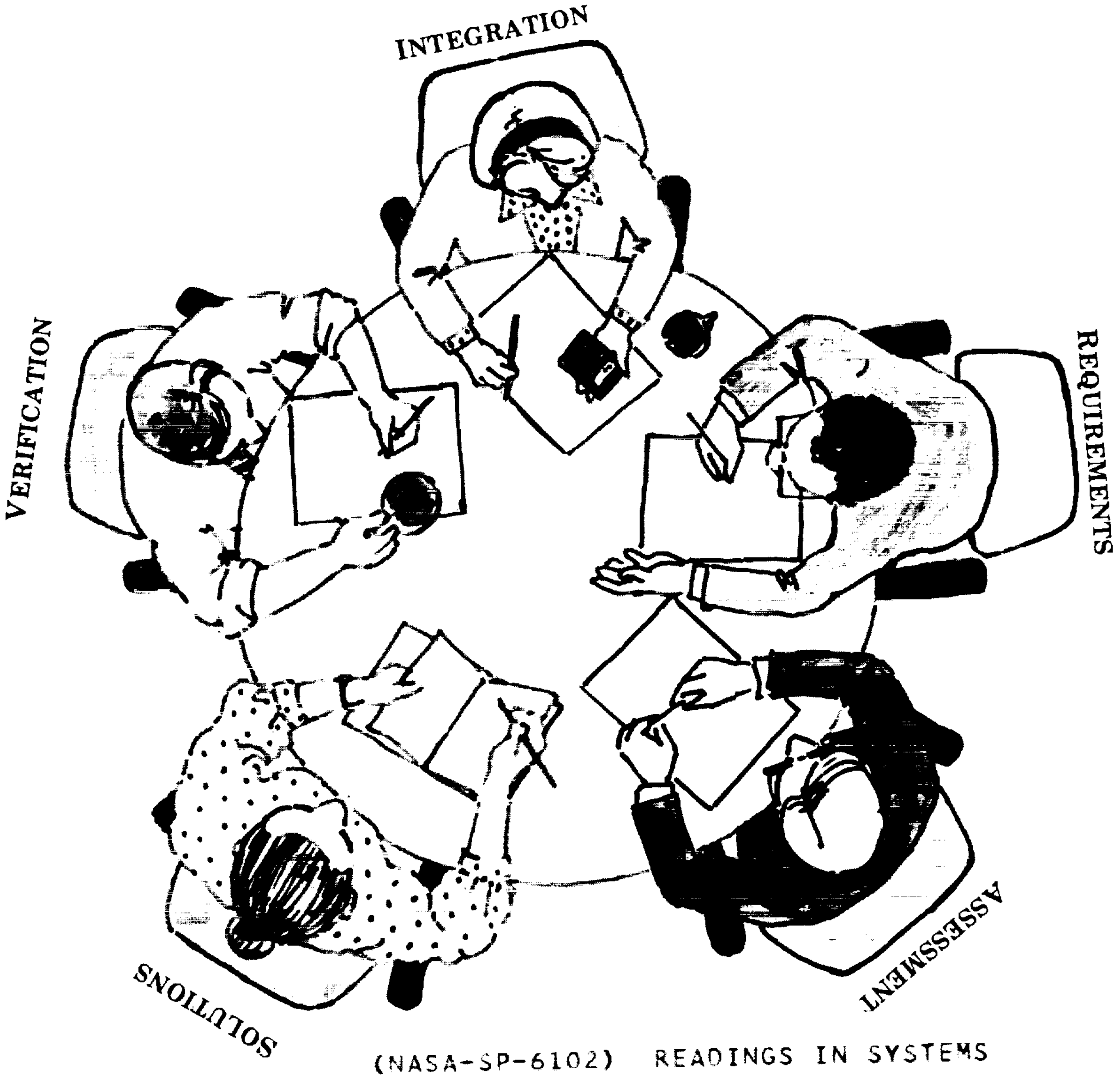
NASA - SP-6102

READINGS IN SYSTEMS ENGINEERING

Edited by Francis T. Hoban
and William M. Lawbaugh



READINGS IN SYSTEMS ENGINEERING

Edited by Francis T. Hoban
and William M. Lawbaugh

NASA National Aeronautics and Space Administration
Scientific and Technical Information Program
Washington, DC 1993

READINGS IN SYSTEMS ENGINEERING

Introduction by Francis T. Hoban and William M. Lawbaugh

com 1 10
p. 9

INTRODUCTION

by Francis T. Hoban and William M. Lawbaugh

Systems engineering is not new—it's been around for quite some time. The ancient engineers who designed and built the pyramids practiced some form of what we today call "systems engineering." Modern systems engineering emerged during and immediately following World War II as weapons grew into weapon systems, due to the degree of complexity in design, development and deployability. The advent of space exploration further increased the need for and use of systems engineering processes and practices.

The Apollo Program is perhaps NASA's best example of the application of systems engineering. During Apollo, systems engineering processes were in place in NASA Headquarters and at all field centers. At some locations the process was formalized; at others it was a back-of-the-envelope application, but it was in place. It was widely practiced, and it sustained a young and vibrant organization during the design, development and operations of the world's greatest engineering feat. NASA systems engineering capabilities grew out of its NACA heritage, bolstered by people from the Department of Defense, industry and academia who joined the team during the Apollo build-up. It should be noted that during the Apollo era, systems engineering was conducted without Agency-wide guidance, standards or lexicon. After Apollo, the various NASA centers continued to implement systems engineering on complex projects but perhaps with less vigor and enthusiasm than that displayed during Apollo.

The discipline again became a priority in NASA when the study team of the National Academy of Public Administration, led by Lt. General Sam C. Phillips, recommended the strengthening of systems engineering in NASA.

This group, in its final report to the NASA administrator on December 30, 1986 also recommended a renewed effort in the education and training of the NASA program and project management workforce. Unwritten but well understood in this recommendation was renewed emphasis on systems engineering training. It was no easy task to build a knowledge base, create a library collection and develop courses and workshops in systems engineering, but the efforts took shape, as one essential part of NASA's Program and Project Management Initiative. This became a continuing education process assisted on an Agency-wide basis by the Systems Engineering Working Group.

Today most large engineering organizations, including NASA, have a systems engineering process containing elements both common and unique to those practiced by other organizations. To document these processes NASA is now involved in the preparation of the first Agency-wide systems engineering manual. The manual addresses common systems engineering practices and tools, as well as those unique to NASA and the aerospace industry. This manual, together with those describing the individual Centers' practices, will fully document systems engineering at NASA and add to the education process.

This present collection was inspired by seven papers prepared by the NASA Alumni League, illustrating the members' systems engineering experience. These papers make up the heart of this collection. We have supplemented them with papers describing industry processes and other governmental practices to illustrate the diversity of systems engineering as it is formulated and practiced. This is one discipline that clearly benefits from cross-fertilization and infusion of new ideas.

There is also a wide variety of tools and techniques described herein, some standard and some unique. It is not unlike an elite crew of talented carpenters showing up for a job, each with some different tools in their respective toolboxes, and each with different tricks or techniques to save time or money—to do each one-of-a-kind job better, cheaper, faster.

If all the authors of *Readings in Systems Engineering* were ever to assemble in one place, there would be some unanimity on basics and essentials but much debate and downright disagreement on the particulars. Nevertheless, the meeting would be lively and interesting—a decent description of the dynamic process of systems engineering itself.

APPROACH

We begin our collection with a now-famous speech delivered by Bob Frosch to a group of engineers in New York in 1969, shortly before his appointment as NASA Administrator. The speech sets the tone best of all for this volume: Frosch urges a common sense approach to systems engineering.

When weapons evolved into weapons systems, the Department of Defense took the lead in systems, engineering. Today the DoD approach is widely recognized, and so we present the newly revised (1990) description of the systems engineering process from the Defense Systems Management College at Fort Belvoir, Virginia. A senior project engineer formerly with Hughes Aircraft, Paul E. Lewkowitcz, then discusses requirement analysis, technology assessment, solution synthesis and performance verification for very large systems. Marshall Space Flight Center does it differently, but one of the best descriptions of Phase A through C can be found in Marshall's systems engineering handbook. To close out this overview section, excerpts from the forthcoming *NASA Systems Engineering Handbook* stress the engineering aspects of successful management of aerospace systems.

In our second section, devoted to specific applications of systems engineering, we begin with two engineers from the Goddard Space Flight Center whose presentations are famous for explaining the process and products of systems engineering in unmanned spacecraft. Tony Fragomeni and Mike Ryschkewitsch are associate chief and chief respectively of Goddard's new Systems Engineering Office. Owen Morris, who was NASA's Lunar module project manager and the Space Shuttle Systems and Engineerng manager, then discusses the history of systems engineering and integration (SE&I) management in manned space programs. Chuck Mathews, past president of the NASA Alumni League, and NASA's manager of the Gemini Program, director of the the Skylab Program and associate administrator for Applications, then focuses upon the systems engineering role in establishing, verifying and controlling top-level program requirements. John D. Hodge, a 25-year veteran of the Department of Transportation and NASA, including the Mercury, Gemini, Apollo and Space Station programs, retiring as an associate administrator, explains cost considerations in the systems engineering process, urging clear definition of requirements, stable management and strong central control to allocate funds properly.

John E. Naugle, retired NASA associate administrator and chief scientist, describes the dual role of "master" and "servant" for the systems engineer, from the requirements phase to preliminary design. Eugene F. Kranz, director of the Johnson Space Center Mission Operations Directorate, and Christopher C. Kraft Jr., former director of the Johnson Space Center, stress manned mission operations and SE&I. Robert O. Aller, recently retired associate administrator for space operations, views operations support as the infrastructure of people, procedures, facilities and systems for flight success. John Yardley, NASA's associate administrator for manned space flight during the Space Shuttle development, asserts that the systems engineer should consider 10 different political and nontechnical groups. Associate David Wensley suggests ways of handling political and institutional factors in the systems engineering process. Loren A. Lemmerman, formerly of the Lockheed-Georgia Company, shows how a large aircraft company fits optimization into systems engineering and the total design process.

Next, we present what today can be looked at as case studies in lost systems engineering opportunities. On May 14, 1973, a minute into flight, Skylab 1 lost its meteoroid shield and one of two solar array systems. The NASA Investigation Board determined that aerodynamic loads were probably not accounted for in design. Likewise, the Seasat Mission Failure Investigation Board described the uncritical acceptance of "standard" or "flight proven" equipment that failed in 1978. But no one wants to end on a negative, so we reproduce a Johnson Space Center engineer's attempt to define and explain "systems engineering" a quarter of a century ago.

This book is primarily for the next generation of systems engineers, so we look ahead. As Bob Aller concludes, "The need for systems engineering is critical to NASA in its preparations for conducting operations in the late 1990s and into the next decade."

The editors gratefully acknowledge the authors for sharing their information with us. We also wish to thank the NASA Alumni League for its most important contribution. We thank the NASA Systems Engineering Working Group and the entire NASA systems engineering family for their encouragement and support.

A CLASSIC LOOK AT SYSTEMS ENGINEERING

by Robert A. Frosch

Editors' Note

Before his term as NASA Administrator, Bob Frosch was an assistant secretary of the U.S. Navy in charge of research, development, test and evaluation of Navy programs. In that capacity, he delivered a controversial and well-remembered speech to the IEEE Group on Aerospace and Electronic Systems during IEEE's international convention in New York on March 26, 1969. Edited portions of that famous speech follow in an effort to preserve what is now considered a classic formulation of systems engineering as an art rather than a science.

In this presentation, I really will be discussing the application of systems engineering to development, and in particular to military systems development (with which I am most familiar). However, from reading various journals and newspapers, I suspect my remarks are of more general applicability. I have said some of these things before, but some bear repeating and some I hope will spark new ideas.

I couple systems engineering, systems analysis and Management (with a capital "M"), because in practice they seem to be closely related terms, referring to the same constellation of systematic practices and attitudes.

- We badly lack: systems engineering of systems engineering; systems analysis of systems analysis.
- And, heaven knows, there is no: management of Management.
- Therefore, I will now preach against home, motherhood and apple pie.

To the charge that I am writing about bad systems engineering, I can only say that I am taking a pragmatic view: the thing is defined by what is done, not what is said; and if what I am describing is bad systems engineering, I can only say that I seldom see any other kind.

What I want to do is discuss briefly a series of antitheses (and perhaps an unbalanced question or two) that pit the systems world against what I believe are some aspects of the real world.

If I plot a graph versus time of what appears to be a recent rising tide of costs, cost overruns, unsatisfactory performance and unhappiness among engineers, I have reason to worry. (If this trend continues, we may have to debate whether the question "whither engineering?" is spelled with one "h" or two.) If I plot on the same graph versus time the rise in talk, directives, and use of "systems engineering," "systems analysis" and "Management," I see high correlation between the two graphs—trouble versus time and the use of systems engineering versus time. This does not prove causation, but it suggests, at least, that the "new techniques" are proving to be a poor substitute for real science and engineering; they are, at the least, not doing what they are advertised as doing, if they are indeed actually not making things worse. It could be that things would be even worse without these new techniques, but I would like to ask some questions and suggest some reasons for believing that systems engineering, systems analysis and Management, as practiced, are likely to be part of the problem, and indeed causative agents.

I believe that the fundamental difficulty is that we have all become so entranced with technique that we think entirely in terms of procedures, systems, milestone charts, PERT diagrams, reliability systems, configuration management, maintainability groups and the other minor paper tools of the "systems engineer" and manager. We have forgotten that someone must be in control and must exercise personal management, knowledge and understanding to create a system. As a result, we have developments that follow all of the rules, but fail.

I can best describe the spirit of what I have in mind by thinking of a music student who writes a concerto by consulting a checklist of the characteristics of the concerto form, being careful to see that all of the canons of the form are observed, but having no flair for the subject, as opposed to someone who just knows roughly what a concerto is like, but has a real feeling for music. The results become obvious upon hearing them. The prescription of technique cannot be a substitute for talent and capability, but that is precisely how we have tried to use technique.

PAPER VS. PEOPLE

My first antithesis pits the systems world of paper and arrangements against the real world of people and hardware. When paper appears in the real-world version of a system, it is generally only as an abstracted commentary. For example, in a very basic sense it really is of no consequence whether the documentation on a weapons system is good, bad or nonexistent; that is only a commentary on whether or why the people and the hardware actually work when called upon, and a tool to help them work. If the systems arrangements on paper and the documentation can help to make the stuff work, then they are of some use. If they are merely the formal satisfaction of a requirement, they are only an interference with engineering. Systems, even very large systems, are not developed by the tools of systems engineering, but only by the engineers using the tools. In looking back at my experiences in development, including watching a number of Navy developments over the past few years, it seems quite clear that in most cases where a system gets into trouble, a competent manager knows all about the problem and is well on the way to fixing it before any management systems ever indicate that it is about to happen. This happens if for no other reason than because the competent manager is watching what is going on in great detail and perceives it long before it flows through the paper system. That is to say, personal contact is faster than form-filling and the U.S. mails. A project manager who spends much time in a Management Information Center instead of roving through the places where the work is being done is always headed for catastrophe. The MIC can assist the people who are not involved in the project toward learning of after-the-fact problems, but that is roughly all that it can do, and its value even for this purpose is frequently questionable.

Blaming deficiencies in management systems for problems that exist in real unknowns, or in the deficiencies of people, is mere foolishness. In a poem called "Bagpipe Music," by Louis MacNeice, the final couplet is:

> *"The glass is falling hour by hour,*
> *the glass will fall forever*
> *But if you break the bloody glass,*
> *you won't hold up the weather."*

LINEARITY VS. THE REAL WORLD

One of the key misassumptions in modern systems engineering and systems analysis is that the total problem can be, and frequently is, decomposed into subproblems; the subproblems can be solved more or less independently, and the total solution can be synthesized by combination of the subsolutions, treating the interactions of the parts as "interfaces." The real world is, however, highly non-linear, and unless real attention is paid to this fact, the linear decomposition treatment will fail catastrophically, because the interaction terms may be as large as the subproblems and not reducible to simple interfaces. The result may well remain decomposed.

This criticism is frequently answered by the comment that problems are unmanageable unless sliced up and, therefore, the procedure is used even though we know it may be seriously in error. This is the case of the man who played in a poker game that he knew to be crooked, because it was the only game in town; or the drunk who looked for his ring under the street lamp even though he had lost it a block away in the dark—the light was better under the street light. I have some difficulty seeing that a bad analysis is really better than an informed judgment, especially since faith in the analysis (and/or the decomposed solution to the problem) is frequently, nay, usually, used as a substitute for seeking or applying any judgment at all. I am often faced with a result that seems absurd, and can even produce a quick analysis that at least makes it obvious that the solution is absurd, but am then given the answer, "Well, that's what the analysis showed."

Such a situation usually indicates room for deep criticism, either of the way in which the problem was divided up, or of peculiarities of the assumptions that drive the problem in curious and unsuspected ways, particularly through the unsuspected (by the systems person) nonlinearities of the problem. It sometimes appears that the only rational subdivision of the problem is to fractionize the blame to the point where approval is sought by default.

I would argue that careful attention to the parts of the problem that do not seem to be easily decomposable into semi-independent parts might be one very good guide to areas involving high risk, since these are likely not to be amenable to our usual rules, procedures and technologies, and hence probably will have to be approached empirically.

SERIAL VS. ITERATIVE MODELS

Systems engineering techniques themselves contribute to disaster because they are all paper techniques and there are only two instead of N dimensions available. What we end up displaying are linear sequential measures of system progress.

The PERT diagram and the milestone chart are excellent examples. These both essentially assume that the progress of development and design consists of doing step A, then step B, then step C, etc. Anyone who has ever carried out a development or a design (as opposed to setting up a management system for doing so) is well aware of the fact that the real world proceeds by a kind of feedback iterative process that looks more like a helix than like a line. That is to say, you do A, then B, then C, then you look at C and go back and change part of A again, and that causes you to fiddle with B and perhaps bring in a B-prime that you bounce against C, and then go back to A and then jump to D, so that there has to be continual adjustment, going back and forth so that the system is adjusted to itself and to its end objectives as it changes and as the design or development proceeds. Because it is difficult to predict this process or to diagram it, or to predict its costs precisely without using competent engineers, the systems engineering procedures simply ignore the iterative, feedback nature of the real world because the process has been degraded to clerical reporting. To a large extent, this tends to constrain project managers from doing work in the real way toward doing it in a way that fits with their management tools. This is clearly nonsense.

As a specific example, doctrine says that one is to consider the "ilities," that is, maintainability, reliability, operability, etc., from the very beginning of the process. This is a vast waste of time and effort. I do not mean that one should not think about these things at the beginning, but it is certainly ridiculous to have a complete plan for the logistics of the maintenance of an object that has not yet been designed. I have seen overruns in expenditure and unnecessary effort generated by the fact that the linear sequencing of milestones had forced development of a complete maintenance and reliability plan for what was no longer the design, and had not been the design for three months. The machinery forced everyone to grind on and on because, after all, the maintenance and reliability milestones could not be missed without disaster and fear of cancellation of the project, even though the plan being worked out had nothing whatever to do with the hardware being designed.

In fact, the point at which to start serious work on configuration control, maintainability and reliability cannot be very well preplanned; it can be roughly preplanned, but it must be adjusted to be at the point at which the design means something and is likely to stay still long enough so that the redesign for the "ilities" will really make some sense. Judgment, not tools, is what is required.

PREDICTION VS. PRODUCTION

This brings me to a related antithesis that I describe as prediction versus production. We have come to a time when meeting certain targets seems to have become more important than producing a satisfactory system. The question is not the development of a system that performs well and was produced at a reasonable cost and in a reasonable time, but rather replacement of this sensible desire by the question, "Does the system perform as predicted, did you produce it for the cost you predicted, and on the schedule you predicted, and did you do it in the way you predicted?" Consequently, looking at what is actually happening in the development has been replaced by measuring it against a simplistic set of predicted milestones. Fulfillment of prediction has been seriously proposed as *the* criterion for judging system managers. It is certainly a *minor* criterion. Fulfillment of a need when fielded continues to be our real objective.

I know of a number of cases where the pressure on prediction has been so great that the project managers were forced to destroy the possibility of having a good system because they were not allowed to adjust what they were doing to the real world; otherwise, they would have been so far off prediction in one or another dimension that the project would have been canceled. We fell between two stools. We had a system that was only approximately what we wanted and the system failed to meet the prediction. Similarly, we have not had the sense to cancel something that met the predictions, but was no damn good.

A QUESTION OF PREDICTABILITY

It is curious that those of us, sophisticated as systems engineers, and having read history (in which no one ever seems to anticipate what really happens), knowing that the prediction time for random noise seen through a bandpass filter is only about one over the bandwidth, should yet seek predictability for the processes with a wide bandwidth of unknown information. No one can predict politics or economics; few of us predict what happens in our own lives. Why then do we assume the predictability of development of the unknown?

Should we expect development milestones to be met? Presumably, the prior probability of meeting the perfectly chosen milestone on time is distributed randomly and symmetrically about the predicted time. If the accomplishment is relatively simple, the distribution is narrow and this is called "low risk;" if the accomplishment is difficult, the distribution is wide and this is called "high risk." However, all development schedules assume success of each process. If we put trouble contingency time allowances into every task, the total contingency allowance would be unacceptably large and the development unacceptably long. This tends to bias the true risk distribution in such a way as to move the peak to the late side. Thus, there is a tendency for the "risk distribution" to peak after the milestone. The contingency allowance should be provided in an unpopular program element, "allowance for stupidity and the unforeseen." Even so, it probably would be eliminated by the efficient review process.

All I am saying is that we only assess the risk of the predictable problems and that there is always a family of unpredictable problems that make things take longer; there are few ("oh, happy few!") cases of luck that make things take less time. We should not expect milestones to be reached, and they never (or hardly ever) are, although milestones are needed to assure adequate program pressure.

This question and my trial answer suggest a signal-to-noise ratio approach to risk and error assessment in development models. I have not tried to carry this further; it is left as an exercise for the developer.

SYSTEMS IN SPACE VS. SYSTEMS IN SPACE-TIME

My next antithesis I would label "systems in space" versus "systems in space-time." We talk about system design and system choice in terms of ten-year life-cycle costs, but the assumption we tend to make is that the system we are costing is a static object once it is designed and produced. In a way, this is forced upon us by the accountant's formalism of dividing costs into investment and recurring costs. Any system managers who say that they are designing their system in space-time, and that they propose to design it so as to facilitate their ability to change it during the course of the ten-year life cycle, will promptly have their project removed from under them because the doctrine says, "This is terribly uneconomical;" furthermore, it says that it is *bad system design.* I would simply like to note here that real-world history tells us that all systems are changed frequently during their lifetime, if for no other reason than that the real requirements and environments and technologies for them change, often in ways that make it stupid to leave them alone. In fact, it is almost true that no military system is ever used for the precise purpose for which it is designed. Consequently, it makes sense to think about the system as something that will have a history in time and that is likely to require change, and to include some thought of this in the design. Change, strangely, is the only truly predictable attribute of the system. Perhaps I am merely going to be enshrined in the next generation of systems engineering doctrine with a special group in every project organization called "changeability management." I hope not.

The question is not whether there will be changes or not, but whether the change process will be under conscious control. Do the developers know "what" and "why" when they allow or make a change? Pretending that no changes are allowable or desirable is merely a way of losing control of the change process.

An example of the consequences of what I mean follows. It is systems engineering doctrine that the system should be matched throughout; that is to say, it is regarded as poor practice to have, for example, high-reliability components matched with low-reliability components since system reliability will really be set by the low-reliability components whereas system cost is likely to be set by the high-reliability components. This ignores the fact that since the system will have to change in time it may be very sensible to build in high-reliability components in some parts of the system, even though the technology does not provide them for other parts of the system. During the course of the lifetime of the system, there may be a high probability of bringing the low-reliability parts up to an equivalent reliability with the higher-reliability parts for a reasonable cost. Thus the system *could* be designed for great improvement in reliability from the very beginning, whereas if everything is matched to the lower reliability, the cost of improvement becomes gigantic, because the changes are extensive. In fact, the rule of thumb may not be good engineering at all if the system is designed considering change with time. We should design for growth and a process of technological leapfrogging in the system.

OPTIMIZATION VS. UNCERTAINTY

One of the fundamental tenets of systems engineering is that the system should be optimized to its purpose. This is dandy if the purpose is very specifically definable and if it is very independent of scenario and enemy behavior. If these requirements are not true, and they almost never are for any military system of any great sophistication, then optimization may merely be the definition of which catastrophe you want to undergo. My analogy is the matching of a narrow-band filter to a specific signal. This is an elegant engineering procedure, provided you can depend on the signal to stay put. If the enemy, for example, has a slight adjustment in *their* frequency, then optimization in the normal sense rapidly becomes nonsense. There is no sense in optimizing the system beyond the accuracy of the definition of requirements, and I never, or almost never, see a definition of requirements with estimated error limits.

This particular kind of catastrophe is most often generated by the portion of systems engineering that the economists like to call systems analysis. That is to say, having chosen some scenario or problem defined in a very specific way, the system prescription follows optimization of this problem to the bitter and ridiculous end. There is a vast reluctance to look at the difficulties and the risks involved in assuming that the chosen problem is the correct problem. I will feel much better about the use of scenarios and prediction of warfare ten years ahead for system choice and optimization if ever I meet a person who can really predict a chess game, or what will happen in the stock market tomorrow. This is not to say the game should be ruled out just because the results cannot be predicted, but rather to reinforce the fact that it is a game and cannot be taken literally.

There is a procedure called sensitivity analysis, but I have rarely seen it applied to the right parameters and variations. It is usually too difficult to do so. One rarely ever considers an error analysis, even when something is known about the error distributions of the input parameters.

A problem related to this is posed by the analysis of multipurpose objects. A tremendous difficulty is generated by the fact that the costs and characteristics must be allocated to the appearance of the system in several different scenarios. Consequently, these systems must be single solutions to several systems engineering requirements. Our usual way of dealing with this problem is to bow three times in its direction and then ignore it, because it is just too hard to solve. Solving it requires solving the systems problems for all the situations in which the multipurpose system appears, then doing all the (nonlinear) interaction cases.

In addition, the cost allocation to the various uses must be attacked. There is simply no methodology available for really trying this and hence the problem is generally ignored. This makes many of the analyses useless, but that is generally ignored too. There is no sense in pretending to solve problems by refusing to address them realistically because they are too difficult, but we go on playing that game.

OBJECTS VS. OBJECTIVES

Finally, we do not distinguish sufficiently between objects and objectives. The working tools and most of the life of systems engineering are spent trying to reach an objective, the objective finally becoming an object. It is important to keep this distinction in mind. The trouble in procurement of a development is that procurement procedures are designed to buy objects, whereas in development there is no object until the end, only an objective, and the two are not the same thing.

For example, what is a specification? A specification is an abstract set intended to describe what is to be produced, but of course it is only a portion of a *total* description. It is a subset of points selected from a continuous portion of an infinite multidimensional space. The object itself and its total future history is the only complete specification. Consequently, the idea of a "complete" specification is an absurdity; we can only produce a partial subset. In fact, it is possible (and we have all seen it happen) for an object that meets the subset of specification points to badly miss being a sensible solution to the problem, because it departs from the required reality between the specification subset points. I hasten to add that sometimes even the object itself, without regard to its future history, is not a sufficient specification, because it does not contain the details of the techniques used to produce it. Let the specifier beware!

Having complained about all of this throughout this article, what do I propose? The only thing I know that works is to obtain a competent person and assistants, and make sure they understand the problem—not the specifications of the problem, not the particular written scenario, but what is really in the minds of those who have a requirement to be solved. Then give them funds, a good choice of managerial and systems engineering tools, and let them work at the problem after reasonably frequent conferences with those who have the requirement.

In this way, the end object may become the best that both parts of the system can produce and not merely the solution to a paper problem, said solution having the best paper properties to match the previous set of paper. (Some paper is water soluble.)

It might do well to bear in mind the following closing thoughts:

- As we are now behaving, we are using up our best people in filling out documentation for their superiors to read, and most of the time no one is running the store.
- We have lost sight of the fact that engineering is an art, not a technique; a technique is a tool. From time to time I am briefed on the results of a systems analysis or systems engineering job in a way that prompts me to ask the questions: "That's fine, but is it a good system? Do you like it? Is it harmonious? Is it an elegant solution to a real problem?" For an answer I usually get a blank stare and a facial expression that suggests I have just said something really obscene.

We must bring the sense of art and excitement back into engineering. Talent, competence, and enthusiasm are qualities of people who can use tools; the lack of these characteristics usually results in people who cannot even be helped by techniques and tools. We can all do better.

THE SYSTEMS ENGINEERING OVERVIEW AND PROCESS
(FROM THE SYSTEMS ENGINEERING MANAGEMENT GUIDE [1990])

Defense Systems Management College

The past several decades have seen the rise of large, highly interactive systems that are on the forward edge of technology. As a result of this growth and the increased usage of digital systems (computers and software), the concept of systems engineering has gained increasing attention. Some of this attention is no doubt due to large program failures which possibly could have been avoided, or at least mitigated, through the use of systems engineering principles. The complexity of modern day weapon systems requires conscious application of systems engineering concepts to ensure producible, operable and supportable systems that satisfy mission requirements.

Although many authors have traced the roots of systems engineering to earlier dates, the initial *formalization* of the systems engineering process for military development began to surface in the mid-1950s on the ballistic missile programs. These early ballistic missile development programs marked the emergence of engineering discipline "specialists" which has since continued to grow. Each of these specialties not only has a need to take data from the overall development process, but also to supply data, in the form of requirements and analysis results, to the process.

A number of technical instructions, military standards and specifications, and manuals were developed as a result of these development programs. In particular, MIL-STD-499 was issued in 1969 to assist both government and contractor personnel in defining the systems engineering effort in support of defense acquisition programs. This standard was updated to MIL-STD-499A in 1974, and formed the foundation for current application of systems engineering principles to military development programs.

In its simplest terms, systems engineering is both a technical process and a management process. To successfully complete the development of a system, both aspects must be applied throughout the system life cycle. From a government's program management point of view, the Defense Systems Management College favors the management approach and defines systems engineering as follows:

> Systems engineering is the management function which controls the total system development effort for the purpose of achieving an *optimum balance of all system elements.* It is a process which transforms an operational need into a description of system parameters and integrates those parameters to *optimize the overall system effectiveness.*

A system life cycle begins with the user's needs, expressed as constraints, and the capability requirements needed to satisfy mission objectives. Systems engineering is essential in the earliest planning period, in conceiving the system concept and defining system requirements.

As the detailed design is being done, systems engineers: 1) assure balanced influence of all required design specialties, 2) resolve interface problems, 3) conduct design reviews, 4) perform trade-off analyses, and 5) assist in verifying system performance.

During the production phase, systems engineering is concerned with: 1) verifying system capability, 2) maintaining the system baseline, and 3) forming an analytical framework for producibility analysis.

During the operation and support (O/S) phase, systems engineering: 1) evaluates proposed changes to the systems, 2) establishes their effectiveness, and 3) facilitates the effective incorporation of changes, modifications and updates.

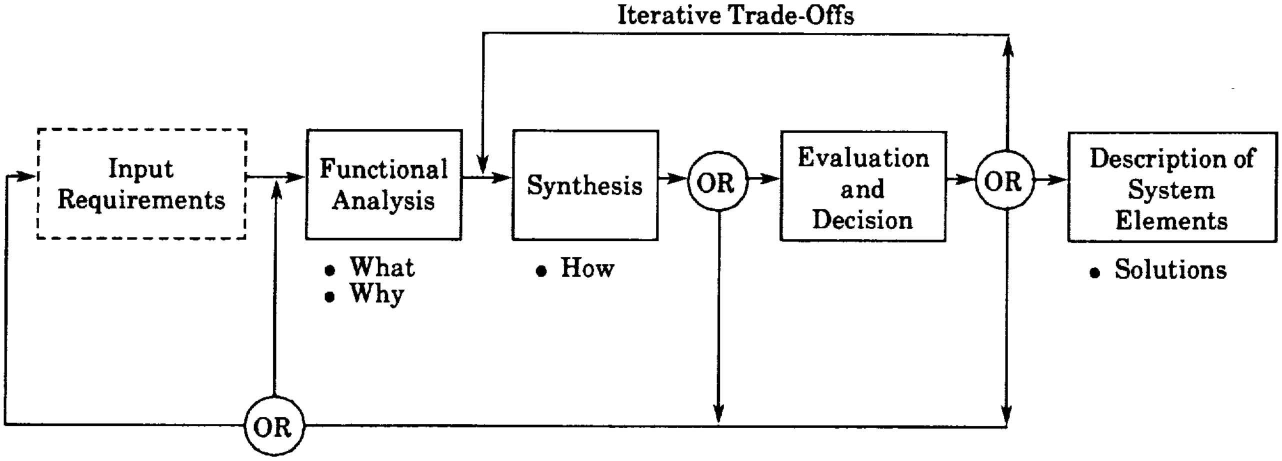


Figure 1 The Systems Engineering Process

THE SYSTEMS ENGINEERING PROCESS

Although programs differ in underlying requirements, there is a consistent, logical process for best accomplishing system design tasks. Figure 1 illustrates the activities of the basic systems engineering process.

The systems engineering process is iteratively applied. It consists primarily of four activities: functional analysis, synthesis, evaluation and decision, and a description of systems elements. The product element descriptions become more detailed with each application and support the subsequent systems engineering design cycle. The final product is production-ready documentation of all system elements.

Since the requirement to implement a systems engineering process may cause major budgetary commitments and impact upfront development schedules, it is important to understand the inherent objectives:

- Ensure that system definition and design reflect requirements for *all* system elements: equipment, software, personnel, facilities and data.
- Integrate technical efforts of the design team specialists to produce an *optimally balanced design.*
- Provide a comprehensive indentured framework of system requirements for use as performance, design, interface, support, production and test criteria.
- Provide source data for development of technical plans and contract work statements.
- Provide a systems framework for logistic analysis, integrated logistic support (ILS), trade studies and logistic documentation.
- Provide a systems framework for production engineering analysis, producibility trade studies, and production and manufacturing documentation.
- Ensure that life cycle cost considerations and requirements are fully considered in all phases of the design process.

Successful application of systems engineering requires mutual understanding and support between the military and contractor program managers. They must be willing to make the systems engineering process the backbone of the overall development program. They must understand the need to define and communicate among the engineering specialty programs. They must recognize the role of formal technical reviews and audits, including the value, objectives and uniqueness of each formal review and audit. They must also know the objectives of the program and possess a thorough interpretation of the user's requirements.

