each state, and to define the loss function. Because of its "worst case" feature, this rule has found some application in military systems.

5.1.4 Trade Study Process: Summary

System architecture and design decisions will be made. The purpose of the trade study process is to ensure

that they move the design toward an optimum. The basic steps in that process are:

- Understand what the system's goals, objectives and constraints are, and what the system must do to meet them — that is, understand the functional requirements in the operating environment.
- Devise some alternative means to meet the functional requirements. In the early phases of the project cycle, this means focusing on system architectures; in later phases, emphasis is given to system designs.
- Evaluate these alternatives in terms of the outcome variables (system effectiveness, its underlying performance or technical attributes, and system cost). Mathematical models are useful in this step not only for forcing recognition of the relationships among the outcome variables, but also for helping to determine what the performance requirements must be quantitatively.
- Rank the alternatives according to an appropriate selection rule.
- Drop less-promising alternatives and proceed to next level of resolution, if needed.

This process cannot be done as an isolated activity. To make it work effectively, individuals with different skills — system engineers, design engineers, specialty engineers, program analysts, decision scientists and project managers — must cooperate. The right quantitative methods and selection rule must be used. Trade study assumptions, models and results must be documented as part of the system archives.

5.2 Cost Definition and Modeling

This section deals with the role of costs in the systems analysis and engineering process, how to measure it, how to control it, and how to obtain estimates of it. The reason costs and their estimates are of great importance in systems engineering goes back to the principal objective of systems engineering: fulfilling the system's goals in the most cost-effective manner. The cost of each alternative should be one of the most important outcome variables in trade studies performed during the systems engineering process.

One role, then, for cost estimates is in helping to choose rationally among alternatives. Another is as a control mechanism during the project cycle. Cost measures produced for project cycle reviews are important in determining whether the system goals and objectives are still deemed valid and achievable, and whether constraints and boundaries are worth maintaining. These measures are also useful in determining whether system goals and objectives have properly flowed down through to the various subsystems.

As system designs and operational concepts mature, cost estimates should mature as well. At each review, cost estimates need to be presented and compared to the funds likely to be available to complete the project. The cost estimates presented at early reviews must be given special attention since they usually form the basis under which authority to proceed with the project is given. Systems engineering must be able to provide realistic cost estimates to project managers. In the absence of such estimates, overruns are likely to occur, and the credibility of the entire system development process, both internal and external, is threatened.

5.2.1 Life-Cycle Cost (LCC) and Other Cost Measures

A number of questions need to be addressed so that costs are properly treated in systems analysis and engineering. These questions include:

- What costs should be counted?
- How should costs occurring at different times be treated?
- What about costs that cannot easily be measured in dollars?

What Costs Should be Counted. The most comprehensive measure of the cost of an alternative is its life-cycle cost. According to NMI 7100.14B, *Major System Acquisitions*, a system's life-cycle cost is "the sum total cost of the direct, indirect, recurring, nonrecurring, and other related costs incurred, or estimated to be incurred in the design, development, production, operation, maintenance, and support [of it] over its anticipated useful life span." A less formal definition of a system's life-cycle cost is the total cost of acquiring, owning and disposing of it over its entire lifetime. System life-cycle cost should be estimated and used in the evaluation of alternatives during trade studies. The system engineer should include in the life-cycle cost those resources, like civil service work-years, that may not require explicit expenditures. A system's life-cycle cost, when properly computed, is the best measure of its cost to NASA.

Life-cycle cost has several components, as shown in Figure 23. Applying the informal definition above, life-cycle cost consists of (a) the costs of acquiring a usable system, (b) the costs of operating and supporting it over its

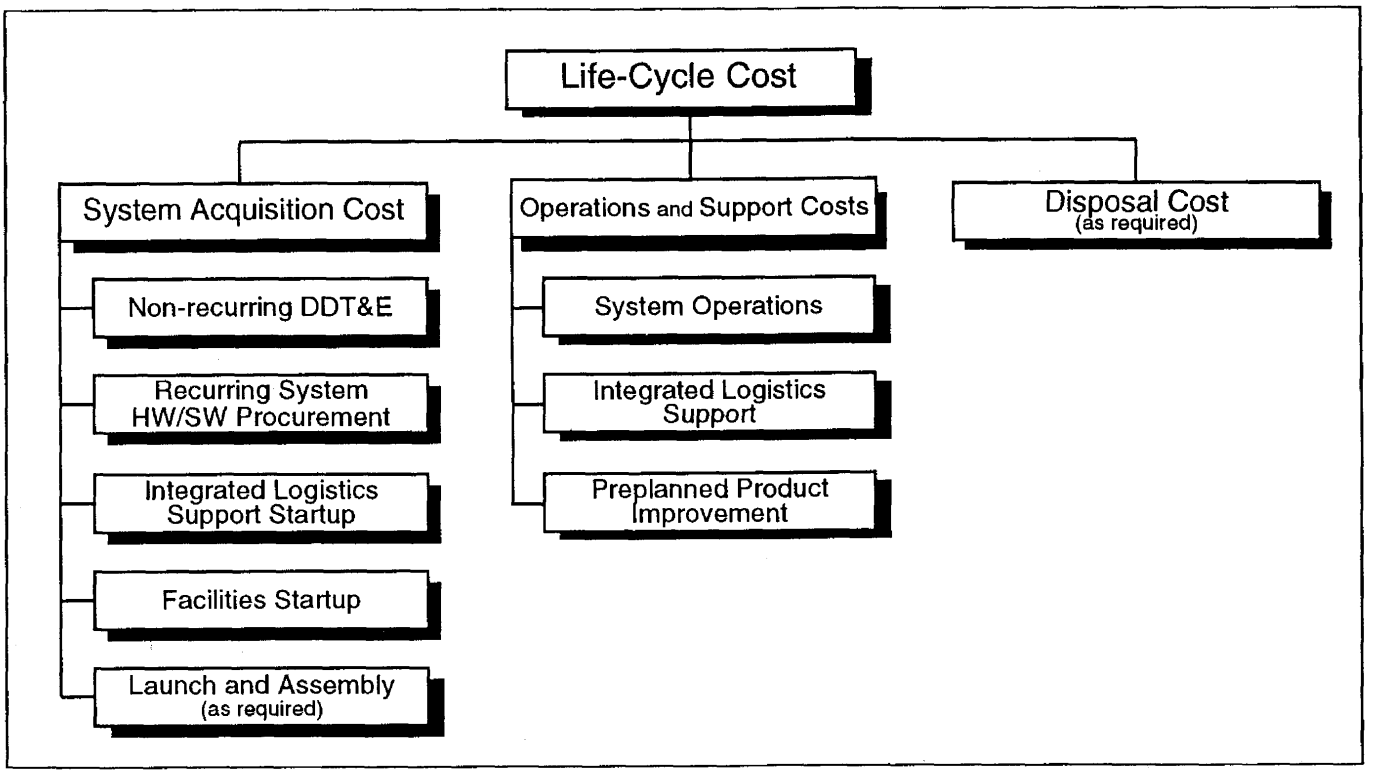


Figure 23 — Life-Cycle Cost Components.

useful life, and (c) the cost of disposing of it at the end of its useful life. The system acquisition cost includes more than the DDT&E and procurement of the hardware and software; it also includes the other start-up costs resulting from the need for initial training of personnel, initial spares, the system's technical documentation, support equipment, facilities and any launch services needed to place the system at its intended operational site.

The costs of operating and supporting the system include, but are not limited to, operations personnel and supporting activities, ongoing integrated logistics support, and pre-planned product improvement. For a major system, these costs are often substantial on an annual basis, and when accumulated over years of operations can constitute the majority of life-cycle cost.

At the start of the project cycle, all of these costs lie in the future. At any point in the project cycle, some costs will have been expended. These expended resources are known as *sunk costs*. For the purpose of doing trade studies, the sunk costs of any alternative under consideration are irrelevant, no matter how large. The only costs relevant to current design trades are those that lie in the future. The logic is straightforward: the way resources were spent in the past cannot be changed. Only decisions regarding the way future resources are spent can be made. Sunk costs may alter the cost of continuing with a particular alternative relative to others, but when choosing among alternatives, only those costs that remain should be counted.

At the end of the system lifetime, some systems may have a positive *residual* or *salvage value*. This value exists if the system can be sold, bartered or used by another system. This value needs to be counted in life-cycle cost, and is generally treated as a negative cost.

Costs Occurring Over Time. The life-cycle cost combines costs that typically occur over a period of several years. Costs incurred in different years cannot be treated the same because they, in fact, represent different resources to society. A dollar wisely invested today will return somewhat more than a dollar next year. Treating a dollar today the same as a dollar next year ignores this potential trade.

Discounting future costs is a way of making costs occurring in different years commensurable. When applied to a stream of future costs, the discounting procedure yields the present discounted value (PDV) of that stream. The effect of discounting is to reduce the contribution of costs incurred in the future relative to costs incurred in the near term. Discounting should be performed whether or not there is any inflation, though care must be taken to ensure the right discount rate is used. (See sidebar on PDV.)

Calculating Present Discounted Value

Calculating the PDV is a way of reducing a stream of costs to a single number so that alternative streams can be compared unambiguously. Several formulas for PDV are used, depending on whether time is to be treated as a discrete or a continuous variable, and whether the project's time horizon is finite or not. The following equation is useful for evaluating system alternatives when costs have been estimated as yearly amounts, and the project's anticipated useful life is T years. For alternative i,

$$PDV_i = \sum_{t=0}^{T} C_{it} (1 + r)^{-t}$$

where r is the annual discount rate and C_{it} is the estimated cost of alternative i in year t.

Once the yearly costs have been estimated, the choice of the discount rate is crucial to the evaluation since it ultimately affects how much or how little runout costs contribute to the PDV. While calculating the PDV is generally accepted as the way to deal with costs occurring over a period of years, there is much disagreement and confusion over the appropriate discount rate to apply in systems engineering trade studies. The Office of Management and Budget (OMB) has mandated the use of a rate of ten percent for NASA systems when constant dollars (dollars adjusted to the price level as of some fixed point in time) are used in the equation. When nominal dollars (sometimes called then-year, runout or real-year dollars) are used, the OMB-mandated annual rate should be increased by the inflation rate assumed for that year. Either approach yields essentially the same PDV. For more information, see OMB Circular A-94, *Discount Rates To Be Used In Evaluating Time-Distributed Costs and Benefits*, March 1972.

In trade studies, different alternatives often have cost streams that differ with respect to time. One alternative with higher acquisition costs than another may offer lower operations and support costs. Without discounting, it would be difficult to know which stream truly represents the lower life-cycle cost. Trade studies should report the PDV of life-cycle cost for each alternative as an outcome variable.

Difficult-To-Measure Costs. In practice, some costs pose special problems. These special problems, which are not unique to NASA systems, usually occur in two areas: (a) when alternatives have differences in the irreducible chances of loss of life and (b) when externalities are present. Two examples of externalities that impose costs are pollution caused by some launch systems and the creation of orbital debris. Because it is difficult to place a dollar figure on these resource uses, they are generally called *incommensurable costs*. The general treatment of these types of costs in trade studies is not to ignore them, but instead to keep track of them along with dollar costs.

5.2.2 Controlling Life-Cycle Costs

Management objectives with regard to the life-cycle cost of a major system are expressed in NMI 7100.14B, *Major System Acquisitions*. These are to:

- Maintain an agency capability to predict, review, assess, negotiate and monitor life-cycle costs for a program
- Be able to assess acquisition cost, schedule and performance experience against predictions, and provide such assessments for consideration by the Administrator at key decision points
- Estimate life-cycle costs to ensure that appropriate tradeoffs among investment (acquisition) costs, ownership costs, schedules and performance are made
- Use independent cost estimates, where feasible, for comparison purposes.

There are a number of actions the system engineer can take to effect these objectives. *Early decisions in the systems engineering process tend to have the greatest effect on the resultant system life-cycle cost.* Typically, by the time the preferred system architecture is selected, between 50 and 70 percent of the system's life-cycle cost has been "locked in". By the time a preliminary system design is selected, this figure may be as high as 90 percent. This presents a major dilemma to the system engineer, who must lead this selection process. Just at the time when decisions are most critical, the state of information about the alternatives is least certain. Uncertainty about costs is a fact of systems engineering.

This suggests that efforts to acquire better information about the life-cycle cost of each alternative early in the project life-cycle (Phases A and B) potentially have very high payoffs. The system engineer needs to understand what the principal life-cycle cost drivers are. Some major questions to consider are: How much does each alternative rely on well-understood technology? Can the system be manufactured using routine processes or are higher precision processes required? What tests are needed to verify and validate each alternative system design, and how costly are they? What reliability levels are needed by

each alternative? What environmental and safety requirements must be satisfied?

For a system whose operational life is expected to be long and to involve complex activities, the life-cycle cost is likely to be far greater than the acquisition costs alone. Consequently, it is particularly important with such a system to bring in the specialty engineering disciplines such as reliability, maintainability, supportability and operations engineering early in the systems engineering process, as they are essential to proper life-cycle cost estimation.

Another mechanism for controlling life-cycle cost is to establish a *life-cycle cost management program* as part of the project's management approach. Such a program establishes life-cycle cost as a design goal, perhaps with sub-goals for annual acquisition costs or operations and support costs. More specifically, the objectives of a life-cycle cost management program are to:

- Identify a common set of ground rules and assumptions for life-cycle cost estimation
- Ensure that best-practice methods, tools and models are used for life-cycle cost analysis
- Track the estimated life-cycle cost throughout the project cycle; and, most important
- Integrate life-cycle cost considerations into the design and development process via trade studies and formal change control assessments.

Trade studies and formal change control assessments provide the means to optimize the effectiveness and life-cycle cost of the system. The complexity of integrating life-cycle cost considerations into the design and development process should not be underestimated, but neither should the benefits, which can be measured in terms of greater cost-effectiveness. The existence of a rich set of potential life-cycle cost trades makes this complexity even greater.

The Space Station *Freedom* Program provides many examples of such potential trades. As one example, consider the life-cycle cost effect of increasing the mean time between failures (MTBF) of *Freedom's* Orbital Replacement Units (ORUs). This is likely to increase the acquisition cost, and may increase the weight, of the station. However, annual maintenance hours and the weight of annual replacement spares will decline. The same station availability may be achieved with fewer on-orbit spares, thus saving precious internal volume used for spares storage. If the ORUs are external to the station, then the amount of extravehicular activity, with its associated logistics support, will also decline. With such complex interactions, it is difficult to know what the optimum point is. At a minimum, the system engineer must have the capability to assess the life-cycle cost of each alternative. (See Appendix B.8 on the effects of ORU MTBF on SSF.)

5.2.3 Cost Estimating

The techniques used to estimate each life-cycle cost component usually change as the project cycle proceeds. Methods and tools used to support budget estimates and life-cycle cost trades in Phase A may not be sufficiently detailed to support those activities during Phase C/D. Further, as the project cycle proceeds, the requirements and the system design mature as well, revealing greater detail in the Work Breakdown Structure (WBS). This should enable the application of cost estimating techniques at a greater resolution.

Three techniques are described below — parametric cost models, analogy and grass-roots. Typically, the choice of technique depends on the state of information available to the cost analyst at each point in the project cycle. Table 4 shows this dependence.

Table 4 — Cost Estimating Techniques by Phase.

Technique	Pre-Phase A and Phase A	Phase B	Phase C/D
Parametric Cost Models	Primary	Applicable	May be applicable
Analogy	Applicable	Applicable	May be applicable
Grass-roots	Not applicable	Applicable	Primary

Parametric (or "top-down") cost models are most useful when only a few key variables are known or can be estimated. The most common example of a parametric model is the statistical Cost Estimating Relationship (CER). A single equation (or set of equations) is derived from a set of historical data relating one or more of a system's characteristics to its cost using well-established statistical methods. A number of statistical CERs have been developed to estimate a spacecraft's hardware acquisition cost. These typically use an estimate of its weight and other characteristics, such as design complexity and inheritance, to obtain an estimate of cost. Similarly, software CERs have been developed as well, relying on judgments about source lines of code and other factors to obtain development costs. (See sidebar on statistical CERs.)

Another type of parametric model relies on accepted relationships. One common example can be found in the application of logistics relationships to the estimation of repair costs and initial and recurring spares costs. The validity of these cost estimates also depends on the quality of the input parameters.

Statistical Cost Estimating Relationships: Example and Pitfalls

One model familiar to most cost analysts is the historically based CER. In its usual form, this model is a linear expression with cost (the dependent variable) as a function of one or more descriptive characteristics. The coefficients of the linear expression are estimated by fitting historical data from previously completed projects of a similar nature using statistical regression techniques. This type of model is analytic and deterministic. An example of this type of model for estimating the first unit cost, C, of a space-qualified Earth-orbiting receiver/exciter is:

$$\ln C = 3.39 + 0.97 \ln W + 0.6523 z$$

where W is the receiver's weight, and z is one if the receiver is intended for geosynchronous orbit, and zero otherwise; ln is the natural logarithm function. (Source: U.S. Air Force Systems Command-Space Division, *Unmanned Space Vehicle Cost Model, Sixth Edition*, November 1988.) CERs are used extensively in advanced technology systems, and have been challenged on both theoretical and practical grounds. One challenge can be mounted on the basis of the assumption of an unchanging relationship between cost and the independent variables. Others have questioned the validity of CERs based on weight, a common independent variable in many models, in light of advances in electronic packaging and composite materials. Objections to using statistical CERs also include problems of input accuracy, low statistical significance due to limited data points, ignoring the statistical confidence bands, and lastly, biases in the underlying data.

The principal advantages of parametric cost models are that the results are reproducible, are more easily documented than other methods, and often can be produced with the least amount of time and effort. This makes a properly constructed parametric cost model (that is, one whose inputs vary with the alternatives under consideration) very effective for use in trade studies.

Analogy is another way of estimating costs. When a new system or component has functional and performance characteristics similar to an existing one whose cost is known, the known cost can be adjusted to reflect engineering judgments about differences.

Grass-roots (or "bottoms-up") estimates are the result of rolling up the costs estimated by each organization performing work described in the WBS. Properly done, grass-roots estimates can be quite accurate, but each time a "what if" question is raised, a new estimate needs to be made. Each change of assumptions voids at least part of the old estimate. Because the process of obtaining grass-roots estimates is typically time-consuming and manpower-intensive, the number of such estimates that can be prepared during trade studies is in reality severely limited.

Whatever technique is used, the direct cost of each hardware and software element often needs to be "wrapped" (multiplied by a factor greater than one) to cover the costs of integration and test, program management, systems engineering, etc. These additional costs are called system-level costs, and are often calculated as percentages of the direct costs.

Using Parametric Cost Models. A number of parametric cost models are available for costing NASA systems. Some of these are shown in Table 5. Unfortunately, none alone is sufficient to estimate life-cycle cost. Assembling an estimate of life-cycle cost often requires that several different models (along with the other two techniques) be used together. To integrate the costs being estimated by these different models, the system engineer should ensure that the inputs to and assumptions of the models are consistent, that all relevant life-cycle cost components are covered, and that the timing of costs is correct.

Table 5 — Some Space Systems Parametric Cost Models.

Model	Source	Application
Unmanned Space Vehicle Cost Model (USCM)*	Air Force Systems Command Space Division	Unmanned Earth-orbiting space vehicles DDT&E, FH, AGE, LOOS**
Programmed Review of Information for Costing and Evaluation (PRICE)	GE/RCA	PRICE/H for electronic and mechanical hardware DDT&E and production, PRICE/S for software
Model for Estimating Space Station Operations Costs (MESSOC)	Space Station Freedom (SSF) Program Office	All mature operations costs for SSF
Software Costing Tool (SCT)	JPL	NASA manned and unmanned flight and ground software development costs
Multi-variable Instrument Cost Model (MICM)*	GSFC (Code 152.0)	Cost of developing and building prototype instruments
Marshall Space Flight Center Historical Cost Models*	MSFC	Subsystem-level DDT&E and FH costs for manned and unmanned spacecraft, and launch vehicles

* Statistically based cost estimating relationships
** FH = Flight Hardware
AGE = Aerospace Ground Equipment
LOOS = Launch and Orbital Operations Support

The system engineer may sometimes find it necessary to make some adjustments to model results to achieve a life-cycle cost estimate. One such situation occurs when the results of different models, whose estimates are expressed in different year *constant* dollars, must be combined. In that case, an appropriate inflation factor must be applied. Another such situation arises when a model produces a cost estimate for the first unit of a hardware item, but the project requires multiple units. In that case, a learning curve can be applied to the first unit cost to obtain the required multiple-unit estimate. (See sidebar on learning curves.)

A third situation requiring additional calculation occurs when a model provides a cost estimate of the total acquisition effort, but doesn't take into account the multi-year nature of that effort. The system engineer can use a set of "annual cost spreaders" based on the typical ramping-up and subsequent ramping-down of acquisition costs for that type of project. (See sidebar on beta curves.)

Although some general parametric cost models for space systems are already available, their proper use usually requires a considerable investment in learning time. For projects outside of the domains of these existing cost models, new cost models may be needed to support trade studies. Efforts to develop these need to begin early in the project cycle to ensure their timely application during the systems engineering process. Whether existing models or newly created ones are used, the SEMP and its associated

Learning Curve Theory

The learning curve (also known as the progress or experience curve) is the time-honored way of dealing with the empirical observation that the unit cost of fabricating multiple units of complex systems like aircraft and spacecraft tends to decline as the number increases. In its usual form, the theory states that as the total quantity produced doubles, the *cost per unit* decreases by a *constant percentage*. The cost per unit may be either the average cost over the number produced, or the cost of the last unit produced. In the first case, the curve is generally known as the cumulative average learning curve; in the second case, it is known as the unit learning curve. Both formulations have essentially the same rate of learning.

Let C(1) be the unit cost of the first production unit, and C(Q) be the unit cost of the Qth production unit, then learning curve theory states there is a number, b, such that

$$C(Q) = C(1)\, Q^b$$

The number b is specified by the rate of learning. A 90 percent learning rate means that the unit cost of the second production unit is 90 percent of the first production unit cost; the unit cost of the fourth is 90 percent of the unit cost of the second, and so on. In general, the ratio of C(2Q) to C(Q) is the learning rate, LR, expressed as a decimal; using the above equation, b = ln (LR)/ln 2, where ln is the natural logarithm.

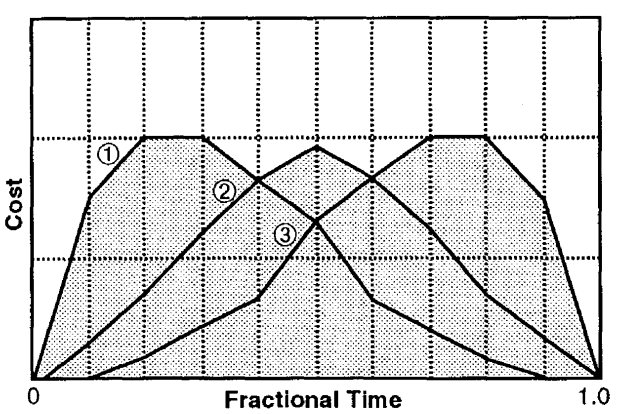
Learning curve theory may not always be applicable because, for example, the *time* rate of production has no effect on the basic equation. For more detail on learning curves, including empirical studies and tables for various learning rates, see Harold Asher, *Cost-Quantity Relationships in the Airframe Industry*, R-291, The Rand Corporation, 1956.

An Example of a Cost Spreader Function: The Beta Curve

One technique for spreading estimated acquisition costs over time is to apply the *beta curve*. This fifth-degree polynomial, which was developed at JSC in the late 60s, expresses the cumulative cost fraction as a function of the cumulative time fraction, T:

$$\text{Cum Cost Fraction} = 10T^2(1-T)^2(A+BT) + T^4(5-4T) \text{ for } 0 \le T \le 1.$$

A and B are parameters (with $0 \le A + B \le 1$) that determine the shape of the beta curve. In particular, these parameters control what fraction of the cumulative cost has been expended when 50 percent of the cumulative time has been reached. The figure below shows three examples: with A = 1 and B = 0 as in curve (1), 81 percent of the costs have been expended at 50 percent of the cumulative time; with A = 0 and B = 1 as in curve (2), 50 percent of the costs have been expended at 50 percent of the cumulative time; in curve (3) with A = B = 0, it's 19 percent.



Typically, JSC uses a 50 percent profile with A = 0 and B = 1, or a 60 percent profile with A = 0.32 and B = 0.68, based on data from previous projects.

life-cycle cost management plan should identify which (and how) models are to be used during each phase of the project cycle.

5.3 Effectiveness Definition and Modeling

The concept of system effectiveness is more elusive than that of cost. Yet, it is also one of the most important factors to consider in trade studies. In selecting among alternatives, the system engineer must take into account system effectiveness, even when it is difficult to define and measure reliably.

A measure of system effectiveness describes the accomplishment of the system's goals and objectives *quantitatively*. Each system (or family of systems with identical goals and objectives) has its own measure of system effectiveness. There is no universal measure of effectiveness for NASA systems, and no natural units with which to express effectiveness. Further, effectiveness is dependent on the context (i.e., project or supersystem) in which the system is being operated, and any measure of it must take this into account. The system engineer can, however, exploit a few basic, common features of system effectiveness in developing strategies for measuring it.

5.3.1 Strategies for Measuring System Effectiveness

System effectiveness is almost always multifaceted, and is typically the result of the combined effects of:

- System output quality
- Size or quantity of system output
- System coverage or comprehensiveness
- System output timeliness
- System availability.

A measure of effectiveness and its measurement method (i.e., model) should focus on the critical facet (or facets) of effectiveness *for the trade study issue under consideration*. Which facets are critical can often be deduced from the accompanying functional analysis. The functional analysis is also very useful in helping to identify the underlying system performance or technical attributes that mathematically determine system effectiveness. (Note that each of the above facets may have several dimensions. If this is the case, then each dimension can be considered a function of the underlying system performance or technical attributes.) Ideally, there is a strong connection between the system functional analysis, system effectiveness measure, and the functional and performance requirements. The same functional analysis that results in the functional requirements flowdown also yields the system effectiveness and performance measures that are optimized (through trade studies) to produce the system performance requirements.

An effectiveness measurement method or model should provide trustworthy relationships between these underlying performance or technical attributes and the measure of system effectiveness. Early in the project cycle, the effectiveness model may embody simple parametric relationships among the high-level performance and technical attributes and the measure of system effectiveness. In the later phases of the project cycle, the effectiveness model may use more complex relationships requiring more detailed, specific data on operational scenarios and on each of the alternatives. In other words, early effectiveness modeling during architecture trade studies may take a functional view, while later modeling during design trade studies may shift to a product view. This is not unlike the progression of the cost modeling from simple parametrics to more detailed grass-roots estimates.

The system engineer must tailor the effectiveness measure and its measurement method to the resolution of the system design. As the system design and operational concept mature, effectiveness estimates should mature as well. The system engineer must be able to provide realistic estimates of system effectiveness and its underlying performance and technical attributes not only for trade studies, but for project management through the tracking of TPMs.

This discussion so far has been predicated on one accepted measure of system effectiveness. The job of computing system effectiveness is considerably easier when the system engineer has a single measure and measurement method (model). But, as with costs, a single measure may not be possible. When it does not exist, the system engineer must fall back to computing the critical,

Practical Pitfalls in Using Effectiveness Measures in Trade Studies

Obtaining trustworthy relationships among the system performance or technical attributes and system effectiveness is often difficult. Purported effectiveness models often only treat one or two of the facets described above. Supporting models may not have been properly integrated. Data are often incomplete or unreliable. Under these conditions, reported system effectiveness results for different alternatives in a trade study may show only the *relative* effectiveness of the alternatives within the context of that trade study. The system engineer must recognize the practical pitfalls of using such results.

high-level, but nevertheless still underlying, system performance or technical attributes. In effect, these high-level performance or technical attributes are elevated to the status of measures of (system) effectiveness (MOEs) for trade study purposes, even though they do not represent a true measure of system effectiveness.

These high-level performance or technical attributes might represent one of the facets described above, or they may be only components of one. They are likely to require knowledge or estimates of lower-order performance or technical attributes. Figure 24 shows how system effectiveness might look in an hierarchical tree structure. This figure corresponds, in some sense, to Figure 23 on life-cycle cost, though rolling up by simple addition obviously does not apply to system effectiveness.

Lastly, it must be recognized that system effectiveness, like system cost, is uncertain. This fact is given a fuller treatment in Section 5.4.

5.3.2 NASA System Effectiveness Measures

The facets of system effectiveness in Figure 24 are generic. Not all will apply to a particular system. The system engineer must determine which performance or technical attributes make up system effectiveness, and how they should be combined, on a system-by-system basis. Table 6 provides examples of how each facet of system effectiveness could be interpreted for specific classes of NASA flight systems. No attempt has been made to enumerate all possible performance or technical attributes, or to fill in each possible entry in the table; its purpose is illustrative only.

For many of the systems shown in the table, system effectiveness is largely driven by continual (or continuous) operations at some level of output over a period of years. This is in contradistinction to an *Apollo*-type project, in which the effectiveness is largely determined by the successful completion of a single flight within a clearly specified time horizon. The measures of effectiveness in these two cases are correspondingly different. In the former case (with its lengthy operational phase and continual output), system effectiveness measures need to incorporate quantitative measures of availability. The system engineer accomplishes that through the involvement of the specialty engineers and the application of specialized models described in the next section.

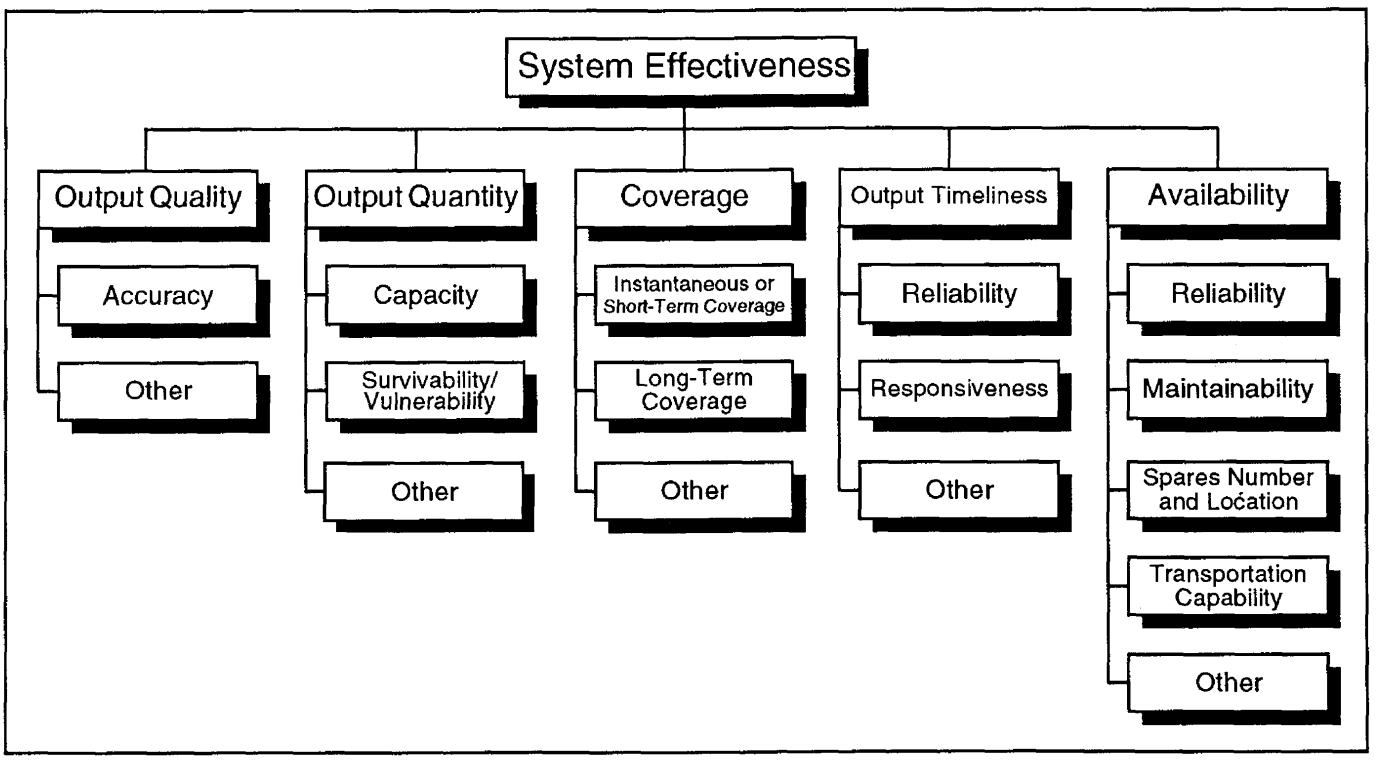


Figure 24 — System Effectiveness Components (Generic).

Table 6 — Facets of Effectiveness for Classes of NASA Flight Systems.

System Class	Output Quality	Output Quantity	Coverage or Comprehensiveness	Output Timeliness	Availability
Launch Systems	Launch reliability; safety during launch; safety during pre-launch processing	User payload capability to LEO, GEO, GTO, etc.		(See availability)	Probability of on-schedule launch (no system-induced postponements)
Inhabited Space Stations	Microgravity environment; operations safety	Annual user-available power, IVA, EVA, pressurized volume, upmass, downmass, CPU time, data storage, uplink, downlink, attach point time, etc.		Data/sample return time	Ratio of operational uptime to total time
Robotic Surface Exploration Rovers		Number of sites/samples	Site/sample diversity	Data/sample return time; probability of meeting launch window	Probability of meeting design life
Astrophysical Observatories	Instrument resolution; bit error rate	Annual observation time	Field of view; instrument synergy; spectral diversity	Data return time; responsiveness to unexpected opportunities	Ratio of operational uptime to total time
Planetary Spacecraft/ Probes	(same as above)	Number of observation targets	(same as above)	Probability of meeting launch window	Probability of meeting design life
Earth Observatories	(same as above)	Annual observation time	(same as above)	Simultaneity of observations	Ratio of operational uptime to total time

5.3.3 Availability and Logistics Supportability Modeling

One reason for emphasizing availability and logistics supportability in this chapter is that future NASA systems are less likely to be of the "launch-and-logistically forget" type. To the extent that logistic support considerations are major determinants of system effectiveness during operations, it is essential that logistics support be thoroughly analyzed in trade studies during the earlier phases of the project cycle. A second reason is that availability and logistics supportability have been rich domains for methodology and model development. The increasing sophistication of the methods and models has allowed the system-wide effects of different support alternatives to be more easily predicted. In turn, this means more opportunities to improve system effectiveness (or to lower life-cycle

Logistics Supportability Models: Two Examples

Logistics supportability models utilize the reliability and maintainability attributes of a particular system design, and other logistics system variables, to quantify the demands (i.e., requirements) for scarce logistics resources during operations. The models described here were both developed for Space Station *Freedom*. One is a stochastic simulation in which each run is a "trial" drawn from a population of outcomes. Multiple runs must be made to develop accurate estimates of means and variances for the variables of interest. The other is a deterministic analytic model. Logistic supportability models may be of either type. These two models deal with the unique logistics environment of *Freedom*.

SIMSYLS is a comprehensive stochastic simulation of on-orbit maintenance and logistics resupply of *Freedom*. It provides estimates of the demand (means and variances) for maintenance resources such as EVA and IVA, as well as for logistics upmass and downmass resources. In addition to the effects of actual and false ORU failures, the effects of various other stochastic events such as launch vehicle and ground repair delays can be quantified. *SIMSYLS* also produces several measures of operational availability. The model can be used in its availability mode or in its resource requirements mode.

M-SPARE is an availability-based optimal spares model. It determines the mix of ORU spares at any spares budget level that maximizes station availability, defined as the probability that no ORU had more demands during a resupply cycle than it had spares to satisfy those demands. Unlike *SIMSYLS*, *M-SPARE*'s availability measure deals only with the effect of spares. *M-SPARE* starts with a target availability (or budget) and determines the optimal inventory, a capability not possessed by *SIMSYLS*.

For more detail, see DeJulio, E., *SIMSYLS User's Guide*, Boeing Aerospace Operations, February 1990, and Kline, Robert, et al., *The M-SPARE Model*, LMI, NS901R1, March 1990.

cost) through the integration of logistics considerations in the system design.

Availability models relate system design and integrated logistics support technical attributes to the availability component of the system effectiveness measure. This type of model predicts the resulting system availability as a function of the system component failure and repair rates and the logistics support resources and policies. (See sidebar on measures of availability.)

Logistics supportability models relate system design and integrated logistics support technical attributes to one or more "resource requirements" needed to operate the system in the accomplishment of its goals and objectives. This type of model focuses, for example, on the system maintenance requirements, number and location of spares, processing facility requirements, and even optimal inspection policies. In the past, logistics supportability models have typically been based on measures pertaining to that particular resource or function *alone*. For example, a system's desired inventory of spares was determined on the basis of meeting measures of supply efficiency, such as percent of demands met. This tended to lead to suboptimal resource requirements from the system's point of view. More modern models of logistics supportability base resource requirements on the system availability effects. (See sidebar on logistics supportability models.)

Some availability models can be used to determine a logistics resource requirement by computing the quantity of that resource needed to achieve a particular level of availability, holding other logistics resources fixed. The line between availability models and logistics supportability models can be inexact. Some logistics supportability models may deal with a single resource; others may deal with several resources simultaneously. They may take the form of a simple database or spreadsheet, or a large computer simulation. Greater capability from these types of models is generally achieved only at greater expense in time and effort. The system engineer must determine what availability and logistics supportability models are needed for each new system, taking into account the unique operations and logistics concepts and environment of that system. Generally both types of models are needed in the trade study process to transform specialty engineering data into forms more useful to the system engineer. Which availability and logistics supportability models are used during each phase of the project cycle should be identified in the SEMP.

Measures of Availability

Availability can be calculated as the ratio of operating time to total time, where the denominator, total time, can be divided into operating time and "downtime". System availability depends on any factor that contributes to downtime. Underpinning system availability, then, are the reliability and maintainability attributes of the system design, but other logistics support factors can also play significant roles. If these attributes and support factors, and the operating environment of the system are unchanging, then several measures of *steady-state availability* can be readily calculated. (When steady-state conditions do not apply, availability can be calculated, but is made considerably more complex by the dynamic nature of the underlying conditions.) The equations below are for four concepts of steady-state availability that the system engineer should recognize.

- Inherent = MTBF / (MTBF + MTTR)
- Achieved = MTBMA / (MTBMA + MTTR + PM)
- General = MTBMA / (MTBMA + MTTR + PM + SPARES + OTHER)
- Operational = (MTBMA + IDLE) / (MTBMA + IDLE + MTTR + PM + SPARES + OTHER)

where:
MTBF = Mean time between failures
MTTR = Mean time to repair (or restore)
MTBMA = Mean time between maintenance actions (corrective and preventive)
PM = Mean downtime for preventive maintenance
SPARES = Mean downtime due to waiting for spares (or supplies)
IDLE = Idle time (stand-by or non-operating time)
OTHER = Mean downtime due to administrative delays, or waiting for maintenance or other resources

These steady-state availability measures can be calculated at a point in time, or as an average over a period of time. A further, but manageable, complication in calculating availability takes into account degraded modes of operation for redundant systems.

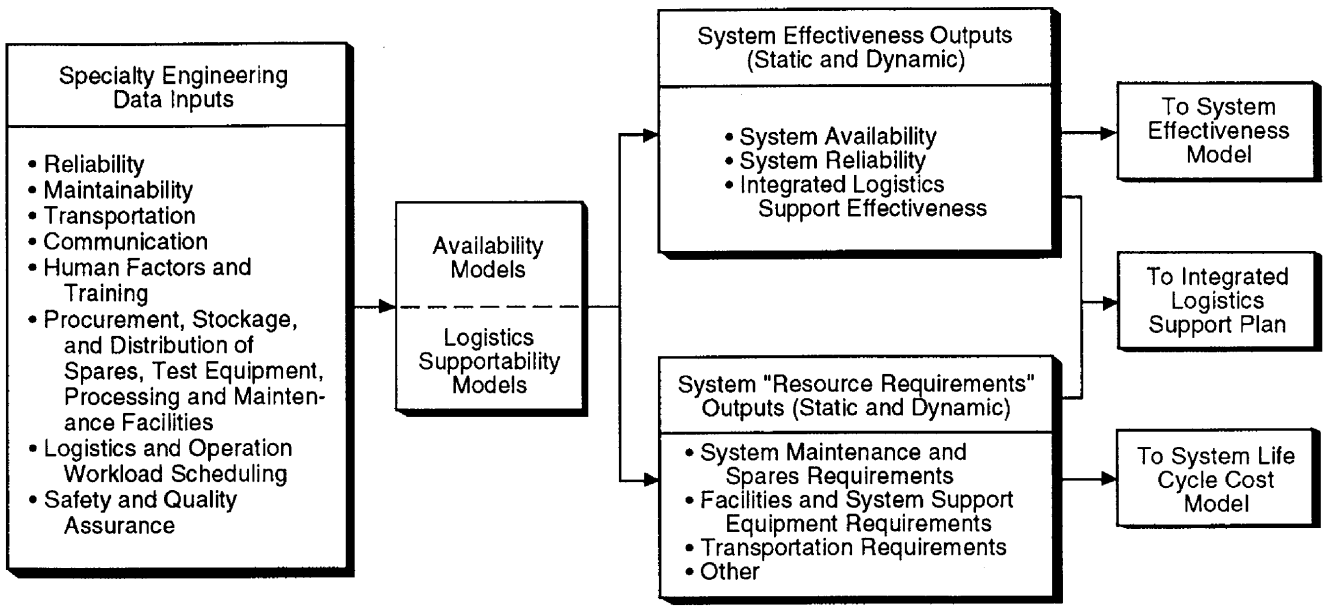


Figure 25 — Roles of Availability and Logistics Supportability Models.