The output of the systems engineering process is documentation. This is the means by which it controls the evolutionary development of the system. Systems engineering prepares a number of technical management and engineering specialty plans that define how each phase of the acquisition cycle will be conducted. Draft plans are usually submitted with the proposal and final plans are delivered in accordance with the Contract Data Requirements List (CDRL). These plans are used by the government to ensure compliance with the contract and used by the contractor to develop detailed schedules and allocation of resources. Specifications are submitted that form the basis for the design and development effort. Top-level specifications are incorporated into the statement of work (SOW) and provided to the developer. The developer will allocate these top-level requirements to lower level system components (hardware and software) and submit the associated specifications and design documents to the government for approval. The status of system development progress is tracked and documented in the form of technical review data packages, technical performance measurement (TPM) reports, analysis and simulation reports and other technical documentation pertinent to the program. In summary, this documentation may include:

- Systems Engineering Management Plan (SEMP)
- Specifications (system, segment, development, product, process, material)
- Design Documentation
- Interface Control Documents (ICDs)
- Risk Analysis Management Plan
- Survivability/Vulnerability (S/V) Hardness Plan
- Mission Analysis Report
- Reliability Plan
- Maintainability Plan
- Integrated Logistics Support Plan (ILSP)
- Software Development Plan (SDP)
- Test and Evaluation Master Plan (TEMP)
- Producibility Plan
- Functional Flow Block Diagrams (FFBD)
- Requirements Allocation Sheets (RAS)
- Audit Reports
- EMI/EMC Control Plan
- Human Engineering Plan
- Trade Study Reports

The systems engineering process is an iterative process applied throughout the acquisition life cycle. The process itself leads to a well defined, completely documented and optimally balanced system. It does not produce the actual system itself, but rather, it produces the complete set of documentation, tailored to the needs of a specific program, which fully describes the system to be developed and produced. Each program's systems engineering process, developed through tailoring and/or adding supplemental requirements, must meet certain general criteria. Although not complete, the following guidelines should be considered in approaching the basic process:

- System and subsystem (configuration item) requirements shall be consistent, correlatable, and traceable both within data produced as basic documentation (e.g., Functional Flow Block Diagram, Requirements Allocation Sheet, and Time Line Sheet) and as related documentation (e.g., work breakdown structure and Logistic Support Analysis Record).
- The concept of minimum documentation shall be evident.
- Acquisition and ownership cost shall be an integral part of the evaluation and decision process.
- Baselines shall be established progressively as an integral part of the systems engineering process.
- The systems engineering process shall result in a design that is complete, at a given level of detail, from a total system element viewpoint.

- The process shall provide for the timely and appropriate integration of mainstream engineering with engineering specialties such as reliability, maintainability, human factors engineering, safety, integrated logistic support, environmental assessments and producibility to ensure their influence on system design.
- The process shall provide for continuing prediction and demonstration of the anticipated or actual achievement of the primary technical objectives of the system. Problems and risk areas shall be identified in a timely manner.
- Formal technical reviews and audits shall be an integral part of the systems engineering process.
- The systems engineering process shall be responsive to change. The impact of changes to system and/or program requirements must be traceable to the lowest level of related documentation in a timely manner.
- Significant engineering decisions shall be traceable to the systems engineering activities and associated documentation upon which they were based.

Figure 2 is an overview of the four basic steps of the systems engineering process.

FUNCTIONAL ANALYSIS

Every engineering effort must begin with a statement of a perceived need. At the beginning of a DOD acquisition effort, this statement will be in the form of a system requirement document, usually developed through a Mission Area Analysis of anticipated threats.

Once the purpose of the system is known, the functional analysis activity identifies what essential functions the system must perform. In order to accomplish this, functional analysis is composed of two primary process segments: functional identification and requirements identification and

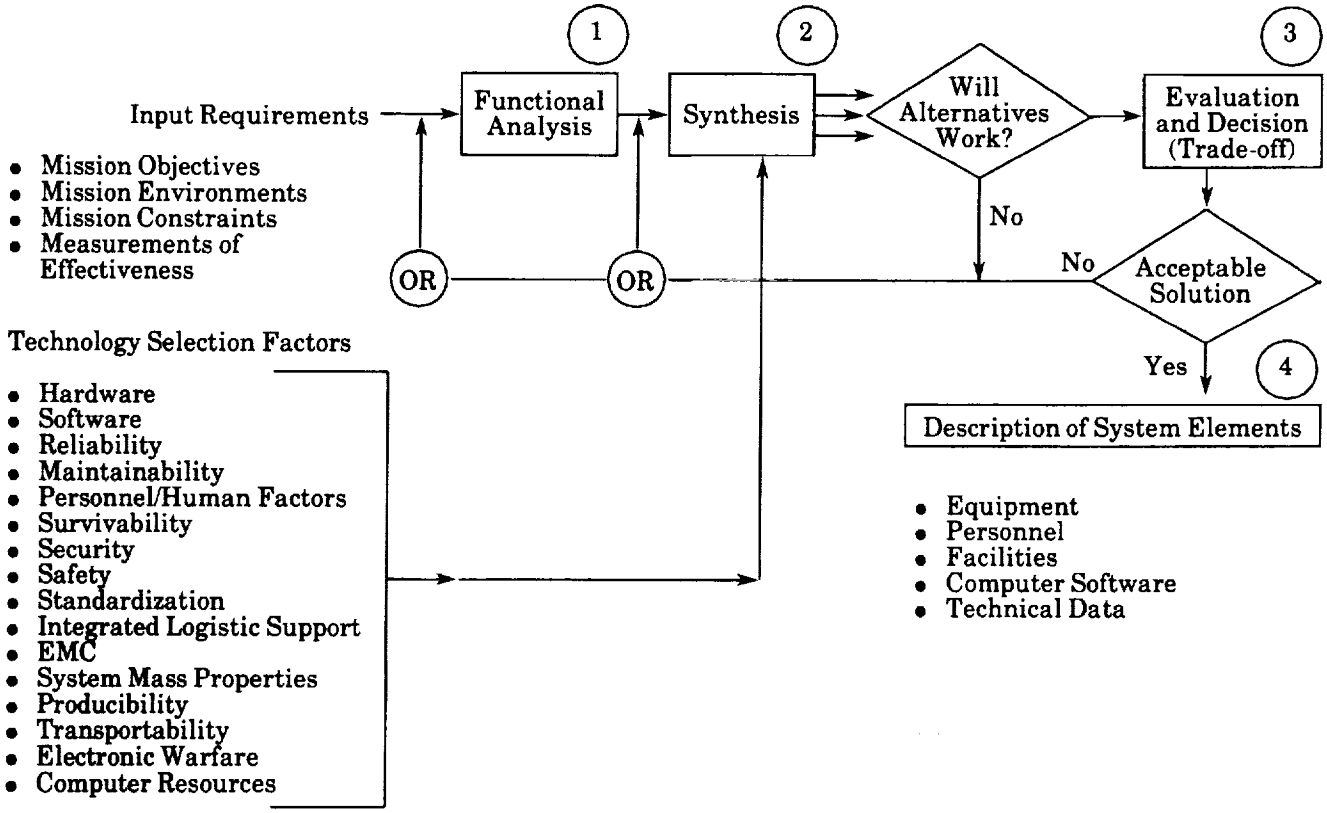


Figure 2 The Systems Engineering Process

allocation (functional performance requirements analysis). It answers the "what" and "why" questions relative to system design.

The basic analytical tool for functional identification is the Functional Flow Block Diagram (FFBD), showing logical sequences and relationships of operational and support functions at the system level. Specific functions will vary from system to system and will be traceable to mission requirements and objectives. Maintenance flow diagrams depicting general maintenance and support concepts will lead to analysis of requirements on an end item/equipment basis. At this level, since functions are more standardized, functional identification is often accomplished using the End Item Maintenance Sheet (EIMS) or Logistic Support Analysis Record (LSAR). Similarly, detailed test requirements are identified using the Test Requirements Sheet (TRS), and productivity requirements are identified using the Production Sheet (PS).

It should be kept in mind that the systems engineering process is always iterative. Each acquisition phase will involve functional analysis to progressively more detail. For example, during the Concept Exploration/Definition (C/E) phase, analysis of support functions will concentrate on Maintenance FFBDs, which will support the establishment of gross maintenance concepts. During Full Scale Development (FSD), emphasis will shift to detailed analysis of the maintenance requirements of specific equipment using the EIMS or LSAR.

The Requirements Allocation Sheet (RAS) is used as the primary analytical tool for requirements identification and allocation, or functional performance requirements analysis as it often is referred to, in conjunction with FFBDs and special purpose documents such as EIMSs, TRSs, and PSs. The RAS serves three purposes in documenting the systems engineering process: 1) initially, it is used to record the performance requirements established for each function; 2) during synthesis, it is used to show the allocation of the functional performance requirements to individual system elements or a combination of elements; and 3) following evaluation and decision, the RAS provides the functionally oriented data required in the description of the system elements.

The Time Line Sheet (TLS) is used to perform and record the analysis of time-critical functions and functional sequences In performing time requirements analysis for complex functional sequences, additional tools, such as mathematical models or computer simulations, may be needed. Time requirements analysis is performed in any or all of the functional cycles of the process to determine whether time is a critical factor. The TLS complements the FFBD in its ability to show a lower level of detail, as well as to illustrate the impact of concurrent functions within a given sequence. TLSs are used to support the development of design requirements for the operation, test and maintenance functions. They identify time-critical functions and depict the concurrency, overlap and sequential relationship of functions and related tasks. Time-critical functions are those that affect reaction time, downtime or availability.

SYNTHESIS

Synthesis supplies the "how" answers to the "what" outputs of functional analysis.

Two documentation tools accomplish and record the synthesis of design approaches or alternative approaches. The Concept Description Sheet (CDS) is used to collect the performance requirements and constraints, as delineated by functional analysis, that apply to an individual subsystem or end item. The CDS also describes at the gross level a design approach for meeting the requirements. The Schematic Block Diagram (SBD) is used to develop and portray the conceptual schematic arrangement of system elements to meet system and/or subsystem requirements. The CDS and SBD are both applicable to all acquisition phases and provide the basis for development of the descriptions of system elements.

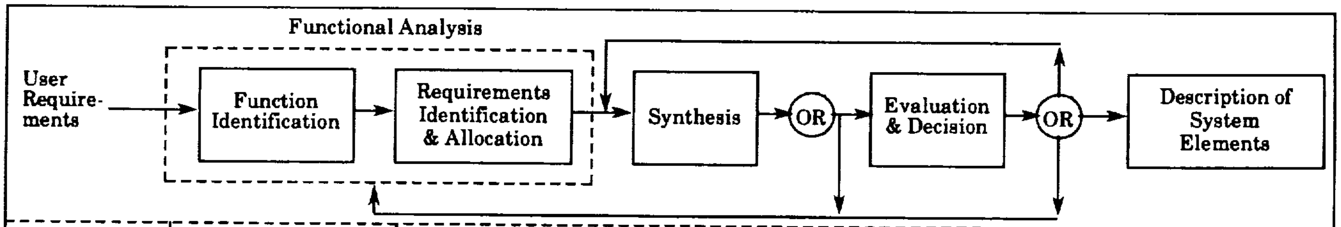
EVALUATION AND DECISION

Since program risk and cost are dependent on practical trade-offs between stated operating requirements and engineering design, continual evaluations and decisions must be made not only at the beginning of the program but throughout the design and development activity.

The Trade Study Report (TSR) is used to summarize and correlate characteristics of alternative solutions to the requirements and constraints that establish the selection criteria for a specific trade study area. The report also documents the rationale used in the decision process and should present risk assessment and risk avoidance considerations. Other tools, such as analytical or mathematical models or computer simulations, may be needed and used in accomplishing the evaluation and decision process.

DESCRIPTION OF SYSTEM ELEMENTS

All systems can be defined by a set of interacting system elements which fall into five categories: equipment (hardware), software, facilities, personnel, and procedural data.



Basic Documentation	Functional Flow Block Diagrams (FFBD) Identify and sequence functions that must be accomplished to achieve system or project objectives. Develop the basis for establishing intersystem functional interfaces and identify system relationships.	Requirements Allocation Sheets (RAS) Define the requirements and constraints for each of the functions and relate each requirement to the system elements of a. Equipment b. Facilities c. Personnel d. Procedural data e. Computer software Time Line Sheets (TLS) Present critical functions against a time base in the required sequence of accomplishment.	Concept Description Sheets (CDS) Constrain the designer to stop at a point in the cycle and create at the gross level a design or synthesis meeting the FFBD, RAS, TLS requirements and constraints. Schematic Block Diagrams (SBD) Develop and portray schematic arrangement of system elements to satisfy system requirements.	Trade Study Reports (TSR) Select, evaluate and optimize promising or attractive concepts. Document the trade-off and supporting rationale. Consider all possible solutions within the framework of requirements.	Design Sheets (DS) Define, describe and specify performance, design and test criteria for the system elements a. Equipment b. Facilities c. Personnel d. Procedural data e. Computer software
Special Purpose Documentation	End Item Maintenance Sheet (EIMS)/Test Reqmt Sheet (TRS)/ Production Sheets (PS)/Logistic Support Analysis Record (LSAR) Identify maintenance, test and production functions on a specific end item, subassembly, and component basis.				Facility Interface Sheets (FIS) Identify environmental and physical interfaces between equipment and facilities on an end item basis.

Indenture is carried to the level required for the selected level of engineering to identify, describe and specify.

Figure 3 Basic and Special Purpose Documentation for Systems Engineering

Two documentation forms are used to describe these system elements: the Design Sheet (DS) and the Facility Interface Sheet (FIS). The DS is used to establish and describe the performance, design and test requirements for equipment end items, critical components and computer software programs. The FIS is used to identify the environmental requirements and interface design requirements imposed upon facilities by the functional and design characteristics of equipment end items. The DS and FIS provide the basis for the formal identification required for configuration management.

The systems engineering process produces the basic and special purpose documentation that controls the evolutionary development of the system. Figure 3 correlates the particular documentation associated with each step of the systems engineering process.

The systems engineering process itself does not actually produce the system, but produces the documentation necessary to define, design, develop and test the system. As such, a variety of engineering and planning documentation is required throughout the acquisition cycle, and systems engineering is the vehicle used to produce that documentation.

Numerous plans are prepared to define which technical activities will be conducted. They address the integration of engineering specialties requirements, "design-for" requirements and organizational resource requirements, and discuss how progress toward system-level goals will be measured. The Systems Engineering Management Plan is the key planning document that reflects these requirements. Contractor compliance with these plans is monitored by government organizations to ensure that standard policies and procedures in the area of systems engineering are employed. Additionally, specifications are prepared as part of the-systems engineering process to form the basis for the design and development effort. The top-level specification (system or segment) is normally approved and draft lower level specifications (configuration items) are developed reflecting allocated system requirements to lower level components or subsystems, which designers and subcontractors translate into hardware and software production plans.

In order to provide a continuing assessment of the system's capability to meet performance requirements, the systems engineering organization prepares technical review data packages, technical performance measurement (TPM) reports, analysis and simulation reports, and other documentation.

The systems engineering process is one approach to providing disciplined engineering during all acquisition phases. Although current application of the process has focused on C/E, D/V, and FSD, systems engineering process techniques and principles are equally applicable to the analysis and definition of production requirements.

The systems engineering process also provides the logic and timing for a disciplined approach, with certain internal assurances of technical integrity such as traceability. Technical integrity ensures that the design requirements for the system elements reflect the functional performance requirements, that all functional performance requirements are satisfied by the combined system elements, and that such requirements are optimized with respect to system performance requirements and constraints.

The DSMC Systems Engineering Management Guide may be purchased from the U.S. Government Printing Office (1991-306-417-QL 3).

SYSTEMS ENGINEERING FOR VERY LARGE SYSTEMS

by Paul E. Lewkowicz

Very large integrated systems have always posed special problems for engineers. Whether they are power generation systems, computer networks or space vehicles, whenever there are multiple interfaces, complex technologies or just demanding customers, the challenges are unique. "Systems engineering" has evolved as a discipline in order to meet these challenges by providing a structured, top-down design and development methodology for the engineer. This paper attempts to define the general class of problems requiring the complete systems engineering treatment and to show how systems engineering can be utilized to improve customer satisfaction and profitability. Specifically, this work will focus on a design methodology for the largest of systems, not necessarily in terms of physical size, but in terms of complexity and interconnectivity.

The literature has generally defined "systems engineering" as in this quote from W.P. Chase in *Management of System Engineering*:

> [Systems Engineering is] the process of selecting and synthesizing the application of . . . knowledge in order to translate system requirements into a system design and . . . to demonstrate that [it] can be effectively employed as a coherent whole to achieve some stated goal or purpose.

This definition points out, in the most general terms, that systems engineering is a process for ensuring that the customer requirements are satisfied. What it also implies is that this satisfaction must be achieved on time and for the agreed-upon price. It is this implicit requirement that is most often unfulfilled in complex engineering projects.

Recent efforts at Hughes Aircraft Company's Space & Communications Group have focused on sharpening the definition of systems engineering and defining standards for improving the implementation of the full systems engineering methodology on large spacecraft programs. Since these programs typically cost in the $100 million range, the pressure to deliver specified performance on time and on budget is enormous. A casual review of programs within the author's experience has shown that the classical approach to systems engineering has been followed throughout, but with varying uniformity and overall success. The question to answer, in the context of even more advanced, more demanding projects, is: "How can it be done better?"

The "classical" method of systems engineering alluded to above consists of requirements definition, technology assessment, solution synthesis and performance verification: four successive steps in the design of the mission solution. Typically, this is an iterative process, since requirements and technology rarely remain static. The customer's mission can be altered by events or even by a better understanding of the technology, risks or costs involved. Synthesized solutions, too, depend on the technology available, as well as the question asked. Often, the proposed technology does not live up to expectations, resulting in a "new" solution and reverification: an embarrassing situation at best, an extremely costly one at worst.

When the verification (or testing) phase of the systems engineering process uncovers a fault, the cause can often be traced to incomplete or improperly stated requirements. An example of this fact is a problem uncovered on one particular series of satellites; an on-orbit failure resulted in the loss of some 16 channels of telemetry data. The failure analysis, performed by the program's

© *1988 IEEE. Reprinted with permission of the author, from IEEE Aerospace Application Conference, Park City, Utah, February 1988*

systems engineering staff, identified the cause as an open circuit in a particular unit. This fault produced an abnormally high telemetry output signal on one channel, which in turn resulted in the degradation of all 16 inputs to the telemetry multiplexer. Had systems engineering levied a requirement to protect against failure-induced overvoltages (via a simple circuit redundancy technique at the unit), only the failed telemetry channel would have been lost, instead of that of 15 other units as well.

The point here is that it is a knowledge of the needs of the whole system that is required, instead of only the needs of the parts. This knowledge exemplifies the principle of "engineering leverage" whereby a few engineers, representing a broad experience base, performing the logical, methodical systems design work, can save money over trial and error or crisis-oriented engineering. It is the concentration of systems knowledge, the "big picture" view, that allows for efficient designs all through the system.

A common question is: "How much systems engineering is required for a given project?" This can usually be interpreted as "How much will this cost?" Clearly a design team with unlimited funds can perform complete requirements analysis, all manner of failure analysis and simulations, and extensive part and unit environmental testing to fully optimize the design of some particular product. But if that product is, say, a ballpoint pen, have they really made it better from the manufacturer's standpoint? Or have they succeeded in making the most expensive writing instrument the world has ever known? The application of systems engineering techniques to a project is a matter of appropriate degree; how much engineering is required to ensure the customer's satisfaction becomes the first question any organization must ask before they can set up a systems engineering program.

This example emphasizes the fact that systems engineering costs are a direct charge to the effort, so the total cost of the engineering must be distributed over the entire production run. Even if the run is large, as in the ballpoint pen case, when the product normally sells for 39 cents, if the engineering costs run into the millions, then the manufacturer could be in serious trouble. For smaller production runs, like a satellite or submarine contract, systems engineering costs can still drive the final sale price, but systems engineering can also reduce the price by preventing errors and rework.

THE SYSTEMS ENGINEERING METHODOLOGY

The procedure followed in systems engineering consists of four distinct phases, described here in the simplest terms: requirements definition, technology assessment, solution synthesis and performance verification. These sobriquets are intended to be mnemonic; the details of what they really signify are presented below.

Requirements Analysis. The initial step consists of defining the problem to be solved and the constraints on the solution set. This is perhaps the single most critical phase of the systems engineering process in that a misunderstanding of the problem to be solved, either in characterizing it or defining the context of the solution, can result in an erroneous conclusion. As in the satellite telemetry example, the customer can be somewhat less than satisfied when a partial solution is delivered.

In large systems, the problem definition is usually described by the contractual documents. The request for proposal (RFP) or the statement of work typically contains directives as to the overall mission of the system, but these are not always completely specific; some interpretation of what the customer really meant is often required.

Another aspect of requirements analysis often underappreciated is that of constraining the solution. The RFP for a program may state that only a certain rocket booster or parts of a specific grade can be used, but the implications of such statements, and especially the implications of the "unstated" or "implied" requirements, can have serious consequences in the final design. These requirements, sometimes called derived or secondary requirements, determine the limits of the parametric trades that can be made in characterizing the problem's solution.

Technology Assessment. Once the basic requirements, both primary and secondary, are in place and understood by the design team, the technology available to solve the problem can be examined for suitability. This step is intuitively obvious for small systems, but when complexity is high, making the appropriate choice is not always easy. Typical activities in this phase include comparative tradeoffs between different processes and materials, architectures and performance. The technology assessment phase may also consider the design and documentation methods and the management organization to be employed on a specific project. Overall, this phase is concerned with selecting the best tools for performing the system design.

Solution Synthesis. This is usually the most time-consuming step in engineering a system to perform complex tasks and meet stringent requirements simply because of the number of choices available. If the requirements are well understood and the available hardware and software appropriate to the task are known, then trade studies can be carried out (on paper) that result in myriad viable combinations. During this phase, compromises are often required in order to satisfy conflicting requirements. For example, in a communications system design, a large antenna may be desired to provide high gain, but this will reduce its coverage capability by reducing the beamwidth. Out of this sea of alternatives must come a single "best fit" solution, meeting all of the original and derived requirements, especially such items pertaining to cost and producibility. If it can't be built or bought, then it's not the right answer.

Performance Verification. Last, but definitely not least, is the performance verification or testing phase. The task here is to prove, with all the rigor possible, that the suggested solution does in fact meet all of the system requirements in a clearly documented way. A standard approach is to utilize specification trees and a verification matrix to show where each requirement from the original customer's source documents is captured in lower level specifications. Additionally, the verification matrix shows how compliance with the requirement is proven, either by inspection, test, demonstration or analysis. In general, the specification system is designed to show a clear, unambiguous flowdown of all system requirements into individual component designs. The verification phase is the test of this flowdown as well as a measure of system performance.

REQUIREMENTS FOR SUCCESSFUL SYSTEMS ENGINEERING

The foregoing text has all been a precursor to this: exactly what does an organization have to do to apply a full-scale systems engineering approach to their work? And, perhaps more importantly, what does it cost that organization? As expected, in systems engineering, as in life, there are no free lunches. This section details the inputs to the process, or what is required by a systems engineering organization in order to function properly.

Formality. First and foremost, a formal, planned approach to the systems engineering process must be in place. Not only must the "generic" methodology for systems engineering be understood by all involved, the detailed program plans for the specific application of systems engineering must reflect this commitment. The major components in the formal system are review procedures, specification generation and maintenance (or "configuration control") procedures, and planning.

As can be deduced from the discussion of the phases of the systems engineering process, some degree of review and checking is inherent to all operations. The establishment of specification and design review teams to examine the documents (e.g., specifications, trade study reports, etc.) and help polish them into complete and correct inputs to the final design cannot be avoided. Without concrete review milestones, the design will often wander and become unfocused with respect to its objectives, which results in inefficient time and money management.

Since the specifications define the problem to be solved and its constraints, it is clear that they must be reliable and well documented. The configuration control function is to provide a routine for the introduction, validation and documentation of new requirements and the updating of old ones within the system. This is an important step in the review process, as well as the design process, in that all parties (customer and contractor alike) need a stable, well-defined basis of judgment for the validation of the system.

Planning is mentioned last in this case only for emphasis: without complete planning for the entire system design effort, from requirements definition through systems engineering, production, and final deployment, the project is doomed to failure. Every management textbook in the world expounds this fact in detail, yet weak planning is still a major cause of cost overruns and poor performance in all types of industry.

Information Exchange. While formality and procedure allow tight control of the requirements, informality and open communications are the key to efficient design and problem resolution. Not only must the contractor communicate effectively with the customer, but the various elements of the contractor's organization (management, system engineers, unit designers, etc.) must all talk to each other in order to completely understand the requirements. In every program there are stated goals and hidden goals, real requirements and perceived requirements; it all depends on where the observer is looking from. Communications and open channels between all participants, regardless of title or rank, are absolutely essential to all phases of the job.

Technology Base. "Technology" in this context means more than the hardware and software that can be employed in a design solution; it encompasses the organizations and information architectures as well. As a system becomes larger and more complex, so too does the technology or "knowledge base" required to fully define the implementation of system requirements. Such a base might include other contractors, national resources (e.g., the Space Transportation System), specialized electronic devices, etc. In short, practically any conceivable problems, and even a few inconceivable ones, can come up in systems design. To deal effectively with them, the systems engineering team must have the knowledge and experience to recognize solutions from a wide selection of possibilities.

Dedication and Staffing. Finally, the one factor that takes system engineering from an abstract concept to a practical reality is the dedication of the people involved. In order to even begin a design for a complex system, a design team is required. Not a single guru and a few part-time acolytes, but a team of committed managers and engineers with experience in real-world problem solving, technical breadth and clearly defined roles in the systems engineering process. Without this core team, the continuity and rigor required by the process to ensure a coherent, effective solution cannot possibly exist.

Just as planning is the key to a successful project, leadership is the key to a successful team. The complexity of the designs under discussion are such that (typically) a wide range of talents are needed to arrive at a solution. This diversity can be dangerous without direction, because diversity is just a polite name for chaos waiting to happen. A group with a broad technical background, when presented a problem without leadership, will always seek to maximize its entropy. The project staff must be directed and focused at all times in order to move through the systems engineering process. After all, efficiency and minimal engineering costs concern the entire group. The depth necessary to perform the detailed designs need not come from the systems staff, however; this is often not possible given the generalist nature required of them. Most companies employ a unit engineering staff to design the components of the complete solution, which simply reflects the top-down design approach of breaking each requirement down into smaller and smaller functional blocks.

An important factor to consider is time. It may take several months or even years to complete the design of a complex system, so continuity becomes a factor in the staffing of the design team. The deleterious effects of change on an organization are well known, and so are those of miscommunication. The training of systems engineers, whether through formal schooling or on-the-job education, is the first step toward building a self-perpetuating, self-replicating design methodology. Experienced staff members are able to produce more and overcome obstacles better than those less experienced; reinventing the wheel is avoided. Additionally, experienced people add synergy to the team by virtue of shared experiences. Synergism in the design process is how the engineering leverage of systems engineering is released, by the magnification of individual efforts. A fringe benefit of this magnification is growth in the individuals involved. The less experienced become more experienced and leadership skills are developed and honed. Not only does the design process (and product) continue to improve but, through continuity and growth, the staff benefits personally as well.

What about the individual roles of the staff members? The need for a broad knowledge base, for generalists, is clear, but what do they do? As in any team-building situation, all members need clearly communicated job descriptions and management expectations; this applies to all members of the project team from the most senior manager to the last clerk. Once the work has started, they need tangible feedback on what is going correctly, according to expectations, and what is not. The immediate benefit to the organization is clear. Job satisfaction increases, and with it, a concomitant rise in overall productivity. Again, the process, when properly managed, feeds upon itself to work more efficiently.

COST VS. BENEFITS OF FULL-SCALE SYSTEMS ENGINEERING

The requirements levied upon systems design for very large projects are simple: provide full customer satisfaction on time and on budget for a set of diverse and complex functional specifications and interconnections. Likewise, the technology appropriate to this task is (hopefully) equally clear: employ a formal, full-scale systems engineering approach to meeting this challenge.

Costs:

- Management must be willing to allow group synergy to make decisions; the "group think" approach is mandatory.
- Personnel must be dedicated and immersed in the systems engineering of a single system. Teamwork and continuity must be fostered and preserved.
- The systems engineering organization can exhibit all the negative aspects of a bureaucracy if not managed precisely.
- Careful, rigorous planning is required for all aspects of the program up-front, before the work begins, which often means extra bidding expense.

Benefits:

+ Customer satisfaction is enhanced through demonstrated performance and the opportunity for full customer involvement in the design process.
+ Manageability is improved by accurate, more complete planning and a well-defined staff structure.
+ Contingencies are worked out in advance, resulting in fewer surprises during the design, test and production phases.
+ Bettar cost performance is achieved due to reduced redesigns, reworks and "patches."

After an analysis of the costs and benefits of implementing a systems engineering solution to a complex design problem, it becomes apparent that the benefits outweigh the costs, especially in terms of the potential for productivity and cost improvements. The chief drawback of this method is that it is difficult to implement in organizations that do not already practice some form of systems engineering, due to the cultural adjustments that are often necessary. Once the need for a rigorous design methodology is apparent, the systems engineering process of requirements analysis, technology assessment, solution synthesis and performance verification can be utilized to provide an efficient, cost-effective solution to the managerial and technical challenges.

The author wishes to thank Dr. Thomas A. Brackey, W. Richard Brown, and Gávien Miyata of the Hughes Aircraft Company for their support and mentorship in several complex design projects.

REFERENCES

Chase, W.P. Management of *System Engineering,* as quoted in Hughes, Seminar.

Defense Systems Management College. *Systems Engineering Management Guide,* U.S. Air Force, 3 October 1983.

Hughes Aircraft Company. *Systems Engineering Seminar for General Motors,* internal memorandum, 1987.

-----. "S&CG Practice 5-0-53," internal memorandum, 21 July 1987.

-----. "Systems Engineering Division Mission, Goals, and Objectives," internal memorandum, 8 October 1987.

IEEE *Standard Dictionary of Electrical and Electronics Terms,* IEEE Press, Third Edition, 1984 (ANSI/IEEE Std 100-1984).

IEEE Spectrum special report, *On Good Design,* Volume 24, Number 5, May 1987.

U.S. Government MIL-STD-499.

WHAT IS A SYSTEM? NASA'S PHASED PROJECT DESCRIPTION

From the MSFC Systems Engineering Handbook (1991)

Systems engineering is defined in MIL-STD-499A as

> . . . the process(es) required to transform an operational need into a description of system performance parameters and a system configuration through the use of an iterative process of definition, synthesis, analysis, design, test and evaluation. It includes the integration of related technical parameters and ensures compatibility of all physical, functional, and program interfaces in a manner that optimizes the total system definition and design. In addition, systems engineering integrates reliability, maintainability, safety, survivability, and other such efforts into the total engineering effort to meet cost, schedule and technical performance objectives. (*Engineering Management*, May 1, 1974)

Systems engineering is a continuous, iterative process that has a built-in feedback mechanism. It is used throughout a project or program's life cycle to arrive at the best system architecture and design possible. Just when systems engineering began to be practiced as a separate discipline is open to debate, but there seems to be general agreement that formal recognition and definition of the process started after World War II. Large, complex post-war development projects such as the first U.S. ballistic missiles and NASA's Apollo program exhibited the characteristics which created the need for systems engineers.

Among these project characteristics are:

- Large design teams with many highly specialized designers
- Many contractors involved, widely separated geographically, complicating communications
- Many hardware and software systems in concurrent development
- Complex operational and logistic support requirements
- Constrained development time
- High level of advanced technology (*Systems Engineering Management Guide*, U.S. Government Printing Office, 1986).

There are many definitions of a system. Two of these are listed below:

- A system is a set of interrelated components working together toward some common objective. (Blanchard, Benjamin S. and Fabrycky, Wolter J., *Systems Engineering and Analysis*, Prentice Hall, Inc., 1990)
- A system is a grouping of parts that operate together for a common purpose. For example, an automobile is a system of components that work together to provide transportation. An autopilot and an airplane form a system for flying at a specified altitude. (Forrester, Jay W., *Principles of Systems*, Wright-Allen Press Inc., 1968).

Systems engineering is a cyclical process as depicted in Figure 1. The terms shown in this figure are explained in the following paragraphs.

1. *Project and Mission Requirements/ Need Definition* can also be termed as "customer engineering." It is the process by which the needs of the customer (the principal investigator or other significant parties, such as Congress or other budgetary authority) are determined. This allows the systems engineer to define requirements for a system that will meet the needs of the customer.

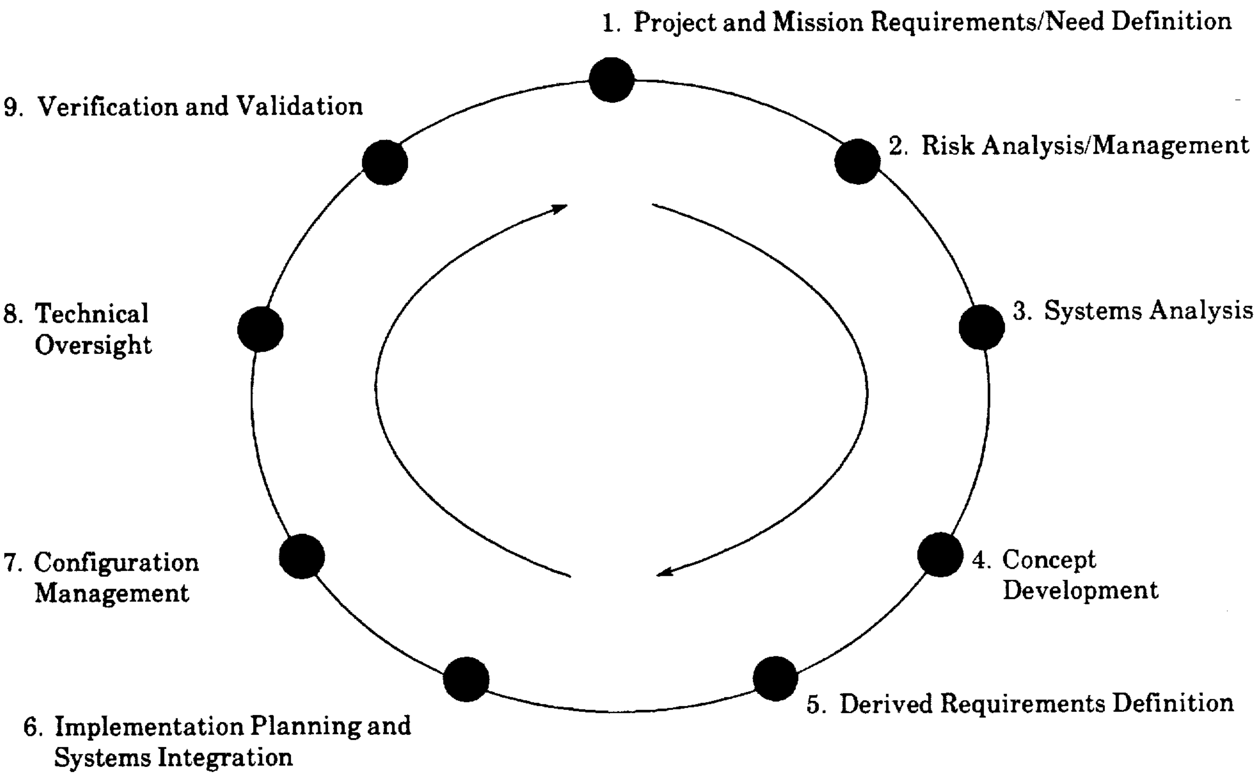

1. Project and Mission Requirements/Need Definition
2. Risk Analysis/Management
3. Systems Analysis
4. Concept Development
5. Derived Requirements Definition
6. Implementation Planning and Systems Integration
7. Configuration Management
8. Technical Oversight
9. Verification and Validation

Figure 1 Systems Engineering Cycle

2. *Risk Analysis/Management* is a continuing process to identify and assess the risks involved with the development and operation of the system. These include technical, schedule, cost and organizational risks. Following the identification of the risks involved, the system engineer then develops an implementation plan to control and, if possible, reduce risks.