Another role for these models is to provide quantitative requirements for incorporation into the system's formal Integrated Logistics Support (ILS) plan. Figure 25 shows the role of availability and logistics supportability models in the trade study process.

Essential to obtaining useful products from any availability and/or logistics supportability model is the collection of high quality specialty engineering data for each alternative system design. (Some of these data are also used in probabilistic risk assessments performed in risk management activities.) The system engineer must coordinate efforts to collect and maintain these data in a format suitable to the trade studies being performed. This task is made considerably easier by using digital databases in relational table formats such as the one currently under development for MIL-STD-1388-2B.

Continuing availability and logistics supportability modeling and data collection through the operations phase permits *operations trend analysis and assessment* on the system (e.g., is system availability declining or improving?) In general, this kind of analysis and assessment is extremely useful in identifying potential areas for product improvement such as greater system reliability, lower cost logistics support, and better maintenance and spares policies.

5.4 Probabilistic Treatment of Cost and Effectiveness

A probabilistic treatment of cost and effectiveness is needed when point estimates for these outcome variables do not "tell the whole story" — that is, when information about the variability in a system's projected cost and effectiveness is relevant to making the right choices about that system. When these uncertainties have the potential to drive a decision, the systems or program analyst must do more than just acknowledge that they exist. Some useful techniques for modeling the effects of uncertainty are described below in Section 5.4.2. These techniques can be applied to both cost models and effectiveness models, though the majority of examples given are for cost models.

5.4.1 Sources of Uncertainty in Models

There are a number a sources of uncertainty in the kinds of models used in systems analysis. Briefly, these are:

- Uncertainty about the correctness of the model's structural equations, in particular whether the functional form chosen by the modeler is the best representation of the relationship between an equation's inputs and output
- Uncertainty in model parameters, which are, in a very real sense, also chosen by the modeler; this uncertainty is evident for model coefficients derived

from statistical regression, but even known physical constants are subject to some uncertainty due to experimental or measurement error; and

- Uncertainty in the true value of model inputs (e.g., estimated weight or thermal properties) that describe a new system.

As an example, consider a cost model consisting of one or more statistical CERs. In the early phases of the project cycle (Phases A and B), this kind of model is commonly used to provide a cost estimate for a new NASA system. The project manager needs to understand what confidence he/she can have in that estimate.

One set of uncertainties concerns whether the input variables (for example, weight) are the proper explanatory variables for cost, and whether a linear or log-linear form is more appropriate. Model misspecification is by no means rare, even for strictly engineering relationships.

Another set of model uncertainties that contribute to the uncertainty in the cost estimate concerns the model coefficients that have been estimated from historical data. Even in a well-behaved statistical regression equation, the estimated coefficients could have resulted from chance alone, and therefore cost predictions made with the model have to be stated in probabilistic terms. (Fortunately, the upper and lower bounds on cost for any desired level of confidence can be easily calculated. Presenting this information along with the cost estimate is strongly recommended.)

The above uncertainties are present even if the cost model inputs that describe a new system are precisely known in Phase A. This is rarely true; more often, model inputs are subject to considerable guesswork early in the project cycle. The uncertainty in a model input can be expressed by attributing a probability distribution to it. This applies whether the input is a physical measure such as weight, or a subjective measure such as a "complexity factor." Model input uncertainty can extend even to a grass-roots cost model that might be used in Phases C and D. In that case, the source of uncertainty is the failure to identify and capture the "unknown-unknowns". The model inputs — the costs estimated by each performing organization — can then be thought of as variables having various probability distributions.

5.4.2 Modeling Techniques for Handling Uncertainty

The effect of model uncertainties is to induce uncertainty in the model's output. Quantifying these uncertainties involves producing an overall probability distribution for the output variable, either in terms of its probability density function (or mass function for discrete output variables) or its cumulative distribution function. (See sidebar on cost S-curves.) Some techniques for this are:

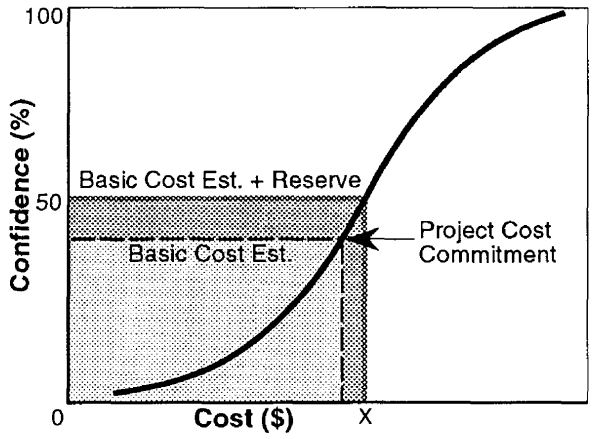
- Analytic solution
- Decision analysis
- Monte Carlo simulation.

Analytic Solution. When the structure of a model and its uncertainties permit, a closed-form analytic solution for the required probability density (or cumulative distribution) function is sometimes feasible. Examples can be found in simple reliability models.

Decision Analysis. This technique, which was discussed in Section 4.6, also can produce a cumulative distribution function, though it is necessary to descretize any continuous input probability distributions. The more probability

The Cost S-Curve

The cost S-curve gives the probability of a project's cost not exceeding a given cost estimate. This probability is sometimes called the budget confidence level. This curve aids in establishing the amount of contingency and Allowance for Program Adjustment (APA) funds to set aside as a reserve against risk.



In the S-curve shown above, the project's cost commitment provides only a 40 percent level of confidence, but with reserves, the level is increased to 50 percent. The steepness of the S-curve tells the project manager how much the level of confidence improves when a small amount of reserves are added.

Note that an Estimate at Completion (EAC) S-curve could be used in conjunction with the risk management approach described for TPMs (see Section 4.9.2), as another method of cost status reporting and assessment.

intervals are used, the greater the accuracy of the results, but the larger the decision tree. Furthermore, each uncertain model input adds more than linear computational complexity to that tree, making this technique less efficent in many situations than Monte Carlo simulation, described next.

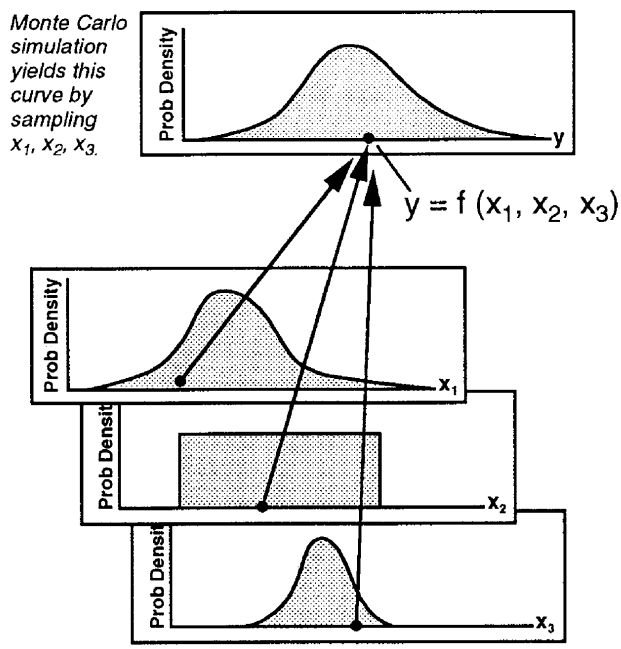


Figure 26 — A Monte Carlo Simulation with Three Uncertain Inputs.

Monte Carlo Simulation. This technique is often used to calculate an approximate solution to a stochastic model that is too complicated to be solved by analytic methods alone. A Monte Carlo simulation is a way of sampling input points from their respective domains in order to estimate the probability distribution of the output variable. In a simple Monte Carlo analysis, a value for each uncertain input is drawn at random from its probability distribution, which can be either discrete or continuous. This set of random values, one for each input, is used to compute the corresponding output value, as shown in Figure 26. The entire process is then repeated k times. These k output values constitute a random sample from the probability distribution over the output variable induced by the input probability distributions.

For an example of the usefulness of this technique, recall Figures 2 (in Chapter 2) and 22 (this chapter), which show the projected cost and effectiveness of three alternative design concepts as probability "clouds." These clouds may be reasonably interpreted as the result of three system-level Monte Carlo simulations. The information displayed by the clouds is far greater than that embodied in point estimates for each of the alternatives.

An advantage of the Monte Carlo technique is that standard statistical tests can be applied to estimate the precision of the resulting probability distribution. This permits a calculation of the number of runs (samples) needed to obtain a given level of precision. If computing time or costs are a significant constraint, there are several ways of reducing them through more deliberate sampling strategies. See *MSFC-HDBK-1912, Systems Engineering (Volume 2)* for a discussion of these strategies.

Commercial software to perform Monte Carlo simulation is available. These include add-in packages for some of the popular spreadsheets, as well as packages that allow the systems or program analyst to build an entire Monte Carlo model from scratch on a personal computer. These packages generally perform the needed computations in an efficient manner and provide graphical displays of the results, which is very helpful in communicating probabilistic information. For large applications of Monte Carlo simulation, such as those used in addressing logistics supportability, custom software may be needed. (See the sidebar on logistics supportability models.)

Monte Carlo simulation is a fairly easy technique to apply, and it offers the potential, as systems analysis and modeling capabilities improve, of greater understanding and communication what uncertainties mean for each alternative system architecture or design. A powerful example of this technique applied to NASA flight readiness certification is found in Moore, Ebbeler, and Creager, who combine Monte Carlo simulation with traditional reliability and risk analysis techniques.

Appendix A — Acronyms

Acronyms are useful because they provide a shorthand way to refer to an organization, a kind of document, an activity or idea, etc. within a generally understood context. Their overuse, however, can interfere with communications. The *NASA Lexicon* contains the results of an attempt to provide a comprehensive list of all acronyms used in NASA systems engineering. This appendix contains two lists: the acronyms used in this document and the acronyms for some of the major NASA organizations.

APA	Allowance for Program Adjustment
AR	Acceptance Review
ACWP	Actual Cost of Work Performed
AGE	Aerospace Ground Equipment
AHP	Analytic Hierarchy Process
BCWP	Budgeted Cost of Work Performed
BCWS	Budgeted Cost of Work Scheduled
C/SCSC	Cost/Schedule Cost System Criteria
CCB	Change Control Board
CDR	Critical Design Review
CER	Cost Estimating Relationship
CI	Configuration Item
CIAR	Configuration Item Acceptance Review
CIL	Critical Items List
CDSR	Critical Design Safety Review
CoDR	Conceptual Design Review
CoDSR	Conceptual Design Safety Review
COTR	Contracting Office Technical Representative
CPM	Critical Path Method
CR	Change Request
CRWG	Computer Resources Working Group
CSM	Center for Systems Management
CWBS	Contract Work Breakdown Structure
DDT&E	Design, Development, Test and Evaluation
DoD	(U.S.) Department of Defense
DSMC	Defense Systems Management College
EAC	Estimate at Completion
ECP	Engineering Change Proposal
ECR	Engineering Change Request
EMC	Electromagnetic compatibility
EMI	Electromagnetic interference
EOM	End of Mission
EVA	Extravehicular Activities
EVM	Earned Value Measurement
FCA	Functional Configuration Audit
FFBD	Functional Flow Block Diagram
FH	Flight Hardware
FMEA	Failure Modes and Effects Analysis
FMECA	Failure Modes, Effects and Criticality Analysis
GOES	Geosynchonous Orbiting Environmental Satellite
GSE	Ground Support Equipment
HQ	NASA Headquarters
I&V	Integration and Verification
ICSEWG	(NASA) Inter-Center Systems Engineering Working Group
ILS	Integrated Logistics Support
IOP	Institutional Operating Plan
IRAS	Infrared Astronomical Satellite
IVA	Intravehicular Activities
LCC	Life-cycle Cost
LEO	Low-Earth Orbit
LMEPO	Lunar/Mars Exploration Program Office
LMI	Logistics Management Institute
LOOS	Launch and Orbital Operations Support
MESSOC	Model for Estimating Space Station Operations Cost
MICM	Multi-variable Instrument Cost Model
MNS	Mission Needs Statement
MOE	Measure of (system) effectiveness
MTBF	Mean Time Between Failures
MTBMA	Mean Time Between Maintenance Actions
MTTR	Mean Time to Repair/Restore
MUL	Material Utilization List
NAR	Non-Advocate Review
NHB	NASA Handbook
NMI	NASA Management Instruction
NOAA	National Oceanic and Atmospheric Administration
OMB	Office of Management and Budget (Executive Branch)
ORU	Orbital Replacement Unit
PBS	Product Breakdown Structure
PCA	Physical Configuration Audit
PDCR	Project Definition and Cost Review
PDR	Preliminary Design Review
PDSR	Preliminary Design Safety review
PDT	Product Development Team
PDV	Present Discounted Value
PERT	Program Evaluation and Review Technique
PM	Preventative Maintenance
POP	Program Operating Plan
PRA	Probabilistic Risk Assessment
PRR	Program/Project Requirements Review
PRSR	Project Requirements Safety Review
RAS	Requirements Allocation Sheet
RTG	Radioisotope Thermoelectric Generator
SAR	System Acceptance Review
SASR	System Acceptance Safety Review
SDF	Software Development Folder
SEB	Source Evaluation Board
SEMP	Systems Engineering Management Plan
SFQR	System Formal Qualification Review

PAGE 86 INTENTIONALLY BLANK

PRECEDING PAGE BLANK NOT FILMED

SI	*Le Système International d' Unités* (the international [metric] system of units)
SIRTF	Space Infrared Telescope Facility
SOFIA	Stratospheric Observatory for Infrared Astronomy
STS	Space Transportation System
SSF	Space Station *Freedom*
TBD	To Be Determined; To Be Done
TDRS	Tracking and Data Relay Satellite
TLA	Time Line Analysis
TLS	Time Line Sheet
TPM	Technical Performance Measure(ment)
TQM	Total Quality Management
TRR	Test Readiness Review
WBS	Work Breakdown Structure
WFD	Work Flow Diagram

NASA Organizations

ARC	Ames Research Center, Moffett Field CA 94035
COSMIC	Computer Software Management & Information Center, University of Georgia, 382 E. Broad St., Athens GA 30602
DFRF	Dryden Flight Research Facility (ARC), P.O. Box 273, Edwards CA 93523
GISS	Goddard Institute for Space Studies (GSFC), 2880 Broadway, New York NY 10025
GSFC	Goddard Space Flight Center, Greenbelt Rd., Greenbelt MD 20771
HQ	NASA Headquarters, Washington DC 20546
JPL	Jet Propulsion Laboratory, 4800 Oak Grove Dr., Pasadena CA 91109
JSC	Lyndon B. Johnson Space Center, Houston TX 77058
KSC	John F. Kennedy Space Center, Kennedy Space Center FL 32899
LaRC	Langley Research Center, Hampton VA 23665
LeRC	Lewis Research Center, 21000 Brookpark Rd., Cleveland OH 44135
MAF	Michoud Assembly Facility, P.O. Box 29300, New Orleans LA 70189
MSFC	George C. Marshall Space Flight Center, Marshall Space Flight Center AL 35812
NASA	National Aeronautics and Space Administration, Washington DC 20546
OAET	NASA Office of Aeronautics, Exploration and Technology (formerly, OAST and OEXP)
OAST	NASA Office of Aeronautics and Space Technology (now OAET)
OCP	NASA Office of Commercial Programs
OEXP	NASA Office of Exploration (now OAET)
OMB	U.S. Office of Management and Budget
OSF	NASA Office of Space Flight
OSSA	NASA Office of Space Science and Applications
SCC	Slidell Computer Complex, 1010 Gauss Blvd, Slidell LA 70458
SSC	John C. Stennis Space Center, Stennis Space Center MS 39529
STIF	Scientific & Technical Information Facility, P.O. Box 8757, BWI Airport MD 21240
WFF	Wallops Flight Facility (GSFC), Wallops Island VA 23337
WSTF	White Sands Test Facility (JSC), P.O. Drawer MM, Las Cruces NM 88004

Appendix B — Systems Engineering Templates and Examples

B.1 A "Tailored" Project Cycle for R&D Projects

Appendix B.1 was contributed by Vincent J. Bilardo, Jr., Chief, Systems Evaluation and Integration Branch, Advanced Life Support Division, NASA/Ames Research Center.

• • •

As an example of the principle of tailoring, a customized project life cycle has been developed for a generic ground-based advanced technology demonstrator testbed project. The technology demonstrator testbed concept is typical of many research and technology development projects that are or will be pursued in order to ready the next generations of technology required for the Space Exploration Initiative. The specific project milestones and data products shown in Figure B-1, Figure B-2 and Figure B-3 are envisioned to be typical of a testbed project with a total life cycle cost, including operating expenses, of $5–20M. Figure B-1 shows a proposed project life cycle for a technology demonstrator testbed. The first feature to note is that the life cycle for the testbed project has been organized into three major phases, rather than the six phases of the generic cycle shown in Figure 5. Each of these three major cycles has in turn been decomposed into three or more sub-cycles, each of which is unique to the needs of the project at that point in its development. There a fewer major review milestones, or "control gates", for the testbed project as compared to the generic project cycle, and the milestones shown in Figure B-1 reflect the unique nature of a ground-based testbed project. Specifically, the testbed project consists of both "technology systems", which are being demonstrated in the testbed, and "support systems", which are primarily facility-oriented but which have to be designed, built and tested nonetheless. The difference in complexity between the $5–20M ground-based testbed, and a generic program or project which can be very large and expensive, such as Space Station *Freedom*, is readily apparent in comparing Figure 5 to Figure B-1.

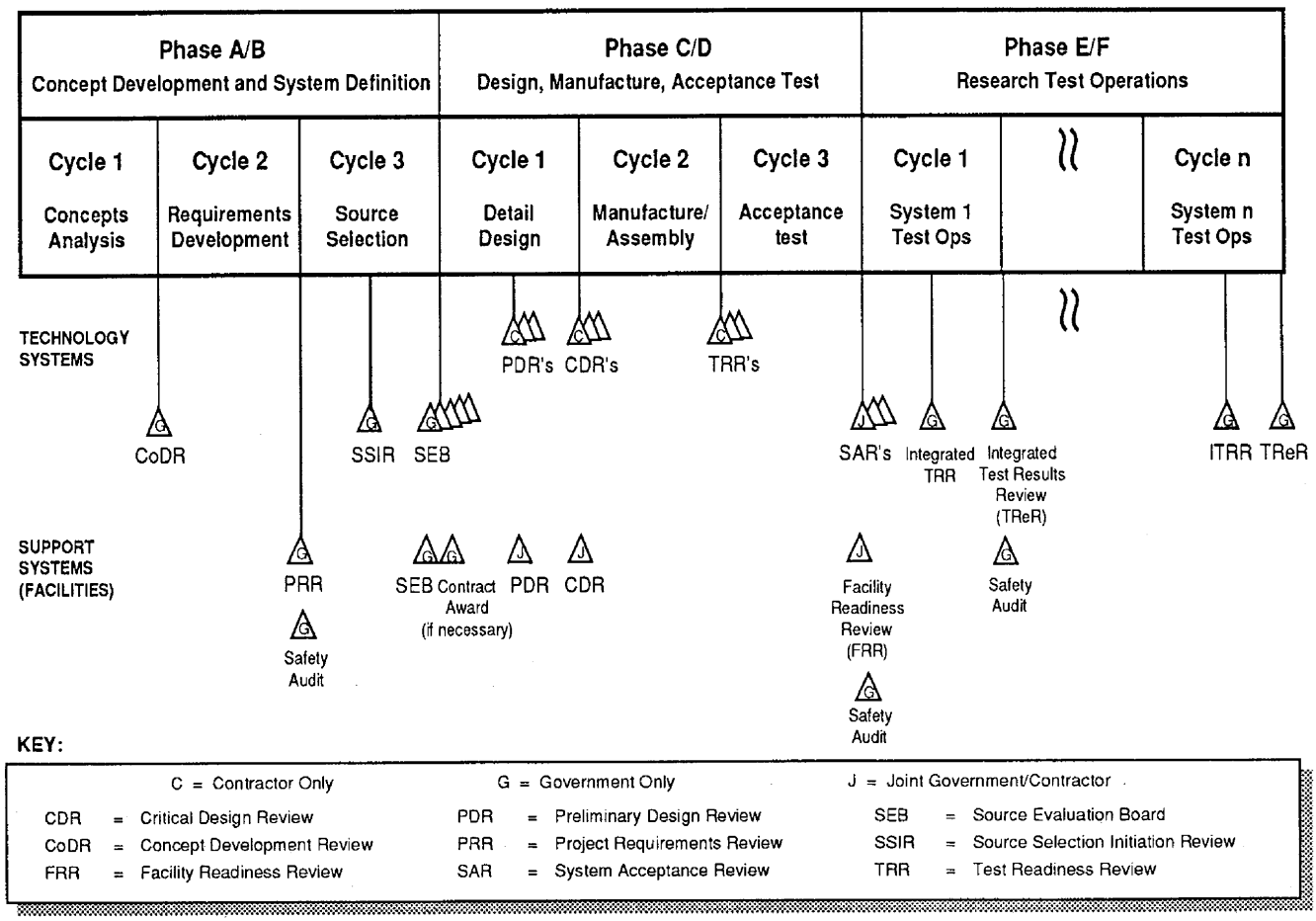


Figure B-1 — Tailored Project Life Cycle, Advanced Technology Testbed Project.

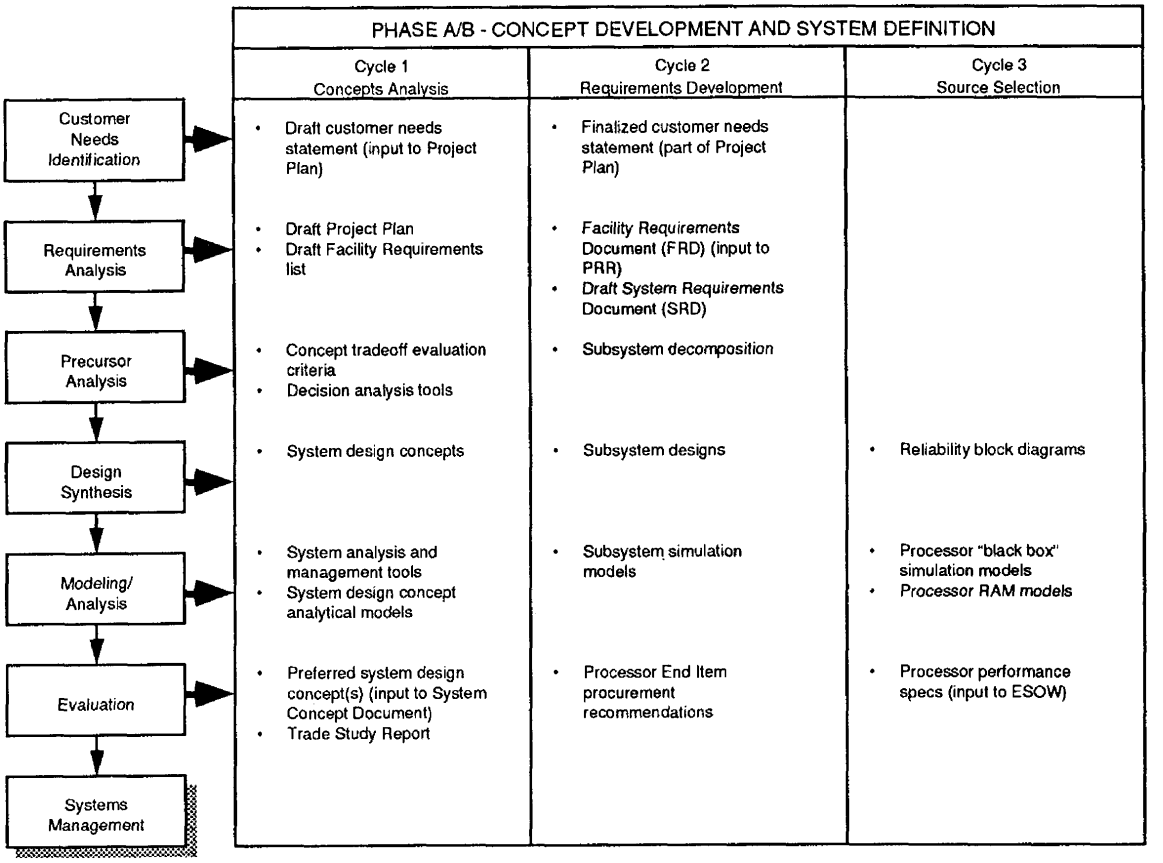


Figure B-2 — Major Products of Generic System Analysis, Advanced Technology Testbed Project.

Thus, the concept of tailoring is seen to be critical to rendering a project, and the systems engineering process which guides it, tractable and affordable.

Once the project's life cycle activities have been established to first order, the next step is to tailor the systems engineering process that is required during each sub-cycle in the project life cycle. One method for accomplishing this tailoring is illustrated in Figures B-2 and B-3. The first step is to assume a generic systems engineering process which consists of three categories of activities: *systems analysis, system management, and system development.* For the purposes of this tailoring example, systems analysis is defined to consist of the steps shown on the left side of Figure B-2, namely: (a) customer needs identification, (b) requirements analysis, (c) precursor analysis (such as risk analysis, functional analysis, or requirements allocation), (d) design synthesis, (e) modeling/analysis, and (f) evaluation. By referring to the complete list of generic data products developed by Forsberg, et al., the products which are desired for each sub-cycle of each phase of the technology testbed project cycle can be identified and aligned with the proper step of systems analysis shown on the left hand side of Figure B-2. Typical data products that are appropriate to the ground-based technology testbed in question are shown.

The same procedure is used to identify the desired products of system management, which is defined in this example to consist of: (a) system baseline and configuration management, (b) requirements flowdown, (c) implementation planning, (d) design review and audit, (e) verification and validation, and (f) program/project milestone review control gate. Typical documents and milestones required for the testbed project as a result of system management activities during the first three cycles of the project are shown in Figure B-3.

Identification of products to be generated during system development can be identified in the same way as before. Note that the typical activities of system development are: (a) detail design, (b) fabrication/procurement, (c) assembly/integration, and (d) test and evaluation.

The process of identifying the data products from the generic NASA Project Cycle that match up with the generic activities of each category of systems engineering is then repeated for each sub-cycle of each phase of the tailored project cycle. Similarly, the generic systems engineering activities outlined herein can, and should, be tai-

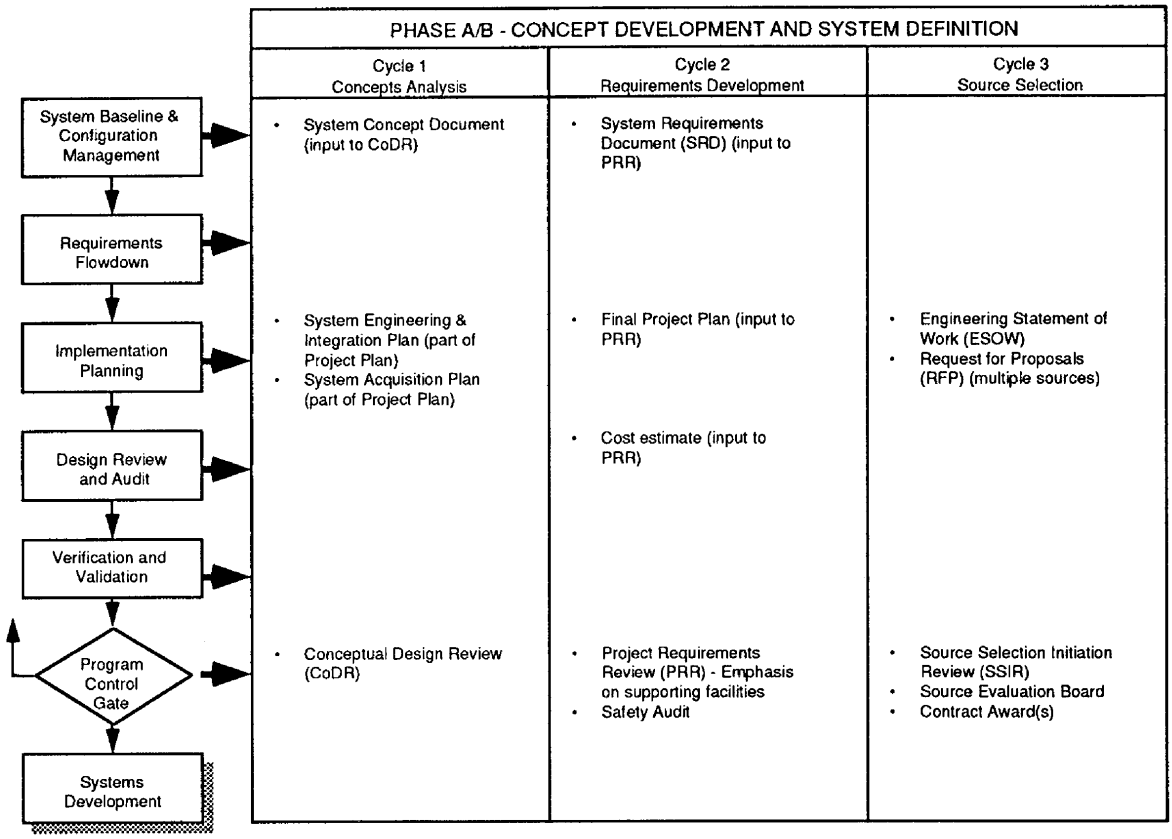


Figure B-3 — Major Products of Generic System Management Activities, Advanced Technology Testbed Project.

lored to meet the specific needs of the program or project of interest. This principle is illustrated in Figure 3, in which there are no products of requirements analysis or precursor analysis that are produced during Cycle 3 — Source Selection. Thus, these steps would not be performed in Cycle 3.

In summary, this example illustrates the important principle of *tailoring* both the project cycle activities and the systems engineering activities during each phase of the tailored project cycle, to the specific needs of the program or project under development. If done properly, tailoring provides a means of optimizing the activities so that resources are not wasted generating unnecessary products or conducting unnecessary reviews. It can help transform the application of a rigorous systems engineering process in a well defined project cycle from an onerous burden to a welcome tool.

Appendix B.2 — A Sample SEMP Outline

An outline recommended by the Defense Systems Management College for the Systems Engineering Management Plan is shown below. This outline is a sample only, and should be tailored for the nature of the project and the risks inherent in the project.

Systems Engineering Management Plan

Title Page

Introduction

Part 1 — Technical Program Planning and Control
- 1.0 Responsibilities and Authority
- 1.1 Standards, Procedures, and Training
- 1.2 Program Risk Analysis
- 1.3 Work Breakdown Structures
- 1.4 Program Review
- 1.5 Technical Reviews
- 1.6 Technical Performance Measurements
- 1.7 Change Control Procedures
- 1.8 Engineering Program Integration
- 1.9 Interface Control
- 1.10 Milestones/Schedule
- 1.11 Other Plans and Controls

Part 2 — Systems Engineering Process
- 2.0 Mission and Requirements Analysis
- 2.1 Functional Analysis
- 2.2 Requirements Allocation
- 2.3 Trade Studies
- 2.4 Design Optimization/Effectiveness Compatibility
- 2.5 Synthesis
- 2.6 Technical Interface Compatibility
- 2.7 Logistic Support Analysis
- 2.8 Producibility Analysis
- 2.9 Specification Tree/Specifications
- 2.10 Documentation
- 2.11 Systems Engineering Tools

Part 3 — Engineering Specialty/Integration Requirements
- 3.1 Integration Design/Plans
  - 3.1.1 Reliability
  - 3.1.2 Maintainability
  - 3.1.3 Human Engineering
  - 3.1.4 Safety
  - 3.1.5 Standardization
  - 3.1.6 Survivability/Vulnerability
  - 3.1.7 Electromagnetic Compatibility/Interference
  - 3.1.8 Electromagnetic Pulse Hardening
  - 3.1.9 Integrated Logistics Support
  - 3.1.10 Computer Resources Lifecycle Management Plan
  - 3.1.11 Producibility
  - 3.1.12 Other Engineering Specialty Requirements/Plans
- 3.2 Integration System Test Plans
- 3.3 Compatibility with Supporting Activities
  - 3.3.1 System Cost-Effectiveness
  - 3.3.2 Value Engineering
  - 3.3.3 TQM/Quality Assurance
  - 3.3.4 Materials and Processes

Appendix B.3 — A "Tailored" WBS for an Airborne Telescope

Figure B-4 shows a partial Product Breakdown Structure (PBS) for the proposed Stratospheric Observatory for Infrared Astronomy (SOFIA), a 747SP aircraft outfitted with a 2.5 to 3.0 m telescope. The PBS has been elaborated for the airborne facility's telescope element. The PBS level names have been made consistent with the sidebar on page 3 of this handbook.

Figures B-5 through B-8 show a corresponding Work Breakdown Structures (WBSs) based on the principles in Section 4.3 of this handbook. At each level, the prime product deliverables from the PBS are WBS elements. The WBS is completed at each level by adding needed service (i.e., functional) elements such as management, systems engineering, integration and test, etc. The integration and test WBS element at each level refers to the activities of unifying prime product deliverables at that level.

Although the SOFIA project is used as an illustration in this appendix, the SOFIA WBS should be tailored to fit actual conditions at the start of Phase C/D as determined by the project manager. One example of a condition that could substantially change the WBS is foreign participation in the project.

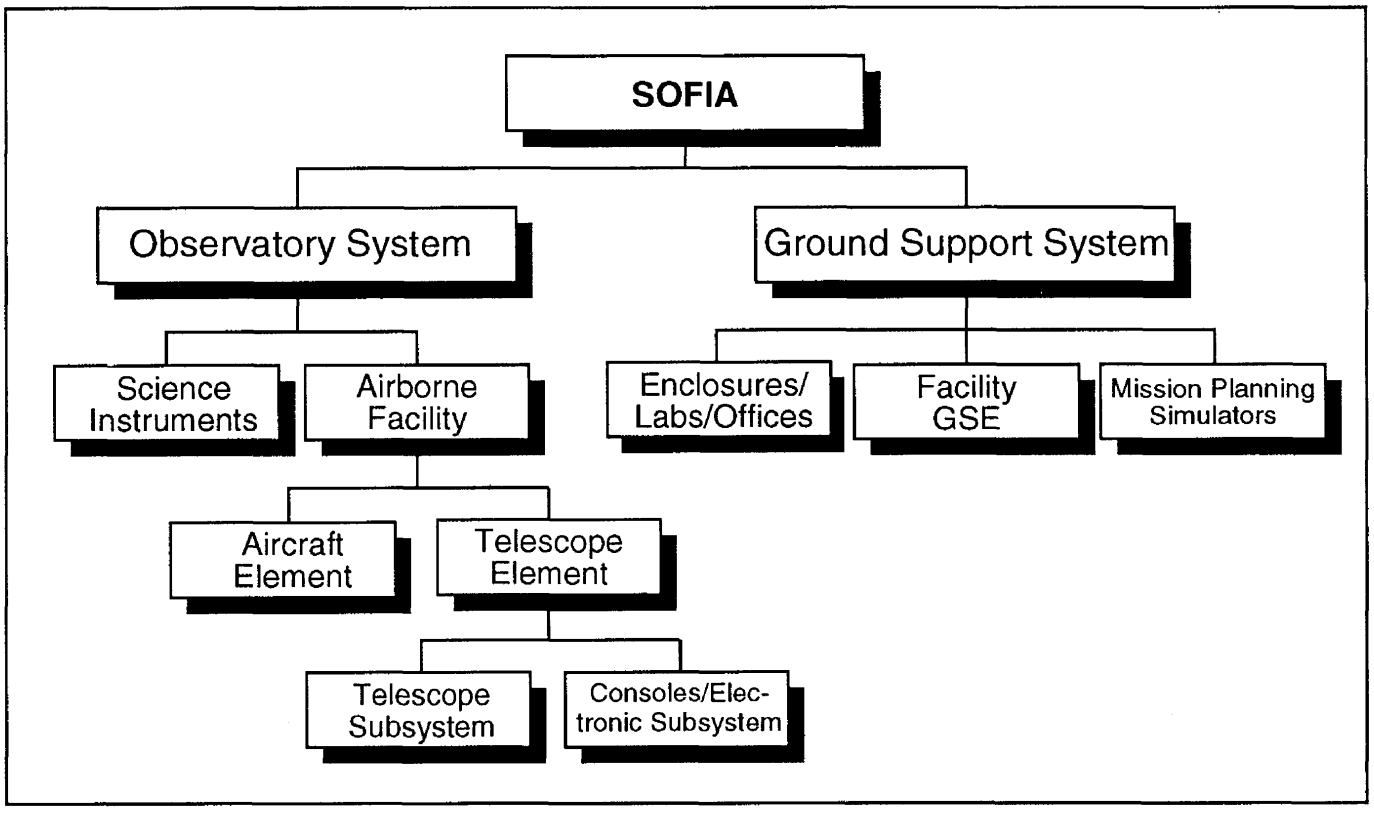


Figure B-4 — Stratospheric Observatory for Infrared Astronomy (SOFIA) Product Breakdown Structure.

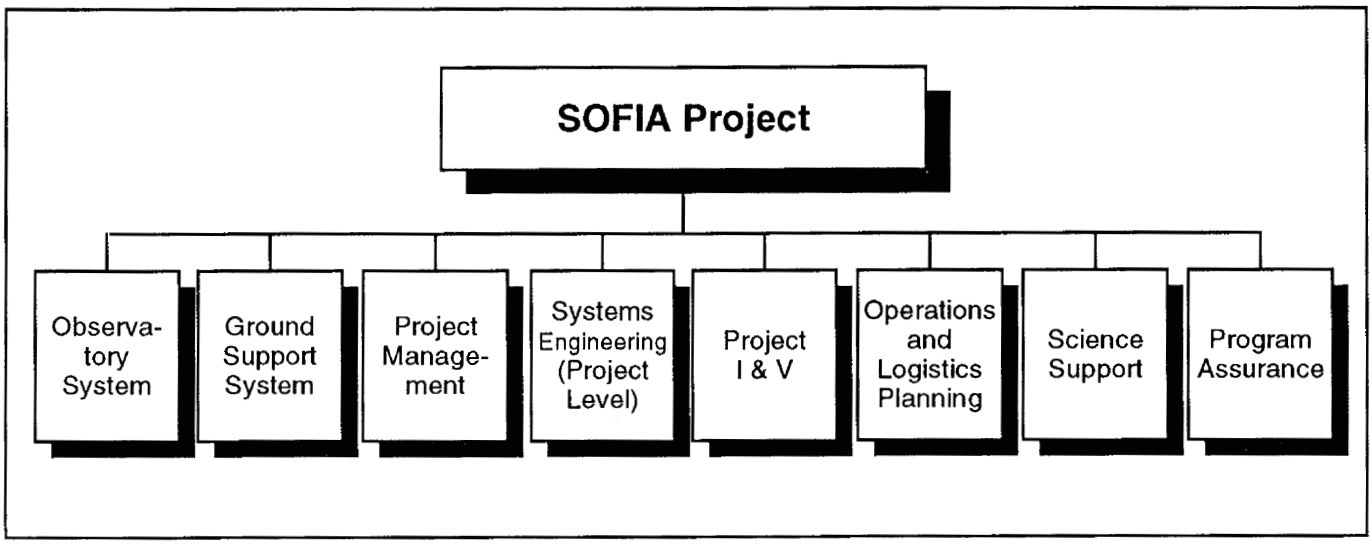


Figure B-5 — SOFIA Project WBS (*Level 3*).

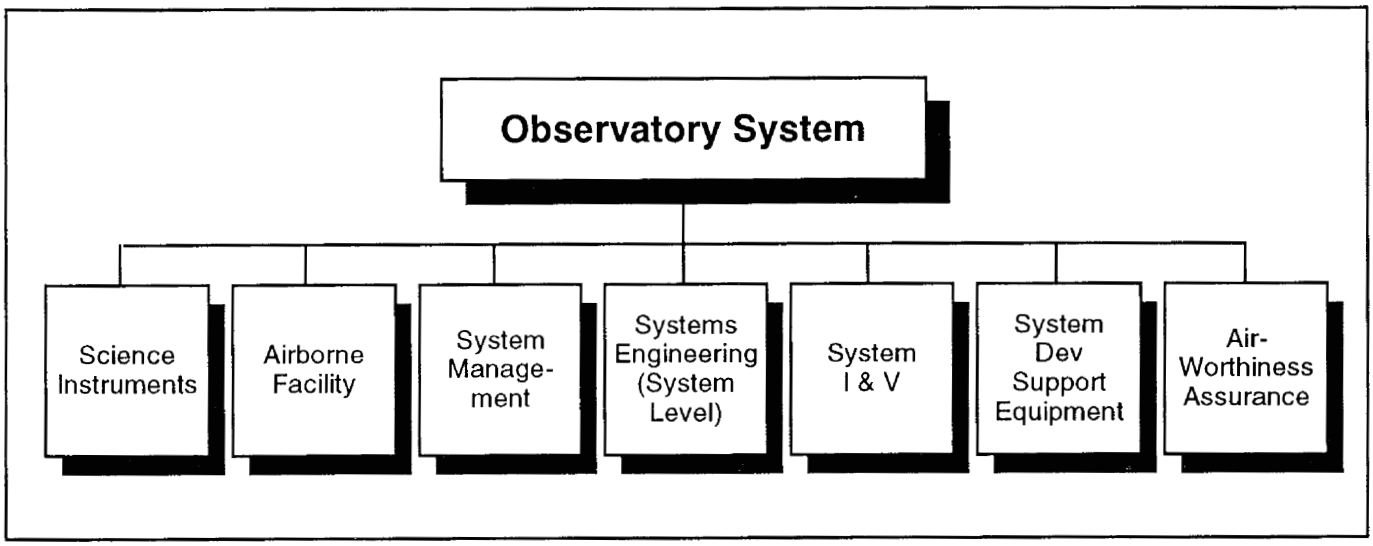


Figure B-6 — SOFIA Observatory System WBS (*Level 4*).