3. *Systems Analysis* involves understanding how the key mission and system functional elements interact. The mission analysis translates the users' needs into functional/performance requirements and design constraints. A functional analysis takes these requirements and breaks them down into specific tasks.

4. *Concept Development* is the process of making informed trade-offs among the various options to select the one that best meets the requirements and design constraints. Preliminary design and performance requirements and implementation architecture are the results.

5. *Derived Requirements Definition* is the process of translating mission and functional analysis results, system operational concepts, and the selected system architecture into a set of system performance and interface requirements. At this level, the requirements must specify either functional or interface criteria only, without presenting design solutions. This gives the detail designers the flexibility needed to arrive at design solutions that meet the requirements.

6. *Implementation Planning and Systems Integration* is a complex activity resulting in a coherent, integrated set of implementation tasks and responsibilities for the design, development, fabrication, verification and operation of the required system. It requires negotiation between the system requirements definition personnel and the system implementation (development) personnel. The plan must also consider the project constraints of schedule and budget while avoiding unnecessary risk.

7. *Configuration Management* activities ensure that controlled definition of all engieering documentation is maintained and correct information is distributed to all appropriate parties in a timely manner. This is one of the most important responsibilities of the systems engineering organization. On larger programs that have large numbers of people involved, this process becomes even more critical. This activity is also the mechanism by which the system development process is documented (i.e., design knowledge capture).

Configuration Management establishes the system to control the requirements and configuration of hardware and software, evaluate changes, and maintain the definition of the configuration via baselined documentation and released drawings.

8. *Technical Oversight* serves two functions. First, it ensures that all the subsystems work together. Second, it implements mechanisms to guarantee that the developed and documented architectural concept is not inadvertently changed during the development process. This allows the developer to certify that the system, which is ultimately tested, will meet the customer's requirements. Technical oversight consists of the technical reviews and audits that gather consensus from all parties involved to ascertain that the effort at any given time is correct and adequately planned for the continuance of the work.

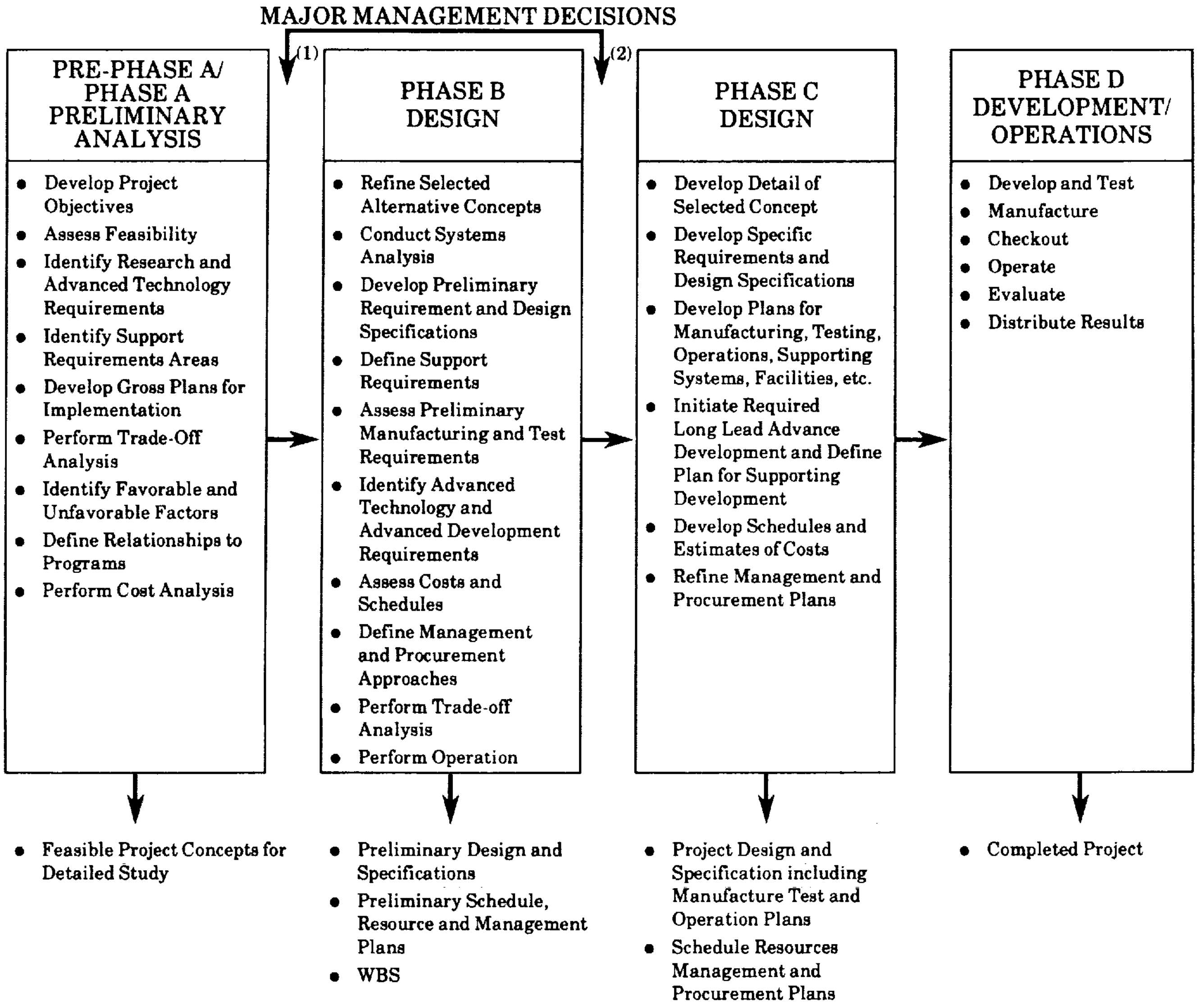
A specific task for the systems engineer to perform is assuring that the systems requirements are understood and correctly implemented by the design organizations. This responsibility requires the systems engineer to work closely with the design organizations throughout the program. At the same time, the systems engineer must recognize that the initial set of systems requirements will not be perfect. During design evolution or because of the inability of a subsystem to meet its intended functional requirements, changes in the systems requirements will be necessary, and the systems engineer should view these changes as a normal part of the design process. Avoid the tendency to view the Systems Requirements Specification as something, once baselined, that is final and unchangeable.

9. During the *Verification and Validation* portion of the development activity, the characteristics and performance of the system are compared to the requirements and specifications. Tests, analyses and demonstrations are performed to verify that the hardware and software satisfactorily meet the performance requirements of the system specifications.

NASA PHASED PROJECT DESCRIPTION

In the planning of major projects, critical requirements must be well defined and the necessary technology must be available. If these criteria are met, there will be an acceptable level of risk in meeting technical goals with reasonable cost and schedule.

To ensure that the program is at a proper level of maturity when Congress approves major funding for design and development, projects go through various phases of analysis and definition. There are five phases in the life cycle of a typical successful project: pre-Phase A (concept study), Phase A (preliminary analysis), Phase B (definition), Phase C (design) and Phase D (development/operations). Depending on the complexity of the system, funding availability and launch schedules, a project may combine phases or add intermediate phases. Common variations would include combining pre-Phase A and Phase A, adding an advanced development phase between Phase B and Phase C, combining Phase C and Phase D into Phase C/D, or moving operations out of Phase D into a separate phase. As a further example, the Space Shuttle program had both a Phase B' (B prime) and Phase B" (B Double-prime) in order to further refine the definition and requirements of the system before proceeding into Phase C. Figure 2 depicts a typical phased project flow in which


MAJOR MANAGEMENT DECISIONS
(1)
(2)
PRE-PHASE A/ PHASE A PRELIMINARY ANALYSIS
• Develop Project Objectives
• Assess Feasibility
• Identify Research and Advanced Technology Requirements
• Identify Support Requirements Areas
• Develop Gross Plans for Implementation
• Perform Trade-Off Analysis
• Identify Favorable and Unfavorable Factors
• Define Relationships to Programs
• Perform Cost Analysis
PHASE B DESIGN
• Refine Selected Alternative Concepts
• Conduct Systems Analysis
• Develop Preliminary Requirement and Design Specifications
• Define Support Requirements
• Assess Preliminary Manufacturing and Test Requirements
• Identify Advanced Technology and Advanced Development Requirements
• Assess Costs and Schedules
• Define Management and Procurement Approaches
• Perform Trade-off Analysis
• Perform Operation
PHASE C DESIGN
• Develop Detail of Selected Concept
• Develop Specific Requirements and Design Specifications
• Develop Plans for Manufacturing, Testing, Operations, Supporting Systems, Facilities, etc.
• Initiate Required Long Lead Advance Development and Define Plan for Supporting Development
• Develop Schedules and Estimates of Costs
• Refine Management and Procurement Plans
PHASE D DEVELOPMENT/ OPERATIONS
• Develop and Test
• Manufacture
• Checkout
• Operate
• Evaluate
• Distribute Results
• Feasible Project Concepts for Detailed Study
• Preliminary Design and Specifications
• Preliminary Schedule, Resource and Management Plans
• WBS
• Project Design and Specification including Manufacture Test and Operation Plans
• Schedule Resources Management and Procurement Plans
• Completed Project

(1) Mission need statement approved
(2) Mission need statement reaffirmed

Source: MM7120.2, Project Management Handbook

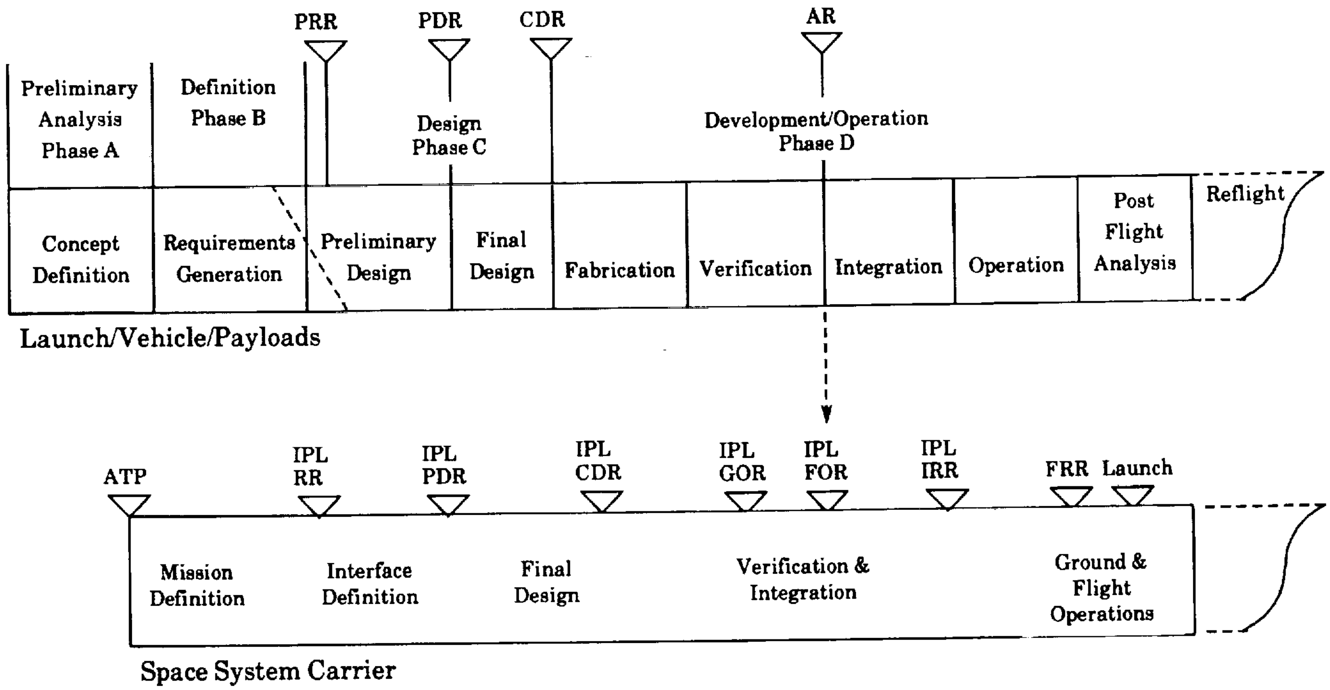
Figure 2 NASA Program Phases

pre-Phase A has been combined with Phase A.

Safety is a critical systems engineering function that must be considered during all program phases and in all studies and analyses. In short, although safety is organizationally the responsibility of S&MA, it is a responsibility of all program participants and should be a primary consideration throughout the systems engineering process.

Figure 2 shows the major activities in each phase, as well as the outputs and major decision points. Note that this description pertains to the typical program, in which NASA contracts with industry to do the Phase C/D activity. Other types of programs include small, contracted efforts, as well as both large and small in-house programs where NASA may retain all or part of the design and development responsibility.

The typical program review phasing includes many more activities and formal reviews than are shown in Figure 2. For completeness, these are introduced here and



Notes:
PRR - Preliminary Requirements Review
PDR - Preliminary Design Review
CDR - Critical Design Review
AR - Acceptance Review
ATP - Authority to Proceed
IPL - Integrated Payload

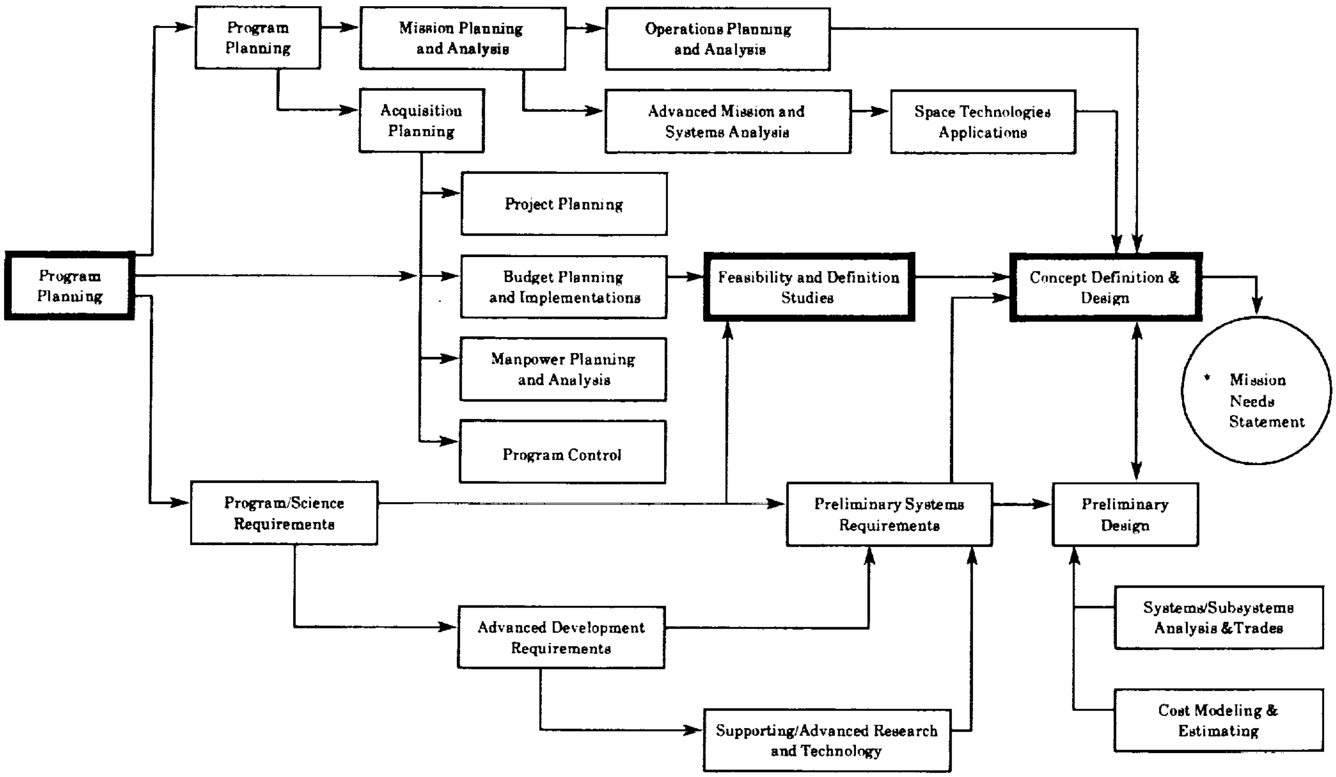
RR - Requirements Review
GOR - Ground Operations Review
FOR - Flight Operations Review
IRR - Integrated Readiness Review
FRR - Flight Readiness Review

Figure 3 Typical Program Review Phasing

shown in Figure 3. This figure also serves to relate the major reviews to the project phases and to show the more detailed integration activities associated with attached payloads and Spacelab-kinds of experiments.

At MSFC, the Program Development (PD) Directorate is responsible for nurturing new projects from idea conception through concept definition supporting preliminary design. Systems engineering is emphasized and utilized throughout this process, both in-house and during contracted studies. Typically, concepts that have matured through this process and gained Congressional new start approval to become official projects are then moved into project offices. The new start review and approval process begins approximately two years in advance of Phase C/D authority to proceed (ATP) at which point funds are applied to begin a major design and development effort. That two-year period is used to execute the definition phase (Phase B) and prepare the request for proposal (RFP) for Phase C/D. The new start approval process includes a definition review or non-advocate review (NAR) generally conducted during the Phase B activity at a time when the project manager, Center management, and Headquarters program office deem appropriate. Results of the NAR are factored into the Phase C/D RFP, as well as the budget approval process. Note that this timeline pertains principally to large programs which include in-house and contracted efforts. The timeframe could be *much shorter* for smaller projects such as experiments. Figure 4 shows the overall systems engineering process flow in Program Development (PD).

In the course of developing the preliminary systems requirements and the conceptual design, PD uses many of the same


Program Planning
Mission Planning and Analysis
Operations Planning and Analysis
Acquisition Planning
Advanced Mission and Systems Analysis
Space Technologies Applications
Project Planning
Program Planning
Budget Planning and Implementations
Feasibility and Definition Studies
Concept Definition & Design
Manpower Planning and Analysis
* Mission Needs Statement
Program Control
Program/Science Requirements
Preliminary Systems Requirements
Preliminary Design
Systems/Subsystems Analysis &Trades
Advanced Development Requirements
Cost Modeling & Estimating
Supporting/Advanced Research and Technology

* Headquarters Approval Lead to Program Initiation Agreement

Figure 4 Systems Engineering Process Flow in Program Development

analysis tools and techniques that are employed by Science & Engineering (S&E) in later program phases. The principal differences in the outputs of the two organizations are the quantity, format and maturity of the documentation and the level of detail in the analyses. In summary, the analyses and trade studies by S&E are to refine, not repeat, the concepts developed by PD in support of design implementation. PD develops the conceptual approach and S&E develops the design implementation.

PRE-PHASE A (CONCEPT STUDY)

A pre-Phase A study may be accomplished within the engineering capability of Program Development or contracted with funding from one of the major NASA Headquarters offices. Successful results from this study would provide justification to initiate a Phase A study or additional pre-Phase A studies. The genesis of new ideas requiring further study can come from a variety of sources: industry, the scientific community, university and research centers, MSFC contractors and associates, or even from within MSFC itself. Typically, such ideas receive a top-level examination by cognizant MSFC/PD personnel. A quick assessment of objectives, requirements and the total mission concept is performed. Often, new ideas are shared with colleagues through proposals (either in response to an RFP or unsolicited), technical papers at professional society meetings, or "white papers" propounding the new idea/concept. From an MSFC in-house weeding out process, concepts are identified for further (Phase A) study.

System functional concept trades are performed during the pre-Phase A period, generally at a fairly cursory level of detail. This process eliminates architectures that are too costly or time-consuming to develop. They are conducted at a level sufficient to support the definition of the top-level system requirements. Architectural options are the result. Some of the primary sources for this identification of concepts include brainstorming, past experience, examination of other systems and intuition.

Cost estimates are developed in pre-Phase A and are usually at a very preliminary level due to the lack of detailed systems definition. These estimates are based primarily on parametrics adjusted for the new program, taking into account differences in mission, size, complexity and other factors.

PHASE A (PRELIMINARY ANALYSIS)

A Phase A study is the preliminary analysis of a space concept. These concepts could have come from a pre-Phase A study or from other sources within or external to NASA. The majority of concepts that are studied at MSFC are assigned by NASA Headquarters and funded accordingly. Documentation in this Phase usually consists of study reports and briefing charts.

Schedules are developed during Phase A studies by Program Development in conjunction with the organization performing the study (contractor, PD, S&E). The schedules include an overall program schedule provided by MSFC and a detailed technical schedule developed by the contractor.

The overall program schedule depicts important milestones that establish the start and finish dates of each study phase, including design, development, launch, and operations. Programmatic milestones are also shown. These are dependent on the federal budget cycle plus proposal preparation and evaluation time. The contractor schedule depicts the major activities and phasing required to develop the hardware in time to meet the scheduled launch date. Since this is a concept study, the detail schedule is still at a relatively high level and would not show activity below the system level.

Cost estimates developed during Phase A are generated using a parametric cost analysis system in conjunction with the cost database discussed above. The has access to several cost estimating systems, both government and commercial. One example is the GE/RCA Price Model. Each model is unique with special capabilities and limitations. Complexity factors and Cost Estimating Relationships are applied to the estimating software using system weight as the independent variable. A factor is applied to the hardware/software costs to account for wraparounds such as project management, test and verification, percent new design, operational complexity, hardware complexity, similarity to other projects or development activities and others. As each system is defined in more detail and the system weight is further refined, the cost estimates become more realistic and provide a higher confidence level in the results.

A cost/risk analysis and assessment is usually completed near the end of each Phase A study. The analysis is accomplished with special software that uses statistical techniques, including a Monte Carlo simulation. The results predict the probability of completing the program within the estimated cost. A risk assessment, which follows the analysis, should identify areas of high risk that require further cost analysis or possibly further trade studies to look at alternate systems that would lower the potential costs without sacrificing technical capability.

As part of the study activity, the contractor provides a detailed risk analysis and assessment to establish a high level of confidence for the program cost. The cost estimate established during this phase will provide NASA Headquarters with the funding requirements to be approved by Congress before the development program can begin.

The processes occurring during Phase A include:

- Development of project objectives
- Assessment of project feasibility
- Identification of research and advanced technology requirements

- Identification of support requirements areas
- Performance of trade-off analyses
- Identification of favorable and unfavorable factors
- Definition of relationships to other programs
- Selection of systems concepts
- Identification of maintenance, technology insertion, and disposal concepts of payload and orbital debris
- Environmental Impact Analysis.

The outputs from Phase A, which become the inputs to Phase B, are in the form of reports or annotated briefing charts and include information on:

- Concept definition
- Preliminary system requirements
- Preliminary configuration layouts
- Point designs
- Preliminary implementation plans
- Preliminary schedules
- Preliminary cost estimates
- Environmental impact.

PHASE B (DEFINITION AND PRELIMINARY DESIGN)

This phase of the project consists of the refinement of preliminary requirements, cost estimates, schedules and risk assessments prior to starting final design and development.

Once the feasibility of an idea is established, the concept definition phase is begun to explore alternatives to meet the documented mission need. Competition and innovation should be employed to ensure that a wide variety of alternatives are identified and examined. Modeling and computer analysis are required to assess the best concepts.

The goal of a concept definition activity is to determine the best and most feasible concept(s) that will satisfy the mission and science requirements. Generally, the requirements available at this point in time are Level I (NASA Headquarters) requirements from preliminary activities.

Level I requirements are broad mission needs and objectives. Occasionally, there may be some Level II (project office level) requirements at this time.

The mission need determination is the first step in a multifaceted preliminary concept definition activity. This is the step that is first performed at a NASA Headquarters or Center level (or industry, university, etc.) and is the precursor to concept development. The mission need determination is that part of early mission planning that identifies a scientific knowledge need or gap that could be met with some kind of NASA sponsored activity. A set of Level I requirements is generally developed during or just prior to the activities described in the following paragraphs.

A feasibility analysis is conducted to determine the viability of the project. The study report usually includes requirements, objectives, problems, opportunities and costs.

A utility analysis is then conducted to determine the value of a project. The following criteria may be considered during this study: the needs met, the scientific knowledge acquired, the political benefits, or potential spinoffs and applications.

Certain satellites and/or instruments are selected for a more detailed level of design. The Preliminary Design Office of Program Development performs these studies. This office is a miniature replication of the capabilities of the laboratories at MSFC: Propulsion, Guidance, Navigation and Control, Electrical Power, Avionics, Structures, Operations, etc. One difference is the emphasis by Program Development in developing credible cost estimates. Cost is an important differential, but often other factors, such as mission risk or incompatibility with other instruments that may be grouped on a common satellite, may predominate.

Throughout the Phase B period the concepts that were developed during Phase A are iteratively reviewed and analyzed. Using trade study techniques, the concepts' capabilities are compared to the system requirements. Those concepts that consistently satisfy the requirements are identified and refined. Any concepts that do not meet performance and other requirements are scrutinized very closely for possible elimination. Following the examination of those that do not perform well, assessments are made regarding their augmentation to discover the degree of change necessary to bring their performance into scope. The concepts that have to change too much or would experience severe budgetary and/or schedule impacts are deleted from the concept definition and analysis cycle. This allows the analysts' energies to be focused on those concepts that are valid and workable.

These trade studies provide a more detailed look at the architectural concepts and result in a narrowing of the field of candidates. Trades performed during this time consider such things as cost, schedule, lifetime and safety. The evaluation criteria used to assess alternative concepts are developed to a finer level of detail than for earlier system trades.

Cost estimates from Phase A are refined as further detailed requirements are identified during Phase B. The cost estimating process is still dependent on parametric analysis. The Program Development cost office works closely with the study contractor in evaluating costing methodology and continuously compares government cost estimates with those of the study contractor. Should a large discrepancy occur, the assumptions and schedule inputs of the study contractor are examined. If this examination yields valid assumptions and schedules, the NASA estimates are adjusted. The cost estimation process goes through continuous iterations during the study to reflect the evolution of detail resulting from trade studies.

Schedules are developed during Phase B by the task team program control personnel and by the study contractors. Schedules developed by the task team are expanded from the Phase A overall program schedules. In addition, other schedules are developed that include Phase C and D procurement strategies, cost phasing and project manning requirements. The study contractor schedules are expanded to lower levels of the work breakdown structure (WBS) to include subsystem development, program management, manufacturing, verification, logistics planning, operations planning and other technical areas. The schedule detail would show the phasing of all major activities through launch and the follow-on operations.

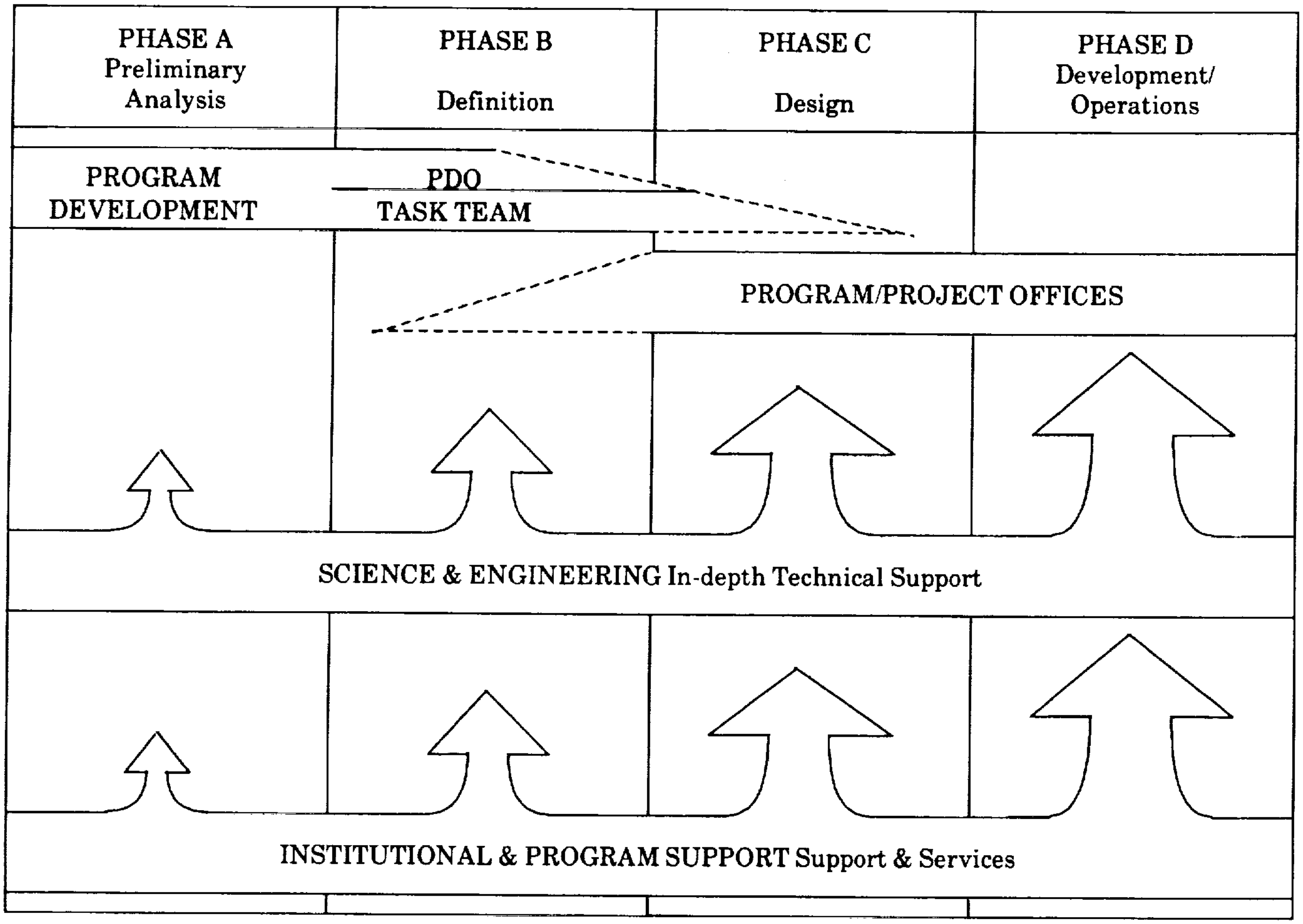
As in Phase A, the typical documentation of this phase consists of reports and briefing charts.

The processes occurring during Phase B include:

- Refinement of selected alternative concepts
- Performance of trade-off analyses
- Performance of system analyses and simulations
- Definition of preliminary system and support requirements
- Definition and assessment of preliminary manufacturing and test requirements
- Identification of advanced technology and advanced development requirements for focused funding
- Refinement of preliminary schedules
- Refinement of preliminary cost estimate and trade study results which support selection of baseline for cost estimate
- Assessment of technical, cost, and schedule risks
- Assessment and refinement of the Mission Need Statement.

The outputs from Phase B, which become the inputs to Phase C, may include (in the form of study reports and annotated briefing charts) information related to:

- Preliminary WBS
- System requirements
- Preliminary interface requirements



Source: *PD Lead Engineer's Guide*

Figure 5 MSFC Support Relationships in Project Phases

- Management and procurement approaches
- Program Implementation Plans
- Request for Proposal (RFP) inputs, where applicable.

Phase B is normally the final phase of activity within Program Development. A separate core of people is selected to form a task team to manage the Phase B contract. At the beginning of Phase B, a chief engineer is appointed to the task team (or project office) to provide consultation to the task team manager on all related engineering matters. The chief engineer also helps ensure that the study contractor uses acceptable engineering practices and sound judgment during the course of the study. The chief engineer is often the deputy to the task team manager and is usually the first Science and Engineering representative substantially involved in the process. The chief engineer's office has personnel resources available to support the project as needed during the study. Additional engineering support from S&E may be used at the discretion of the chief engineer. The chief engineer plays a key role in determining the state of technical maturity of the project for starting the design and development phase.

At the conclusion of Phase B, the task team is converted to a project office, and it is no longer under the direction of program development. On large projects, such as Space Station, a project office might be created prior to Phase B; in that case, Program Development (PD) support becomes minimal (such as cost estimating and limited programmatics) and S&E plays a major role in the Phase B engineering activities.

At MSFC, it is not uncommon to have more than one directorate providing engineering or technical support to a project throughout its life cycle. The transition of engineering support is depicted in Figure 5.

Figure 5 shows that Program Development typically performs most, if not all, of the technical support during Phase A. As the project life cycle evolves, the S&E Directorate takes on a larger and larger role as PD's involvement tapers off. The exact point at which S&E gets involved varies depending on the size and priority of the project at MSFC, as well as the availability of S&E manpower resources. In every case, however, Phase C and D activities are exclusively the domain of S&E.

The extent of information and the level of detail available at the end of Phase B to begin the Phase C design are variable and become a function of the time and money made available to the PD organization for the conduct of Phase B studies. As a result, significant efforts may be needed at the beginning of Phase C to refine many of the Phase B analyses.

The hand-over of technical responsibility from PD to S&E is an interface which requires a great deal of attention to minimize transition problems and project disruptions. A key issue to be addressed is the type and content of documentation produced in Phases A and B. Since these early phases typically have limited funding and PD's manpower resources are limited, requirements and specifications resulting from Phase B may require substantial refinement and rework by S&E at the beginning of Phase C. It is important that Phase C planning and schedules account for this activity.

PHASE C (DESIGN)

This phase requires Congressional budget approval for projects large enough to be separate line items in the NASA budget submission. Funding must be approved and available at the start of Phase C. Detailed design is accomplished and plans are refined for final development, fabrication, test and operations.

The processes occurring during Phase C include:

- Refinement of work breakdown structure
- Development of Systems Requirements Specification
- Development of design and contract end item specifications
- Development of interface requirements documents
- Completion of preliminary and detail design
- Development of preliminary interface control documents (ICDs)
- Performance of detailed system analyses
- Development of manufacturing, testing verification, integration, operations, supporting systems and facilities plans
- Definition of a development plan
- Refinement of schedules and cost estimates
- Refinement of management and procurement plans.

The outputs from Phase C, which become the inputs to Phase D, include:

- Updated system requirements documentation
- Updated detail design and CEI specifications
- Baseline.

It is typically at the beginning of Phase C, when industry is heavily involved in design and project funding is increased dramatically, that many formal documentation requirements are contractually imposed. This can contribute to large cost increases over previous estimates in Phases A and B, and dictates the need for early inputs from the S&E engineering organization to assure that design and performance requirement specifications and data requirements are incorporated into initial cost estimates.

PHASE D (DEVELOPMENT/OPERATIONS)

During this phase of a project, flight hardware and software are developed, manufactured/coded, tested and qualified for flight. In addition, support is provided for the follow-on flight operations.

The processes occurring during Phase D include:

- Development and test of prototype and protoflight hardware
- Verification/Validation - qualification of hardware and software for flight
- Manufacture and integration of flight hardware
- Checkout of flight systems
- Launch operations
- Flight operations
- Retrieval or disposal of payload and orbital debris.

The outputs from Phase D include:

- A successful mission,
- Documentation and evaluation of the results and anomalies
- Documentation of lessons learned.