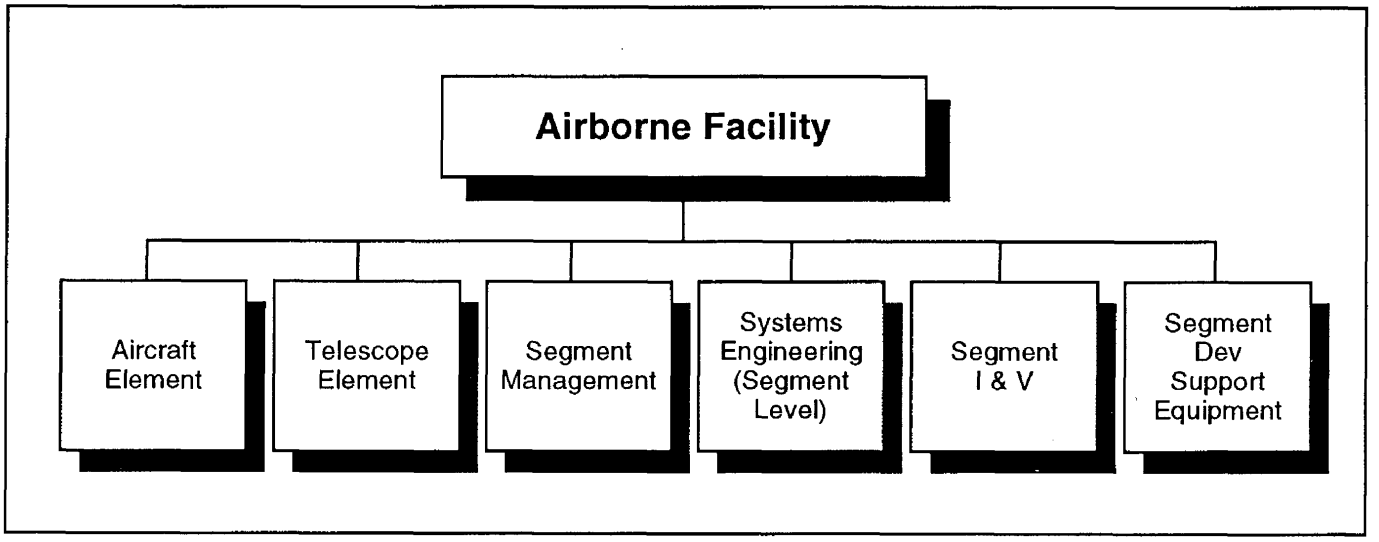


Figure B-7 — SOFIA Airborne Facility WBS (*Level 5*).

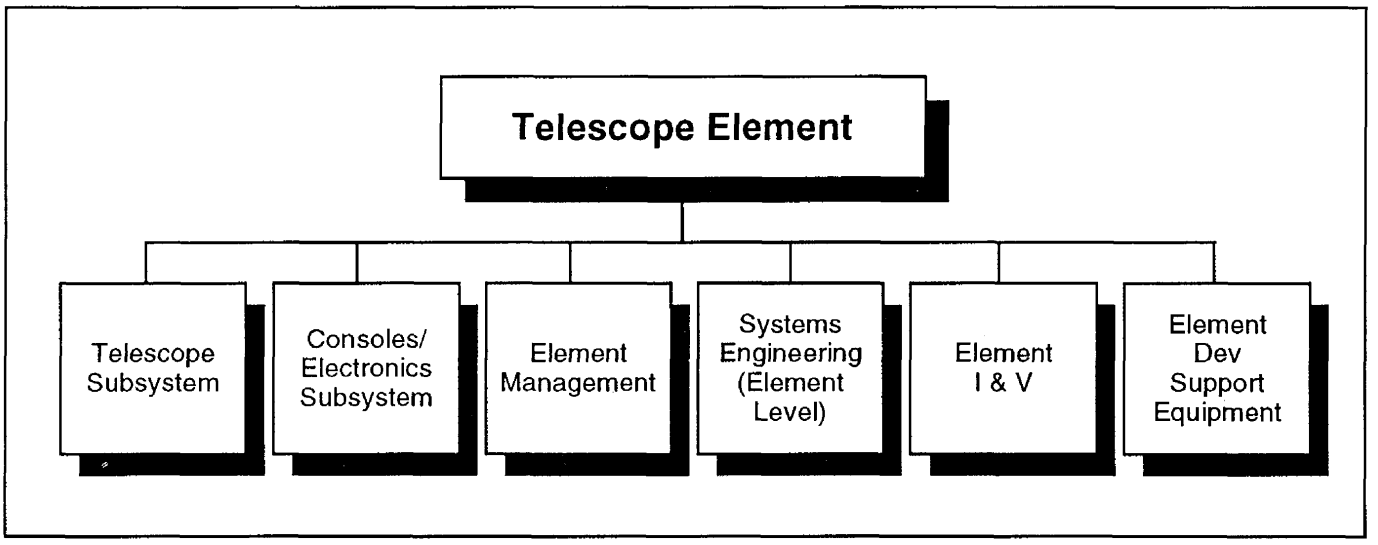


Figure B-8 — SOFIA Telescope Element WBS (*Level 6*).

Appendix B.4 — A Sample Configuration Management Plan Outline

1.0 Introductions
- 1.1 Description of the CIs
- 1.2 Program Phasing and Milestones
- 1.3 Special Features

2.0 Organization
- 2.1 Structure and Tools
- 2.2 Authority and Responsibility
- 2.3 Directives and Reference Documents

3.0 Configuration Identification
- 3.1 Baselines
- 3.2 Specifications

4.0 Configuration Control
- 4.1 Baseline Release
- 4.2 Procedures
- 4.3 CI Audits

5.0 Interface Management
- 5.1 Documentation
- 5.2 Interface Control

6.0 Configuration Traceability
- 6.1 Nomenclature and Numbering
- 6.2 Hardware Identification
- 6.3 Software and Firmware Identification

7.0 Configuration Status Accounting and Communications
- 7.1 Data Bank Description
- 7.2 Data Bank Content
- 7.3 Reporting

8.0 Configuration Management Audits

9.0 Subcontractor/Vendor Control

Appendix B.5 — Characterization, Mission Success and SRM&QA Cost Guidelines for Class A–D Payloads

Appendix B.5 is Attachment A of NMI 8010.1A, *Classification of NASA Payloads.*

	Class A	Class B	Class C	Class D
Characterization	High priority, minimum risk	High priority, medium risk	Medium priority, medium/high risk	High risk, minimum cost
Typical factors used to determine payload classifications	High national prestige; long hardware life required; high complexity; highest cost; long program duration; critical launch constraints; retrieval/ reflight or in-flight maintenance to recover from problems is not feasible.	High national prestige; medium hardware life required; high to medium complexity; high cost; medium program duration; some launch constraints; retrieval/reflight or in-flight maintenance to recover from problems is difficult or not feasible.	Moderate national prestige; short hardware life required; medium to low complexity; medium cost; short program duration; few launch constraints; retrieval/reflight or in-flight maintenance to recover from problems may be feasible.	Little national presitge; short hardware life required; low complexity; low cost; short program duration; non-critical launch time/orbit constraints; re-flyable or economically replaceable; in-flight maintenance may be feasible.
Achievement of mission success criteria	All affordable programmatic and other measures are taken to achieve minimum risk. The highest practical product assurace standards are utilized.	Compromises are used to permit somewhat reduced costs while maintaining a low risk to the overall mission success and a medium risk of achieving only partial success.	Moderate risks of not achieving mission success are accepted to permit significant cost savings. Reduced product assurance requirements are allowed.	Significant risk of not achieving mission success is accepted to permit minimum costs. Minimal product assurance requirements are allowed.
Estimated relative [1] SRM&QA cost factors	1.0	0.7 × Class A	0.4 × Class A	.1 × Class A

Note [1]: There are wide variations in the methods for specifying and accounting for "SRM&QA costs". For Class A programs, these costs are typically in the range of 10–15% of the total program cost. The relative SRM&QA cost factors specified here are intended to require substantive differences in the SRM&QA programs (and the associated costs) for the various program classifications in order to establish a meaningful ladder of cost/risk levels.

Appendix B.6 — An Example of a Critical Items List

```
                    SHUTTLE CRITICAL ITEMS LIST - ORBITER

 SUBSYSTEM :LANDING DECELERATION      FMEA NO 02-1  -001   -1    REV:02/09/82
.ASSEMBLY  :MAIN LANDING GEAR         ABORT:               CRIT. FUNC:   1
.P/N RI    :MC621-0011                                     CRIT.  HDW:   1
.P/N VENDOR:1170100 MENASCO              VEHICLE     102   099   103   104
.QUANTITY  :2                         EFFECTIVITY:    X     X     X     X
.          :LEFT HAND                 PHASE(S)    PL  LO    CO    DO X LS
.          :RIGHT HAND
.                                  REDUNDANCY SCREEN:  A-N/A   B-N/A   C-N/A
.
.PREPARED BY:                 APPROVED BY:                APPROVED BY (NASA):
.DES          L L RHODES      DES _________________       SSM _________________
.REL          A L DOBNER      REL _________________       REL _________________
.
.
.ITEM: MLG STRUT
.  MLG SHOCK STRUT INNER AND OUTER CYLINDER AND LOAD CARRYING MEMBERS.
.FUNCTION:
.  MLG LOAD CARRYING MEMBERS CYLINDER - DAMPER, WHERE A PASSAGE OF
   HYDRAULIC FLUID THROUGH AN ORFICE ABSORBS THE ENERGY OF IMPACT AND
   WHERE DRY NITROGEN IS USED AS THE ELASTIC MEDIUM TO RESTORE THE
   UNSPRUNG PARTS TO THEIR EXTENDED POSITION.
.FAILURE MODE:  STRUCTURAL FAILURE
.
.CAUSE(S):
.  STRESS CORROSION.  PIECE-PART STRUCTURAL FAILURE.  OVERLOAD.
.EFFECT(S) ON (A)SUBSYSTEM (B)INTERFACES (C)MISSION (D)CREW/VEHICLE:
.  (A) LOSS OF SUBSYSTEM FUNCTION.   (B) NONE.   (C) NONE.   (D) PROBABLE
   LOSS OF VEHICLE IF MAIN STRUT FAILS ON LANDING.
.DISPOSITION & RATIONALE (A)DESIGN (B)TEST (C)INSPECTION (D)FAILURE HISTORY:
.  (A) UNDER WORST CASE LOADING (FLAT STRUT) THE STRUT IS CAPABLE OF
   WITHSTANDING ONE LANDING AT THE NORMAL LANDING DESIGN GROSS WEIGHT OF
   207,000 LBS. AND SINK SPEED OF 9.6 FEET PER SECOND WITH CORRESPONDING
   LANDING ROLLOUT AND BRAKING CONDITIONS, WITH NO YIELDING OF THE
   STRUCTURAL MEMBERS.  (B) ACCEPTANCE INCLUDES VERIFICATION THAT
   CERTIFIED  MATERIALS AND PROCESSES WERE USED.  CERTIFICATION INCLUDES A
   FATIGUE LOAD TEST SPECTRUM (REF MC62-0011 TABLES 10-11) REPRESENTING THE
   EQUIVALENT LOADING FOR THE LIFE OF EACH LANDING GEAR WITH A SCATTER
   FACTOR OF 4.0.  THE STATIC LOAD TESTS INCLUDED A TAXI BUMP (65K
   PAYLOAD), VEHICLE WEIGHT 227 KIPS/AND A RIGHT TURN/WHICH IS THE WORST
   CASE CONDITIONS WITHOUT FAILURE.  (C) DURING TURNAROUND-VISUALLY INSPECT
   FOR DAMAGE.  USE NDE TO SPPORT SUSPECT AREAS.  AT MANUFACTURER-RAW
   MATERIAL VERIFIED-VISUALL INSP./ID PERFORMED-PARTS PROTECTION, COATING
   AND PLATING PROCESSES VERIF. BY INSPECTION.-MANUF., INSTL. AND ASSY.
   OPERATIONS VERIF. BY SHOP TRAVELER MIPS-CORROSION PROTECTION PROVISIONS
   VERIF. NDE OF SURFACE AND SUB-SURFACE DEFECTS VERIF. BY INSPECTION.
   PROPERLY MONITORED HANDLING AND STORAGE ENVIRONMENT VERIFIED. MATL. AND
   EQUIPMENT CONFORMANCE TO CONTRACT REQMTS. VERIFIED BY INSP.-FINDINGS
   VERIFIED BY AUDIT 9-25-78.  (D) DURING DROP TEST PROGRAM, THE OUTER
   GLAND NUT FAILED.  MENASCO REDESIGNED AND CHANGED FROM ALUMINUM TO STEEL
   MATL.  THE SNUBBER RING P/N 1170134-1 WAS REDESIGNED.  UPPER BEARING
   1170107-1 WAS REPLACED BY A SOLID ALUMINUM-BRONZE BEARING.
```

Appendix B.7 — Techniques of Functional Analysis

Appendix B.7 is reproduced from the *Defense Systems Management Guide*, published January 1990 by the Defense Systems Management College, Ft. Belvoir, VA.

• • •

System requirements are analyzed to identify those functions which must be performed to satisfy the objectives of each functional area. Each function is identified and described in terms of inputs, outputs, and interface requirements from top down so that subfunctions are recognized as part of larger functional areas. Functions are arranged in a logical sequence so that any specified operational usage of the system can be traced in an end-to-end path. Although there are many tools available, functional identification is accomplished primarily through the use of 1) functional flow block diagrams (FFBDs) to depict task sequences and relationships, 2) N^2 diagrams to develop data interfaces, and 3) time line analyses to depict the time sequence of time-critical functions.

B.7.1 Functional Flow Block Diagrams

The purpose of the FFBD is to indicate the sequential relationship of all functions that must be accomplished by a system. FFBDs depict the time sequence of functional events. That is, each function (represented by a block) occurs following the preceding function. Some functions may be performed in parallel, or alternate paths may be taken. The duration of the function and the time between functions is not shown, and may vary from a fraction of a second to many weeks. The FFBDs are function oriented, not equipment oriented. In other words, they identify "what" must happen and do not assume a particular answer to "how" a function will be performed.

FFBDs are developed in a series of levels. FFBDs show the same tasks identified through functional decomposition and display them in their logical, sequential relationship. For example, the entire flight mission of a spacecraft can be defined in a top level FFBD, as shown in Figure B-9. Each block in the first level diagram can then be expanded to a series of functions, as shown in the second level diagram for "perform mission operations". Note that the diagram shows both input (transfer to operational orbit) and output (transfer to space transportation system orbit), thus initiating the interface identification and control process. Each block in the second level diagram can be progressively developed into a series of functions, as shown in the third level diagram on Figure B-9. These diagrams are used both to develop requirements and to identify profitable trade studies. For example, does the spacecraft antenna acquire the tracking and data relay satellite (TDRS) only when the payload data are to be transmitted, or does it track TDRS continually to allow for the reception of emergency commands or transmission of emergency data? The FFBD also incorporates alternate and contingency operations, which improve the probability of mission success. The flow diagram provides an understanding of total operation of the system, serves as a basis for development of operational and contingency procedures, and pinpoints areas where changes in operational procedures could simplify the overall system operation. In certain cases, alternate FFBDs may be used to represent various means of satisfying a particular function until data are acquired, which permits selection among the alternatives.

B.7.2 N^2 Diagrams

The N^2 diagram has been used extensively to develop data interfaces, primarily in the software areas. However, it can also be used to develop hardware interfaces. The basic N^2 chart is shown in Figure B-10. The system functions are placed on the diagonal; the remainder of the squares in the N x N matrix represent the interface inputs and outputs. Where a blank appears, there is no interface between the respective functions. Data flows in a clockwise direction between functions (e.g., the symbol F_1 F_2 indicates data flowing from function F_1 to function F_2). The data being transmitted can be defined in the appropriate squares. Alternatively, the use of circles and numbers permits a separate listing of the data interfaces as shown in Figure B-11. The clockwise flow of data between functions that have a feedback loop can be illustrated by a larger circle called a control loop. The identification of a critical function is also shown in Figure B-11, where function F_4 has a number of inputs and outputs to all other functions in the upper module. A simple flow of interface data exists between the upper and lower modules at functions F_7 and F_8. The lower module has complex interaction among its functions. The N^2 chart can be taken down into successively lower levels to the hardware and software component functional levels. In addition to defining the data that must be supplied across the interface, the N^2 chart can pinpoint areas where conflicts could arise.

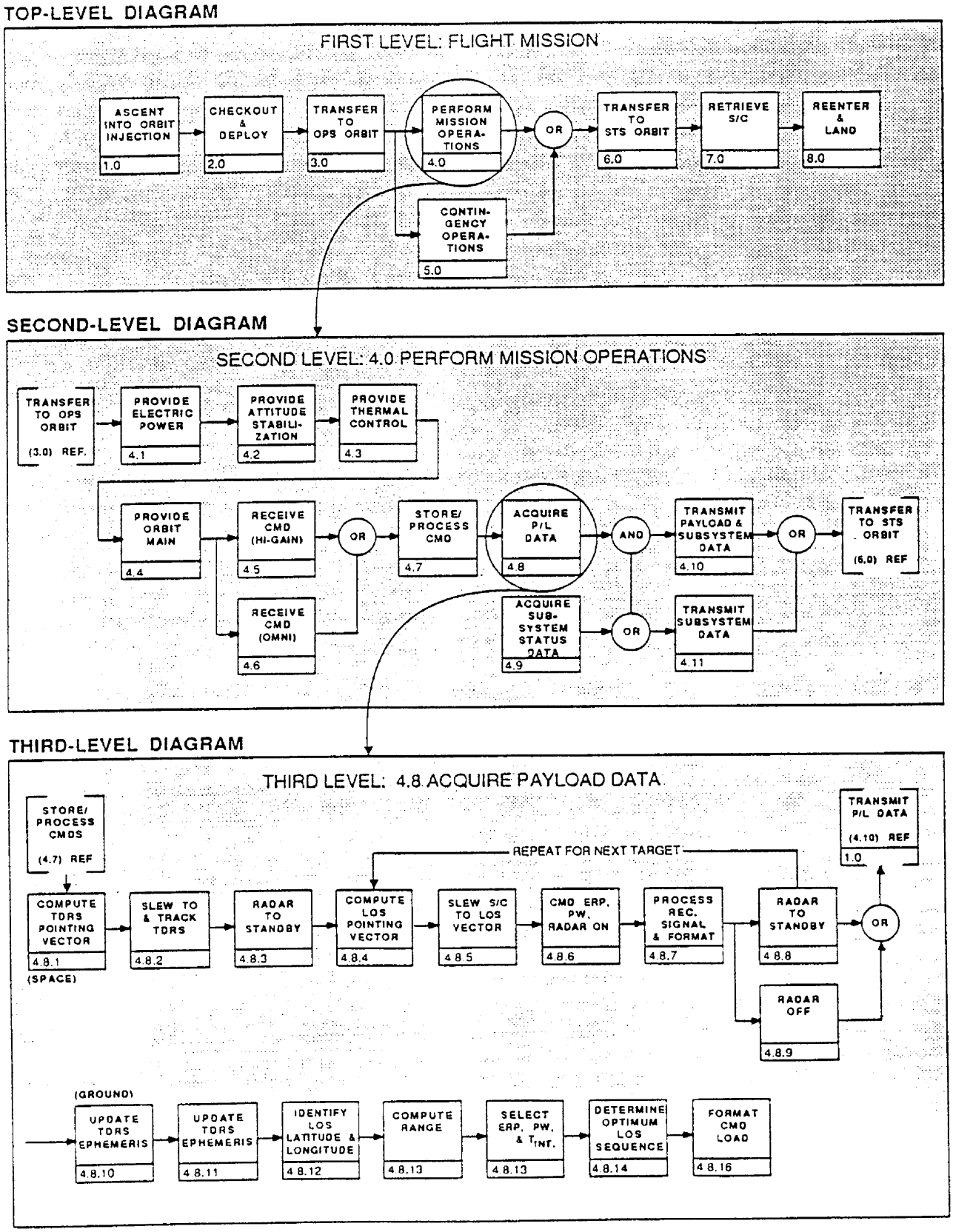


Figure B-9 — Development of Functional Flow Block Diagrams.