In the early days of spaceflight, MSFC provided expendable propulsion systems, so most project activity terminated when launch operations were complete. As the mission of MSFC evolved into payloads and experiments, its role in the area of mission operations and maintenance greatly expanded and now provides an important function in present projects such as Spacelab, the National Space Transportation System, Hubble Space Telescope, the Advanced X-Ray Astrophysics Facility, and Space Station Freedom. These programs involve 15 to 30 years of technology insertion, operations and maintenance activities that would justify a separate independent phase in their life cycles.

At MSFC, the design phase is normally combined with the development and operations phase to form a Phase C/D. The resulting contract takes the Phase B data, refines it into a final design, develops and fabricates the hardware, tests and flight qualifies it, and supports the flight and mission operations.

MANAGEMENT ISSUES IN SYSTEMS ENGINEERING

by Robert Shishko and Robert G. Chamberlain
with contributions by
Robert Aster, Vincent Bilardo, Kevin Forsberg, Hal Mooz, Lou Polaski and Ron Wade

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'etre*: 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 systems engineer. The interaction between the buyer and systems 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 systems 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.

The work breakdown structure (WBS) is also a hierarchical 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 WBS and the schedules by which that work is done.

The project's organizational structure describes 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 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-linking. The archives should contain all of the rest of the project's information that is worth keeping, 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.

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 the requirements tree.

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 systems engineer concentrates on managing its technical aspect. This requires that the systems 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.

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 first. 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.

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
- 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
- Human engineering
- Maintainability
- Safety
- Survivability/vulnerability
- Integrated logistics
- Quality assurance.

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 on work breakdown structures and network schedules.)

SEMP Lessons Learned from DoD Experience

- A well-managed project requires a coordinated SEMP 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.

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.

Managing the Systems Engineering Process: Summary

Systems engineering organizations, and specifically project-level systems engineers, are responsible for managing projects through the technical aspect of the project cycle. This responsibility includes managing the decomposition and definition sequence, managing the integration, verification and validation sequence and controlling 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.

THE WORK BREAKDOWN STRUCTURE

A 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 1 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.

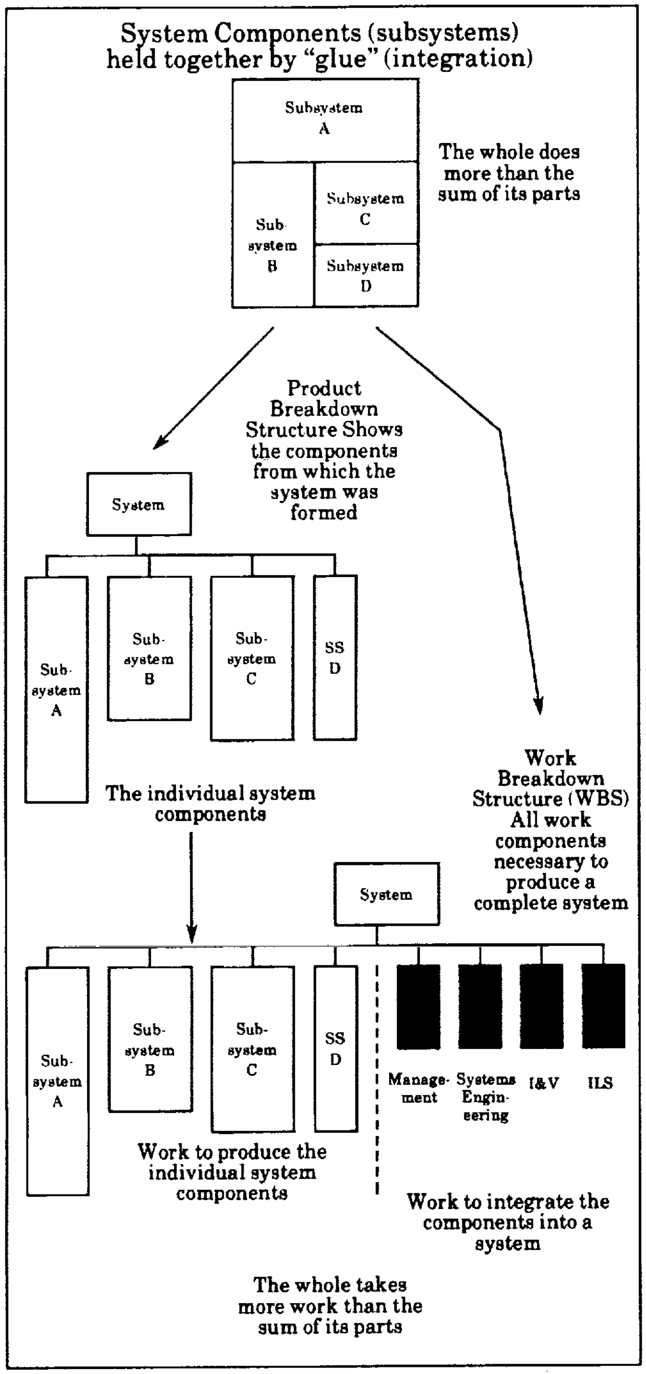

System Components (subsystems)
held together by "glue" (integration)
Subsystem A
Sub system B
Subsystem C
Subsystem D
The whole does more than the sum of its parts
Product Breakdown Structure Shows the components from which the system was formed
System
Sub-system A
Sub-system B
Sub-system C
SS D
The individual system components
Work Breakdown Structure (WBS) All work components necessary to produce a complete system
System
Sub-system A
Sub-system B
Sub-system C
SS D
Manage-ment
Systems Engin-eering
I&V
ILS
Work to produce the individual system components
Work to integrate the components into a system
The whole takes more work than the sum of its parts

Figure 1 The Relationship between a System, a Product Breakdown Structure, and a Work Breakdown Structure

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.

This section provides some techniques for developing a WBS, and points out some mistakes to avoid.

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 work force, 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.

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 systems 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 recursive process is repeated 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. Some of the terms for projects that provide multiple deliveries, are "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) that identifies each delivery. At any 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 all produced on a routine basis. A PBS for an operational facility might consist 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.

Common Errors in Developing a WBS

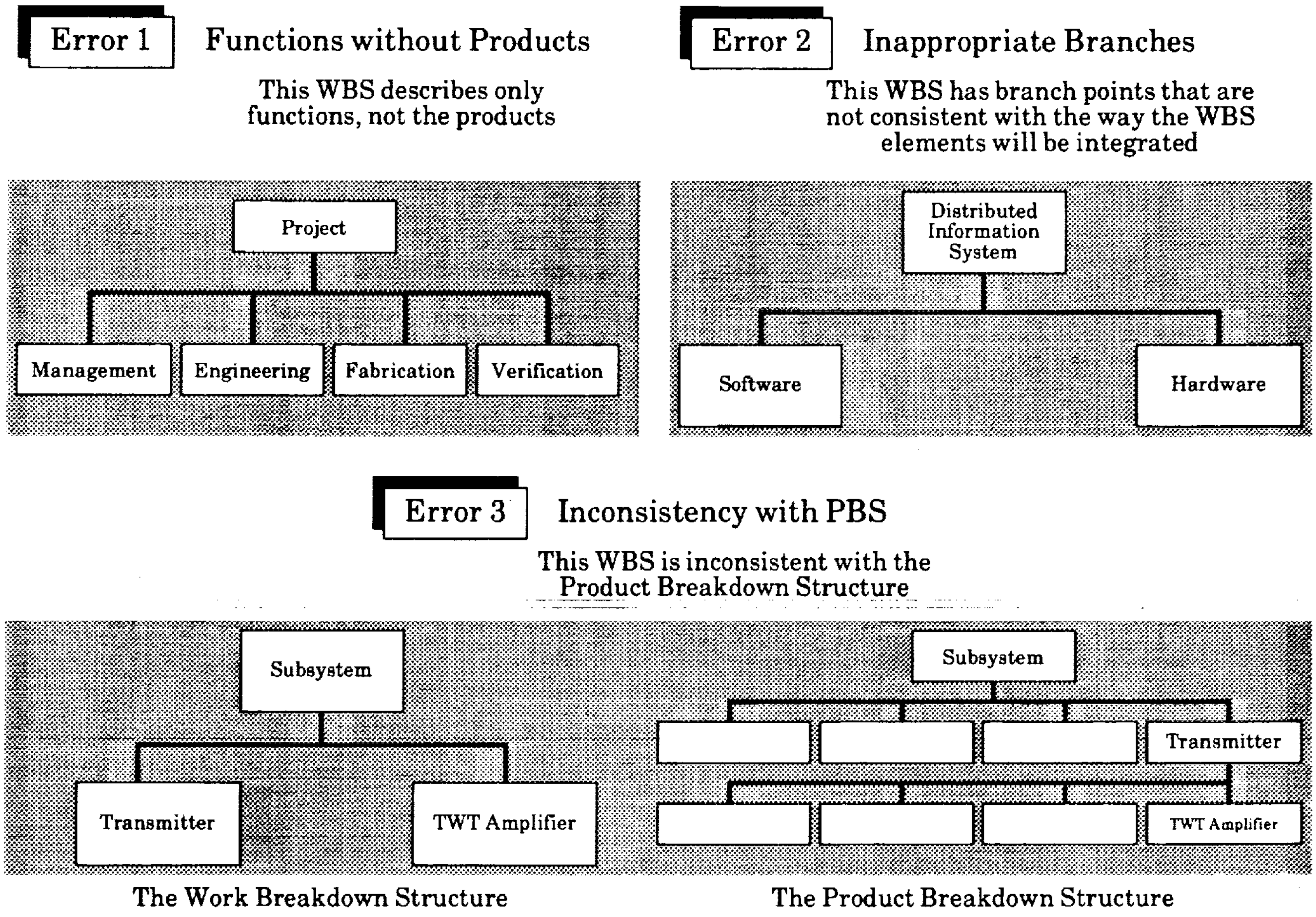
There are three common errors found in WBSs:

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

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 2. 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.



Error 1 Functions without Products

This WBS describes only functions, not the products

Error 2 Inappropriate Branches

This WBS has branch points that are not consistent with the way the WBS elements will be integrated

Error 3 Inconsistency with PBS

This WBS is inconsistent with the Product Breakdown Structure

The Work Breakdown Structure

The Product Breakdown Structure

Figure 2 Examples of WBS Development Errors

NETWORK 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.

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. An understanding of the project's schedule is a prerequisite for accurate project budgeting.

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.

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.

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 3, 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.

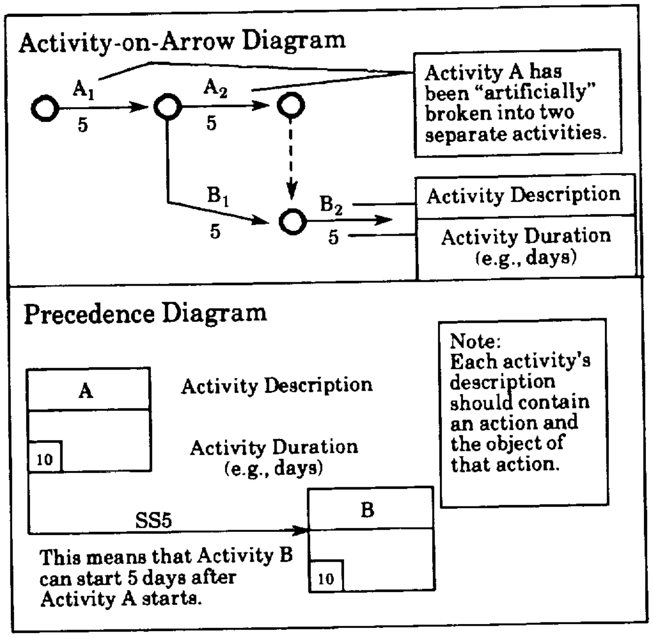

Activity-on-Arrow Diagram
A1
5
A2
5
B1
5
B2
5
Activity A has been "artificially" broken into two separate activities.
Activity Description
Activity Duration (e.g., days)
Precedence Diagram
A
10
Activity Description
Activity Duration (e.g., days)
SS5
B
10
This means that Activity B can start 5 days after Activity A starts.
Note: Each activity's description should contain an action and the object of that action.

Figure 3 Activity-on-Arrow and Precedence Diagrams for Network Schedules

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).

Each of these three activity relationships may be modified by attaching a lag (+ or −) to the relationship, as shown in Figure 3.

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.

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 (work force, 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 systems engineer to review the schedule logic. It is not unusual at this point for some iteration of steps one to four to be required in order to obtain a satisfactory schedule. Reserve will also 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., at this point. The critical path for the project is discovered at this step in the process.

Step 6: Review the work force level and funding profile over time, and make final adjustments to logic and durations so that work force levels and funding levels are reasonable. Adjustments to the logic and the durations of activities may be needed to conform to the schedule targets established at the project-level. This may include adding more activities to some WBS element, deleting redundant activities, increasing the work force 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.

Reporting Techniques

Summary data about a schedule is usually described in Gantt charts, a good example of which is shown in Figure 4. 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.

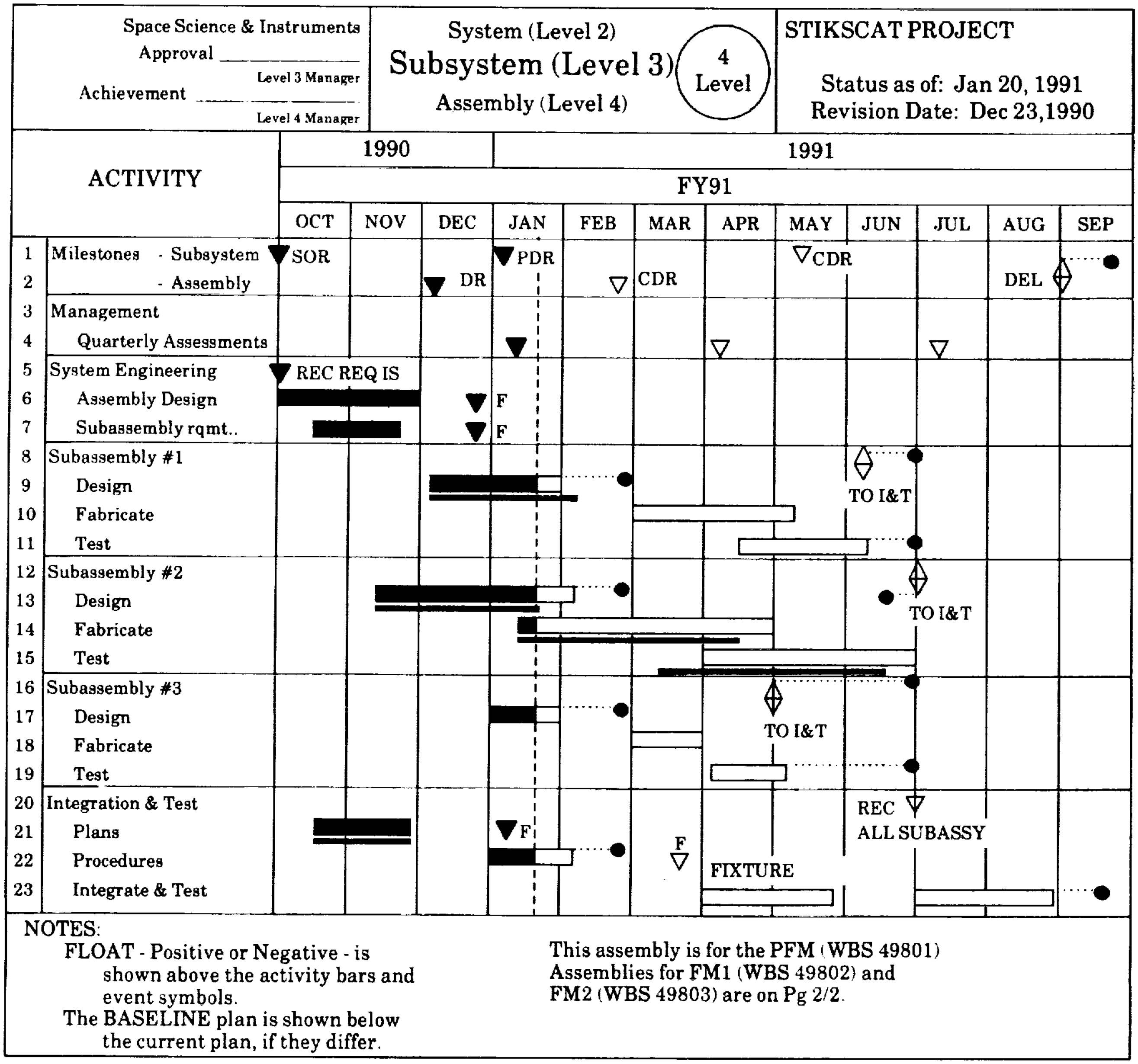
Good scheduling systems provide capabilities to show resource requirements over time, and to make adjustments so that the schedule is feasible with respect to resource constraints over time. Resources may include work force level, funding profiles, important facilities, etc. Figure 5 shows an example of an unleveled resource profile. The objective is to move the start dates of tasks that have float to points where

Desirable Features in Gantt Charts

The Gantt chart shown in Figure 4 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:
  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 to those items most pertinent at the time of status reporting.



NOTES:
FLOAT - Positive or Negative - is shown above the activity bars and event symbols.
The BASELINE plan is shown below the current plan, if they differ.

This assembly is for the PFM (WBS 49801) Assemblies for FM1 (WBS 49802) and FM2 (WBS 49803) are on Pg 2/2.

Figure 4 An Example of a Gantt Chart

the resource profile is feasible. If that is not sufficient, then the assumed task durations for resource-intensive activities should be re-examined and, accordingly, the resource levels changed.

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 systems 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

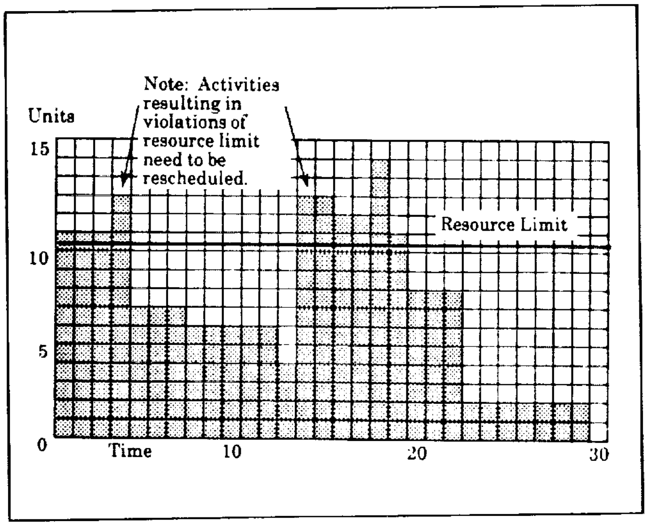


Figure 5 An Example of an Unleveled Resource Profile

the WBS and network schedule. This specifically involves combining the project's work force and other resource needs with the appropriate work force rates and other financial and programmatic factors to obtain cost element estimates. These elements of cost include:

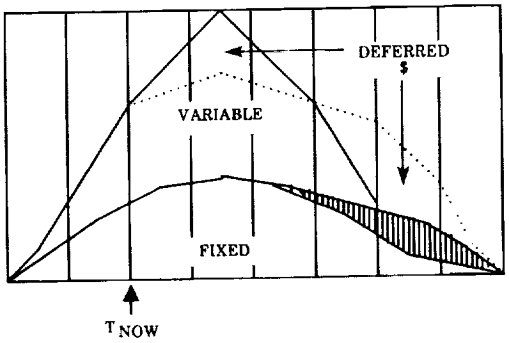
- Direct labor costs
- Overhead costs
- Other direct costs (travel, data processing, etc.)
- Subcontract costs
- Material costs
- General and administrative costs
- Cost of money (i.e., interest payments, if applicable)
- Fee (if applicable)
- Contingency

When there is a cost cap or a fixed cost profile, there are additional logic gates that must be satisfied before the systems 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 systems 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.

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. 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.

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 work force 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-years) dollars.

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.

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.

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

- 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 systems 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 6.

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 elucidate 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

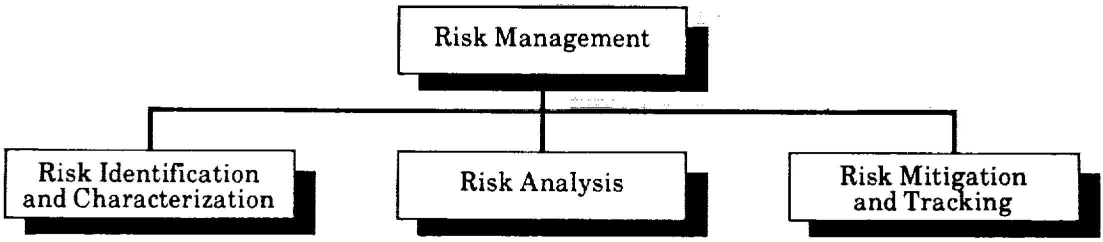

Risk Management
Risk Identification and Characterization
Risk Analysis
Risk Mitigation and Tracking

Figure 6 Risk Management Structure

systems 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 systems 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 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 systems 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 are shown in Table 1. The systems engineer needs to choose the techniques that best fit the unique requirements of each project.

Risk Identification and Characterization	Risk Analysis	Risk Mitigation and Tracking
Expert interviews	Decision analysis	Watchlists/ milestones
Independent assessment (cost, schedule and technical)	Probabilistic 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		

Table 1 Techniques of Risk Management

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 ongoing 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.*

Types of Risks

There are several ways to describe the various types of risk a project manager/systems engineer faces. Traditionally, project managers and systems 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 systems 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 systems engineer's subjective probability estimates form a valid basis for risk decision making.

Risk Identification and Characterization Techniques

A variety of techniques is 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 later.

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 4245.7M, *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.

Risk Analysis Techniques

The tools and techniques of risk analysis rely heavily on the concept and "laws" (actually, axioms and theorems) of probability. The systems 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.

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.

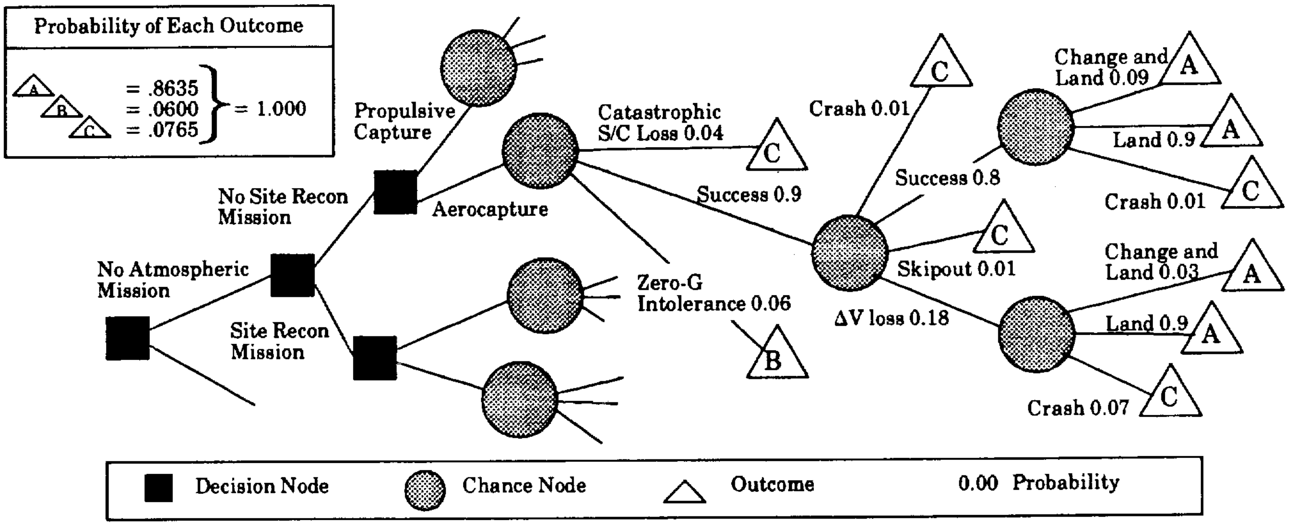
In brief, decision analysis is a technique that allows:

- 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

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.

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.

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

Probabilistic Risk Assessment Pitfalls

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

$$R = \Sigma_s P_s C_s$$

where P_s is the probability of outcome s, and C_s is the consequence of outcomes. 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.

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. This approach explicitly recognizes that single point values for these variables do not adequately represent the risk conditions inherent in a project.

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 systems 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 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 systems engineer should recommend the establishment of reasonable financial and schedule contingencies and technical margins.

The strategy and underlying rationale selected for a specific risk 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 deal 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 work arounds, 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. (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 used extensively on the Shuttle program to track items with significant system safety consequences.

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 later.

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 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, managers and systems engineers 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.

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 systems 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 systems engineer is responsible for ensuring the completeness and technical integrity of the technical baseline. The content of the technical baseline includes:

- Definition (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.

Baseline Evolution

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

Risk management activity 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.

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 7.)

Configuration management is essential to the execution of 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.

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

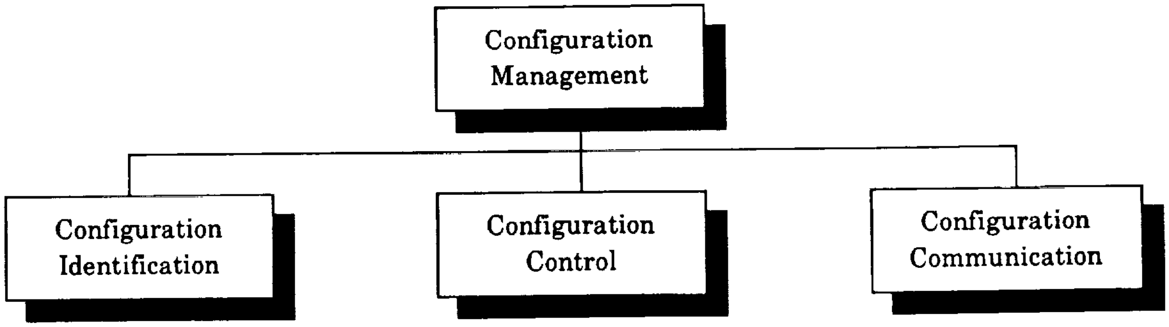


Figure 7 Configuration Management Structure

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.

Change Control Board Conduct

Objective: To review evaluations and then approve or disapprove proposed changes to the project's technical, operations or business baseline.
Participants: Project manager (chair), project-level systems 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 systems engineer for completeness prior to presentation.
Decision: The CCB members discuss the Change Request (CR) and formulate a decision. Project manager agrees or overrides.

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.

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 systems 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 8. 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

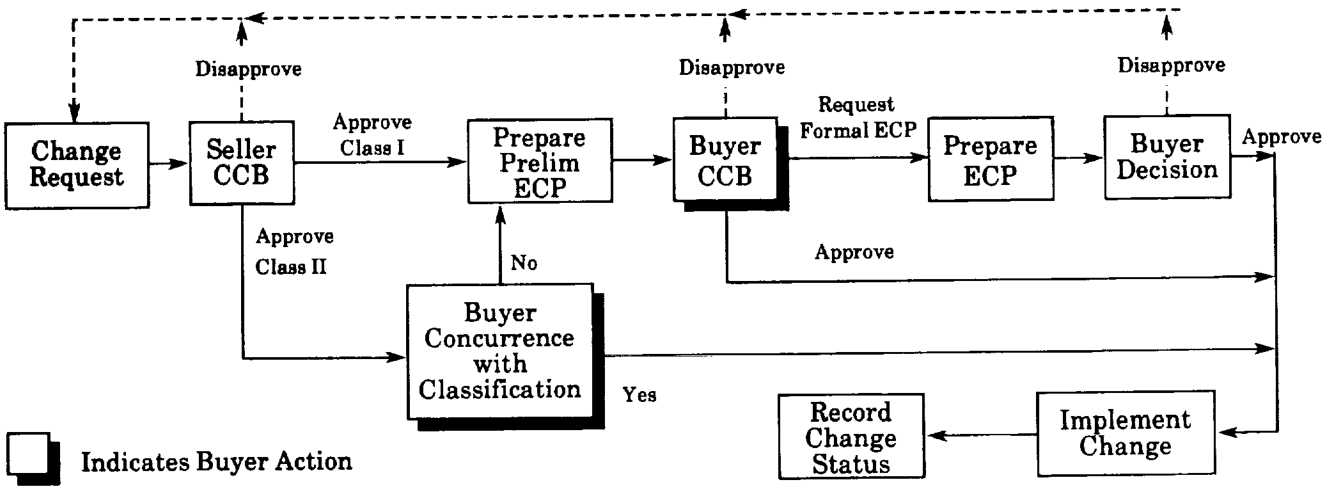


Figure 8 Contract Change Control Process

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.

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 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 systems 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 database or paper files, and cannot degrade with time. This is not a trivial requirement, 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, 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 systems 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.

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.

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. 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.

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.

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.

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 are 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/her 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-disciplinary 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 (ECR) 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.

SPECIFIC TYPES OF REVIEWS

This section describes the types, purpose, timing and content of most of the reviews that 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 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 and 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
- Nonelectric 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
- Nonconformance 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
- Nonconformance 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 here. However, the systems 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 and 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
- Nonconformance reports/status
- Personnel training requirements
- Personnel training status
- Present status of safety activities, including cost and technical developments
- Risk management activities.

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 earlier. 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 9. 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

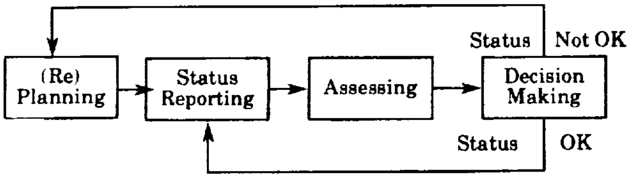


Figure 9 Planning and Status Reporting Feedback Loop

taken down the hierarchy. Managers at each level determine (consistent with policies establi:hed at the next higher level of the project hierarchy) how often, and in what form, reporting data 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.

Cost and Schedule Control Measures

Status reporting and assessment on costs and schedules provides the project manager and systems 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/systems engineer may choose to evaluate the completeness and quality of these reporting systems against criteria established by the project manager/systems 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 10 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 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*.

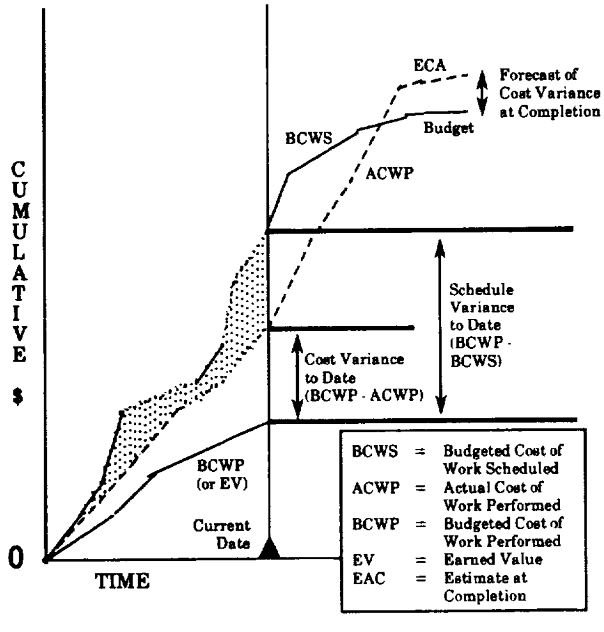


Figure 10 Cost and Schedule Variances

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.

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 Systems 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 systems 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 systems engineer.

Computing the Estimate at Completion

EAC can be estimated at any point in the project. The appropriate formula depends upon the 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 + ACEP - 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 × (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 .

In a large project, a good EAC is the result of a variance analysis that may use 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.

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.
- "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 systems engineer needs to decide which generic and unique TPMs are worth tracking at each level of the PBS. The systems 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.

In selecting TPMs, the systems 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 systems 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.

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.

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 n^{th} unit.

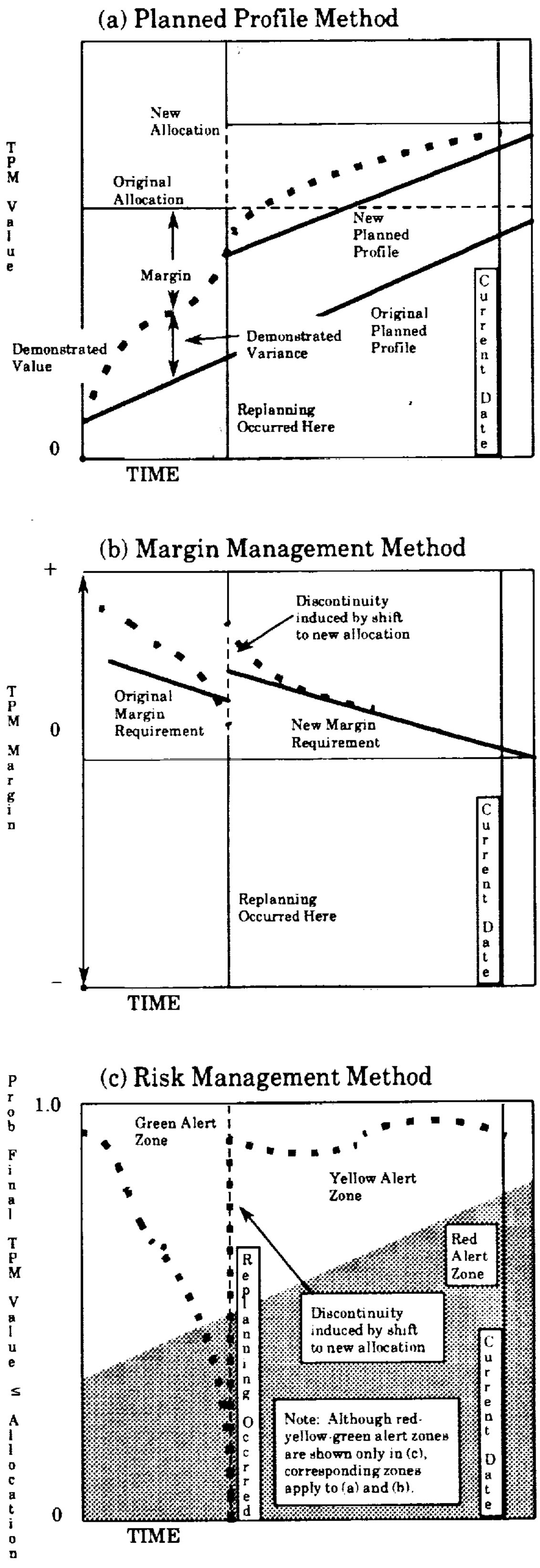
A closely related method of assessing a TPM relies on establishing a time-phased margin requirement for it and comparing the actual margin against that requirement. The margin is generally defined as the difference between a TPM's demonstrated value and its allocated requirement. The margin requirement 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 systems 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 11(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 11(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 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 11(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. Early in the TPM tracking program, when the demonstrated



Note: Although red-yellow-green alert zones are shown only in (c), corresponding zones apply to (a) and (b).

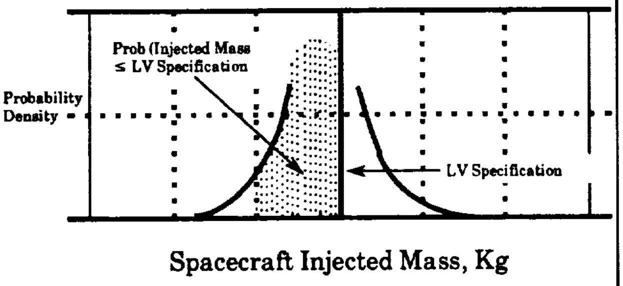
Figure 11 Three TPM Assessment Methods

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 11(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.

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 and are integrated into spacecraft. 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.



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 11(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!

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 systems 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.

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 earlier.

Systems engineering process metrics try to quantify the effectivity and productivity of the systems engineering process and organization. Within a single project, tracking these metrics allows the systems 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 systems engineer may focus on metrics dealing with systems engineering staffing, project risk management progress and major trade study progress. A subsystem systems engineer may focus on subsystem requirements and interface definition progress and verification procedures progress. It is useful for each systems engineer to focus on just a few process metrics. Which metrics should be tracked depends on the systems 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. That 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

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. related	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
	V&V procedures completed per systems engineering hour	P
	Processing of trouble reports	Q
Reviews	Processing of Review Item Discrepancies (RIDs)	Q
	Processing of action items	Q

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

Table 2 Systems Engineering Process Metrics

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 recognize 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 of open ECRs, or by examining the number of ECRs opened last month versus the total number open. The systems engineer should apply personal judgment in picking the status reporting and assessment method.

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 personal efforts.

N93-24683
158575
p. 8

Spacecraft Systems Engineering: An Introduction to the Process at GSFC

by Tony Fragomeni and Mike Ryschkewitsch

Systems engineering means different things to different people. Some say it applies only to one spacecraft or a total mission. Others say it applies only to hardware and not to software, but that assumption is flatly wrong. Still others say it is electrically oriented while others say it is mechanically oriented; that depends upon whether you talk to an electrical or a mechanical engineer. Systems engineering is often equated with systems management and systems design. Some would reduce it to a purely analytical process and others would reduce it to mere hands-on physical integration.

Systems engineering is all of these and much more. It encompasses such terms as the system approach, system analysis and systems integration. It includes systems requirements analysis and functional analysis. The Goddard Space Flight Center's Code 400 *Project Manager's Handbook* says it is "one of the most important technical efforts of a project and . . . assures the design adequacy of the complete system to meet the stated user/experimenter requirements for a mission." These efforts include both the ground and flight segments, launch vehicle interface, and the end-to-end data system from collection of raw data on orbit to reduced data on the ground ready for analysis. The handbook says: "The Systems Manager of a project serves as Chief Engineer and provides a focal point for the systems engineering effort throughout all phases of the project."

As a succinct definition, that is as good as any but not really very helpful in understanding the systems engineering process, especially in the development of spacecraft. The concept becomes much clearer and richer when we ask why we need systems engineering, who a systems engineer is, what the systems engineer does and what are some of the products.

But first we can state what systems engineering is not. It is not one, single, isolated process. The whole process of systems engineering is better described as an attitude . . . a plan of attack . . . a way of thinking. Consider, for example, the difference between a chemist adding one ingredient to a fixed solution to achieve a predictable result, and a doctor who must consider a variety of uncertain and ever changing physical and emotional factors in the diagnosis and treatment of a patient.

As shown in Figure 1, systems engineering is not a process that is easily contained in a single manual or cookbook. Rather, it is the systematic use of many time-tested and experience-verified disciplines, tools and human resources needed to identify, define and solve problems. Which tools to use or expertise required depends not only on the mission under consideration but also the phase or stage of the project. The process thus demands a great deal of versatility and flexibility.

Finally, systems engineering is not always one individual or even one organization. Instead, it is a flexible process which makes the development and design meet the requirements and constraints imposed by the user and the system environment. It is a process characterized by multiple starts and stops, frequent shifts and alternate approaches, as opposed to a clear-cut path or a simple recipe for success.

Systems engineering is clearly a dynamic process that cannot and will not be pinned down into a simple procedural formula. This process, however, is generally the same for different kinds of projects. In these times of increasingly constrained budgets, it is

PRECEDING PAGE BLANK NOT FILMED

78 INTENTIONALLY BLANK

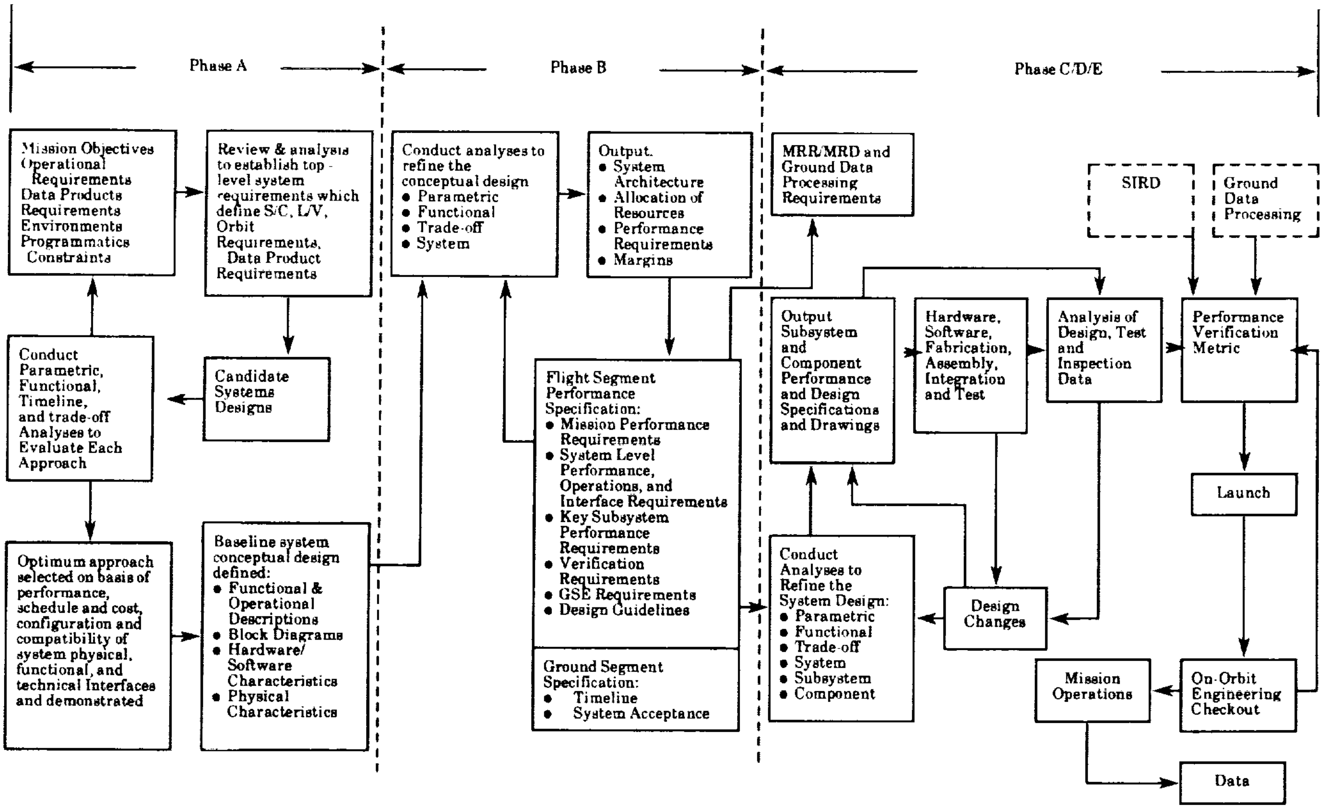

Phase A
Phase B
Phase C/D/E
Mission Objectives Operational Requirements Data Products Requirements Environments Programmatics Constraints
Review & analysis to establish top - level system requirements which define S/C, L/V, Orbit Requirements, Data Product Requirements
Conduct analyses to refine the conceptual design
• Parametric
• Functional
• Trade-off
• System
Output.
• System Architecture
• Allocation of Resources
• Performance Requirements
• Margins
MRR/MRD and Ground Data Processing Requirements
SIRD
Ground Data Processing
Conduct Parametric, Functional, Timeline, and trade-off Analyses to Evaluate Each Approach
Candidate Systems Designs
Flight Segment Performance Specification:
• Mission Performance Requirements
• System Level Performance, Operations, and Interface Requirements
• Key Subsystem Performance Requirements
• Verification Requirements
• GSE Requirements
• Design Guidelines
Ground Segment Specification:
• Timeline
• System Acceptance
Output Subsystem and Component Performance and Design Specifications and Drawings
Hardware, Software, Fabrication, Assembly, Integration and Test
Analysis of Design, Test and Inspection Data
Performance Verification Metric
Launch
Optimum approach selected on basis of performance, schedule and cost, configuration and compatibility of system physical, functional, and technical Interfaces and demonstrated
Baseline system conceptual design defined:
• Functional & Operational Descriptions
• Block Diagrams
• Hardware/ Software Characteristics
• Physical Characteristics
Conduct Analyses to Refine the System Design:
• Parametric
• Functional
• Trade-off
• System
• Subsystem
• Component
Design Changes
Mission Operations
On-Orbit Engineering Checkout
Data

Figure 1 SE Process in the Evolution of a Mission

incumbent upon the systems engineer to optimize the systems design and to do things efficiently and *not* just effectively. Systems engineers are called upon to identify the risks in increasingly complex projects, and then attempt to minimize the impact of those risks. In very complex spacecraft, which are expected to perform delicate and ultrasophisticated functions, a minor intrasystem perturbation can have a major performance impact across multiple systems. Systems engineering is a disciplined technical approach that forces us to do our homework up front and early on, to uncover problems before they become showstoppers. Although we cannot conclusively test for everything, we are expected to identify and verify realities and adequate margins.

In a sense, we have always had systems engineering in NASA, but it may aptly be termed "informal." Certainly, we recall engineers and managers who had a big-picture perspective, looking at all functions and how they interrelate, but more often than not, their trade studies were on isolated scratch pads and the logic kept in their heads or in a desk drawer. You can almost hear them say: "This is the way we've always done it."

Sometimes this informal system worked, especially on small, relatively simple projects. But as the spacecraft became more complex and development time elongated, a more formal process of systems engineering emerged. In simple terms, it starts with functional analysis and leads to functional requirements and then design requirements. It starts at the top and works down, fully documented at each step and traceable. The greater the complexity and duration of a project, the greater the penalty for not catching errors early on, and the greater the need for a well understood and well documented process. The SE process should ensure that all fixes be made before the start of hardware fabrication when the cost of fixes is relatively inexpensive. To wait until later is costly, and it can be prohibitive at the interval between acceptance testing and launch.

SE ROLES AND RESPONSIBILITIES

The main objective in systems engineering is to devise a coherent *total* system design capable of achieving the stated requirements. Requirements should be rigid. However, they should be continuously challenged, rechallenged and/or validated. The systems engineer must specify *every* requirement in order to design, document, implement and conduct the mission. Each and every requirement must be logically considered, traceable and evaluated through various analysis and trade studies in a *total* systems design. Margins must be determined to be realistic as well as adequate. The systems engineer must also continuously close the loop and verify system performance against the requirements.

The fundamental role of the systems engineer, however, is to engineer, not manage. Yet, in large, complex missions, where more than one systems engineer is required, someone needs to manage the systems engineers, and we call them "systems managers." Systems engineering management is an *overview* function which plans, guides, monitors and controls the technical execution of a project as implemented by the systems engineers. As the project moves on through Phases A and B into Phase C/D, the systems engineering tasks become a small portion of the total effort. The systems management role increases since discipline subsystem engineers are conducting analyses and reviewing test data for final review and acceptance by the systems managers.

REQUIREMENTS

The name of the game in systems engineering is requirements. The statement, traceability and eventual verification of requirements is probably the most important aspect of systems engineering. Requirements are initially derived from user needs, i.e., the customer. It is understood that for each requirement there is an associated margin that must continually be challenged. As the project nears completion, the amount of available margin is expected to decrease since the margins are updated based on "actuals."

- **Functional Requirements** provide a description of the functions and subfunctions required to conduct the mission. These are generally derived from functional analysis and allocation.

- **Performance Requirements** or source requirements define what the system must accomplish and how well the system must perform. These requirements are initially derived from user needs and requirements statements and refined through requirements analyses and trade studies. They are defined during each application of the systems engineering process based on outputs from previous iterations of the process, program decisions and updates to user requirements. They provide the metrics that must be verified through appropriate analyses, demonstrations and tests.

- **Derived Requirements** are lower level (subsystem and components) performance requirements resulting from an analysis of the user stated performance requirements and the definition of functional requirements. These derived requirements are used by subsystem discipline engineers in characterizing the subsystem performance requirements necessary to ensure the attainment of the user-stated performance or source requirements.

- **Reflected Requirements** are requirements placed on other subsystems or on the higher level systems which must be provided to each of the subsystems to ensure proper performance of the subsystem and the eventual attainment of the user stated performance or source requirements.

- **Design Requirements** are described by drawings, material lists, process descriptions and other supporting documents for the fabrication, production or manufacturing of a system element. These are generally derived from the synthesis of a solution for one or more higher level requirements.

The systems engineer must be able to demonstrate the traceability of each requirement through each level, right up to the contractually binding source requirements. User requirements are determined and refined during Phase A studies. A host of considerations are made in order to produce the best set of "integrated performance requirements," considering technical performance, first as mitigated by cost and schedule. Systems engineers should not and do not make cost and schedule decisions, especially in the later phases, but in Phases A and B, cost and schedule are trade-off parameters that must be considered in determining the best course of action.

PHASE A - MISSION ANALYSIS

In Phase A Mission Analysis, systems engineers will translate user needs or goals into a quantifiable set of functional requirements that can be translated into design requirements. User requirements are defined as a "set of objectives" that are quantified in broad terms and basic functions. The user should also state performance measures in terms of preferences as well as trade evaluation criteria. The systems engineers will conduct functional, parametric and system analyses to define and refine mission requirements and to generate alternative candidate system designs. Baseline system conceptual designs should emerge as design drivers are identified, as well as high risk areas and offsets. Common system drivers include size, weight, power, data rate, communications, pointing, orbital altitude, mission operations coverage (geometry and timing) and scheduling. Trade-off studies are conducted to balance the requirements, but even the optimal technical approach may not be the best way when the design is evaluated in terms of cost, schedule *and* risks. Since all projects will undergo cost, schedule and technical perturbations during development, it is imperative that a good system be developed. However, contractual, legal and fiscal requirements dictate that the technical approach must be agreed to by the start of Phase C/D. The overall system architecture must be established during Phase A; this includes the apportionment of functions between the flight and ground segments. It is imperative that proper studies and analyses be done to result in the correct structure since this affects the remainder of the project up through the operations phase.

Phase A outputs or products include a Phase A Report, a Science Requirements Document, preliminary Instrument Interface Requirements Documents, cost, schedule and a Project Initiation Agreement (PIA). The Phase A Report includes functional and operational descriptions, hardware and software distribution, design requirements, system/subsystem descriptions, mission description, a preliminary work breakdown structure (WBS) and recommendations for Phase B. The Phase A Report must have sufficient data to answer questions such as these:

- Do the conceptual design and operational concept meet the overall mission objectives?
- Is the design technically feasible?
- Is the level of risks acceptable?
- Are schedules and budget within the specified limits?
- Do preliminary results show this option to be better than all others?

PHASE B - DEFINITION PHASE

Assuming that each crucial question is answered affirmatively during Phase A, the systems engineer will continue development of the system requirements by conducting more detailed analyses to refine the baseline system conceptual design. These Phase B tasks must result in technical requirements and operational functions that are reflected in Interface Control Documents, performance and design specifications and statements of work that are used to produce the hardware during Phase C.

Specifications are defined as "a description of the technical requirements for a material or product that includes the criteria for determining whether the requirements are met." Basically, there are four types of specifications:

- Functional - describes only the ultimate end use; contractor is responsible.
- Performance - describes quantitatively what it must do; contractor is responsible.
- Design - what to make and how to make it; buyer is responsible.
- Levels of Effort - used only for support services.

The statement of work (SOW) describes the work needed to carry out the entire mission as well as how and where the work is to be done. The work breakdown structure (WBS) is used for reporting progress, performance and engineering evaluations. The WBS will structure the family of specifications and drawings resulting from the progressive stages of systems engineering. The final result of the Phase B process is a system definition in sufficient depth of detail to allow beginning the detailed design process for each of the individual subsystems.

PHASE C/D - EXECUTION PHASE

During Phase C/D, systems engineering provides technical oversight during design, development, test and evaluation to ensure that timely and appropriate intermeshing of all technical disciplines are reflected in the overall design. Technical performance requirements and margins are continually reaffirmed through analyses and tests during this phase. Phase C/D outputs or products will also include a variety of analytical and test reports on hazards, faults, single-point failures and failure modes for "what-if" or worst-case scenarios. Trade-offs and other analyses continue but in greater detail at the subsystem and component levels to ensure proper conversion of performance requirements into the design and into the hardware.

PHASES E AND F - PRE-MISSION AND MISSION OPERATIONS

Phases E and F, Pre-mission and Mission Operations, also involve systems engineering, although to a lesser degree since the most important SE work is done early on. However, the final verification of a space flight, system can only be done in flight, on-orbit. The systems engineering team is full time with the flight operations team during initial on-orbit engineering checkout and on call during mission operations. The final product is the "On-Orbit Engineering Performance Report" which measures mission performance against requirements. This document becomes useful in subsequent projects, especially if it contains lessons learned. Finally, the systems engineer's job is only completed when the user has the final delivered product, e.g., scientific data, in hand.

SYSTEMS ENGINEERING ANALYSES

Systems engineering is a highly analytical process. Throughout the entire project (not just at the beginning) the systems engineer will conduct or review numerous analyses to establish strong performance and design parameters as well as to continually evaluate design approaches and options. A systems engineer is expected to establish performance parameters and margins, verify them with test and inspection data, and compare the actual to the predicted. Everything must be "what-ifed" to the lowest necessary level, not just once but continually, so that there are few if any surprises.

One tool used by the systems engineer is functional analysis. This is a top-to-bottom effort done in all phases and at every hardware level. The systems engineer takes a performance requirement (function) at one hardware level of assembly and, after thorough analysis, determines the optimum distribution and implementation of the requirement at the next lower hardware level. Functional analysis is also used to determine whether a particular function is best accomplished in flight or on the ground. Functional analysis results in a hierarchical structure (i.e., architecture) that progressively divides and allocates how a function is to be accomplished, down to the lowest common denominator. This is extremely useful in deciding where to cut the interface, especially in view of verification, accountability and jurisdictional (i.e., contractual) boundaries.

Another top-to-bottom systems engineering analysis done in all phases is the requirements flowdown and allocation analysis. This can be described as an equitable, attainable and realistic distribution of system-level performance requirements and resources, including margins, to successively lower levels of hardware assemblies. To verify the validity and distribution of tolerances and margins, continued analysis and review are required throughout the project. This starts during Phase A and continues through every on-orbit checkout. Distribution should be compared to actuals, and estimates should be quantified as a function of design maturity.

Trade-off studies and analyses also define margins and identify potential problem areas. They are done on all systems and for all technical disciplines to select the configuration that best satisfies a user requirement. Alternative technologies are examined to satisfy functional and design requirements, including those with moderate to high risk. Trade-off studies also support make-or-buy decisions and help manage technical risk. In Phases A and B, they establish system architecture and configuration. In Phase C/D, they evaluate alternate solutions in system/subsystem/component design. After critical design review (CDR), however, trade-off studies are conducted only during the evaluation of design changes or responses to failures. All factors that affect the function or requirement must be studied: performance, reliability, safety, cost, risk, schedule, maintainability, servicing, power, weight, thermal, complexity, etc.

System parametric and sensitivity modeling and analyses are used to develop confidence that a design satisfies higher level requirements, and to provide traceability of functional, performance and design requirements. This is accomplished by varying a particular performance parameter between its established worst-case limits and as perturbed by worst-case environmental stresses to determine the resultant effect on successively higher assembly levels or performance parameters. These analyses can serve as a primary vehicle for conducting trade studies and to assess the whole system effectiveness of synthesized design options and alternatives. Like all other studies and analyses, these analyses are done during all phases and are updated based on actual test data.

RISK ASSESSMENT

Risk assessment is approached from different but related directions. During Phases A and B, the systems engineer will want to do sufficient analyses to ensure that the technical approach is valid and that any new developments or state-of-the-art items and their risk offsets have been identified. During Phase C/D, sufficient analysis must assure that performance requirements and margins are adequate and are in fact satisfied. Throughout the entire project life cycle, risk assessment and particularly Failure Mode Effects

Analyses and fault tree analyses should be used as design tools to enhance the overall system design and make it immune to failures, both hardware and human.

Risk assessment is the identification and evaluation of the impact upon the technical performance of those system elements that appear to possess an inherent probability of failing to meet some critical performance or design requirement essential for the successful accomplishment of the intended mission. Systems engineering identifies the potential failures, establishes margins and quantifies the risk. Risk taking gets down to knowing what your margins are and how they are distributed. How do you know what the margins are? By doing lots of analyses and backing them up with tests. Two of the best tools are Failure Mode Effects Analysis (FMEA) and hazards analyses.

The FMEA assures that the failure modes of a system are known and can be addressed in an orderly fashion. Initially the analysis must identify all critical functions and the effects of the impairment of those functions on mission success. Following this, a detailed component and system interaction study is conducted to determine all the ways a function could be impaired, the effect on mission success and how such an impairment could be detected. The impact of these failures and the probability of occurrence must be evaluated in light of the user requirements and the desired level of reliability.

The FMEA is also used in compiling the system-level fault tree used by the flight operations team (FOT) during mission operations. The fault tree is a listing of every plausible anomaly or failure that may occur on orbit. It starts out with the detection of the anomaly or failure as observed by the FOT via telemetry. It then provides a road map used by the FOT in isolating the cause of the anomaly and taking the required corrective action or operational work-around so that the mission can proceed. The fault tree analysis and the development of the FMEA should be done together.

Systems safety hazards analyses are also considered a systems engineering function. The intent of the systems safety hazards analysis is to identify design deficiencies that could directly — or indirectly through operator error — result in personnel injury or damage to the flight hardware. In this case, any potential hazards that could result in death, severe injury or illness must be eliminated. The impact of a major system loss or damage must be evaluated in light of user requirements.

Operations hazards analyses look at possible failures occurring during testing, handling and transportation that could jeopardize the hardware or personnel. All catastrophes and critical hazards resulting in death, severe injury or illness, or major system loss or damage must be eliminated. Marginal hazards may be tolerated if they can be rationally justified and accepted.

REVIEWS, PERFORMANCE ASSESSMENT AND VERIFICATION

The systems engineer is best advised to start early and stay late in reviewing and assessing performance requirements and the associated verification methods employed to prove the requirement has been satisfied. Reviews must be done at all levels. Non-advocate reviews (NARs) should be conducted at the end of Phase B to evaluate the technical, cost and schedule approach for accomplishing the mission. System-level reviews and lower-level hardware design and test reviews should be conducted continually. Peer reviews are vital at all levels and must be conducted by "looking at the drawings and not the viewgraphs." Trend analysis is needed on all critical performance parameters, from box level acceptance through on-orbit to enable the early identification of potential problem areas. Technical performance measurement (TPM) is one proven method of assessing compliance to requirements and the level of technical risk. TPM is defined as the continuing analysis, test and demonstration of the degree of anticipated and actual achievement of selected technical measures and performance parameters. TPM involves analysis of the differences among the achievement to date, current estimate and the required or target value for the parameter.

SUMMARY AND SOME ADVICE

Systems engineering is much more than a one-person job. It is best described as "the technical conscience of a project." As such, systems engineering is a highly structured and disciplined engineering process that cuts across all technical disciplines to ensure interface design compatibility, both intersystem and intrasystem. It organizes at the system level — not at the subsystem level, where compromises may be made. It establishes performance requirements and margins. Systems engineering evaluates the validity of hardware through analysis and review of test data. It identifies risk and offers approaches for the project manager to eliminate or reduce the impact. One eye of the system engineer is on how the end product is used during mission operations; the other is focused on how analyses and tests can prove it can do the job within acceptable margins. Both eyes work in tandem, together, clearly and in focus. Remember:

1. Perform sound systems analyses and design; consider *all* options.
2. Don't box yourself in with unnecessary and undue constraints.
3. Exercise extreme care in system design, especially incorporating appropriate (to the risks) redundancy and provisions for late design changes and on-orbit operational work-arounds, and factor in testing ability.
4. Institute the discipline to ensure painstaking attention to details — great and small.
5. Maintain a total dedication to quality — quality is *designed in*, it does *not* accidentally happen.
6. Ensure rigorous pre-launch testing to establish that requirements are in fact satisfied, and any workmanship or marginal designs are uncovered.
7. Insist on inexhaustible diligence in testing — allow an unexplained or random failure only after all reasonable and practical steps to isolate are taken.
8. Attempt to design backwards — satisfy mission requirements first.
9. Conduct extensive reviews — look at the drawings, not viewgraphs.
10. Have adequate documentation to know where you are going, how you are getting there, where you have been and *when you are there.*
11. Have an open door policy to foster strong intra-project technical communications.
12. Ensure total openness regarding problem identification and resolution.

Systems Engineering & Integration and Management for Manned Space Flight Programs

153576

P-18

by Owen Morris

The development of systems engineering and program management in NASA manned space programs has grown in a largely uncoordinated manner over the last 30 years. However, the systems and practices that have been developed form a proven pattern for successfully integrating large, technically complex programs executed in several geographical locations. This development has not been recorded in a comprehensive manner, and much of the reasoning behind the decisions made is not obvious.

For the purposes of this discussion, systems engineering is defined as the interdisciplinary engineering that is necessary to achieve efficient definition and integration of program elements in a manner that meets the system-level requirements. Integration is defined as the activity necessary to develop and document the systems' technical characteristics, including interface control requirements, resource reporting and analysis, system verification requirements and plans, and integration of the system elements into the program operational scenario.

This paper discusses the history of SE&I management of the overall program architecture, organizational structure and the relationship of SE&I to other program organizational elements. A brief discussion of the method of executing the SE&I process, a summary of some of the major lessons learned, and identification of things that have proven successful are included.

History

NASA, then the National Advisory Committee for Aeronautics (NACA), participation in the management of major aerospace programs began shortly after World War II with the advent of the X series research aircraft. In these projects, essentially all of the technical responsibility was delegated to one of the Centers, which were primarily expert in the technical area being explored (i.e., aerodynamics, stability, control and structures) but did not have experts in the development of hardware. Accordingly, NACA entered into agreements with the Air Force or Navy to manage the actual development of the aircraft. The NACA Centers focused their direction on the technical requirements and performance characteristics to be demonstrated by the aircraft. The contractor's responsibility was similar to that for the development of any aircraft, and the contractor usually furnished test pilots for early demonstration flights.

With the formation of NASA and the start of major manned space programs, it was necessary for NASA to develop the capability to manage complex development activities. Very little SE&I capability existed within the functional organizations of the NASA Centers. As a result, SE&I expertise was developed within each of the program offices. In particular, the Gemini program office was set up with autonomous capability to manage SE&I and direct the development contractor.

With the advent of the Apollo program, SE&I was again managed from the project offices at the development centers. The project offices used specialized technical capability from the Center functional organizations and prime contractors and initiated the practice of hiring support contractors to assist in implementing SE&I. After the Apollo I fire, a review committee was established to determine the cause of the fire and recommend modifications to the program. One of the recommendations made was that NASA acquire a technical integration and engineering support contractor to assist in accomplishing SE&I activity. The Washington program office selected Boeing as the contractor and managed the contract for this activity; however, a large portion of the work force was located at the Centers. The contractor's responsibilities included monitoring the development and operational activities at the Centers, forming integrated assessments of the activity, and making recommendations to the program director for improvements. As the program matured, the contract focus was changed, and the contractor provided a significant number of personnel to directly support the Centers in SE&I and systems development activities.

With the initiation of the Space Shuttle program and the adoption of the Lead Center concept, it was decided to manage the Level II integration activity, including SE&I, by providing a small management core within the program office and using many of the Centers' functional organizations to provide technical support in a matrix fashion. At the Johnson Space Center (JSC), the lead person from the functional organization was generally a branch head or an assistant division chief. JSC had a relatively large staff to draw from to provide the specific technical expertise and the level of effort needed to accomplish a given task.

The Space Station Freedom program was started using the Space Shuttle program as a model. As the Lead Center, JSC managed integration. Later, the Level II function was moved near Washington, D.C., under the deputy program director, and an independent contractor was brought in to assist the integration process. The Space Station Freedom management organization will be discussed in more detail in the next section.

PROGRAM MANAGEMENT ORGANIZATIONAL STRUCTURE

A single NASA Center largely managed early NASA manned space flight programs, which allowed for a relatively simple organizational structure to accomplish program integration. JSC, then called the Manned Space Center, managed both development and flight operational aspects of the Mercury and Gemini programs with the checkout and preflight testing being performed by support elements at Cape Canaveral.

Apollo became organizationally more complex (Figure 1). The spacecraft development was managed by JSC, the launch vehicle development by Marshall Space Flight Center (MSFC), the prelaunch activities by Kennedy Space Center (KSC)—by then an independent NASA Center—and the flight operations by JSC. In all of these programs, the responsibility for the development of the flight hardware was delegated to the Centers, and the interfaces between projects were intentionally kept as simple as possible. The Washington office, under direction of the program director, was responsible for overall direction of the program including budgetary allocations, congressional relations, and management of development issues between the project offices at the different Centers. The actual integration activity (SE&I) was coordinated by a series of panels and working groups in which individuals from the Washington program office served as either chairperson or members, with the program director overseeing the activity. In the early programs (Mercury and Gemini), this activity was the responsibility of a single Center, and the Washington office was coordinated in an informal manner, but by the end of the Apollo program, the management of the panel and working group activity was relatively formal. In all of these programs the Center directors took an active part and personally felt responsible for the technical excellence of the work performed by their Centers. This intercenter involvement was accomplished primarily through the management council and major program reviews where Center directors personally participated in major decisions.

In part of the Apollo program, the Washington office retained the responsibil-

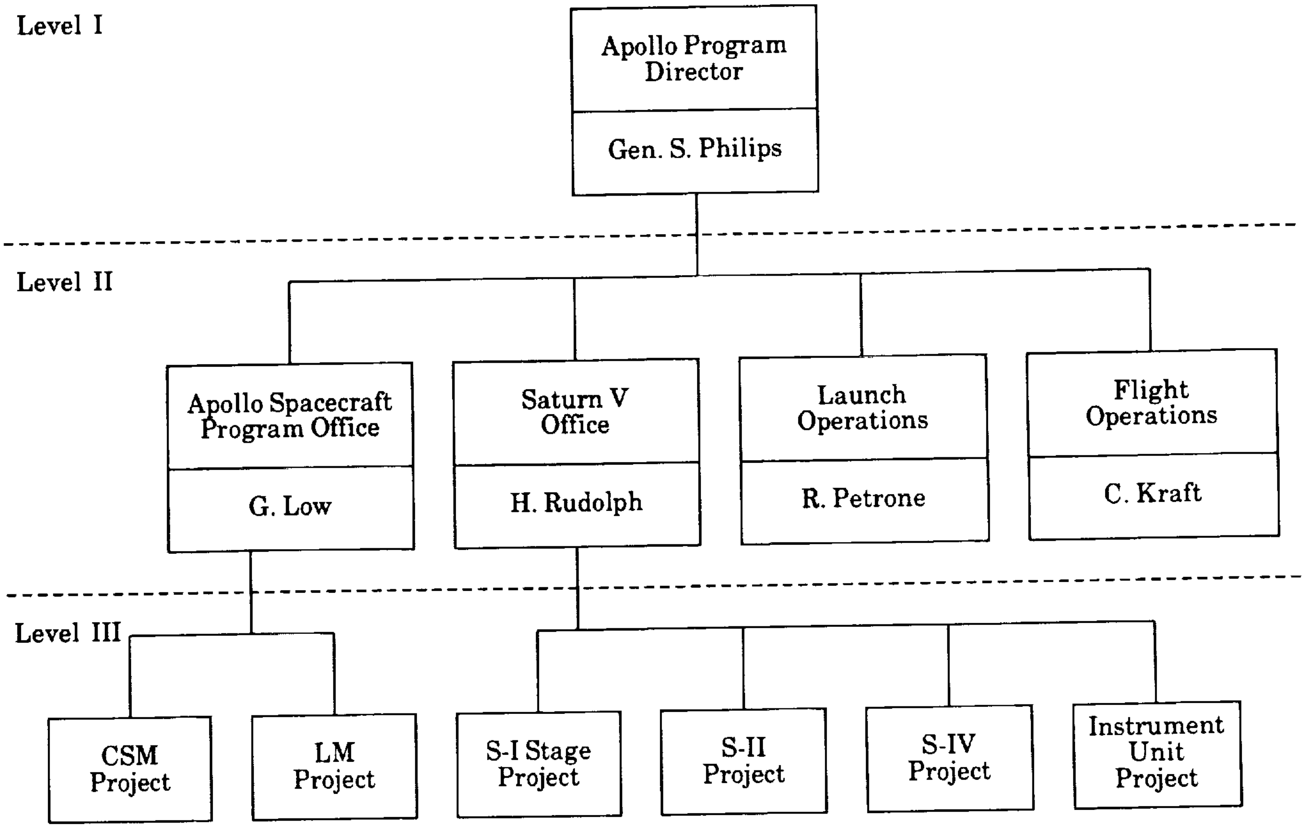


Figure 1 Apollo Program Management Organization

performing the SE&I activity with the actual work being led by Bellcom, a division of Bell Laboratories. Ultimately, this approach was abandoned, at least partly because much of the Center director's responsibility was lost, and an adversarial relationship between the program director and the Center organizations developed. The execution of the SE&I was returned to the Centers with management and coordination of intercenter activities achieved through the use of working groups, panels and management reviews.

At the outset of the Space Shuttle program (Figure 2), the management of SE&I was markedly changed. Some of the more important changes were adoption of the Lead Center management concept in which one of the participating Centers was delegated the management of program level integration including SE&I activities; the adoption of a configuration with functional and physical interfaces of much greater complexity; and the employment of one of the major hardware development contractors as the integration support contractor. The complex interfaces made SE&I activity voluminous and involved and required the commitment of a larger percentage of the program resources to this activity.