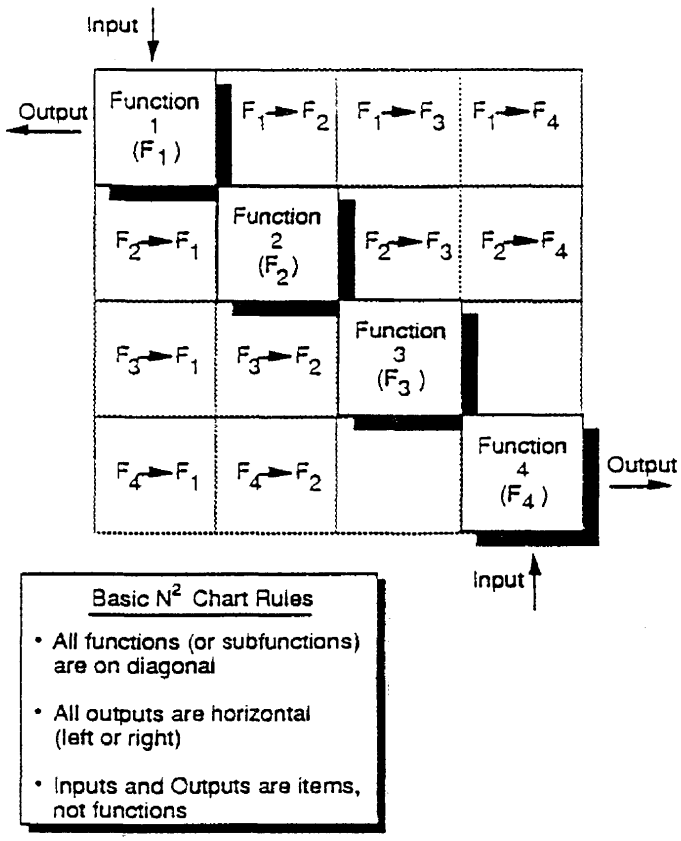


Figure B-10 — N^2 Chart Definition.

B.7.3 Time Line Analysis

Time line analysis adds consideration of functional durations and is used to support the development of design requirements for operation, test and maintenance functions. The time line sheet (TLS) is used to perform and record the analysis of time critical functions and functional sequences. Additional tools such as mathematical models and computer simulations may be necessary. Time line analysis is performed on those areas where time is critical to the mission success, safety, utilization of resources, minimization of down time, and/or increasing availability. Not all functional sequences require time line analysis, only those in which time is a critical factor. The following areas are often categorized as time critical: 1) functions affecting system reaction time, 2) mission turnaround time, 3) time countdown activities, and 4) functions requiring time line analysis to determine optimum equipment and/or personnel utilization. An example of a high level TLS for a space program is shown in Figure B-12.

For time critical function sequences, the time requirements are specified with associated tolerances. Time line analyses play an important role in the trade-off process between man and machine. The decisions between automatic and manual methods will be made and will determine what times are allocated to what subfunctions. In addition to defining subsystem/component time requirements, time line analysis can be used to develop trade studies in areas other than time consideration (e.g., should the spacecraft location be determined by the ground network or by onboard computation using navigation satellite inputs? Figure B-6 is an example of a maintenance TLS which illustrates that availability of an item (a distiller) is dependent upon the completion of numerous maintenance tasks accomplished concurrently. Furthermore, it illustrates the traceability to higher level requirements by referencing the appropriate FFBD and requirement allocation sheet (RAS).

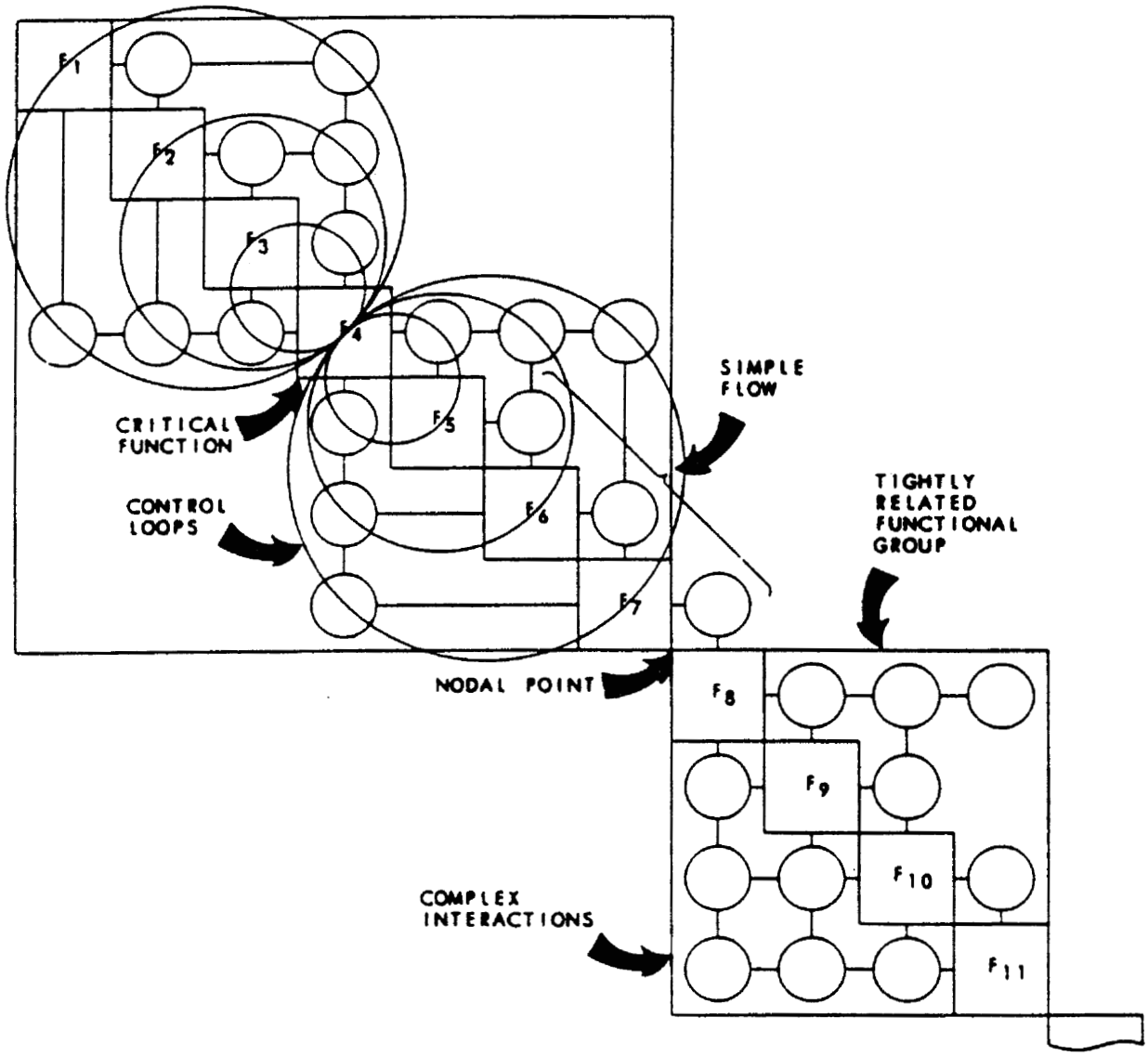


Figure B-11 — N^2 Chart Key Features (from "The N^2 Chart", R. Lano, © 1977 TRW Inc.)

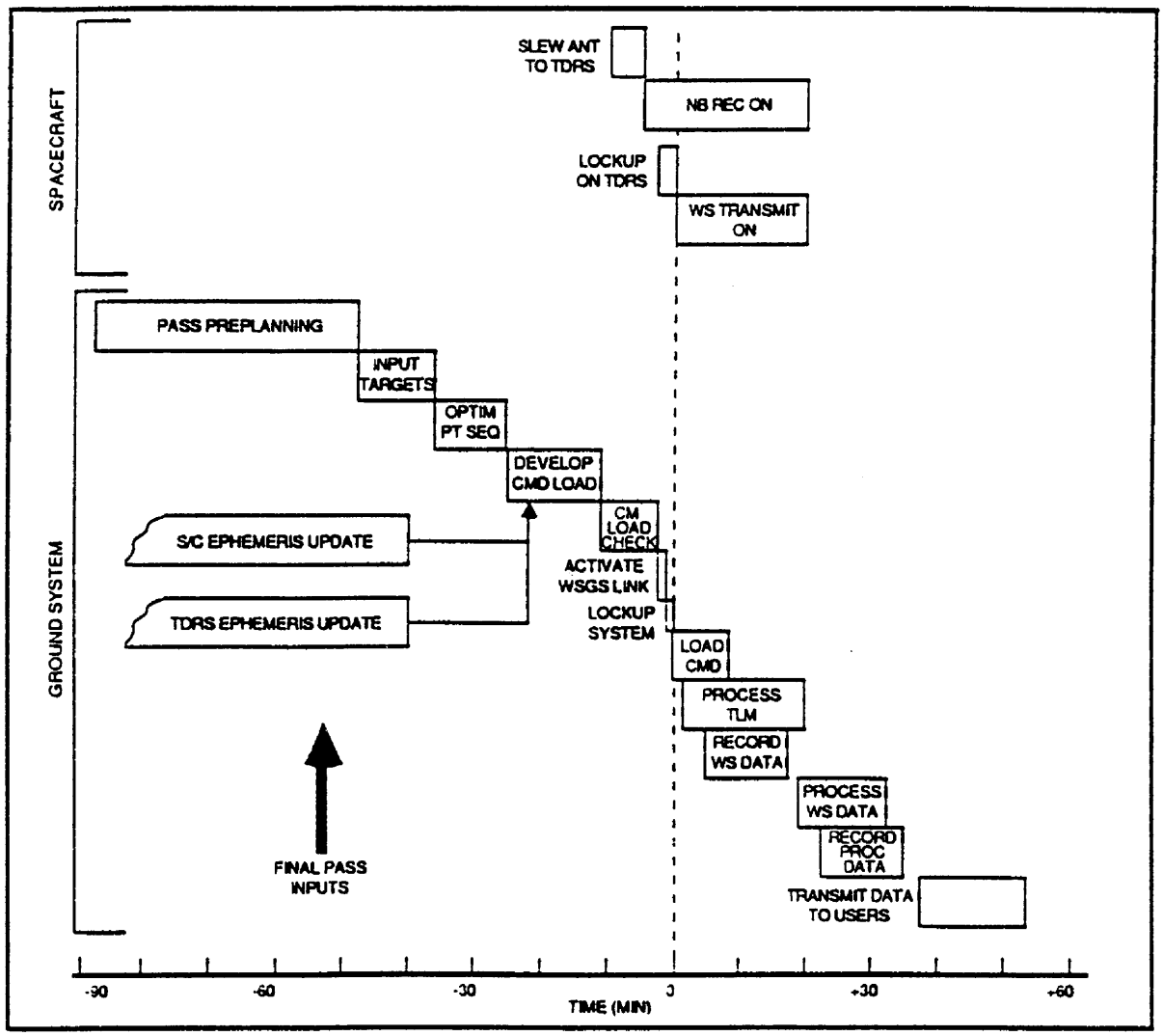


Figure B-12 — Flight Mission Time Lines.

Appendix B.8 — The Effect of Changes in ORU MTBF on Space Station *Freedom* Operations

The reliability of Space Station *Freedom*'s (SSF) Orbital Replacement Units (ORUs) has a profound effect its operations costs. This reliability is measured by the Mean Time Between Failure (MTBF). One study of the effects, by Dr. William F. Fisher and Charles Price, was *SSF External Maintenance Task Team Final Report* (JSC, July 1990). Another, by Anne Accola, et al., shows these effects parametrically. Appendix B.8 excerpts this paper, *Sensitivity Study of SSF Operations Costs and Selected User Resources* (presented at the International Academy of Astronautics Symposium on Space Systems Costs Methodologies and Applications, May 1990).

• • •

There are many potential tradeoffs that can be performed during the design stage of SSF. Many of them have major implications for crew safety, operations cost, and achievement of mission goals. Operations costs and important non-cost operations parameters are examined. One example of a specific area of concern in design is the reliability of the ORUs that comprise SSF. The implications of ORU reliability on logistics upmass and downmass to and from SSF are great, thus affecting the resources available for utilization and for other operations activities. In addition, the implications of reliability on crew time available for mission accomplishment (i.e., experiments) vs. station maintenance are important.

The MTBF effect on operations cost is shown in Figure B-13. Repair and spares costs are influenced greatly by varying MTBF. Repair costs are inversely proportional to MTBF, as are replacement spares. The initial spares costs are also influenced by variables other than MTBF. The combined spares cost, consisting of initial and replacement spares are not as greatly affected as are repair costs. The five-year operations cost is increased by only ten percent if all ORU MTBF are halved. The total operations cost is reduced by three percent if all ORU MTBF are doubled. It would almost appear that MTBF is not as important as one would think. However, MTBF also affects available crew time and available upmass much more than operations cost as shown in Figures B-14 and B-15.

Available crew time is a valuable commodity because it is a limited resource. Doubling the number of ORU replacements (by decreasing the MTBF) increases the maintenance crew time by 50 percent, thus reducing the amount of time available to perform useful experiments or scientific work by 22 percent. By halving the ORU replacements, the maintenance crew time decreases by 20 percent and the available crew time increases by eight percent.

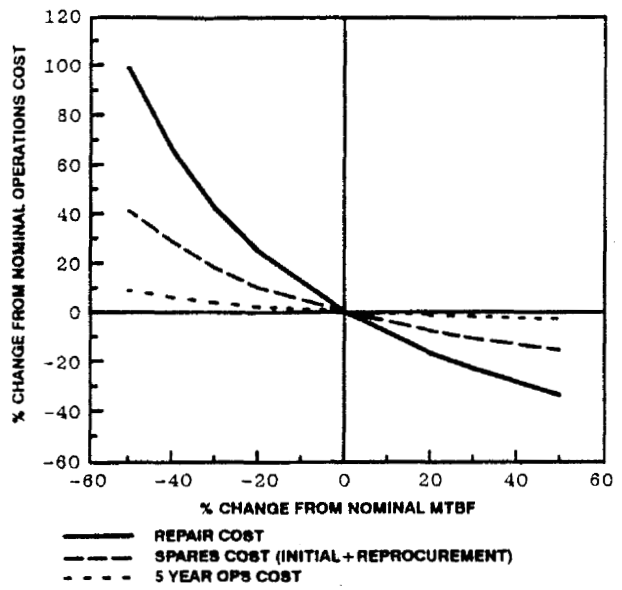


Figure B-13 — Effect of MTBF on Operations Cost.

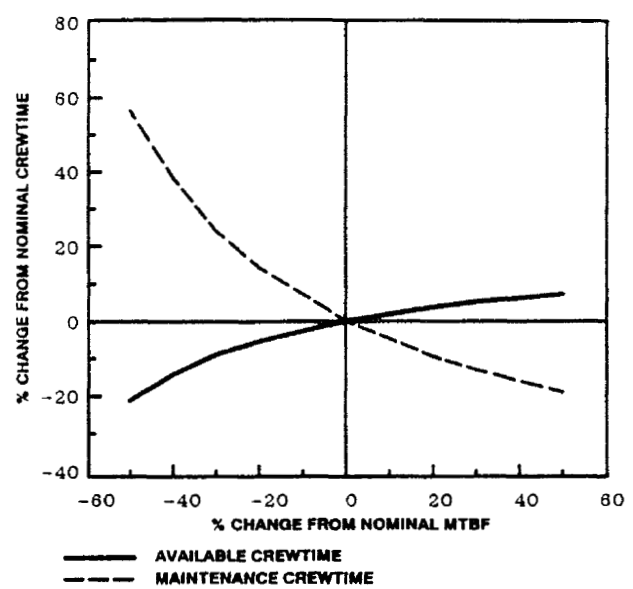


Figure B-14 — Effect of MTBF on Crew Time.

Available upmass is another valuable resource because a fixed number of Space Shuttle flights can transport only a fixed amount of payload to the SSF. Extra ORUs taken to orbit reduces available upmass that could be used to take up experimental payloads. Essentially, by doubling the number of ORU replacements, the available upmass is

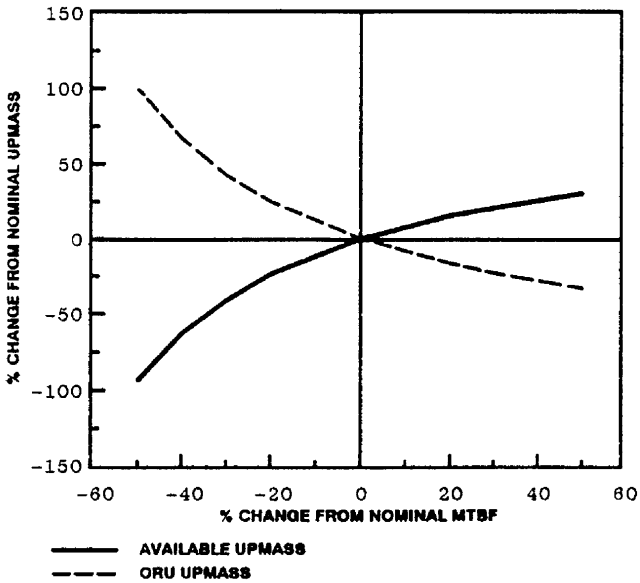


Figure B-15 — Effect of MTBF on Upmass.

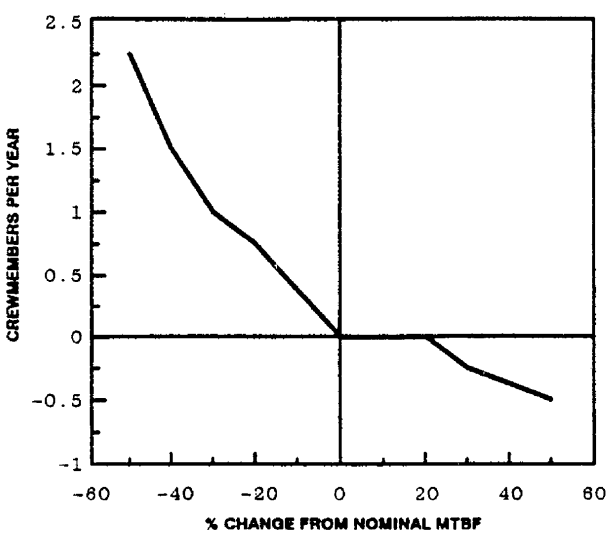


Figure B-16 — Effect of MTBF on Number of Crew (Available Crew Time Maintained).

driven to zero. Conversely, halving the number of ORU replacements increases the available upmass by 30 percent.

Although the effects of MTBF on resources is interesting, it is a good idea to quantify the effectiveness of the scenarios based on total cost to maintain the nominal re-

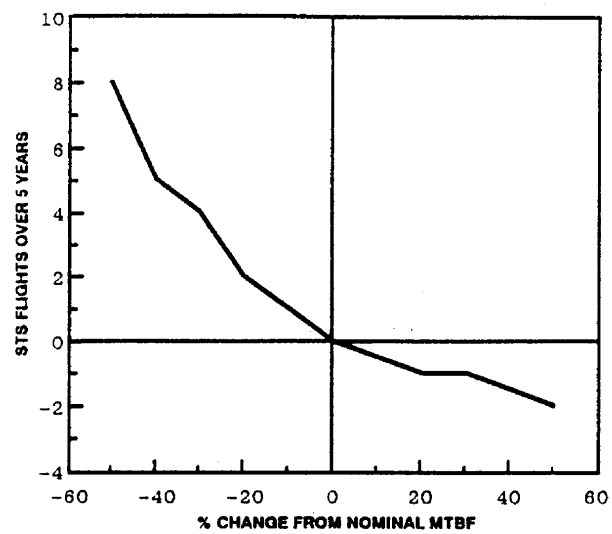


Figure B-17 — Effect of MTBF on Number of STS Flights (Available Upmass Maintained).

sources. Figure B-16 shows the number of crew members needed each year to maintain the available crew time. The figure shows that to maintain the nominal available crew time after doubling the number of ORU replacements, the Station would need two extra crew members. It should be noted that no attempt was made to assess the design capability or design cost impacts to accommodate these extra crew members. The savings of crew due to halving the number of ORU replacements is small, effectively one less crew member for half the year.

Figure B-17 shows the number of Space Shuttle flights over five years needed to maintain the nominal available upmass. The Space Shuttle flights were rounded upward to obtain whole flights. Doubling the number of ORU replacements would mean eight extra Space Shuttle flights would be needed over five years. Halving the ORU replacements would require two fewer Space Shuttle flights over five years. No attempt was made to assess the Space Shuttle capability to provide the extra flights or the design cost impacts to create the ORUs with the different reliabilities.

Figure B-18 shows the effect of assessing the cost impact of the previous two figures and combining them with the five-year operations cost. The influence of MTBF is effectively doubled when the resources of available upmass and crew time are maintained at their nominal values.