The Space Station Freedom program was structured so that the interface activity between the work packages was even more complex than that of the Shuttle program. Initially, the Lead Center approach to SE&I activity was adopted, but the implementation was not effective. As a result of recommendations made by study groups and the committee reviewing the Challenger accident, it was decided to transfer the responsibility for program integration activity, including SE&I, to the deputy program director in Reston, Virginia, and to bring on a contractor to provide program integration

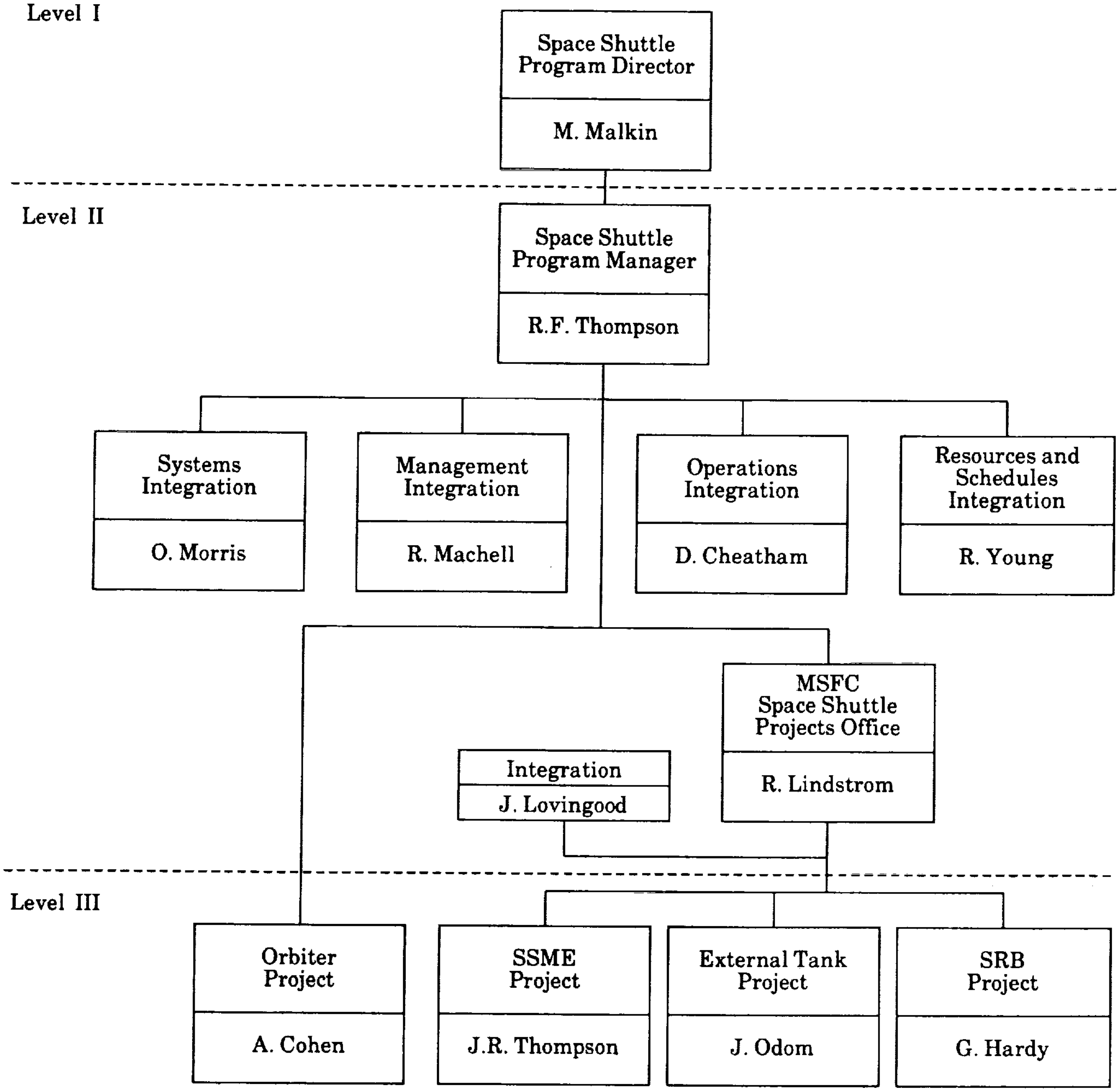


Figure 2 Space Shuttle Program Management Organization

support (Figure 3). Contractors having significant hardware development contracts were excluded from the contract competition. The first approach was to provide detailed management of SE&I activity by the Reston civil service personnel with the integration contractor providing support in executing the activity. Additionally, it was thought that much of the technical integration could be accomplished by having the work package contractors negotiate the definition and execution of much of the detailed integration process directly between themselves. This proved ineffective, however, because there was no clear lead responsibility and no clear way to resolve differences. As a result, because of the complexity of program integration and the lack of in-depth backup capability, this management approach has not been completely effective.

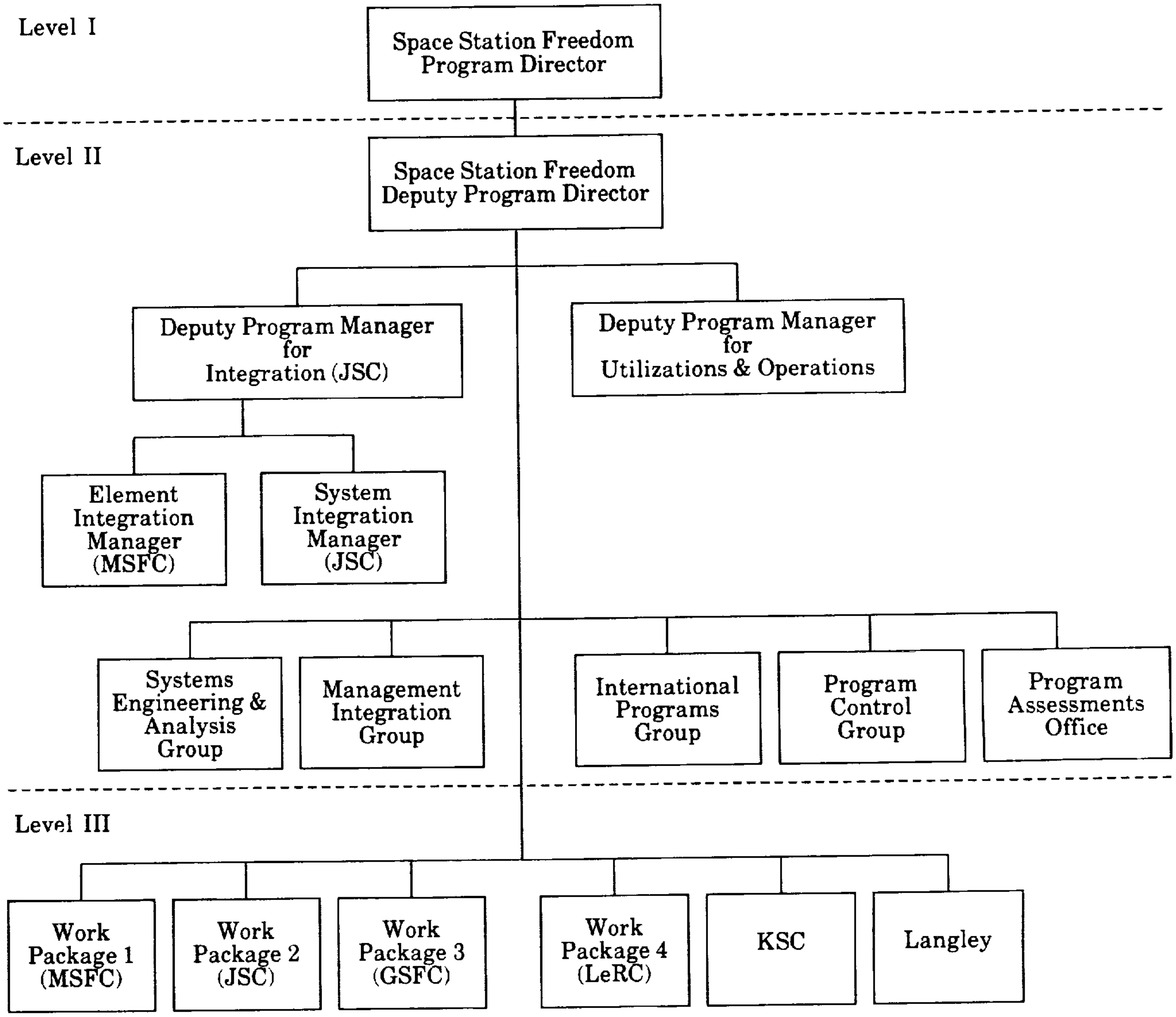

Level I
Space Station Freedom Program Director
Level II
Space Station Freedom Deputy Program Director
Deputy Program Manager for Integration (JSC)
Deputy Program Manager for Utilizations & Operations
Element Integration Manager (MSFC)
System Integration Manager (JSC)
Systems Engineering & Analysis Group
Management Integration Group
International Programs Group
Program Control Group
Program Assessments Office
Level III
Work Package 1 (MSFC)
Work Package 2 (JSC)
Work Package 3 (GSFC)
Work Package 4 (LeRC)
KSC
Langley

Figure 3 Space Station Freedom Program Management Organization (1990)

Recently, it was decided to give the integration support contractor direct responsibility for the integration of the program but without authority to directly manage the work packages or their contractors. In an attempt to obtain more in-depth capability, the program director and deputy program director decided to execute the systems integration portion of the SE&I activity at two of the Centers with the deputy director for integration physically located at one of the Centers. Since these functions were still retained organizationally within the program office, they were under the control of the deputy program director and, at the same time, had the advantage of drawing from the in-depth technical capability residing at the Centers. Simultaneously, the integrating contractor's work force at the Centers was increased in both responsibilities and numbers.

GROWING PROGRAM COMPLEXITY

One of the major factors determining the efficiency of the integration of a program is the methodology used to delegate the engineering and development responsibilities to the project offices at the Centers. It has been found that less complex organizational structures and simple interfaces are extremely important to allow efficient management of SE&I activities. Each of NASA's manned space programs has been organizationally more complex than its predecessor and has had more complex interfaces. In both the Mercury and Gemini programs, the flight elements were divided into two parts, spacecraft and launch vehicle, and the physical and functional interfaces between the two were quite simple. The induced environmental interfaces were somewhat more complex but readily amenable to experimental and analytical determination.

The Apollo program involved a major increase in program complexity. The spacecraft was divided into two project offices and the launch vehicle was divided into four project offices. By assigning the four launch vehicle projects to the same Center (MSFC), the integration between launch vehicle stages could be accomplished at the Center level. Similarly, both spacecraft projects were assigned to one center (JSC) for the same reason. The physical and functional interfaces between the spacecraft and launch vehicle, and hence between Centers, was relatively simple. In a 1971 paper titled "What Made Apollo a Success," George Low stated: "Another important design rule, which we have not discussed as often as we should, reads: minimize functional interfaces between complex pieces of hardware. Examples in Apollo include the interfaces between the spacecraft and launch vehicle and between the command module and the lunar module. Only some 100 wires link the Saturn launch vehicle and the Apollo spacecraft, and most of these have to do with the emergency detection system. The reason that this number could not be even smaller is twofold: redundant circuits are employed, and the electrical power always comes from the module or stage where a function is to be performed. For example, the closing of relays in the launch vehicle could, in an automatic abort mode, fire the spacecraft escape motor. But the electrical power to do this, by design, originates in the spacecraft batteries. The main point is that a single person can fully understand this interface and can cope with all the effects of a change on either side of the interface. If there had been 10 times as many wires, it probably would have taken a hundred (or a thousand?) times as many people to handle the interface." However, the operational complexity of the Apollo vehicle demanded a more extensive integration activity between the Centers and for the first time posed the problem of accomplishing detailed technical coordination between Centers.

One of the basic tenets of the Space Shuttle was to have an integrated vehicle that would recover the most expensive elements of the system for reuse. This led to a design concept that placed a great majority of the electronics and major components of the main propulsion systems in the orbiter. This design concept led to very large increases in interface complexity between the program elements and, more importantly, between the Centers. For instance, the number of electrical wires running between the external tank and the orbiter was more than an order of magnitude greater than between the spacecraft and launch vehicle of Apollo, and for the first time, major fluid systems ran across the interfaces. This represented a formidable increase in the effort required to successfully accomplish the SE&I activity. As previously noted, a new program management structure (Figure 1) was adopted to accommodate the increase. The accomplishment of program-level SE&I was given to a "Lead Center." The program director at Headquarters was still responsible for program budgetary control, Congressional relations and a technical staff sufficient to assure that the program technical activity was being properly implemented. At JSC, which was the Lead Center for the Shuttle program, a Level II program office was established totally separate from the Level III orbiter project office located at the same Center.

The development of the flight hardware was delegated to four project offices with the orbiter office located at JSC, as mentioned above, and the other three—the Space Shuttle main engine office, the external tank office, and the solid rocket booster office—located at MSFC. In addition to the hardware development project offices, a prelaunch processing office was formed at KSC. All of the project offices reported to the Level II program manager for all programmatic direction except budget allocation, which was retained by the program director at Headquarters.

The SE&I activity was delegated to the systems integration office located within the JSC Level II office. The orbiter contractor, Rockwell International, was selected to be the integration support contractor, but to increase objectivity, the integration activity was made a separate exhibit to the contract and technical direction was delegated to the Level II systems integration office. The MSFC Space Shuttle project office appointed an integration manager to manage the integration of the Marshall Space Shuttle projects and to serve as the primary interface to the Level II systems integration office.

The flight hardware developmental delegation of the Space Station Freedom program was formulated in an even more complex manner (Figure 4). End-to-end developmental responsibility for each of the major functional systems was delegated to one of four project offices called work package offices in the Space Station Freedom program. Responsibility for assembling and delivering the flight hardware was broken down by launch elements, again assigned to one of the work package offices. Each of these launch elements incorporates components of most of the distributed systems, necessitating the transfer of an extremely large number of hardware and software items between work packages prior to their delivery to the Government. This resulted in another major increase in the complexity of the program-level SE&I process and directly contributed to the difficulty of implementing a satisfactory SE&I process in the Space Station Freedom program.

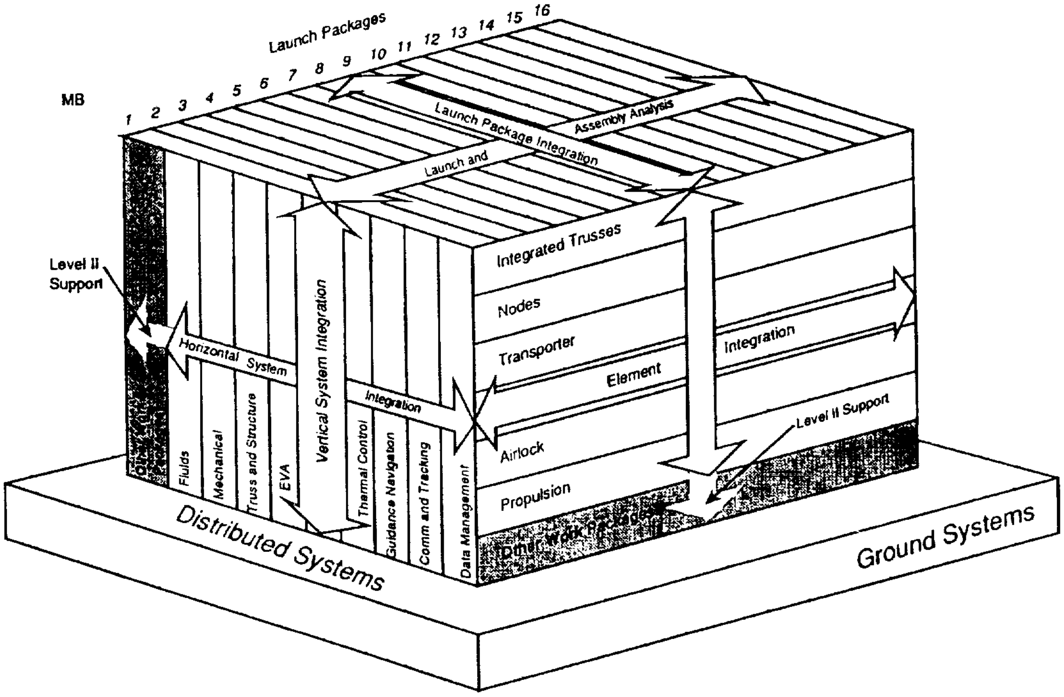


Figure 4 Space Station Integration Job

SE&I SCENARIO

As a program develops from concept to operational status, the characteristics of the SE&I activity vary greatly. Early in the program, conceptual SE&I is intimately involved in defining systems that will meet the overall program objectives and in evaluating the relative merits of each. This is usually accomplished in NASA manned programs by the civil service organizations, often in concert with Phase A/B contracts with industry. After the general systems specification has been developed and a detailed evaluation of systems concepts has been completed, SE&I provides a lead in the preparation of the procurement specifications for the Phase C and D activities and is usually directly involved in the source selection process. After award of the Phase C and D contracts and final selection of the design approach chosen for implementation, SE&I is responsible for preparing system-level technical specifications, which define the performance requirements to be satisfied by each of the major program elements. SE&I then develops the system characterization process to be used (discussed in detail later) and starts an initial analysis cycle. The results of this cycle are extremely important in verifying the validity of the system technical specifications and providing a technical basis for conducting the Program Requirements Review (PRR). After completion of the PRR and updating of the technical specifications, SE&I starts the definition of the interface control document tree and the initial document drafts. Another system characterization cycle is started, based on the updated specifications and the hardware and software concepts chosen to assess the adequacy of the proposed preliminary design approach.

By this time in the program, the ad hoc organizational structure should be well in place and functioning routinely. The communication and management overview provided by this structure of working groups, panels and reviews is central to accomplishing horizontal integration among the project offices and is discussed in more detail later.

In preparation for the preliminary design review (PDR), SE&I defines the minimum content required in the PDR data packages and is responsible for preparing system-level documents supporting the Integrated System PDR. During the PDR process, SE&I representatives participate in the project-level reviews with particular emphasis on the compliance of the project to the system-level requirements. During the Integrated System PDR, emphasis is placed on assuring that the preliminary designs proposed by the projects are compatible across the interfaces and that the integrated system is capable of meeting the operational requirements of the program. The SE&I organization is intimately involved with the evaluation and disposition of review item discrepancies (RIDs) that are submitted during the review.

As a result of the PDR process, changes to the requirements and modifications to the preliminary design of the elements are incorporated. A new characterization cycle is then initiated to evaluate the compatibility between the modified requirements and proposed system capabilities. At this time, the drafts of the interface control documents are expanded and quantitative detail is added to assure that the documents are mature enough to become baseline requirements in the program. This maturation process inevitably results in the identification of physical and functional disconnects among the elements and in a significant number of changes to the baseline.

In a similar manner, the verification plans of the elements and the integrated system are refined and baselined. The responsibility for executing the test and analysis required by the integrated system verification plan are delegated to appropriate organizations that prepare detailed plans for accomplishing the assigned portions of the verification.

Detailed mission operational scenarios and timelines are prepared by the operations organizations, and the operations and SE&I organizations jointly conduct an analysis of the system capabilities to support the scenarios. Concurrently, the acceptance test and prelaunch operations requirements and plans are prepared and delegated for execution.

In preparation for the critical design review (CDR), another system characterization cycle is performed, based upon the detailed design of the elements. This cycle typically uses mature models to synthesize the hardware and software systems and also incorporates the results of tests performed to that time. SE&I participates in the conduct of the CDR in a manner similar to that of the PDR. After completion of the CDR, the system requirements and design changes resulting from the CDR are incorporated into the documentation, and another complete or partial system characterization cycle validates the decisions made during CDR.

After CDR, the primary activity of the SE&l organization is to analyze test results and conduct analysis to verify the capability of the system that is being manufactured. Particular emphasis is given to verifying the interface characteristics of the elements as defined by the interface control documents. This activity directly supports the preparation for the design certification review (DCR), and provides interface information necessary to allow acceptance of the system hardware and software by the Government.

The DCR is conducted similarly to the PDR and CDR but addresses the as-built hardware and software. Successful completion of the DCR certifies the acceptability of the as-built elements and the ability to be integrated into an overall system that will satisfy the initial program operational requirements. Final operational certification of the system is obtained by a combination of the DCR process and analysis of information obtained during early flight operation of the system.

The SE&I organization's participation throughout the program development cycle supports operational planning and real-time operations. SE&I is the repository of corporate knowledge of the details of system capability, which is vital to the effective and efficient operation of the system.

Relationship of SE&I to Other Program Functions

To effectively accomplish the SE&I task, the SE&I management organization must maintain good communications and obtain the support of other program office organizations. Some of the more important interactions are discussed below.

Configuration Management. The interaction between SE&I and configuration management is particularly strong. As the developers and keepers of the systems specifications, SE&I has an interface with the configuration management function that is extremely active throughout the life of the program. The SE&I office recommends the baselining of the technical requirements as they become sufficiently mature and then serves as the office of primary responsibility for defining and evaluating most of the proposed changes to this baseline. The SE&I office, after proper coordination throughout the integration function, also recommends the processing of noncontroversial changes outside of the formal control board meetings, where appropriate. This significantly reduces the board's workload and conserves the time of the key managers who are members of the change control board. As significant issues are referred to the board, SE&I presents an analysis of the issues involved and makes appropriate recommendations for action.

Program Control. SE&I supports the program control function in the development of program schedules and budgets. The key to making this support effective is the use of the SE&I logic networks and estimates of the manpower required to accomplish the activities. Because of SE&I's interdisciplinary nature, SE&I can assist in planning activities in many areas of the program.

Early in the program, SE&I helps define the content and schedule milestones of each project to permit coherent development of project-level schedules and cost estimates. SE&I also provides program control with the engineering master schedules (EMS) and associated budget estimates for incorporation in the overall schedule and budget system. SE&I also works with program control in planning major program reviews; provides technical leadership in conducting the reviews; and frequently chairs the screening groups and pre-boards.

Operations. In all of the NASA manned space programs to date, the SE&I function has been managed in an organization different from the operations definition and planning function. Although this is undoubtedly the best choice in the later phases of the program, it may result in a less thorough incorporation of operational requirements in the systems specifications and other SE&I products early in the program. It may be desirable to combine the management of SE&I and operations in the same office early in the program and then separating them later, perhaps at the completion of the preliminary design review. The stated reason for separating the functions in the past has been that they serve as a check and balance on each other; however, the separation also disconnects the detailed interfaces between the two functions.

SR&QA. The interactions between SE&I and the system reliability and quality assurance (SR&QA) functions depend on how responsibility for executing the program is delegated. If a large part of the SR&QA activity is accomplished within the SR&QA organization, SE&I is used as a reservoir of information or to perform specific tasks as requested by SR&QA. However, if the SR&QA office is responsible for setting the requirements for SR&QA activities and for evaluating the outcomes—while other organizations are delegated the responsibility for executing the work—then SR&QA must define and obtain baseline approval of task requirements, monitor execution of the task by SE&I, and evaluate the results to assure satisfactory achievement.

The former mode of operation was exemplified during the early Apollo program, in which the SR&QA activities were largely accomplished within the SR&QA office using basic engineering information obtained from SE&I and other program organizational offices. Later in the Apollo program, the second mode of execution was adopted; the engineering offices, primarily SE&I, actually performed the work and made a first-level analysis before formally transmitting the results to SR&QA for authentication. This latter method was considered more effective primarily because problems and discrepancies were often discovered by the originating engineering office and corrected even before the task was completed.

SE&I EXECUTION

Techniques developed in past NASA manned programs have proven effective and have become an integral part of implementing SE&I activities. The following paragraphs describe, in no particular order, some of the most important techniques in planning and implementing new programs.

Importance of SE&I Early in a Program. In the early stages of complex programs, comprehensive SE&I support helps determine the architecture to be used to delegate project responsibility. This is accomplished by dividing the program into the next lower level of management, the project offices. The primary outputs are comprehensive and clear program requirement specifications, identification of major programmatic interfaces, development of the ad hoc SE&I management structure, definition of operating concepts, and preparation of initial specifications for the hardware to be delegated to each project office.

The SE&I organization is responsible for managing technical integration both vertically between different levels of the management organizations and horizontally across the organizations at each level. To efficiently achieve both dimensions of integration, it is necessary to develop logic diagrams of the major SE&I activities to be accomplished by each of the organizational elements and then to determine the interrelations between them. By developing these diagrams and playing them against different organizational structures, it is possible to evaluate the proposed organizations in simple terms and easily define the interactions between the organizational elements, thus helping to choose the most efficient management structure. The importance of the logic diagrams will be discussed later.

Development and Use of Ad hoc Integration Structure. To manage the definition and implementation of the SE&I activities in manned space programs, NASA has developed an effective ad hoc organizational structure. The structure consists of a series of reviews, panels and working groups that address the definition and management of integration functions throughout the program. Each organization has members who represent all of the organizations interested in the particular integration function being managed. In the Space Station Freedom program, the working group structure is formed by technical disciplines and distributed systems, such as Guidance, Navigation and Control, Robotics, and Loads and Dynamics. The panels are formed to address specific programmatic management areas (i.e., assembly requirements and stage definition, system design integration, and element design integration) that span a number of organizations. The reviews are formed to address relatively broad program areas as shown in Figure 5.

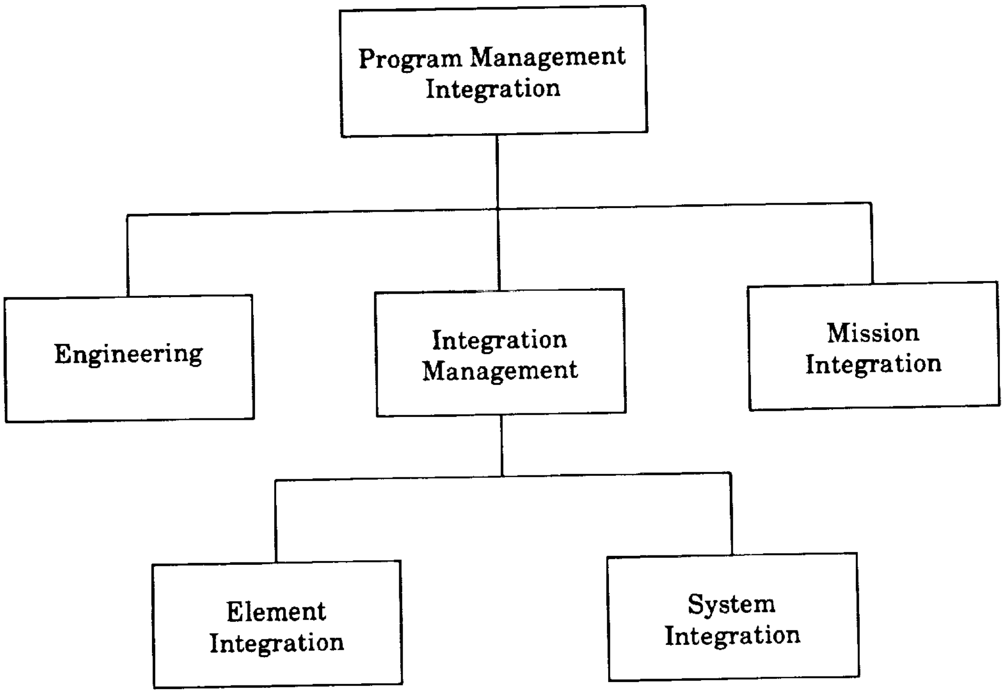


Figure 5 Space Station Freedom Technical Review Structure (1990)

Each organization is responsible for developing the integration plan in its area of responsibility, monitoring the execution of the tasks, identifying problem areas, and either resolving them or submitting them to the overall program management structure for resolution. Although these organizations by their nature do not perform work, the members, by working back through their functional organizations, greatly influence the work being accomplished in their particular area of expertise. As rapport develops between members, many potential problems and issues are identified and resolved without being referred to formal management decision channels. In addition, the quality of the work materially improves. This ad hoc organizational structure also provides obvious places for program elements to present any issue for deliberation and resolution. All of the panels and working groups support each review as needed, and submit their open issues to the most appropriate review for resolution.

The reviews address broad issues and serve as a communication channel between the panels and the working groups. Since the reviews cover all of the panels and working groups, they provide an excellent way of assessing and recommending to management the interdisciplinary priorities of the program.

Chairpeople of the panels and working groups are the most qualified individuals available in a particular discipline. Only secondary consideration is given to selecting a person from a specific organizational element. As a result of their recognized stature, the chairpeople provide leadership, which makes their recommendations and decisions more credible. The panels and working groups also call in outside expertise when needed, but such outside inputs are filtered by the panels and working groups before making a recommendation to the reviews or other management organizations.

Internal vs. Matrix SE&I Staffing. As already noted, SE&I has been staffed and accomplished in different ways in different NASA manned programs. In the early manned space programs, the personnel required to accomplish the SE&I activity were assigned directly to the program and project offices. During the Apollo and Shuttle programs, the program office had only the people necessary to manage the SE&I activity, and most of the work was accomplished by technical experts assigned from the Centers' functional organizations in a matrix fashion. Although each method has its advantages and disadvantages, the matrix approach generally has more advantages in that manpower can be increased or decreased as needed by pulling support from the matrix organizations without reassigning the people involved. The primary disadvantage is that the leader of a particular area does not report functionally to the program or project office, which means that the line of direction is not as strong. The importance of this negative factor, however, is inversely proportional to the working relationship between the organizations. In the Space Shuttle program, this relationship and the matrix approach worked well. In other programs, the relationship was not as good and direction through the matrix was less effective. On occasion, program management appointed all panel and working group chairpeople from the program office staff, giving less regard to the individual's personal qualifications. This led to a marked decrease in the stature of the ad hoc structure, which then resulted in a lack of support from the functional organizations and a decrease in the quality of the integration activity and products. As in many areas of SE&I, effective implementation relies heavily on the quality of the leadership and the maintenance of free and open communications among the organizations involved.

2

Logic Networks. As the NASA manned space programs have become increasingly complex, it has become difficult to define the specific content and tasks needed to accomplish the SE&I function. Central to the development of a comprehensive SE&I plan is the development of detailed logic networks, which form the basis for planning, executing and evaluating the SE&I activities.

As used in the Space Shuttle program, logic networks covered all of the SE&I activities that had to be accomplished by all elements of the program organization. Thus, these networks were able to interrelate SE&I activities both vertically and horizontally throughout the program management structure. The basic summary logic networks were developed for the entire program duration, to identify all major activities required as a function of time, and were instrumental in developing cost and manpower forecasts for the entire duration of the program. Detailed logic networks were then prepared for the near-term in the Shuttle program for 12 months, identifying in greater detail the specific activities to be accomplished by each organizational element during that period. The networks were revised every six months to extend the detail planning horizon; in addition, the summary networks were reviewed and modified as needed on an annual basis. The logic networks were a primary input to the development of the engineering master schedules discussed in the next paragraph.

Engineering Master Schedules (EMS) and Associated Dictionary. The activities identified in the SE&I integration logic networks were then assigned to specific organizations for execution and presented as a schedule for each organization involved. By using a numbering system for the activities, the logic network and the schedule could be easily correlated. The schedules allowed cost and manpower estimates to be prepared for each organization and provided an excellent means of determining status and managing activities in real time.

Associated with the EMS, a dictionary was prepared with an entry for each activity. Each entry identified all input information required to allow the accomplishment of the activity; described the contents of the products; and identified the primary user of each product, the scheduled completion date, and the person responsible for preparing the product. The EMS and the dictionary were the primary tools for defining and communicating SE&I activities throughout the entire program structure.

As would be expected, the content of the EMS changes character over the life of the program and accordingly, requires various technical capabilities over time. Early in the program, the design activities involve a large number of trade studies and the development of synthesis tools to be used in evaluating the capabilities of the proposed design. As the program matures and the design solidifies, the activities become more involved with exercising the system models, conducting tests and analyzing data. As the flight phase approaches, the activities are predominated by operational considerations, including the development of operational data books, mission requirements, certification of system readiness, and support of mission planning and real-time mission operations.

System Characterization Process. A major SE&I activity throughout the program life span is the assessment of the capability of the system to meet specified requirements. In the NASA manned space program, this has been accomplished in an analytic sense by synthesizing the vehicle characterizations in the form of either models or simulations, and then developing detailed performance characterizations by exercising the models against selected mission timelines and significant mission events.

The methodology used to perform the system synthesis is central to the development of the logic networks and schedules described earlier. An examination of the system usually reveals scenarios useful in conducting the overall system evaluation; after selecting the most desirable scenario, it forms the nucleus of the overall SE&I logic. In the Space Shuttle program, the scenario chosen was (1) develop the necessary models and simulations; (2) determine the structural modal characteristics; (3) determine the loads on each of the system elements; and (4) perform stress analysis of the system when subjected to these loads. Using this scenario it was relatively easy to define and interrelate the SE&I activities of other disciplines, such as GN&C, propulsion, and thermal, among others. After defining all of the required activities, a document was prepared to identify the models to be used, and the mission events to be analyzed and to define the configuration to be used. The sequence described above formed an analysis cycle of a specific configuration subjected to specific operational requirements. In the Shuttle program, it was termed an integrated vehicle baseline characterization cycle (IVBC). As previously described in the SE&I scenario, several characterization cycles are needed during the program: as the program matures, the cycles have additional synthesis detail, more definitive configuration information, and better operational information.

At the completion of each of the characterizations cycles, system deficiencies are identified and modifications to either the system specifications or the requirements are made. For program management purposes, it is usually convenient to schedule the completion of one of the characterization cycles to occur just before each of the major program-level review milestones.

Program Reviews. SE&I has a large input to each of the program-level reviews, such as system requirements review, preliminary design review, critical designreview, design certification review, and flight readiness reviews. As mentioned above, completion of one of the system characterization cycles is an excellent indicator of whether the system design meets the specified requirements. The engineering master schedule gives a graphic representation of whether the integration progress is being achieved. Reports produced by the SE&I activity, such as resource allocation status and margins, interface control document status, design reference mission maturity, and system operational data books indicate the maturity of the element participation in the system-level SE&I process.

Design Reference Missions. Most of the manned space programs had to be capable of performing a relatively large number of diverse missions, and the specifications are written to allow hardware and software systems and elements that are flexible enough to satisfy all of the missions. For analytical purposes, however, it is convenient to define and adopt one or more design reference missions (DRMs) that stress all of the systems capabilities to a significant extent. The DRMs are used as the primary mission requirements in the system characterization cycles, and in evaluating the ability to meet performance specifications. In addition to evaluating the baselined configuration against the DRMs, other specification requirements are evaluated by the accomplishment of specific analyses or tests, as necessary.

The DRMs also allow the user community to evaluate whether the system is capable of meeting specific user needs and whether these needs are specifically in the system specifications. The DRM is used by mission planners to determine the system's capability of performing any specific mission under consideration.

Verification. Verification plays a major role in program planning and in the ultimate cost of the system. Although most of the verification is delegated to projects, SE&I is responsible for identifying the overall verification requirements and specific system-level verification tests and simulations, which frequently require specialized facilities and significant amounts of system hardware and software. Since these system-level verification tests are frequently complex and expensive, planning for them needs to start very early in the program. The system-level verification network is developed as an integral part of the program SE&I logic networks and is baselined early in the program.

Final verification of some system requirements can only be accomplished in the real flight environment, and these are demonstrated in early operations before final certification of system operational capability is accomplished. It is also important to integrate the system-level verification planning and the operations planning to promote the maximum synergism possible between system verification and operational training.

In manned space programs, all of the major system level verification tests have been assigned to program or functional organizational elements other than SE&I for implementation. This has helped to assure that the management of SE&I can remain objective in the evaluation of overall certification adequacy.

DCR Process. One of the most significant activities of SE&I its role in the certification of the system design prior to the start of the flight operations and then later, prior to committing the system to operating throughout the entire design envelope. SE&I is instrumental in setting the overall requirements for the DCR and is directly responsible for the system-level portion of the review. This process becomes the final major system characterization cycle, using a synthesis of the as-built vehicle hardware and software capabilities and results of tests and analyses. DCR results also form the basis for the system operational data books that are used to plan and conduct the operational phase of the program. The DCR requires that all system requirements be evaluated against all of the as-built system capabilities, and where possible, the system margins are quantified to assist the operations organization in planning and conducting flight operations.

ICD Development. As the program management organizational structure is determined and responsibility for developing hardware and software is delegated, it is necessary to start the development of the interface control document (ICD) tree, which identifies each required ICD and the content to be presented. As previously noted, 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. The early development of strawman ICD trees can greatly assist in optimizing the overall program management structure. As the program progresses and the system configuration becomes better defined, the content of each ICD is developed in more detail and ICD working groups are formed to quantify the environmental, physical, functional and operational characteristics in detail. In most manned programs, the ICDs have been baselined at a relatively early point in the program and have usually contained a large number of TBDs (to be determined). After baselining the ICDs, working groups continue their work to arrive at specific values for each of the TBDs and to continually assess the adequacy of the ICDs as the design matures.

The ICDs are primary documents at each program review and provide a basis for evaluating the adequacy of the items being reviewed to satisfactorily function as part of the total system.

Program Management Organizational Structure. The efficiency of program management is greatly influenced by the organizational structure selected. Organizational structures that are compact and simple promote effective program management. Compactness is measured vertically by the number of levels of the program management organization and horizontally by the number of organizations at each level. Each additional organizational element significantly increases the manpower and costs of achieving program integration, including SE&I. If each organizational element must interface with all others in the program, the number of interfaces increases rapidly as organizations are added. Adding management levels increases the complexity for delegating the execution of the program. This factor was evident to the Augustine Commission in their recent summary report *The Future of the U.S. Space Program,* in which they recommended that "multicenter projects be avoided wherever possible, but when this is not practical, a strong and independent project office reporting to Headquarters be established near the Center having the principal share of the work for that project; and that this project office have a systems engineering staff and full budget authority."

In addition to keeping the management structure compact, it is also very important to select an architecture that divides the program into project offices, to enable simple interfaces between projects and delegation that is all-encompassing. All of the deliverable hardware assigned to a given project should be the responsibility of that project to design and manufacture. In all manned programs prior to the Space Station, there was little transfer of hardware and software between projects—with one exception, that being the development flight instrumentation in the Apollo program.

Early in Apollo, a decision was made to establish a civil service project office to develop, procure and deliver the specialized development flight instrumentation to the prime spacecraft contractors for installation and integration in the early spacecraft. Coordination of the large volume of interface information required the development and maintenance of the complex bilateral schedules and support required. The complexity of providing support after the transfer of the instrumentation was a significant management problem throughout the entire time that the development flight instrument was used. In the Space Station Freedom program, considering the many hardware and software items that must be passed between work packages, it will be difficult to develop, coordinate and maintain all of the interface information required.

Objectivity In Management. To promote objectivity in managing SE&I, one of the basic ground rules in the Shuttle program was that the SE&I function would not be responsible for the development of any flight hardware or software products; thus, they had no conflicting pressure to make their development job easier at the expense of another organization. It was found that any bias, either perceived or real, immediately brings the objectivity of management into question and rapidly destroys the confidence between organizational elements.

Need for Good Communication. The nature of SE&I is such that most of the program elements and many other agency organizations are involved in the execution of SE&I tasks. To facilitate accomplishment of the work, the importance of free and open communication cannot be overstressed. One of the ways of accomplishing this is "to live in a glass house." All decisions and, of equal importance, the logic behind those decisions must be communicated to all parties involved if they are to understand their role and how it fits into the overall picture. All parties must feel that their inputs are included in the decision-making process. This openness, and the accompanying feeling of vulnerability, is often not welcomed and requires faith and confidence between the organizations involved. The fact that mistakes will be made must be accepted, and all organizations involved must constructively assist in correcting them. Frequent open meetings of the ad hoc organizational elements described above have proven to be an effective tool in developing rapport between peers and communicating information and decisions throughout the program structure. As noted earlier, however, such meetings become increasingly time-consuming and expensive as the complexity of the organizational structure is increased.

Importance of Margins. At the time programs are initiated, they are frequently sold on the basis of optimistic estimates of performance capability, cost and schedules. This often results in reducing margins to low levels at program initiation and solving early program costs and schedule problems by reducing weight, power and other resource margins. As a consequence, margins are reduced to zero or negative values early in the program, making it necessary to modify the program to either reduce requirements or introduce program changes that will reestablish positive margins. The recovery of the margin inevitably leads to significantly higher ultimate program costs in both dollars and days. Minimum life cycle costs are achieved by holding relatively large margins early in the program and then allowing them to be expended at a prudent rate during the program life cycle.

THINGS THAT HAVE WORKED WELL

In the management of the manned space programs' SE&I activities, several approaches have been particularly successful. Some of the most important, have been discussed previously but are readdressed here because of their assistance in the management of SE&I.

Ad hoc Organizations. The use of ad hoc organizations to coordinate SE&I activities has proven to be a valuable tool. The effectiveness of SE&I depends heavily on good communications between organizations and the assurance that all organizational elements take a common approach to the implementation of SE&I. This is difficult to accomplish using the normal program office organizations because they cannot directly address inter-organizational communications and have difficulty managing across organizational lines. The ad hoc organizational structure, on the other hand, is made up of specialists from each of the affected organizations, and their activities directly promote inter-organizational communications. Using this technique, technical peers can plan and monitor the execution of specific SE&I activities. When a resolution cannot be reached within the ad hoc organization, the issue can be referred to the proper program management office for decision.

Standard Organization Structure within the Program and Project Offices. During the Apollo program, the program director decided to have all of the program management offices at both Level II and Level III adopt a standard organization structure: five offices reported to the program manager and the same five offices reported to each project manager. This technique assured that the work breakdown structure was similar for all offices, that direct counterparts could be identified in each of the offices, and that budget allocations flowed down in a uniform and predictable manner. All of these features resulted in less cross-linking between organizations and made the required program management activity more rational and predictable. Although the specific office structure chosen would be different for each program, having uniformity between the Level II and Level III management offices should be considered for future programs.

System Characterization Cycles. Constructing the SE&I plan and identifying the required tasks is a very complex undertaking in large programs. As previously described, it is best to have a well-defined core of activity that, when completed, will characterize the capability of the system to meet the specified requirements. Analysis of the results reveals deficiencies and allows modifications to either the requirements or the system design to be identified, thus assuring an adequate margin of performance. Building on this core analysis cycle, it is relatively easy to plan the other SE&I tasks in a consistent manner, and create a complete characterization of the system capability.

Matrix Management Organizational Approach. The concept of staffing the program management office with a small number of people who serve as managers only and then augmenting their capability with personnel drawn from other Center organizations in a matrix fashion has significant advantages. Manpower can be brought in from the organizations only when it is actually needed, and the technical composition can be changed over time to satisfy programmatic needs. The quantity of personnel can be augmented to meet program needs, i.e., during major program reviews; the personnel involved can be assured of a career path in their parent organization; and the individuals involved can continually replenish their expertise by participating in the R&D activities of their parent organization.

This mode of operation has been quite successful and has demonstrated several additional advantages, such as reducing friction and undesired competition between the program office and Center functional organizations, improving technical communications across programs being implemented simultaneously, and providing an efficient way of phasing the development program into an operational role. In particular, the assignment of program-level SE&I to a Lead Center, coupled with the execution of this assignment using Center functional organizations in a matrix fashions, allows the program to take advantage of both the quality and quantity of technical expertise available throughout the Center.

Use of a Prime Development Contractor to Provide SE&I Support. In the Shuttle program, the SE&I support contractor was also the prime contractor for the development of the Space Shuttle orbiter. Although there was considerable concern about the ability of the contractor to maintain objectivity in supporting SE&I, this concern was reduced to an acceptable level by separating the direction channels of the development and integration activity both within NASA and within the contractor's organization. The support contract was also set up with an award fee structure in which SE&I was responsible for providing inputs for the SE&I activities. There were many advantages in this arrangement:

a) The integration personnel were familiar with one of the major program elements and did not need to become familiar with that element or the general program structure.

b) Technical experts could be made available for both activities as needed.

c) Many of the synthesis tools required by both activities were similar, and frequently one model could be used for both purposes with only minor modifications.

d) Uniformity in approach assured ease of comparison of results from both project-level and program-level activities.

The management of SE&I in NASA manned space programs has developed over the last 30 years to satisfactorily integrate relatively complex programs. Some of the approaches and techniques described in this paper may be helpful in integrating future programs. Careful consideration of the organizational structure and systems architecture at a start of a program has an overriding effect on the effort required to accomplish the SE&I activity.

THE SE ROLE IN ESTABLISHING, VERIFYING AND CONTROLLING TOP-LEVEL PROGRAM REQUIREMENTS

by Charles W. Mathews

People working in the field of systems engineering have differing views as to the range and depth of this subject. Without venturing into the controversial arena of specific definitions, I will assert that systems engineering has much to do with the definition, evaluation and control of the technical effort aimed at achieving the objectives of a program. Efforts in the field of systems engineering may in fact go well beyond purely technical considerations, e.g., when cost or political considerations impact the technical approach to a program. In this context, the systems engineering process must function to maximize the probability that a program's technical requirements can be met while at the same time recognizing and including other program factors and constraints. New constraints as well as technical problems can be encountered at all stages of a program, often necessitating some adjustment to the program objectives and requirements. Such activities are part of the systems engineering process, which must begin immediately at the start of a program and continue throughout the life of the program.

Sometimes a program manager will concentrate on insuring that hardware elements perform well and all play well together, assuming that this alone will enable the program requirements to be met. Then on entering the operational phase, while the system may indeed perform, it may not do what was intended. This situation frequently occurs because many engineers, scientists, managers, and yes, even administrators tend to be intrigued by and want to concentrate on configuration selection and design problems. It is the responsibility of the top-level systems engineering professionals to be the conscience of all participants in making sure that program requirements are met or properly adjusted.

The need is to focus on program requirements during all phases and facets of a program, e.g. definition, development, manufacturing, testing, operations, growth and, most important, effective use or mission accomplishment. The effort just described involves the entire systems engineering task; however, the main emphasis of this paper is the interaction of the systems engineering process with the top-level program requirements. This aspect of systems engineering is often given inadequate attention during certain phases of a program. This paper will endeavor to answer such questions as:

What is meant by top-level program requirements, and who generates them?

How are these requirements validated, altered, and controlled by the systems engineering process?

What capabilities are needed to accomplish such efforts effectively?

WHAT ARE TOP-LEVEL PROGRAM REQUIREMENTS?

Top-level program requirements are directly related to program objectives or systems uses determined and stated early during the program definition. Probably the most remembered program objective of the past was to "land men on the moon and return them safely to Earth." The program requirements that emerged from early studies included, among others, one to two-week mission durations, lunar landing, extravehicular activities, launch from a remote site, rendezvous, and reentry from near escape velocity, all of which had never been accomplished at the time of President Kennedy's statement.

These requirements in turn highlighted the need to define and validate specific technical approaches—redundancy concepts, simple system interfaces, new technology requirements (e.g., fuel cells), operational demonstrations such as Gemini, entirely new configurations such as the LM, and the nature of the flight program buildup. Incidentally, many of the program requirements for Apollo determined the mission objectives for the earlier Gemini program. In any event, program requirements must be established early and stated distinctly so that all necessary steps for meeting and validating them can be determined. This effort is a fundamental systems engineering function.

Types of Program Objectives and Requirements

The program objectives and requirements described in the preceding paragraphs emphasize mission demonstrations. Obtaining desired science or applications information is another type of program objective. The program requirements then state the need for specific data, usually specifying a particular instrument or instrument set; the operating conditions under which the data is to be obtained (e.g., orbit altitude, field of view, and pointing accuracy); and the data handling and use. Conversely, a new instrument may be conceived or created with the program objective to establish its use potential. The Multispectral Scanner employed in the Landsat program is an example.

Another space program category includes service functions such as Earth-to-orbit transportation or a space laboratory. In the first case, the program objective might be economical and an easy access to the space environment for the using community. Program requirements then include such parameters as dollars per pound to orbit, launch frequency and payload integration lead times. Conversely, in this case, the program objectives might also be stated in terms of capability demonstrations such as the reentry of a winged spacecraft, ground landing and reusability. The program requirements then are related to system performance in accomplishing these mission and configuration demonstrations.

It is important to firmly establish which of the above two categories reflect the real program objectives because a capability demonstration has a higher potential for success than a tightly specified use commitment. The systems engineering organization should be providing top-level program management with the information to make such determinations. The program objectives may vary during program implementation because of early "selling" pressures or because of unforeseen technical problems When this happens, the systems engineering organization should provide concrete evidence to management so that a strategy can be developed to properly inform the outside world, e.g., Office of Management and Budget (OMB), Congressional committees and the media; if the outside elements are not made to understand and accept such changes in a timely way, support can be alienated, placing extreme pressure on the program.

Establishing Priorities

When a large number of objectives and associated requirements are included in a given program, an additional complication occurs. Several past programs qualify including programs as early as Gemini and space station programs such as Skylab. Even Apollo, with its simply stated mission objective, had many secondary objectives associated with lunar exploration and lunar science. It is very important to establish priorities without precluding the accomplishment of objectives of lower priority. For example, the two top priorities in the Gemini program were demonstration of long duration flight and rendezvous, but large quick-opening hatches were incorporated to accommodate extravehicular activities (EVA) and the spacecraft structure was designed to permit the firing of a large propulsive stage once docked to it. Most of these secondary objectives were accomplished. In fact, because of the way the actual flight program developed, EVA was one of the first accomplishments. The secondary program objectives also afforded some flexibility; the paraglider system planned for use in ground landing, for example, was dropped from Gemini in order to meet cost and schedule objectives.

To summarize what has been stated thus far, a number of classes of top-level program requirements exist. They can be associated with mission objectives, scientific investigations or space services, among others. In addition, different ways of looking at top-level program requirements include demonstrations as compared with tightly specified commitments. Many programs have multiple requirements. Nevertheless, it is important to 'zero in' on these requirements early in the systems engineering process, i.e., during Phase A. Most important, they must combine to realistically meet the stated objectives of the program; they must be prioritized when necessary; and they must be clearly stated and documented in the Program Requirements Document.

These requirements may have to be changed, adjusted or reprioritized as the program proceeds, and any changes must be carefully controlled and formally approved at the top level of the program throughout its life. If program objectives are affected, a decision by the administrator is required (at least for medium-to-large programs). The outside world needs to be kept abreast of significant changes in objectives or top-level requirements so that no sudden surprises occur that affect support.

The systems engineering function should provide the initial evaluation and validation of the top-level program requirements and should continue to evaluate proposals or events that would produce any change. The effort should occur at the top level of a distributed systems engineering function and guide upper level program management and the administrator.

WHO GENERATES TOP-LEVEL PROGRAM REQUIREMENTS?

A program objective can be conceived and stated initially by almost anyone working at any level, from the President, as in Apollo, to others on down. If considered seriously, such an objective is studied to determine its validity, practicality and usefulness. Sometimes it takes a short time to obtain a go-ahead; sometimes it takes many years, as on the Space Station. One of the fallouts of these efforts should be a clear statement of top-level program requirements.

The involvement of the right people in the generation of top-level program requirements is extremely important. Depending on the nature of the program, this involvement can include customers, users, operators and, of course, designers and developers. Program managers and directors, however, should guard against limiting involvement in this activity to just the latter two. Systems engineering, should be involved early to assure a reasoned and logical approach to the generation and iteration of program requirements.

In the space science and applications arenas, program requirements are frequently generated by a process that begins with a program objective or a flight system capability being stated in an "Announcement of Flight Opportunity." Investigators are then selected through evaluation of the responses obtained. The experiments selected determine the actual requirements of the flight program. Other inputs are often required, as adjustments may be needed in consideration of technical limitations or program costs, for example. The analysis and resulting output of the systems engineering group usually gives rise to an iteration of the program requirements, which again involves the science team. Frequently, a selected proposal provides for excellent science but does not deal adequately with other constraining technical considerations and the cost implications associated with the overall effort.

Hierarchical Consideration in Requirements Generation

In all classes of space flight programs, the systems engineering organization should work closely with groups having expertise in and cognizance over program requirements. In Apollo, because the primary program objective was oriented to the accomplishment of a specific mission demonstration, operational personnel—particularly those involved in flight operations—tended to be near the top of the program requirements hierarchy. Even though science requirements existed and science teams and advisory committees were active, the science requirements were of lower priority, at least until after the first lunar landing was accomplished. In contrast, a program such as Skylab always included the solar scientists and Earth resources investigators, among others, at the top of the requirements hierarchy, even though the engineering and operations personnel may have been somewhat confused by this arrangement.

The Space Shuttle involves still another situation. The operations groups can be perceived to be the customers for the system, but the real users at the top of the hierarchy are the scientists, commercial firms, industrial experimenters and NASA engineers who provide the payloads that fly on the Space Shuttle or conduct related experiments or other use functions. This is similar to the relationship between passengers and shippers, the airlines, and the commercial airplane developer in the air transport industry. In addition to general operating efficiency, consideration must be given to user accommodation from the start. Such needs are now quite successfully accomplished in commercial aviation. Naturally, expectations are less in the case of the Space Shuttle because of its experimental nature, but it is fair to say that user accommodation has not been accomplished to the degree desired.

The foregoing discussion is not meant to imply that successful hardware design, development and systems integration is not an important facet of systems engineering. There are instances where these considerations are at the top of the requirements hierarchy. An instrument demonstration such as the Multispectral Scanner is one case in point, and the Advanced Communications Technology Satellite (ACTS) is another technology demonstration of this type. In most respects, the research airplanes such as the X-1 and X-15 fit into this category. However, this case does not fit the situations occurring in most NASA programs. It is therefore critical for top-level program management to examine its program, determine who the main contributors or generators of the program requirements are, and assure that they are interfacing adequately with the systems engineering function. This need exists at the outset of the program but should continue through the design and development phases, for as hardware and software systems problems are encountered, the tendency is to focus on them, and top-level program requirements can be altered or even disappear without due consideration.

WHO VALIDATES TOP-LEVEL PROGRAM REQUIREMENTS?

Activities that validate top-level program requirements are mostly of a systems engineering nature. This validation, is an important, though small, part of the total systems engineering job. In total, systems engineering, particularly during design and development, is a distributed activity. Spacecraft hardware systems such as electrical power, attitude control and communications all have to be systems engineered. Total systems elements (e.g., a launch vehicle stage, a checkout facility, a launch complex and a flight control center) all have to be systems engineered to correctly perform their functions. In the end, all elements involved in a program—the total flight system, the operational support facilities, the mission planning, and the user integration, among others—need to be brought together in a timely fashion to meet the program objectives and requirements. An effort of this nature, even for a very modest program, is too complex to be handled in a purely top-down fashion. The cardinal rule is that all the interfaces at any particular level, both horizontally and vertically, should be as clean and simple as is practical.

Validation Efforts During Program Definition

At the start of program evolution, practically all of the mainstream effort is of a systems engineering character and is more top-down than later in the program. The validation effort begins in pre-Phase A, where options are examined for meeting the program objectives as well as certain initially stated program requirements. These requirements should endeavor to incorporate most of the major program factors but are usually general and often are quite optimistic. All aspects of the technical and programmatic approach should be studied. Although effort is limited in this phase, a determined attempt must be made to establish and to ascertain the feasibility of meeting the program requirements. This work should usually be accomplished by a team working at a single location, although supporting effort and information can be obtained from groups in other locations. There have been cases where alternative approaches are studied by separate teams, which has proved to be effective in some pre-Phase A efforts. In all likelihood, the program requirements will be changed and expanded to account for such factors as technology readiness, knowledge of the operating environment, mission complexity and similar factors. A need for additional technology development or operational verifications may be identified as well. Any pre-Phase A study that is completed with everything looking rosy should be viewed with caution.

Phase A efforts are aimed at selecting and analyzing a single programmatic and technical approach, at least in theory, to best meet the requirements of the program. Again, the Phase A activity is chiefly a systems engineering effort usually conducted by a single team at a single location. If a work breakdown structure with clear interfaces can be established at this time, then systems engineering at multiple locations may be possible. In any case, the group that worked during the pre-Phase A study needs to be augmented considerably, and the support of one or more contractors is frequently obtained.

In this phase, emphasis should be placed not only on hardware but on validating the mission design and other operational or use aspects of the program. This emphasis is particularly important where the operational life of the program is envisioned to be very long, e.g., Space Shuttle, Hubble Telescope, Space Station and the Earth Observing System (EOS). It is important to clearly establish what is required in the operational phase and to establish with adequate confidence the feasibility of accomplishing the programs with realistic operational costs and schedules.

At the time the program enters Phase B, a complete work breakdown structure should be established, including all facets of the program with simple and clear interfaces and as little overlap as possible. Program work assignments will be made. For moderate to large programs, these assignments may involve program groups at different geographic locations, including parts of the total systems engineering effort. The top-level program requirements should have been established in adequate detail, and each program organizational element should regard these requirements as program constraints.

The program requirements or even the objectives can be changed because of unforeseen events or other activities occurring throughout the course of the program, but they should be subject to formal change control. Obviously this particular change control activity deals with top-level program requirements and must occur at the highest level in the program; in certain cases, the administrator should be informed of an impending change and must be informed when program objectives are significantly impacted.

Validation Efforts During Design, Development and Operations

Although the top-level systems engineering effort in the definition phases of a program is important, this function is critically important in Phases C/D, the design and development phases. It is during this time that most of the technical difficulties and other program limitations surface. There is a strong tendency to focus on the flight hardware and to get it delivered and flying. These situations sometimes allow the top-level requirements to "fall through the cracks," later producing surprises, embarrassments and undue pressures, which can contribute to the potential for accidents and failures in the operational phase.

Systems engineering must continue throughout the operational phases of a program. Although the character of the top-level activities change, there still is a need to deal with program requirements and their alteration. Some of the possible subjects are the rate and nature of the flight program buildup, working around performance deficiencies or failures, and adjustments to mission objectives. On the positive side, the top-level systems engineering in the operational phase involves the incorporation of new system capabilities and mission extensions, including the development and control of the associated program requirements.

Support to the activities just described is accomplished by a systems engineering group also operating at the highest level in the program's organizational structure. This group is the guardian and conscience of the top-level program requirements but by no means includes the total systems engineering effort. The group should be composed of engineering personnel, each of whom has considerable technical experience in one or more of the applicable areas and possesses a natural talent and desire to deal with all aspects of the program. The individuals should be selected so the group encompasses as many of the technical, scientific and programmatic disciplines involved in the program as possible, but the group does not have to be large. By selecting people with the right backgrounds and talents, the work can be done in part by obtaining information from other elements of the program—in particular, other systems engineering groups.

HOW ARE TOP-LEVEL PROGRAM REQUIREMENTS CONTROLLED?

Control of top-level program requirements is extremely critical to program success. This is not to say that such requirements cannot be changed. Almost without exception changes will occur, but they must be carefully controlled by a well-defined process that establishes the change impact on the program, particularly its objectives. This process also must inform program participants inside and outside the program organizational structure, including those having responsibilities or scrutiny from above.

The program director is the individual who is personally responsible for the integrity and control of the top-level program requirements. As such, the program director must assure that a Program Requirements Document is produced during Phase A and that it is properly updated immediately following a change. This effort is supported mainly by the program director's systems engineering group described in previous sections. This group is responsible for analyzing any proposed change that could potentially impact the top-level program requirements.

The analysis can be done by the group itself or by support groups, including contractors. The analysis must specifically include in writing how the affected requirement(s) would be changed and the determination of other impacts such as cost or schedule, which could be either positive or negative.

Change Control of Program Requirements

Change proposals are brought before a standing committee, usually called a change board, selected by the program director. There will be other change boards throughout the program, but this one should deal only with top-level program requirements. Anyone who proposes a legitimate change in the program requirements should be able to come before this board. In general, individuals who have a significant input should also be invited. The proposed change is usually presented by its sponsor and is followed by a presentation of the analysis of the systems engineering group. Following discussion, the program director makes the disposition, which can include acceptance, rejection, or a requirement for further analyses or information. Following an acceptance, the Program Requirements Document should be changed immediately. Regardless of the nature of the decision, the affected elements of the program organization need to be informed immediately. Affected elements outside the program should also be informed in a timely manner but only after an appropriate strategy is developed.

One of the chief difficulties associated with this change control activity is that events that impact the top-level program requirements can occur at any place, at any time and at any level in the program, and there is a natural tendency to try to fix a problem at its source without passing on information. Several things can be done to alleviate this difficulty as it relates to the activities of the top-level systems engineering group. Individuals in the group must attend design reviews and other program reviews associated with all the program elements. They must be able to have free information exchange with other program and project personnel and to accompany them on visits to contractors when the occasion demands. These activities are best accomplished if the group and its members operate with a low profile. They should not give or imply directions or conclusions in discussions with program and project people. All direction as a result of their work should come from the program director. Naturally, these individuals must be able to request and analyze program documentation, but all such activities should be done in a way to maintain good rapport with other groups working in the program.

TOP-LEVEL PROGRAM REQUIREMENTS IN PREVIOUS PROGRAMS

In general, most of NASA's past major programs have successfully met their program objectives and must have fulfilled their program requirements. Some brief observations of the results obtained during some of the previous manned programs may provide useful insight into future programs. Although the very early programs were not explicitly divided into program phases, in retrospect, it is possible to discuss them within a phased context.

The Mercury Program

The Mercury program objective was to place a manned spacecraft in orbit around Earth and return safely. In pre-Phase A, several winged (lifting) configurations were studied as well as the so-called "capsule." The capsule was selected on the basis of greater technical simplicity and limitations on launch vehicle payload capability. In Phase A, in addition to developing the spacecraft systems specifications, safety requirements were emphasized, including the proper positioning and support of the crew to handle launch and reentry accelerations, which were demonstrated on a centrifuge; the concept of a system to escape from the launch vehicle if necessary; and the layout of a worldwide tracking and monitoring network. In Phase B, a full-scale demonstration of the reentry heat protection system was conducted, and the results produced minor design changes. The concepts of flight control and recovery were evolved, including a mission control center and flight controller deployment to remote sites, worldwide communication for near real-time surveillance of the missions, and recovery procedures involving ship deployment.

The spacecraft configuration and specifications proved to be satisfactory although considerable development problems were encountered. The biggest systems engineering problem was associated with the lack of appreciation of the difficulties in conducting factory and preflight checkout. The checkout required more or less continuous human presence in the extremely confining interior of the spacecraft, producing wire breakage and other damage. These conditions were severe enough to curtail the flight program, although six manned flights were made, building up to a duration of approximately one day in orbit.

The Gemini Program

The pre-Phase A activity concentrated largely on correcting some of the basic problems encountered in Mercury, i.e., a Gemini spacecraft design that had most of the equipment outside the pressure vessel and was also checkable from the outside, allowing a relatively clear cockpit area. The spacecraft was enlarged to provide for a two-man crew, but the basic external configuration and heat protection system of the Mercury spacecraft was retained.

Most of the Phase A activity involved defining the mission objectives, in support of Apollo, and the related program requirements associated with rendezvous and long duration flight, e.g., the Atlas-Agena target vehicle, orbit maneuvering system, rendezvous radar, fuel cells, and the cryogenic storage of hydrogen and oxygen in a supercritical state. Again, considerable development problems emerged, largely associated with the newer systems, such as ablative thrusters and fuel cells. Problems were also encountered in the flight program. The initial rendezvous exercise revealed inadequate attention to mission design, which was later corrected, and several classes of rendezvous were successfully demonstrated. The extravehicular activities revealed deficiencies in training, and neutral buoyancy simulation was introduced late in the program.

One significant systems engineering achievement emphasized the checkout systems and checkout procedures, and the delivery of flight ready spacecraft. To gain confidence, many of the checkout personnel at the Cape were sent on temporary duty (TDY) to the factory to participate in the factory checkout of the early spacecraft. This approach allowed the ten manned flights to take place on about two-month cycles and contributed immensely to the on-time launches required for rendezvous.

The Apollo Program

The Apollo Program was characterized by a disjointed definition program. Because of the obvious schedule pressures, certain contracts involving Phase B-type effort were let before either the mission design or the lunar landing mode had been selected. For example, the command and service module contract was awarded while questions about the use of Earth orbit rendezvous, lunar orbit rendezvous, and the so-called direct ascent were still being debated. Sufficient pre-Phase A effort was completed to enable a decision to go with the lunar orbit rendezvous route in the spring of 1962, but the Phase A work on the lunar module, even when accomplished in-house on a highly accelerated schedule, did not allow the lunar module contractor to be selected until nearly a year after the selection of the command and service module contractor. This situation proved to be very distracting to the latter and resulted in major inefficiencies in the contracted effort caused by premature work force buildup.

What saved the situation was the maintenance of simple interfaces between the two spacecraft. In fact, not much more than a docking interface existed; however, there was also an important structural interface recalling that service module propulsion was used to place the docked configuration in lunar orbit. No support was required between the two spacecraft except status monitoring, and no commonality of systems was specified, although by some rationales, this approach appears inefficient. The simple system organizational and programmatic interfaces obtained greatly benefited the program. It was also the approach taken in connection with other elements of Apollo.

The operational phase of the Apollo program provides good illustrations of systems capability extension and mission extensions. The major extensions to the lunar surface stay-time of the lunar module is an example. The decision to accomplish this was made about the time of the first lunar landing, and a Headquarters systems engineering group provided the impetus for the validation. Another capability extension was the addition of the lunar rover contract, awarded about six years after the Apollo start but before the first lunar landing. Both these added capabilities greatly enhanced the lunar surface science and exploration aspects of the Apollo program.

The Skylab Program

The definition activities of the program that ultimately became Skylab proceeded in what must be described as a highly confused state; most of the program objectives and user-oriented program requirements, however, remained stable for the entire duration of the program. The program first known as Apollo Applications started out as a series of single-mission flights involving a larger number of scientific and technical experiments. This program concept was the basis for approval in the President's budget for FY 1968. About the same time, a command decision was made to incorporate these experiments in a concept known as the "wet workshop," in which a spent upper stage of the Saturn V would be left in orbit, purged, occupied and outfitted to perform the experiments. Many believed the concept could not work, but the program proceeded to preliminary design and, in many cases, detailed design. In the spring of 1969, a decision was made to go to a "dry workshop" wherein all the flight hardware elements would be assembled and checked out on the ground and launched using the first two stages of the Saturn V as the· launch vehicle. It took another four years of design and development to bring the program to flight readiness. The flight program was quite successful in the accomplishment of the many experiments. The data obtained from a large solar telescope, for example, the Apollo Telescope Mount (ATM), was regarded as outstanding by the scientists involved. This capability was included in the earliest program requirements.

CONCLUDING REMARKS

This paper has endeavored to highlight the importance of generating top-level program requirements at an early stage in the program evolution or Phase A definition phase. These requirements should include all the factors involved in meeting the program-objective(s) and should be stated with clarity so a determination can be made as to whether they can or are being met. Depending on the nature of the program, these requirements can relate to uses of a capability, a mission objective or other factors, including a simple hardware demonstration such as a test of a new instrument. It is critical to understand whether specific performance requirements are to be met or only a demonstration of capability is entailed, for the latter provides more flexibility for program adjustments.

The establishment of program requirements usually requires input and involvement of people both inside and outside of the program organization. Determination of just what disciplines are involved is important, particularly for the users and operators.

Validation of the top-level program requirements is a systems engineering function. At the outset, the systems engineering organization works with entities responsible for generating the requirements in an iterative process to assure their validity. This activity continues throughout the life of the program because of unforeseen events that impact the program effort. At times, this will necessitate changes to top-level program requirements. Changes should be under formal change control, and the systems engineering organization operating at the top of the program organization structure should be responsible for the validation effort. Systems engineering is a program-distributed activity that allows the top-level systems engineering organization to be relatively small because it depends on others for most of the required analysis. It should operate with a low profile.

Past programs serve to illustrate the range of program requirements considerations and the associated systems engineering effort. In the early manned programs, safety was a dominant consideration. Experience in these programs showed that preflight checkout is an important consideration, as is mission design, training, and simulation, all of which can impact the hardware design.

The top-level program requirements and the associated systems engineering activities should obtain and maintain simple interfaces between program elements, even though this produces some apparent program inefficiency. At least one past program, Skylab, has shown that top-level program requirements can be maintained even when considerable fluxing occurs with regard to the hardware and mission design.

THE IMPORTANCE OF COST CONSIDERATIONS IN THE SYSTEMS ENGINEERING PROCESS

by John D. Hodge

One of the most vexing aspects of managing large programs within NASA (or any other high technology government programs) is how to allocate program funds in a way that is best for the program. One of the major reasons is that the role of cost changes throughout the phases of the program. Another reason is that total cost is not all that easy to define; yet another is that funding, which is based on annual appropriations, is almost never consistent with fiscally efficient program spending rates. The net result is that program costs almost always escalate and inordinate amounts of time are spent controlling costs at the expense of maintaining performance or schedule.

Many studies have been performed to try and understand this problem. They show that program costs will escalate by at least a factor of three, from approval to completion. The studies suggest a number of guidelines that should be followed if costs are to be kept down, including clear definition of requirements, stable management and strong central control. Unfortunately, these factors are not always under the control of the program manager.

This paper examines the question of cost, from the birth of a program to its conclusion, particularly from the point of view of large multi-center programs, and suggests how to avoid some of the traps and pitfalls. Emphasis is given to cost in the systems engineering process, but there is an inevitable overlap with program management. (These terms, systems engineering and program management, have never been clearly defined.) In these days of vast Federal budget deficits and increasing overseas competition, it is imperative that we get more for each research and development dollar. This is the only way we will retain our leadership in high technology and, in the long run, our way of life.

BASIC PRINCIPLES

The principles are simple. First, define very carefully what it is you are trying to do. Check everything you do against that baseline, even if it has to be changed, and resist change once the decisions have been made. Second, break up the program into manageably sized chunks of deliverables that can be measured in terms of cost, schedule and performance, and define the interfaces between the chunks. Third, continuously assess the risks to success as the program proceeds, and modify only as necessary.

REQUIREMENTS TRACEABILITY

Most studies have shown that the primary reason for cost escalation is that not enough time or resources are spent in defining the program. It is clear that you cannot control what you have not or cannot define. It is during this period that some of the most elegant systems engineering should be performed, especially in understanding the cost of every requirement and its systems implication. Even if the definition is adequate during the early phases of the program, it is imperative that great vigilance be exercised in maintaining the baseline definition of the program and the fundamental reasons for doing the program. This process establishes a small but influential part of the program office, preferably within the systems engineering organization, dedicated to the traceability of requirements and to ensuring that a clear path exists from program rationale to program requirements to systems requirements to systems design. Too often, once a design has been established, changes are proposed and enacted that bear little relationship to the original premises of the program. As will be discussed later in this paper, there are many reasons for change, but where possible, changes should be considered during the formulation of the program and not later when the program structure is in place and the program is in progress. Change is almost always costly; requirements traceability provides a bulwark against which the program manager and the systems engineer can stand and defend.

BASELINE COST, SCHEDULE AND PERFORMANCE

The three main parameters in the control process—cost, schedule and performance—are the program manager's bread and butter. Again, program definition is vital and necessary from the very beginning. It may be argued that clear definition is not possible, particularly early in the program; nevertheless, an approved, traceable baseline, although it may alter, must be known at any given time, and must include everything in the program. The "I forgots" can kill you.

The key to success in handling these three parameters is to manage the balancing act between them. Cost, schedule and performance are usually dependent variables and at various times, one or another may assume greater or lesser importance. A single variable, however, should never be changed without knowing the impact on the other two. Within the NASA culture, performance is generally the predominant factor, and schedule is a distant second. Cost tends to be considered mostly in the context of the annual appropriation, but from the point of view of the program manager, all three parameters must be defined and approved continuously, which is a function of the systems engineering process.