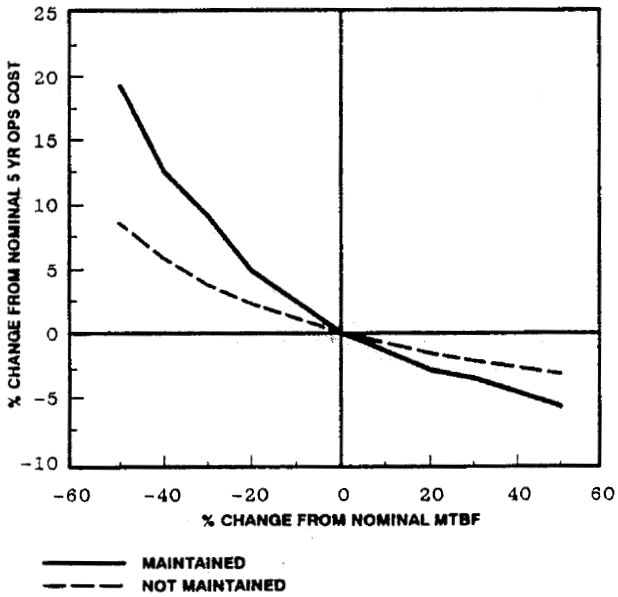


Figure B-18 — Effect of MTBF on Five-year Operations Cost (Maintaining vs. Not Maintaining Available Upmass and Crew Time).

Appendix C — Use of the Metric System

C.1 NASA Policy

It is NASA policy (see NMI 8010.2A) to:

- Adopt the International System of Units, known by the international abbreviation *SI* and defined by ANSI/IEEE Standard 268-1982, as the preferred system of weights and measurements.
- Use the metric system for all major flight program new starts (unless a waiver is granted).
- Use the metric system in procurements, grants and business-related activities to the extent economically feasible.
- Establish a plan for transition of all NASA activities to the use of the metric system, except to the extent that such use is impractical or will cause significant inefficiencies or loss of markets to U.S. firms.
- Permit continued use of the inch-pound system of measurement for existing systems.

C.2 Definitions of Units

The content of this section is reproduced from IEEE/ANSI 268-1982, *IEEE Standard for Metric Practice*, copyright © 1982 by the Institute of Electrical and Electronics Engineers, Inc., with the permission of the IEEE.

• • •

Outside the United States, the comma is widely used as a decimal marker. In some applications, therefore, the common practice in the United States of using the comma to separate digits into groups of three (as in 23,478) may cause ambiguity. To avoid this potential source of confusion, recommended international practice calls for separating the digits into groups of three, counting from the decimal point toward the left and the right, and using a small space to separate the groups. In numbers of four digits on either side of the decimal point the space is usually not necessary, except for uniformity in tables. To conform with the international practice, this section uses spaces — rather than commas — in number groups.

C.2.1 SI Prefixes

The names of multiples and submultiples of SI units may be formed by application of the prefixes and symbols shown in the sidebar. (The unit of mass, the *kilogram*, is the only exception; for historical reasons, the *gram* is used as the base for construction of names.)

Prefixes for SI Units

Factor	Prefix	Sym.	Pronunciation**
10^{18}	exa	E	EXa (*a* as in *a*bout)
10^{15}	peta	P	PETa (*e* as in pet, *a* as in *a*bout)
10^{12}	tera	T	as in *TERR*ace
10^{9}	giga	G	JIGa (*i* as in *j*ig, *a* as in *a*bout)
10^{6}	mega	M	as in *MEG*aphone
10^{3}	kilo	k	KILLoh
10^{2}	hecto*	h	HECKtoe
10	deka*	da	DECKa (*a* as in *a*bout)
1			
10^{-1}	deci*	d	as in *DEC*imal
10^{-2}	centi*	c	as in *CENT*ipede
10^{-3}	milli	m	as in *MIL*itary
10^{-6}	micro	μ	as in *MIC*rophone
10^{-9}	nano	n	NANoh (*AN* as in *AN*t)
10^{-12}	pico	p	PEEKoh
10^{-15}	femto	f	FEMtoe (*FEM* as in *FEM*inine)
10^{-18}	atto	a	as in an*AT*omy

* The prefixes that do not represent 1000 raised to a power (that is, *hecto*, *deka*, *deci*, and *centi*) should be avoided where practical.

** The first syllable of every prefix is accented to assure that the prefix will retain its identity. (*Kilometer* is *not* an exception.)

C.2.2 Base SI Units

ampere (A) The *ampere* is that constant current which, if maintained in two straight parallel conductors of infinite length, of negligible circular cross section, and placed one meter apart in vacuum, would produce between these conductors a force equal to 2×10^{-7} newton per meter of length.

candela (cd) The *candela* is the luminous intensity, in a given direction, of a source that emits monochromatic radiation of frequency 540×10^{12} Hz and that has a radiant intensity in that direction of 1/683 watt per steradian.

kelvin (K) The *kelvin*, unit of thermodynamic temperature, is the fraction 1/273.16 of the thermodynamic temperature of the triple point of water.

kilogram (kg) The *kilogram* is the unit of mass; it is equal to the mass of the international prototype of the kilo-

gram. (The international prototype of the kilogram is a particular cylinder of platinum-iridium alloy which is preserved in a vault at Sèvres, France, by the International Bureau of Weights and Measures.)

meter (m) The *meter* is the length equal to 1 650 763.73 wavelengths in vacuum of the radiation corresponding to the transition between the levels $2p_{10}$ and $5d_5$ of the krypton-86 atom.

mole (mol) The *mole* is the amount of substance of a system which contains as many elementary entities as there are atoms in 0.012 kilogram of carbon-12. **Note:** When the mole is used, the elementary entities must be specified and may be atoms, molecules, ions, electrons, other particles, or specified groups of such particles.

second (s) The *second* is the duration of 9 192 631 770 periods of the radiation corresponding to the transition between the two hyperfine levels of the ground state of the cesium-133 atom.

C.2.3 Supplementary SI Units

radian (rad) The *radian* is the plane angle between two radii of a circle which cut off on the circumference an arc equal in length to the radius.

steradian (sr) The *steradian* is the solid angle which, having its vertex in the center of a sphere, cuts off an area of the surface of the sphere equal to that of a square with sides of length equal to the radius of the sphere.

C.2.4 Derived SI Units with Special Names

In addition to the units defined in this subsection, many quantities are measured in terms of derived units which do not have special names — such as *velocity* in **m/s**, *electric field strength* in **V/m**, *entropy* in **J/K**, and so on.

becquerel (Bq = 1/s) The *becquerel* is the activity of a radionuclide decaying at the rate of one spontaneous nuclear transition per second.

degree Celsius (°C = K) The *degree Celsius* is equal to the kelvin and is used in place of the kelvin for expressing Celsius temperature defined by the equation $t = T - T_0$, where t is the Celsius temperature, T is the thermodynamic temperature, and $T_0 = 273.15$ K (by definition).

coulomb (C = A·s) *Electric charge* is the time integral of electric current; its unit, the *coulomb*, is equal to one ampere second.

farad (F = C/V) The *farad* is the capacitance of a capacitor between the plates of which there appears a difference of potential of one volt when it is charged by a quantity of electricity equal to one coulomb.

gray (Gy = J/kg) The *gray* is the absorbed dose when the energy per unit mass imparted to matter by ionizing radiation is one joule per kilogram. (The *gray* is also used for the ionizing radiation quantities: specific energy imparted, kerma, and absorbed dose index.)

henry (H = Wb/A) The *henry* is the inductance of a closed circuit in which an electromotive force of one volt is produced when the electric current in the circuit varies uniformly at a rate of one ampere per second.

hertz (Hz = 1/s) The *hertz* is the frequency of a periodic phenomenon of which the period is one second.

joule (J = N·m) The *joule* is the work done when the point of application of a force of one newton is displaced a distance of one meter in the direction of the force.

lumen (lm = cd·sr) The *lumen* is the luminous flux emitted in a solid angle of one steradian by a point source having a uniform intensity of one candela.

lux ($lx = lm/m^2$) The *lux* is the illuminance produced by a luminous flux of one lumen uniformly distributed over a surface of one square meter.

newton ($N = kg{\cdot}m/s^2$) The *newton* is that force which, when applied to a body having a mass of one kilogram, gives it an acceleration of one meter per second squared.

ohm (Ω = V/A) The *ohm* is the electric resistance between two points of a conductor when a constant difference of potential of one volt, applied between these two points, produces in this conductor a current of one ampere, this conductor not being the source of any electromotive force.

pascal ($Pa = N/m^2$) The *pascal* [which, in the preferred pronunciation, rhymes with *rascal*] is the pressure or stress of one newton per square meter.

siemens (S = A/V) The *siemens* is the electric conductance of a conductor in which a current of one ampere is produced by an electric potential difference of one volt.

sievert (Sv = J/kg) The *sievert* is the dose equivalent when the absorbed dose of ionizing radiation multiplied by the dimensionless factors Q (quality factor) and N (product of any other multiplying factors) stipulated by the International Commission on Radiological Protection is one joule per kilogram.

tesla ($T = Wb/m^2$) The *tesla* is the magnetic flux density of one weber per square meter. In an alternative approach to defining the magnetic field quantities the *tesla* may also be defined as the magnetic flux density that produces on a one-meter length of wire carrying a current of one ampere, oriented normal to the flux density, a force of one newton, magnetic flux density being defined as an axial vector quantity such that the force exerted on an element of current is equal to the vector product of this element and the magnetic flux density.

volt (V = W/A) The *volt* (unit of electric potential difference and electromotive force) is the difference of electric potential between two points of a conductor carrying a constant current of one ampere, when the power dissipated between these points is equal to one watt.

watt (W = J/s) The *watt* is the power that represents a rate of energy transfer of one joule per second.

weber (Wb = V·s) The *weber* is the magnetic flux which, linking a circuit of one turn, produces in it an electromotive force of one volt as it is reduced to zero at a uniform rate in one second.

C.2.5 Units in Use with SI

Time The SI unit of time is the second. This unit is preferred and should be used if practical, particularly when technical calculations are involved. In cases where time relates to life customs or calendar cycles, the minute, hour, day and other calendar units may be necessary. For example, vehicle speed will normally be expressed in kilometers per hour.

minute (min) 1 min = 60 s
hour (h) 1 h = 60 min = 3600 sec
day (d) 1 d = 24 h = 86 400 sec
week, month, etc.

Plane angle The SI unit for plane angle is the radian. Use of the degree and its decimal submultiples is permissible when the radian is not a convenient unit. Use of the minute and second is discouraged except for special fields such as cartography.

degree (°) $1° = (\pi/180)$ rad
minute (′) $1' = (1/60)° = (\pi/10\,800)$ rad
second (″) $1'' = (1/60)' = (\pi/648\,000)$ rad

Area The SI unit of area is the square meter (m^2). The hectare (ha) is a special name for the square hectometer (hm^2). Large land or water areas are generally expressed in hectares or in square kilometers (km^2).

Volume The SI unit of volume is the cubic meter. This unit, or one of the regularly formed multiples such as the cubic centimeter, is preferred. The special name *liter* has been approved for the cubic decimeter, but use of this unit is restricted to volumetric capacity, dry measure, and measure of fluids (both liquids and gases). No prefix other than *milli-* or *micro-* should be used with *liter*.

Mass The SI unit of mass is the kilogram. This unit, or one of the multiples formed by attaching an SI prefix to *gram* (g), is preferred for all applications. The megagram (Mg) is the appropriate unit for measuring large masses such as have been expressed in tons. However, the name *ton* has been given to several large mass units that are widely used. The term *metric ton* should be restricted to commercial usage, and no prefixes should be used with it.

metric ton $1\ t = 10^3$ kg

Energy The SI unit of energy, the joule, together with its multiples, is preferred for all applications. The *kilowatt hour* is widely used, however, as a measure of electric energy. This unit should not be introduced into any new areas, and eventually it should be replaced by the megajoule.

kilowatt hour 1 kWh = 3.6 MJ (exactly)

Others ANSI/IEEE Standard 268 lists the *kilowatthour* in the category of "Units in Use with SI Temporarily". In that same category, it also defines the *barn* ($1\ b = 10^{-28}\ m^2$) for cross section, the *bar* ($1\ bar = 10^5$ Pa) for pressure, the *curie* ($1\ Ci = 3.7 \times 10^{10}$ Bq) for radionuclide activity, the *roentgen* ($1\ R = 2.58 \times 10^{-4}$ C/kg) for X- and gamma-ray exposure, and the *rad* (1 rd = 0.01 Gy) for absorbed dose.

C.3 Conversion Factors

One of the many places a complete set of conversion factors can be found is in ANSI/IEEE Standard 268. The abridged set given here is taken from that reference. Symbols of SI units are given in bold face type and in parentheses. Factors with an asterisk (*) between the number and its power of ten are exact by definition.

To convert from	to	Multiply by
acre foot	meter^3 (**m^3**)	1.233 5 E+03
acre	meter^2 (**m^2**)	4.046 873 E+03
astronomical unit	meter (**m**)	1.495 979 E+11
atmosphere (standard)	pascal (**Pa**)	1.013 25*E+05
barrel (for petroleum, 42 gal)	meter^3 (**m^3**)	1.589 873 E–01
board foot	meter^3 (**m^3**)	2.359 737 E–03
British thermal unit (International Table)	joule (**J**)	1.055 055 852 62*E+03
calorie (International Table)	joule (**J**)	4.186 8*E+00
centimeter of mercury (0 °C)	pascal (**Pa**)	1.333 22 E+03
centimeter of water (4 °C)	pascal (**Pa**)	9.806 38 E+01
cup	milliliter (**mL**)	2.366 E+02
curie	becquerel (**Bq**)	3.7*E+10
day	second (**s**)	8.64*E+04
day (sidereal)	second (**s**)	8.616 409 E+04
degree (angle)	radian (**rad**)	1.745 329 E–02
degree Celsius	kelvin (**K**)	$T_K \equiv t_{°C} + 273.15$
degree Fahrenheit	degree Celsius	$t_{°C} \equiv (t_{°F} - 32)/1.8$
degree Fahrenheit	kelvin (**K**)	$T_K \equiv (t_{°F} + 459.67)/1.8$
degree Rankine	kelvin (**K**)	$T_K \equiv T_{°R}/1.8$
dyne	newton (**N**)	1*E–05
electronvolt	joule (**J**)	1.602 19 E–19
erg	joule (**J**)	1*E–07
fathom	meter (**m**)	1.828 8 E+00
foot	meter (**m**)	3.048*E–01
foot of water (39.2 °F)	pascal (**Pa**)	2.988 88 E+03
footcandle	lux (**lx**)	1.076 391 E+01
footlambert	candela per meter^2 (**cd/m^2**)	3.426 259 E+00
ft·lbf	joule (**J**)	1.355 818 E+00
ft·lbf/s	watt (**W**)	1.355 818 E+00
ft·poundal	joule (**J**)	4.214 011 E–02

To convert from	to	Multiply by
g (standard acceleration of free fall)	meter per second^2 (**m/s^2**)	9.806 65*E+00
gallon (US liquid)	meter^3 (**m^3**)	3.785 412 E–03
gauss	tesla (**T**)	1*E–04
grain	kilogram (**kg**)	6.479 891*E–05
horsepower (550 ft·lbf/s)	watt (**W**)	7.456 999 E+02
hour	second (**s**)	3.6*E+03
hour (sidereal)	second (**s**)	3.590 170 E+03
inch	meter (**m**)	2.54*E–02
inch of mercury (32 °F)	pascal (**Pa**)	3.386 38 E+03
inch of water (60 °F)	pascal (**Pa**)	2.488 4 E+02
kilogram-force (kgf)	newton (**N**)	9.806 65 *E+00
kilowatt hour (kW·hr or kWh)	joule (**J**)	3.6*E+06
kip (1000 lbf)	newton (**N**)	4.448 222 E+03
knot (international)	meter per second (**m/s**)	5.144 444 E–01
lambert	candela per meter^2 (**cd/m^2**)	1/π*E+04
langley	joule per meter^2 (**J/m^2**)	4.184*E–04
light year	meter (**m**)	9.460 55 E+15
liter	meter^3 (**m^3**)	1*E–03
maxwell	weber (**Wb**)	1*E–08
mho	siemens (**S**)	1*E+00
micron	meter (**m**)	1*E–06
mil	meter (**m**)	2.54*E–05
mile (international)	meter (**m**)	1.609 344*E+03
mile (US statute)	meter (**m**)	1.609 3 E+03
mile (nautical)	meter (**m**)	1.852*E+03
ounce (avoirdupois)	kilogram (**kg**)	2.834 952 E–02
ounce (troy or apothecary)	kilogram (**kg**)	3.110 348 E–02
ounce (US fluid)	meter^3 (**m^3**)	2.957 353 E–05
parsec	meter (**m**)	3.085 678 E+16
pica (printer's)	meter (**m**)	4.217 518 E–03
pound (mass)(avoirdupois)(lb or lbm)	kilogram (**kg**)	4.535 923 7*E–01
poundal	newton (**N**)	1.382 550 E–01
pound force (lbf)	newton (**N**)	4.448 221 615 260 5*E+00
quad	joule (**J**)	1.055 E+18
quart (US dry)	meter^3 (**m^3**)	1.101 221 E–03
quart (US liquid)	meter^3 (**m^3**)	9.463 529 E–04
rad (absorbed dose)	gray (**Gy**)	1*E–02
rem (dose equivalent)	sievert (**Sv**)	1*E–02
roentgen	coulomb per kilogram (**C/kg**)	2.58 E–04

To convert from	to	Multiply by
slug	kilogram (**kg**)	1.459 390 E+01
tablespoon	milliliter (**mL**)	1.479 E+01
teaspoon	milliliter (**mL**)	4.929 E+00
therm (US)	joule (**J**)	1.054 804*E+08
ton (explosive energy of TNT)	joule (**J**)	4.184*E+09
ton of refrigeration (12 000 Btu/h)	watt (**W**)	3.517 E+03
ton (short, 2000 lb)	kilogram (**kg**)	9.071 847 E+02
year (sidereal)	second (**s**)	3.155 815 E+07
year (tropical)	second (**s**)	3.155 693 E+07

Bibliography

Agrawal, Brij N., *Design of Geosynchronous Spacecraft*, Prentice-Hall, Inc., Englewood Cliffs, NJ 07632, 1986. **Referred to on page(s) 1.**

ANSI/ASTM E380-89a, *Standard Practice for Use of the International System of Units (SI) (The Modernized Metric System)*, American Society for Testing and Materials, 1916 Race St., Philadelphia, PA 19103, 1989.

Asher, Harold, *Cost-Quantity Relationships in the Airframe Industry*, R-291, The Rand Corporation, 1956. **Referred to on page(s) 78.**

Barclay, Scott, et al., *Handbook for Decision Analysis*, Decisions and Designs, Inc., McLean, VA, September 1977. **Referred to on page(s) 43.**

Biernacki, J., et al., "Applications of Modern Systems Analysis Paradigms to the Development of Complex Systems", in "Systems Engineering for the 21st Century", *Proceedings of the Second Annual Symposium of the NCOSE*, July 20–22, 1992, Seattle, WA, pp 581–587.

———, "Application of Enhanced Modern Structured Analysis Techniques to Space Station Freedom Electric Power System Requirements", *26th Intersociety Energy Conversion Engineering Conference*, Boston, MA, August 1991.

Bilardo, Vincent J. Jr., *Systems Analysis Plan (Revision 1)*, Systems Evaluation & Integration Branch, Advanced Life Support Division, NASA Ames Research Center, Moffett Field, CA, June 1990.

Blanchard, B.S., and W.J. Fabrycky, *Systems Engineering and Analysis*, Prentice-Hall, Inc. Englewood Cliffs, NJ, Second Edition 1990. **Referred to on page(s) 1.**

———, *Systems Engineering Management*, Wiley Interscience, 1991. **Referred to on page(s) 1.**

Boehm, Barry W., "A Spiral Model of Software Development and Enhancement", *Computer*, pp 61–72, May 1988. **Referred to on page(s) 13.**

Chamberlain, Robert G., George Fox and William H. Duquette, *A Design Optimization Process for Space Station Freedom*, JPL Publication 90-23, June 15, 1990. **Referred to on page(s) 18.**

Chestnut, Harold, *Systems Engineering Tools*, John Wiley & Sons, Inc., New York, 1965. **Referred to on page(s) 1.**

———, *Systems Engineering Methods*, John Wiley & Sons, Inc., New York, 1965. **Referred to on page(s) 1.**

Churchman, C. West, Russell L. Ackoff and E. Leonard Arnoff, *Introduction to Operations Research*, John Wiley & Sons, Inc., New York, 1957.

DeJulio, E., *SIMSYLS User's Guide*, Boeing Aerospace Operations, February 1990. **Referred to on page(s) 81.**

de Neufville, R., and J.H. Stafford, *Systems Analysis for Engineers and Managers*, McGraw-Hill, New York, 1971. **Referred to on page(s) 1, 43.**

Defense, Department of, *Transition from Development to Production*, DoD 4245.7-M, 1985. **Referred to on page(s) 40.**

———, *Metric System, Application in New Design*, DOD-STD-1476A, 19 November 1986.

Defense Systems Management College, *Systems Engineering Management Guide*. **Referred to on page(s) 1.**

———, *Scheduling Guide for Program Managers*, GPO #008-020-01196-1, January 1990.

———, *Risk Management: Concepts and Guidance*, 1987.

Dixon, Bernard, and Paul Villone, *Goddard Multi-variable Instrument Cost Model (MICM)*, Resource Analysis Office, NASA Goddard Space Flight Center, Research Note #90-1, May 1990. **Referred to on page(s) 77.**

Fisher, Gene H., *Cost Considerations in Systems Analysis*, R-490-ASD, The Rand Corporation, December 1970. **Referred to on page(s) 63.**

Forsberg, Kevin, and Harold Mooz, "The Relationship of System Engineering to the Project Cycle", Center for Systems Management, 5333 Betsy Ross Dr., Santa Clara, CA 95054; also available in *A Commit-*

ment to Success, Proceedings of the first annual meeting of the National Council for Systems Engineering and the 12th annual meeting of the American Society for Engineering Management, Chattanooga, TN, 20–23 October 1991. **Referred to on page(s) 20.**

Green, A.E., and A.J. Bourne, *Reliability Technology*, Wiley Interscience, 1972.

Griffin, Michael D., and James R. French, *Space Vehicle Design*, AIAA 90-8, American Institute of Aeronautics and Astronautics, c/o TASCO, P.O. Box 753, Waldorf, MD 20604-9961, 1990. **Referred to on page(s) 1.**

Hickman, J.W., et al., *PRA Procedures Guide*, The American Nuclear Society and The Institute of Electrical and Electronics Engineers, NUREG/CR-2300, Washington, DC, January 1983. **Referred to on page(s) 42.**

Hillebrandt, P., et al., *Unmanned Space Vehicle Cost Model, Sixth Edition*, Air Force Systems Command/Space Division, SD TR-88-97, November 1988. **Referred to on page(s) 77.**

Hillier, F.S. and G.J. Lieberman, *Introduction to Operations Research*, 2nd Edition, Holden-Day, Inc., 1978.

Hodge, John, "The Importance of Cost Considerations in the Systems Engineering Process", in the *NAL Monograph Series: Systems Engineering Papers*, NASA Alumni League, 922 Pennsylvania Ave. S.E., Washington, DC 20003, 1990. **Referred to on page(s) 14.**

Hood, Maj. William C., *A Handbook of Supply Inventory Models*, Air Force Institute of Technology, School of Systems and Logistics, September 1987.

Hughes, Wayne P., Jr. (ed.), *Military Modeling*, Military Operations Research Society, Inc., 1984.

IEEE, *American National Standard Metric Practice*, ANSI/IEEE Std 268-1982 (supersedes IEEE Std 268-1979 and ANSI Z210.1-1976), American National Standards Institute (ANSI), 1430 Broadway, New York, NY 10018. **Referred to on page(s) 107.**

Jet Propulsion Laboratory, *The JPL System Development Management Guide*, Version 1, JPL D-5000, Jet Propulsion Laboratory, November 15, 1989.

Keeney, R.L., and H. Raiffa, *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*, John Wiley and Sons, Inc., New York, 1976. **Referred to on page(s) 72.**

Kline, Robert, et al., *The M-SPARE Model*, Logistics Management Institute, NS901R1, March 1990. **Referred to on page(s) 81.**

Lano, R., *The N^2 Chart*, TRW Inc., 1977. **Referred to on page(s) 64, 99.**

Leising, Charles J., "System Architecture", in *System Engineering at JPL*, course notes (contact: Judy Cobb, Jet Propulsion Laboratory), 1991. **Referred to on page(s) 10.**

Lexicon — Glossary of Technical Terms and Abbreviations Including Acronyms and Symbols, Draft/Version 1.0, produced for the NASA Program/Project Management Initiative, Office of Human Development (Code ND), NASA Headquarters, by DEF Enterprises, P.O. Box 590481, Houston, TX 77259, March 1990. **Referred to on page(s) 87.**

Marshall Space Flight Center, *Systems Engineering Handbook, Volume 1 — Overview and Processes; Volume 2 — Tools, Techniques, and Lessons Learned*, MSFC-HDBK-1912, Science and Engineering, Systems Analysis and Integration Laboratory, Systems Analysis Division, NASA George C. Marshall Space Flight Center, February 1991. **Referred to on page(s) 71, 85.**

McCormick, Norman, *Reliability and Risk Analysis*, Academic Press, Orlando, FL, 1981.

Military Standards

———, *Systems Engineering*, MIL-STD-499B, Department of Defense, currently in revision. **Referred to on page(s) 1.**

———, *Logistics Support Analysis Record*, MIL-STD-1388-2B, currently in revision, Department of Defense. **Referred to on page(s) 83.**

———, *Failure Modes, Effects, and Criticality Analysis*, MIL-STD-1629A, Department of Defense. **Referred to on page(s) 40.**

Miles, Ralph F., Jr. (ed.), *Systems Concepts — Lectures on Contemporary Approaches to Systems*, John Wiley & Sons, New York, 1973. **Referred to on page(s) 1.**

Moore, N., D. Ebbeler and M. Creager, "A Methodology for Probabilistic Prediction of Structural Failures of Launch Vehicle Propulsion Systems", American Institute of Aeronautics and Astronautics, 1990. **Referred to on page(s) 85.**

Morgan, M. Granger, and Max Henrion, *Uncertainty: A Guide to Dealing with Uncertainty in Quantitative Risk and Policy Analysis*, Cambridge University Press, Cambridge, England, 1990.

Morgan, William C., *Systems Engineering/Integration Process Briefing*, Planet Surface Systems Office, NASA Johnson Space Center, May 1991.

Morris, Owen, "Systems Engineering and Integration and Management for NASA Manned Space Flight Programs", in *NAL Monograph Series: Systems Engineering Papers*, NASA Alumni League, 922 Pennsylvania Ave. S.E., Washington, DC 20003, 1990. **Referred to on page(s) 9, 10.**

NHB 5103.6B, *Source Evaluation Board Handbook*, NASA Handbook 5103.6B, Office of Procurement (Code H), NASA Headquarters, October 1988. **Referred to on page(s) 71.**

NHB 9501.2B, *Procedures for Contractor Reporting of Correlated Cost and Performance Data*, NASA Handbook 9501.2B, Financial Management Division (Code BF), NASA Headquarters, February 1985. **Referred to on page(s) 56.**

NMI 7100.14B, *Major System Acquisitions*, Program Operations Division (Code HS), NASA Headquarters, February 27, 1990. **Referred to on page(s) 3, 13, 14, 17, 73, 75.**

NMI 7120.3, *Space Flight Program and Project Management*, Office of Management (Code N), NASA Headquarters, February 6, 1985. **Referred to on page(s) 17.**

NMI 8010.1A, *Classification of NASA Payloads*, Safety Division (Code QS), NASA Headquarters, 1990. **Referred to on page(s) 39, 97.**

NMI 8010.2A, *Use of the Metric System of Measurement in NASA Programs*, Office of Safety and Mission Quality, NASA Headquarters, June 11, 1991. **Referred to on page(s) 107.**

NMI 8070.4A, *Risk Management Policy*, Safety Division (Code QS), NASA Headquarters, undated. **Referred to on page(s) 37, 43.**

OMB Circular A-94, *Discount Rates To Be Used In Evaluating Time-Distributed Costs and Benefits*, Office of Management and Budget, March 1972. **Referred to on page(s) 75.**

OSSA, *Initial OSSA Metrication Transition Plan*, Office of Space Science and Applications, NASA Headquarters, September, 1990.

Pace, Scott, *U.S. Access to Space: Launch Vehicle Choices for 1990-2010*, The Rand Corporation, R-3820-AF, March 1990.

Ruskin, Arnold M., "Project Management and System Engineering: A Marriage of Convenience", Claremont Consulting Group, La Cañada, California; presented at a meeting of the Southern California Chapter of the Project Management Institute, January 9, 1991. **Referred to on page(s) 10.**

SP-7012 (NASA Special Publication), *The International System of Units; Physical Constants and Conversion Factors*, E.A. Mechtly, published by the NASA Scientific and Technical Information Office, 1973; U.S. Government Printing Office, (stock number 3300-00482).

Saaty, Thomas L., *The Analytic Hierarchy Process*, McGraw-Hill, New York, 1980. **Referred to on page(s) 71.**

Shishko, R., *Catalog of JPL Systems Engineering Tools and Models (1990)*, JPL D-8060, Jet Propulsion Laboratory, 1990.

———, *MESSOC Version 2.2 User Manual*, JPL D-5749/Rev. B, Jet Propulsion Laboratory, October 1990. **Referred to on page(s) 77.**

Sivarama Prased, A.V., and N. Somasehara, "The AHP for Choice of Technologies: An Application", *Technology Forecasting and Social Change*, Vol. 38, pp. 151–158, September 1990.

Smith, Jeffrey H., Richard R. Levin and Elisabeth J. Carpenter, *An Application of Multiattribute Decision Analysis to the Space Station Freedom Program — Case Study: Automation and Robotics Technology Evaluation,* JPL Publication 90-12, Jet Propulsion Laboratory, May 1, 1990. **Referred to on page(s) 72.**

Steuer, R., *Multiple Criteria Optimization*, John Wiley and Sons, Inc., New York, 1986.

Stewart, Rodney D., *Cost Estimating*, 2nd ed., John Wiley and Sons, Inc., New York, 1991.

Wagner, G. M., *Principles of Operations Research — With Applications to Managerial Decisions*, Prentice Hall, 1969.

Wertz, James R., and Wiley J. Larson (eds), *Space Mission Analysis and Design*, Kluwer Academic Publishers, 101 Philip Drive, Norwell, MA 02061, 1991. **Referred to on page(s) 1.**

Index

Acceptance review 19, 20, 45, 52, 53
Advanced projects 13
Analytic Hierarchy Process (AHP) 71
Apollo 9, 10, 80
Architecture — see system architecture
Audits 46, 48, 52, 53
Availability
    measures of 82
    as a facet of effectiveness 79–81
    models of 81–83
Baselines 4, 8, 14, 17–20, 35–37
    control and management of 10, 21, 27–30, 39, 44–48, 90, 96
    in C/SCS 56
    in reviews 50–54
Bayesian probability 40
Budget cycle, NASA 17, 20
Budgeting, for projects 31, 35, 37, 55

Change Control Board (CCB)
    composition of 46
    conduct of 45–47
Change Request (CR) 45, 46, 49, 61, 66
Concurrent engineering 21, 22, 27–29
Configuration control 8, 11, 17, 44–47, 51, 96
Configuration management 4, 10, 17, 45, 46, 50, 90, 96
Congress 3, 17, 20
Constraints 3, 5, 8–11, 14, 18, 28, 35, 46, 50, 58, 63–68, 70, 71, 73, 97
Contingency planning 43, 44
Control gates 13–22, 27–29, 45, 46, 48, 49, 89, 90
Cost (see also life-cycle cost) 4, 5, 9, 10
    account structure 27, 30, 31, 33
    caps 35, 37
    estimating 76–79
    fixed vs variable 37
    in trade studies 63–65, 70, 73
    operations 77, 104–106
    overruns 17, 73
    spreaders 78
Cost-effectiveness 4, 5, 9, 10, 17, 92
    in trade studies 63–65, 68, 70, 73
Cost Estimating Relationship (CER) 76, 77, 84
Cost/Schedule Control System (C/SCS) 56
Critical Design Review (CDR) 18, 45, 46, 51, 52, 54, 55
Critical Item/Issue List (CIL) 39, 44, 54, 55, 98
Critical path 13, 33, 36

Decision analysis 39, 41, 72
Decision sciences 7, 73
Decision support package 10, 66
Decision trees 41, 66, 70
Design engineer(ing) 6, 7, 22, 28, 49, 73
Design reference mission — see reference mission
Design trades — see trade studies
Digraphs 41
Discounting — see present discounted value
Dynamic programming 68

Earned value 7, 31, 56, 59
Effectiveness 4, 5, 9, 10
    facets of 79–81
    in TPM 58
    in trade studies 63–65
Engineering Change Request (ECR) — see change request
Engineering specialty disciplines 6, 44
    in concurrent engineering 23
    in SEMP 29, 92
    in trade studies 65, 73, 76, 80
Estimate at Completion (EAC) 31, 56, 57, 84
Event trees 41, 42

Failure Modes and Effects Analysis (FMEA) 39–41, 54
Failure Modes, Effects, and Criticality Analysis (FMECA) 39–41
Fault tree 41, 42
Feasibility 14, 21, 22, 50, 58
Feasible design 5, 17, 18
Figure of merit 70, 71
Freedom — see Space Station *Freedom*
Functional Flow Block Diagram (FFBD) 64, 99–101

Game theory 7
Gantt chart 35, 36
Goddard Space Flight Center (GSFC) 77

Heisenberg — see uncertainty principle

IEEE 42, 107, 109, 110
Improvement
    continuous 60
    product or process 11, 18, 22, 47, 83
Inflation, treatment of 74, 75, 78
Institutional Operating Plan (IOP) 20
Integrated Logistics Support (ILS) 29, 30, 74, 82
    plan 19, 21, 92
Integration, conceptual 11
Integration, system 4, 11, 18, 25, 29, 30, 32, 33, 53, 92
Inter-Center Systems Engineering Working Group (IC-SEWG) *vii*, 3, 13, 21

Interface
requirements 6, 9–11, 17, 44, 50, 51, 64, 92, 99
control of 28, 60, 92, 96

Johnson Space Center (JSC) 43, 78

Learning curve 78
Lessons learned 11, 19, 30, 39, 40, 53
Lexicon, NASA 87
Life-cycle cost (see also cost) 8, 10, 73–79
components of 73, 74
controlling 75, 76
Linear programming 7, 68

Make-or-buy 29
Margin 42, 44, 59, 60
Marshall Space Flight Center (MSFC)
handbook 71, 85
historical cost models 77
Mean Time Between Failure (MTBF) 76, 82, 104, 105
Mean Time to Repair (or Restore) (MTTR) 82
Metric system
conversion factors for 110–112
definition of units in 107–109
Military standards 1, 4, 40, 83
Mission analysis 7, 14
Mission Needs Statement (MNS) 14, 17, 45
Models, mathematical
characteristics of good 68, 69
of cost 42, 77, 78
of effectiveness 42, 79–81
pitfalls in using 68, 84
relationship to SEMP 29, 78, 79
types of 67, 68
use in systems engineering 6, 9, 11, 21, 63–67, 83–85
Monte Carlo simulation 84, 85
Multi-attribute utility theory 71, 72

Network schedules 33–35, 42
Non-advocate Review (NAR) — see Project Definition and Cost Review
NASA Management Instruction (NMI) *vii*, 1, 3, 13, 14, 17, 37, 39, 43, 73, 75, 107

Objective function 4, 10, 70
Objectives, of a mission, project, or system 3, 4, 8, 11, 14, 17, 34, 35, 37, 39, 50, 55, 59, 63–67, 70–73, 79, 82
Office of Management and Budget (OMB) 20, 75
Operations concept 14, 17, 64, 65, 73
Operations research 7
Optimization 3, 6, 7, 9, 13, 50, 63, 68, 76, 79, 92

Orbital Replacement Unit (ORU) 76, 81, 104, 105

Parametric cost estimation 76–78
Partitioning — see interfaces
Payload 17, 19, 39, 50, 51, 57, 81, 97, 104
PERT 34, 39, 42
Precedence diagram 34
Preliminary Design Review (PDR) 18, 45, 46, 50, 51, 54, 55
Present Discounted Value (PDV) 74, 75
Probabilistic Risk Assessment (PRA) 39, 41, 42, 44, 72
Probability distribution 5, 9, 10, 40–42, 59, 60, 84, 85
Producing system 1, 27, 56, 60
Product Breakdown Structure (PBS) 27, 30–33, 55, 58, 93
Product development process 7, 13, 18, 20–22, 65
Product development teams (PDT) 23
Product system 1, 27
Program, level of hierarchy 3
Program/project control *vii*, 44, 55–57
Program Operating Plan (POP) 20
Project
level of hierarchy 3
management (see also system management) *vii*, 27, 37, 55, 79
plan 17–19, 28–30, 50
termination 48
Project cycle
NASA 13–20, 89, 90
technical aspect of 20–25
Project Definition and Cost Review (PDCR) 17
Project Initiation Agreement (PIA) 14, 17
Prototype 13, 77

Quality
of systems engineering process 60, 61, 71
as a facet of effectiveness 79, 81
Quality assurance 6, 29, 49–52, 92
Quality Function Deployment (QFD) 7
Queueing theory 7

Red team 49
Reference mission 9, 65
Reliability 2, 6, 22, 39, 41, 49–51, 75, 76
in effectiveness 68, 81–85, 104
in SEMP 29, 92
in TPM 57, 58
Reporting — see status reporting and assessment
Requirements 3, 6, 11, 14, 17–19, 21, 22, 25, 28, 30, 37, 45, 46, 51–54, 60, 61, 67, 92
allocation of 11, 90, 92, 101
analysis 7, 9, 79, 90–92, 99
as part of the baseline 4, 8, 17, 18, 44

Requirements (continued)
design 9, 28, 51, 54, 101
documentation 14, 18, 45
functional 9, 51, 65, 66, 73, 79
interface 9, 17, 50, 51, 99
performance 9, 28, 52, 56, 58, 64, 69, 73, 74
reviews 17, 45, 49, 50
role of 27
traceability 17, 28, 30, 48, 101
Reserves
project 17, 37, 42, 44, 56, 84
schedule 17, 35, 42
Resource leveling 35
Resource planning — see budgeting
Risk
analysis 38, 39, 41, 42
aversion 41, 71
identification and characterization 38, 39–41
management 29, 37–46, 50–55, 84, 92, 97
mitigation 38, 39, 42–44
templates 40
types of 39, 40

Safety reviews 17, 19, 53–55
Scheduling 33–35, 55
S-curve, for costs 84
Selection rules, in trade studies 6, 10, 64, 65, 69–73
Simulations 29, 67, 68, 81, 82, 84, 85
SOFIA 93
Software 3, 6, 13, 19, 21, 22, 44, 46, 47, 50–52, 54, 65, 74, 96, 99
cost estimating 76,77
in WBS 30, 32, 34
off-the-shelf systems engineering 35, 41, 71, 85
Source Evaluation Board (SEB) 71
Space Shuttle 3, 40, 44, 47, 98, 104, 105
Space Station *Freedom* 8, 11, 39, 40, 72, 76, 77, 81, 89, 104
Specialty disciplines — see engineering specialty disciplines
Specifications 8, 9, 17, 18, 25, 29–31, 44, 45, 48–52, 54, 55, 57, 59, 61, 92, 96
Status reporting and assessment 31, 55–61, 84
Successive refinement, doctrine of 7–11, 27
Supportability 42, 81–83
Symbolic information
desireable characteristics of 48
in systems engineering 27
System architecture 6, 8, 11, 14, 17–20, 27, 31, 64, 65, 68, 69, 72, 73, 75, 79, 85
System engineer
role of 44
dilemma of 6, 75
System management (see also project management) 4, 6, 90
Systems analysis, role of 6, 7, 57, 63
Systems approach 7
Systems engineering
objective of 4–6
metrics 60, 61
process *vii*, 5, 10, 20, 27–30, 33, 38, 63–67, 73, 75, 76, 78, 92
Systems Engineering Management Plan (SEMP) 28–31, 39, 40, 60, 66, 78, 82, 92
Systems Engineering Working Group — see Inter-Center Systems Engineering Working Group

Tailoring
by each Center 1
of effectiveness measures 79
of project cycle 13, 28, 89–91
of SEMP 29
of systems engineering process metrics 60
Technical Performance Measure(ment) (TPM)
assessment methods for 45, 59, 60, 84
relationship to effectiveness measures 79
relationship to SEMP 60, 92
role and selection of 31, 39, 44, 57, 58
Test(ing) (see also verification) 3, 6, 11, 13, 14, 17–19, 22, 25, 33, 42, 45, 46, 49–53, 58–60, 65, 75, 77, 90, 92, 93
Test Readiness Review (TRR) 19, 30, 51, 52
Total Quality Management (TQM) 7, 60, 92
Trade study
process 9, 17, 18, 63–67, 72
progress as a metric 60, 61
reports 10, 18, 67
Trade tree 65, 66

Uncertainty, in systems engineering 5, 6, 19, 38–44, 65, 75, 83–85
Uncertainty principle 39

Validation 11, 14, 26, 28–30, 57, 61, 90
Variances, cost and schedule 56, 57
Verification 4, 11, 17–19, 26, 28–30, 44, 46, 90
as part of reviews 51–53
relationship to status reporting 57, 60, 61

Waivers 52, 54
Work Breakdown Structure (WBS) 4, 17, 18, 27, 34–37, 50, 56, 76, 77, 92
development of 30–33
errors to avoid 32, 33

Work Breakdown Structure (continued)
example of 93–95
Work flow diagram 34