PROGRAM RISK ANALYSIS

In recent years, especially since the Challenger accident, program risk analysis has come to be used largely in the context of crew safety, but this is only a part of program risk. Basically, program risk analysis assesses the probability of meeting requirements as changes occur. A number of analytical tools now available can be used to understand the relationships between cost, performance and schedule. Again, a small group within the systems engineering organization should be dedicated to understanding the impact of any change on all three parameters. Armed with this information, risk can be reduced in many ways. Adding more money, reducing the performance requirements, or extending the schedule are most often used. A competent systems engineer will know the relationships between these three variables and the impact of any situation on the total program.

THE ROLE OF COST IN PHASED PROCUREMENT

The most common form of procuring high technology capability within the Federal Government is known as phased procurement. The theory behind this procurement method is that commitment to the program gradually increases with time and in discrete stages. Within NASA, there are four standard phases; others are beginning to creep in as the ability to establish new programs becomes more difficult and the duration and cost of operations becomes a more significant part of total program costs. The role of cost is different in each of the phases. The phases are:

Pre-phase A: This is a very unstructured period that examines new ideas, usually without central control and mostly oriented toward small studies. This period can last for a decade or more and produces the list of ideas and alternatives from which new programs are selected.

Phase A: Sometimes called the feasibility phase, this is a structured version of the previous phase. Usually a task force or program office is established, and multiple contracts will be awarded. The goal of this phase, which may last for several years but usually is limited to one or two years, is to decide whether a new program will be started and what its purpose and content should be. This phase represents less than one percent of the total program costs. Nevertheless, it is largely a systems engineering effort and sets the stage for everything that follows.

Phase B: Sometimes known as program definition, this phase is the most important in establishing the basic parameters of the program. By the time this phase is finished (a period of two or three years), the program rationale, cost, schedule, performance, management style and the most likely technical solution will have been established. This phase usually involves multiple contracts to establish a variety of ideas and a competitive environment, should the program proceed. Cost is continuously assessed as a function of design solutions relative to basic requirements. Studies indicate that from five percent to ten percent of the total program costs will need to be expended if control is to be maintained over the program during Phases C and D.

Phase C/D: Originally separate phases, this period covers design, development, test and evaluation. Contracts may be open to all qualified bidders or only to those involved in the previous phase. Although competition is not usually open between Phases C and D, commitment to Phase D depends on a successful and acceptable design. In past programs, two-thirds of the total program cost was expended during this period. The systems engineering role has begun to shift toward systems specification and systems interfaces. The secret to cost control is a sound definition of end items and their interfaces with a tight hold on changes.

Phase E: In most past programs, the operations costs were less than 20 percent of the total cost. This was because there was a definite end to a relatively short-term program. In recent years, particularly in the manned programs, the length of the operational phase has increased significantly. In the case of the Shuttle, it could be conceived as indefinitely long. For this reason, life cycle costs should be a major consideration from the beginning.

SELLING THE PROGRAM

The definition of a new start within NASA varies by program and organization but can generally be said to occur at the beginning of Phase B. Prior to that time, the program manager is selling the program. The total expenditure of funds during the selling period is usually far less than one percent of the final program costs; this is, however, when the basic parameters of the program are established. It is a time of building constituents both inside and outside the Agency. Assuming that a feasible technical solution is available and an acceptable management scheme can be provided, much of the debate about whether a program should be approved centers largely around the question of cost. Of course, with only preliminary designs available, only cost estimates can be made and these are obtained from standard cost models.

COST ESTIMATING

During Phase A of the program, when the most rudimentary designs are available, it is essential that program cost estimates are made before the program start can be authorized. Estimates are made using cost models that have been developed on the basis of past experience on similar programs. These models are among the most arcane devices invented by engineers, so a few words on how they work are appropriate.

Past experience is captured by documenting the cost of each system on the basis of weight. Regression analysis is performed to determine a straight line log relationship. Once the weight of the system has been estimated, the cost can be determined. This estimate is multiplied by a complexity factor to allow for the risk associated with the selected technology and may vary from as little as 0.50 to 2 or more. This is repeated for each system, and the total becomes the baseline cost. This total is multiplied by a factor to allow for systems engineering and testing by the prime contractor. This is known as the "prime wrap" factor and is again determined based on all relevant past experience. All prime contractor estimates are added and then multiplied by a second factor known as the "nonprime wrap." This is the cost of government work. Finally, a reserve factor is used to allow for problems during the program. There are separate cost models for manned and unmanned programs, which are significantly different. In general, for the unmanned programs, about 40 cents of every dollar goes to hardware, and in the manned programs, about 20 cents.

These cost models pose a great many problems. First, they are normalized on the basis of weight. Clearly this is not valid in all cases, particularly structure. Second, they do not explain why the costs are what they are. Factors such as management style, procurement strategy and test philosophy are not differentiated. Third, they include all past experience, including errors and overruns. In this respect, these cost models assume no learning curve. As it was in the beginning, is now, and forever shall be! They must therefore be used with great caution. From the systems engineer's point of view, these cost models can be used to assess the relative costs of various design solutions; on an absolute basis, however, they are of little use.

So far we have been able to make a tentative estimate of the cost of the flight system. To this must be added the cost of new facilities, including launch, test beds, flight operations, networks and data reduction, among other factors, and finally the cost of operations.

It is at this point that the program manager faces the first dilemma: What should be included in the program cost? That sounds like a simple question, but it is complicated by the fact that not all costs are under the control of the program manager nor is he or she responsible for justifying all of the associated appropriations. For example, launch costs are provided by the Office of Space Flight, network costs are provided by the Office of Operations, and civil service costs are provided by the research and program management fund managed by the Office of the Comptroller. New buildings are provided under the construction of facilities budget. In addition, most new program managers are surprised to find that a tax based on the number of civil servants working on the program varies from Center to Center, and neither the number of people nor the level of tax is under the control of the program manager. Taxation without representation! Despite this dilemma, the systems engineer should include all of these factors in the cost estimate because the chosen design will affect all of them; overall program costs are as important to the Agency as direct program costs.

Program costs tend to be presented as only those costs that are under the control of the program manager. No matter how much this limitation is stated in presentations, it is assumed that it is the total program cost (especially when it is a popular program) that has the support of the Executive branch, the Congress and other constituencies. It is no wonder that the average program increases in cost by a factor of about three from the time of approval to completion and that most program managers during this period are accused of everything from naiveté to self-deception to outright lies. There is the added ethical question that if all costs were presented, the program would not be approved!

DEFINING THE PROGRAM

This phase of the program, usually known as Phase B, will take from one to two years. The purpose is to take the various concepts considered in Phase A and select a single valid solution. By the time Phase B is over, a clear set of requirements should be available with a complete set of functional specifications and a cost estimate based on preliminary design concepts rather than on cost models. These are primarily produced by the systems engineering organization and include at least one preliminary design and selected technologies with well-understood risks associated with those technologies. Don Hearth, who recently retired from NASA as director of the Langley Research Center, performed a study on how much this phase has cost for various past programs as compared to the success of the program in later phases. Success was measured as the ability to maintain performance, schedule and cost as determined at the end of Phase B. He concluded that the most successful programs spent between five percent and ten percent of the total program cost in Phase B. The scope was limited to unmanned programs, but the rationale can reasonably be extended to manned programs.

Apart from establishing a credible functional system specification, it is essential to determine the management structure, the procurement strategy and a baseline cost for the life of the program, including the cost of operations. Once again, the primary method for cost estimating is the cost model, but there should be sufficient detail available to check the model with bottom-up costs based on feasible design solutions. The systems engineer is responsible for comparing these two cost estimating techniques. It is unwise to proceed to the next phases unless some bottom-up cost estimating has been performed.

Perhaps the most important product of this phase is a complete work breakdown structure. Again, this is largely the responsibility of the systems engineering organization. The axiom to be followed is, "You cannot control what you have not defined."

WORK BREAKDOWN STRUCTURE

Too often a program will be approved without a well-established work breakdown structure (WBS) describing the whole program, which inevitably results in large cost overruns. The WBS is the basis for the procurement strategy and often for the management structure. Without it, program changes will take place after the contractors are in place and have to be paid. Overlaps between contracts, as well as missing elements and contract changes, are always expensive.

The following simple rules have to be followed:

1. Each element of the WBS should contain a deliverable that can be defined.

2. The sum of the WBS elements must be the total program. (Note that a given program manager may not have the responsibility for all elements, but they should each be defined and allocated.)

3. Each deliverable should be accompanied by a cost and a schedule. The cost should include a reserve based on the estimated risk associated with that element, and the cost should be allocated to that element.

As simple as these rules sound and as much as NASA requires contractors to adhere to them, the internal track record is dismal. We can go a long way toward containing costs if discipline is established early and maintained.

One last word of caution. A WBS element should never be established on the basis of function or organization. These elements are not end items. Other mechanisms exist for identifying these elements, which in general could be defined as program overhead and not entirely the responsibility of the program manag:r. They should be recognized for what they are and identified, but they should not be included in the WBS.

MANAGING THE PROGRAM

We have now reached the time in the program when promises have been made, deals have been struck, and the program has been approved. All that remains is to deliver. A custom within NASA stipulates that new managers are installed with the belief that the skills required to sell a program and to define it are different than those required to run it. Certainly some changes can be expected, but I believe that such changes are better if they occur sometime after a phase has been entered and the basic management structures have been established. What the program needs at this time is ownership of the concept, and changes in management will usually result in program changes that inevitably will lead to increased costs. This is particularly true of the systems engineering group that has carefully balanced the requirements against the design and is familiar with the "why" of a decision as well as the "what." So far the total expenditure has been relatively low, but once the contractors are onboard and the manpower begins to build up, costs can escalate at an alarming rate. In a very short time, increases or decreases in performance, extensions or reductions in schedule, and decreases in annual funding will all increase cost.

Design to cost. There is much talk about design to cost but very little action, and for this there are a number of reasons. Earlier, I mentioned that within NASA there is a tendency to order the three variables by performance first, schedule second, and only then worry about cost. So by tradition, cost tends to be put on the back burner. One of the reasons for this is that during the Apollo program, the cost function was transferred to the budget and program control groups. In a program where the technical problems were so difficult and the budgets were ample, this was understandable, but this is no longer the case. This situation resulted in a shift away from making the design engineer accountable for cost as well as performance and schedule. The second problem occurs when the cost is not allocated at the WBS element level, where it can readily be traded against performance and schedule and easily traced. I believe that cost must be allocated to the lowest possible level (a little scary for the program manager), but unless this is done, it will be impossible to hold the designer accountable and unlikely that overall costs will be held in check. The third problem is that in an organization that prides itself on technical excellence, it is very difficult not to make things a little better; consequently, there are always plenty of ideas around. The credo of the systems engineer should therefore be: "The better is the enemy of the good."

Design to life cycle cost. Over the past decade, the operational costs of NASA programs have steadily risen as a percentage of total program costs. This is largely due to the fact that programs have a longer life in the operational phase. Whereas 20 years ago operational costs amounted to no more than 20 percent of costs, they are now approaching half of the NASA budget. It is time to place design to life cycle cost on an equal footing with design to cost. The dilemma is that a design that allows low-cost operations will usually demand higher development costs and in turn, this means larger front-end program costs. It is essential that the systems engineer make these assessments. The simplest thing for a program manager to do is walk away from this dilemma and let the operations people worry about it later. As this is becoming an overall problem for the Agency, the ability to make new starts will depend on the ability to ensure that a sufficient percentage of the budget is available for operations. Unfortunately, it is difficult to get enough operations people to participate early in the program, but I believe it is essential. Some kind of veto power should be established when it comes to making design decisions; too many program managers do not feel responsible for operations costs and perhaps, what is worse, are not held accountable for it. Let there be no doubt that operational costs are unacceptably high. An operational concept must therefore be developed early enough in the program to have an effect on the design process.

Change control. Once a program is underway, the program manager's responsibility is controlling change, which is inevitable. Earlier I said that you cannot control what you have not defined. It is equally true that you cannot control changing something that is not defined. First know what it is! A complete WBS with allocated schedule and cost is, once again, the key. Change requests must not be limited to solving a technical problem. They must be accompanied by cost and schedule impacts and, just as important, life cycle cost impacts. In addition, there is always a rippling cost impact caused by change. Other WBS elements may be affected, including items in different contracts or in totally different NASA codes, or line items. For these reasons, change must be assessed at the systems engineering level as well as at the WBS level. Perhaps the overriding rule is that changes should be difficult to approve but easy to implement once the decision is made.

Managing cost reserves. A qualified cost estimator would not let a program get started without making provision for cost overruns or reserve. The many uncertainties in a development program make it essential. An analysis of past programs allows a fairly accurate estimate to be made of what is a reasonable total amount as a percentage of total costs, assuming that the programs are similar. Determining how and when the allowance should be allocated is much more difficult. One school of thought says that reserves should be held at the highest level in the program and applied only to correct unforeseen occurrences. The problem is that this tends to bail out poor performers. I believe that the reserve should be determined based on the perceived risk of the element when the WBS is formulated. The manager of the element should then be held responsible for the stewardship of the reserve. In order for this to work, some sort of reward system must be established for the manager who does not spend the reserve. In any case, it would be prudent to maintain some reserve at the central level for those things that cannot be anticipated. Just to keep the system honest, a very simple tracking program can be established to follow the expenditure of the reserves at the WBS element level after the fact. I would like to see an in-depth study done on this subject.

TRAPS AND PITFALLS

So far we have talked about where cost fits into the program management and systems engineering processes. There are a few areas that may catch the program manager unprepared and a few ideas that may be used to make life a little easier in the future. It may not be possible to implement all of them, but it is worth a try.

Buying in. If you are involved in the selling of the program, the easiest trap to fall into is underpricing the program. Despite stories to the contrary, I do not believe that this is a matter of deliberate low bidding. Although I once heard a distinguished gentleman say that we do business the old fashioned way, we do underbid and make up on change requests. The fact is that every program manager I have ever met was convinced that he or she could do it for less than the past record would suggest. Unfortunately, this usually involves changing the way we do business. I believe that there are less expensive ways, but you should tackle this one at your own risk and only if you have the support of the very top of the organization. The systems engineer must be the conscience of the program manager during this period.

Design to budget. Let us assume that we have completed a perfect Phase B and that everything is in place, including the rate of expenditure by year. It is a virtually certain that two things will happen. First, with eloquent rationales and spreadsheets by the ton, the various element managers will find a need to increase their funding allocation. One favorite argument will be that the sellers of the program, who are no longer in charge, will be blamed for not understanding the problem. In addition, Congress may add a requirement or two. Second, the budget will be cut in the Agency, at the Office of Management and Budget (OMB), and finally in Congress. At this point, the intricate patterns of dependency between performance, cost and schedule begin to unravel. In the first year, this is not devastating because you can always delay bringing the prime contractors on board. But by the time they arrive, the trap has been set for the most insidious form of management, design to budget. Unfortunately, a fact of life is that very few research and development (R&D) programs have multi-year funding, and annual budgets will be less than planned. The net effect is that program costs will escalate, and enormous pressures will attempt to bring down the annual funding. The first remedy is to stretch the schedule, and the second is to reduce the scope of the program. You will no doubt find yourselves in this position, and you will receive a great deal of advice from the nonparticipants, but you should beware of "descoping." A cursory examination of the cost models will show that in the manned programs, only 20 cents of every dollar go to hardware. (In the unmanned programs, the number is closer to 40 cents.) Once the management structure is in place and the contracts have been awarded, virtually all of the other costs are fixed or very difficult to reduce. Take out all the content and the program cost will still be 80 percent of the estimate! The lesson is that if you are forced to remove content, you should be sure to take out every cent that is associated with that content: prime wraps, nonprime wraps, test beds, personnel, and, if necessary, the kitchen sink. It will be difficult to find, but it will be worth the effort. If this were a mystery novel, it might well be called "The Case of the Missing 80 Percent." Where does it all go, and why is it only 60 percent for unmanned programs? Much of this is valid and accounts for systems engineering and integration at all levels of the program, including test and evaluation, operations, and many other things. But it also accounts for duplication of test facilities, overlaps between assignments, management style, inefficiencies and a host of hidden costs associated with maintaining the institutions that are often invisible to the program manager. The systems engineer is responsible for ferreting out the good from the bad. It is a simple fact that the first one percent reduction in these wraps (80 percent to 79 percent) increases the amount of hardware by five percent (20 percent to 21 percent)! A 20 percent improvement in the wraps (80 percent to 60 percent) results in a doubling of the hardware (20 percent to 40 percent) or cutting the program costs in half for the same amount of hardware! "Thar's gold in them thar hills."

The UPN System. The NASA budget is prepared and submitted using a system of breakdowns known as the unique project number (UPN) system. All parts of the agency are required to report their annual needs on the basis of this system, including the program offices. From a program point of view, a fatal flaw in this process is the numbering system, which generally describes functions rather than end items and is therefore not in consonance with the principles of a WBS system. It is essential that the program manager be able to trace the equivalence of the UPN number and its corresponding WBS element. This will require a joint effort between systems engineering and the program control people. Without this traceability matrix, the program manager will not know what is being asked for or where the money is going. Too often the UPN number is perceived as directly equivalent to the WBS element, but this is very seldom the case unless the WBS is not end-item oriented. (The latter happens more often than it should.) One way to avoid this situation is to make the annual budget call for the program using the WBS system and then translate it to the UPN system for the purpose of aggregating the total NASA budget. I have never seen this happen.

The cost of operations. I mentioned earlier that the costs of operations are now about 50 percent of the NASA budget. This is partly due to the increase in the operational life of a program and to the fact that we have not learned to design systems for operability. It has not been necessary in the past. It is also true that the productivity of the operations infrastructure has not been high on the program manager's list. If we are to reduce total program costs, which are vital to the Agency and to the program, it is time to strike a new level of cooperation between these two normally separate parts. The program and the systems engineer must assume a large part of the responsibility.

THE INSTITUTION AND THE PROGRAM

Although not directly related to the systems engineering process, a number of things bear directly on the program and have a major effect on the ability to perform the various program functions. These generally concern the relationship between the program and the institution. NASA was originally established using the resources of the National Advisory Committee for Aeronautics (NACA), an aeronautical research organization that was seldom involved in large development programs. The budget was relatively small, and there were few contractors. In fact, all contracts were signed at the Washington office, the NACA equivalent of Headquarters. It quickly became apparent that, in addition to the research centers, a development center was needed. The Goddard Space Flight Center (GSFC) was established to perform this function. This was rapidly followed by the Lyndon B. Johnson Space Center (JSC) in Houston, the George C. Marshall Space Flight Center (MSFC) in Huntsville, and the Jet Propulsion Laboratory (JPL) in Pasadena. Almost immediately, GSFC and JPL became responsible for multiple unmanned programs, which were largely contained within a single Center, and JSC and MSFC became responsible for multicenter manned programs. In both cases, program offices were established and the Centers provided the resources, both personnel and facilities, to support the program. With the exception of JPL, which was a federally funded research and development center and operated outside the civil service system, all NASA personnel and basic facilities are funded separately from the programs in line items known as Research and Program Management (RPM) and Construction of Facilities (CoF). Program-specific facilities are funded by the program and these facilities are most often operated by support contractors, also funded by the program. This system was established so that the programs would be managed by government personnel who would rotate from program to program and carry their experience with them. This worked very well until the late 1960s when the budget began to fall rapidly, and there was a significant reduction in NASA personnel. By the early seventies, both the budget and the number of personnel had been cut in half, but the number of Centers remained essentially the same. The cost of maintaining the institution could not longer be sustained by the RPM and CoF line items. The solution was to tax the programs based on the number of personnel that were applied to the program. Unfortunately, the program manager does not decide how many people should work on the program, which, by tradition, is the responsibility of the Center director. Neither does the program manager participate in determining the level of the tax. These decisions, again by tradition, are made by the comptroller.

MAINTAINING THE INSTITUTION

Unless the basic system of funding personnel is changed, the programs will most certainly be responsible for funding some of the institutional costs that are not related to the program; the RPM budget will never be allowed to grow to compensate for this. The question is rather how large the institution needs to be to support the program and how that decision is made. I mentioned earlier that the WBS should represent the totality of the program and should always describe deliverables; this problem runs counter to that principle. I believe that the solution lies in accepting this cost for what it is, negotiating the level of tax with the program manager for the duration of the program, and taking it off the top each year. It may not be controllable in the normal sense, but at least it is a known number.

Personally, I believe that the Agency would be better served if the development centers were managed using an industrial funding system similar to JPL and many other government facilities, including the Navy labs. But until that happens, it will be necessary to find some balance between the institutional and program needs.

MANAGEMENT STABILITY

Every program will change management during its life cycle. The common practice in NASA has been to make these changes deliberately between phases. It is not uncommon to see as many as four different managers during a program, including a specialist in closing off completed programs. The positive side to this is that it is possible to match the needs of each phase of a program to the special capabilities within the agency. The negative side is that each manager has a different style, each program has different management needs, and often these do not match when the change-over occurs between phases. One way is not always right and another always wrong, but each is different, and changes even in management style can, and usually do, increase the cost of the program. The secret then is to stick with a team as long as possible, particularly the systems engineering team, something that is easy to say and difficult to do in these times of declining internal expertise and increasing retirements.

THE TYRANNY OF EXPERIENCE

Too often, you will find resistance to change in the way things are done. "We have always been successful (measured by performance) doing it this way, and its very dangerous to change winning ways." "If it ain't broke, don't fix it." "You get no credit for an on-time failure." All true and at the same time, destructive to valid new ways of doing business, especially when it comes to introducing more efficient or less expensive ways. When the space program started, we had no experience and what followed was the most innovative and exciting period in the history of high technology programs. But now we have all that experience, and it has become a burden. By all means, you should keep the wise heads around (they may still save you), but take advantage of the explosion in new technologies and capabilities, which allows for things that we could only dream of 30 years ago. You should be careful before you introduce a change, but you should not dismiss it out of hand.

DOES IT MATTER?

We have been in the civilian space business for almost 40 years, and time after time we have shown that we can rise to any challenge and lead the competition, provided we have the resources. Time and time again the Federal Government has provided the resources. We have been the envy of the world. We have written the book on the subject, both from a technical and a management sense.

Until now, it was enough to know that we were the best. There was no established competition, most of the money was spent internally, and cost efficiency was second to performance. Some have characterized it as a Works Projects Administration (WPA) for the technologists! The problem is that in this era of budget deficits and trade deficits, there is not enough discretionary money to go around. Even without international competition, it would be imperative to get more out of our research dollars. The trouble is that we have learned profligate ways, as neither the government nor the contractors give rewards for cost efficiency. And while we were basking in this glory, the rest of the world has been catching up. They have been reading the book, and the competition, supported by their governments, is getting good and fierce.

But there is a difference; the competition believes that the space business is here to stay. I said space business, but I meant commerce, and in commerce cost efficiency is paramount. Do we still want to stay at the top, or are we ready to leave it to the rest of the world? Are we prepared to do what is necessary to stay in the game? After all, it's only a space program. Does it matter? You bet!

CAN ANYTHING BE DONE?

In this paper, I have attempted to show where cost fits into the space program's engineering and management business. A combination of things have placed cost at the bottom of the priority ladder except in matters of the inexorable annual budget. There are many ways to improve cost efficiency, some of which are available to the program manager. In the long run, it will take a concerted effort by all of us to make a difference. The Executive branch and Congress, together with industry and academia, must work as before, when we perceived that we were second. In the meantime, I hope that I have been able to give the budding systems engineer and program manager a few tips to do something about the problem of cost considerations. We can only do something about it if we want to!

Systems Engineering and the User: Incorporation of User Requirements into the SE Process

by John E. Naugle

A scientific mission goes through two distinct stages, each with its own special requirements for systems engineering. A division director at NASA Headquarters, assisted by a program chief and a program manager, conducts the first stage. These three people, assisted by committees and working groups, define the mission, formulate its objectives, establish its rough boundaries and manage the selection of the experiments. The division director practices a rough and ready kind of systems engineering, balancing the desire of the scientist for the most complex sophisticated instrument possible against the desire of the Office of Management and Budget and Congress to reduce the NASA budget. If the division director's systems engineering is done well, the mission will be supported and scientific results obtained. If, on the other hand, the systems engineering is poor, the mission may be canceled either because the scientific community concludes the scientific objectives do not merit the cost or because the Office of Management and Budget or Congress thinks the cost is too high.

After the experiments have been selected, the action shifts from Headquarters to one of the NASA Centers, and the second stage begins. A project manager, assisted by a project scientist and supported by an engineering and a financial staff, is in charge of the second stage. The second stage begins with the preliminary design phase and ends when the last scientific paper has been published. All the hardware for the mission is constructed, tested and operated in the second stage.

Systems engineers incorporate the scientists and their instruments into the systems engineering process during the preliminary design phase. At the conclusion of the preliminary design phase, the project manager conducts a preliminary design review to assure everyone—the scientists, the project management, the Center management and NASA Headquarters—the scientific objectives and requirements have been incorporated into the systems engineering process.

This paper is organized into four parts. In the Gestation Phase, I describe the process of starting a new mission and establishing its rough boundaries. Next I show how the scientific experiments are selected. Then we enter the Preliminary Design Phase, where we incorporate the scientist's instruments into the systems engineering process. Finally, I show how the Preliminary Design Review (PDR) assures NASA management and the scientists that the scientific requirements have been incorporated into the systems engineering process to everyone's satisfaction.

Throughout I emphasize the dual role of servant and master that the systems engineer plays with respect to the scientist and the project manager. As servant, the systems engineer works to assure the scientists that the project will meet the requirements of their experiment and their instrument; as master, the systems engineer works to assure the project manager that the scientists and their instrument will meet the requirements of the project. A glossary of terms appears at the end of this paper.

I emphasize the need for the systems engineering process to consider all of the pieces of hardware that the mission will require and all the activities that must be conducted during the entire mission. It is easy, in the early phases of a mission, to focus on the spacecraft and the instruments and to ignore or push into the background those activities and facilities that will be needed later or are the responsibility of other offices. The associate administrator for the Office of Space Science and Applications needs to know, before committing to undertake a mission, that the entire mission has been thought through, that the facilities will be available, and that the funding is adequate to procure all the flight and ground-based hardware and to pay for all the work that will be required.

I arbitrarily end this paper with the PDR. Clearly there will be continuous interaction between the scientists and the systems engineers throughout the remainder of the mission. However, the main purpose of the PDR is to see that the user requirements have been properly incorporated into the system. Other papers discuss the role of systems engineers in later phases of the mission.

THE GESTATION PHASE

If we are to successfully incorporate user requirements into the systems engineering process, we need to know how NASA creates a new mission and establishes its principal boundaries; we need to know who selects the scientific instruments and when.

New missions get started in a variety of ways. A person with a new idea may initiate a new space mission. A scientist at a NASA Center or a university may make a discovery, ask a new question or invent a new instrument. An engineer at a NASA Center or in industry may invent a new control system enabling more precise measurements to be made. A technology may mature.

New missions have been started this way in the past, but now, more and more, new missions either come from a group of people convened by NASA specifically to think about new missions or are logical follow-ons to existing or completed missions. The Hubble Space Telescope was started as a logical step after the Orbiting Astronomical Observatories. Its scientific objectives were laid down in 1964 during a summer study conducted for NASA by the National Academy of Sciences Space Science Board. The Advanced X-Ray Astronomical Facility continues the x-ray observations begun with Uhuru and High Energy Astronomy Observatory. Ulysses continues the study of the Sun begun by HELIOS. Some missions are precursors to later more complex missions. Surveyor and the Lunar Orbiter were precursors to Apollo. The Lunar Observer and the Mars Observer, in addition to increasing our knowledge of the Moon and Mars, will be designed to provide data needed to design manned lunar bases and manned missions to Mars.

Applications missions result from a need for additional coverage, better resolution, more complete coverage of the electromagnetic spectrum or a new operational spacecraft.

Although there is no set process by which a new mission gets started, once it begins, there is a fairly predictable process by which it moves from concept to design to flight. Usually a new mission gets underway when a dedicated advocate devotes the time and energy required to get the idea accepted within NASA. This advocate may be located at a Center, a university, another federal agency, an aerospace company or in NASA Headquarters. The advocate prepares a rough design of the spacecraft and a list of potential instruments. With these in hand, the advocate buttonholes scientists, engineers, Center and Headquarters personnel to persuade them to become supporters of the mission. At a Center, the advocate may bootleg some feasibility studies at the Center before taking the concept to Headquarters. At some point, the advocate must describe the mission to the director of the appropriate division in NASA Headquarters and persuade the director that NASA should undertake the mission. If it is an astronomy mission, the advocate must convince the director of the Astrophysics Division; if a planetary mission, the director of the Solar System Exploration Division; if an Earth science or applications mission, the director of the Earth Science and Applications Division. The director may ask the advocate and supporters to describe the concept to the appropriate NASA advisory committee or to a summer study sponsored by the Space

Science Board. The director may ask a Center or a contractor to make a feasibility study of the mission before committing to the 5- or 10-year effort that is required to get a new mission underway. The advocate may appeal to the associate administrator for the Office of Science and Applications to tell a reluctant division director to undertake the mission, but until the director is convinced that the mission is worth doing, it is almost impossible to get a new mission started.

Once the division director becomes enthusiastic about the mission, it will be incorporated into the director's long-range plan, and the groundwork will be prepared for approval by NASA senior management, Office of Management and Budget and Congress. Once the division director includes a description of the mission in the division's advanced program, the advocate's work is over; the mission takes on a life of its own. The division director provides funds for studies and for research and development and may provide funds to several scientists to begin work on potential instruments for the mission.

Applications missions are started by an agreement between the division director at NASA Headquarters and the division director's counterpart at the National Oceanic and Atmospheric Administration, or whichever agency needs the mission. They agree that the mission has merit and that they should begin to jointly plan for the mission. Agreements are made as to what research and development will be conducted, who will conduct it, and which agency will pay for it. They will produce a mutually acceptable plan of action by which they will seek approval and funds.

SETTING THE BOUNDARIES

The scientific or applications objectives establish some but not all of the boundaries of a space mission. Other factors, such as the kind of transportation or the funds available help set the boundaries. Nonscientific criteria may have influenced the scientific objectives themselves. The initial diameter of the Hubble Telescope, four meters, was chosen in the mid-sixties because that was the diameter of the largest spacecraft that could be put inside the shroud of the Saturn V launch vehicle. Later, the diameter was reduced to 3.2 meters to take advantage of existing manufacturing, test and calibration equipment. The broad boundaries of the Viking mission were set by the capability of the Titan launch vehicle. As a matter of fact, in its formative stage, Viking was called the Titan Orbiter-Lander Mission. An earlier Mars orbiter-lander mission, Voyager, had been planned for a Saturn V; this big Voyager was canceled by Congress because it was too large and too expensive and because the scientists involved would not support such an expensive mission at that stage in the exploration of Mars. The competition with the Soviets also helped set the boundaries for Viking. The scientific returns from Viking had to be sufficient to justify the cost of the mission, even though the Soviets might land a spacecraft on Mars before Viking got there. National needs—foreign policy, security, development of new technology and the maintenance of an institution or a capability—may influence the size, scale and timing of a mission. For a decade scientists unsuccessfully tried to persuade NASA to start a mission to study the interplanetary medium near the Sun. After President Johnson offered to undertake a joint space mission with Germany, it took NASA just 24 hours to establish the HELIOS Mission to make a close flyby of the sun. The need to test the Titan IIIC launch before the launch of the Viking mission dictated that HELIOS would use the unproven Titan IIIC rather than the existing Atlas-Centaur.

The actions of the members of Congress as they review, authorize and appropriate funds for a mission may help establish the boundaries of a mission. A key chairperson or a powerful committee member may decide that a particular mission is worth $500 million but not $750 million; the chairperson may decide to support a mission if it will increase employment or prevent the closure of a facility in the chairperson's district.

Purists may argue that systems engineering should focus on technical constraints and need not take into account nebulous political and managerial constraints. Unfortunately, such constraints have been with us since the first time two people joined together to accomplish a task neither could do alone. Incorporation of such constraints into the systems engineering process is just as important as incorporating the purely technical constraints. The division director, however, must keep the political and technical constraints separate and should never attempt to justify a political constraint with some flimsy technical justification. If this happens, the rest of the participants in the mission will become confused and the division director will lose credibility. If the participants are kept straight, then later, if relief is needed from some such constraint, the division director will know who must be persuaded to get relief and the kind of justification that must be prepared.

In the early days of NASA, with a powerful administrator and with space exploration a major national goal, a project manager could ignore factors other than the scientific and technical requirements. Today, the assembly and maintenance of the necessary support for the mission are so difficult that these other factors may become as important, if not more important, than the requirements derived from the objectives of the mission.

Out of this combination of political and technical considerations, the major boundaries are set for a mission. The launch vehicle is selected, the project management center is picked, the trajectory and a tentative launch date identified, and a rough idea formed of the kind and number of instruments that will make up the payload. The availability of transportation and the support of the Office of Operations is established. A rough cost estimate is made.

THE ROLE OF THE PAYLOAD AND THE TECHNICAL WORKING GROUPS

As soon as the broad boundaries of a mission are established and the division director is confident about obtaining approval, the groundwork begins for selecting principal investigators—the scientists who will perform the mission experiments. To make the selection, the division director first needs to know how many and what kind of instruments can be placed on the spacecraft, an analysis accomplished by two working groups: a Payload Working Group and a Technical Working Group. The Payload Working Group consists of NASA and academic scientists from the scientific disciplines involved in the mission, and the Technical Working Group of system engineers and discipline engineers representing all the engineering disciplines and subsystems required to design, build and operate the spacecraft. Working together, these two groups will design a trial payload that will accomplish the scientific objectives of the mission and a spacecraft capable of supporting that payload. In this joint activity, we begin to incorporate the user requirements into the systems engineering process.

The trial payload and the spacecraft emerge through an iterative process. The members of the Payload Working Group select a trial payload—a group of instruments that accomplish the objectives of the mission. In assembling this trial payload, the Payload Working Group may invite scientists to come to a meeting to describe instruments they hope to fly on the mission. They may invent new instruments that are needed to accomplish the objectives. The Payload Working Group will estimate the weight, volume, power and communication needs, and specify the orientation and stabilization requirements for each instrument. One or more members of the Technical Working Group will attend the meetings of the Payload Working Group to help them develop the requirements and to design the spacecraft and bring back to the Technical Working Group a better understanding of the payload that is emerging.

Meanwhile, the Technical Working Group will use the scientific objectives and broad constraints of the mission and design a hypothetical spacecraft for the mission. The Technical Working Group then takes the first trial payload prepared by the Payload Working Group and integrates it into the spacecraft. The two groups then hold a joint session where the Technical Working Group reviews the fit between the payload and the spacecraft, and the descriptions of changes that must be made either in the spacecraft or in the payload to make them compatible. Additional power may be required, the structure of the spacecraft modified, or one or more instruments may have to be redesigned or eliminated. At the conclusion of the joint meeting, the two groups agree on the actions each will take during the next iteration with the mutual objective of making the payload and the spacecraft compatible. The Payload Working Group refines the payload and the Technical Working Group refines the design of the spacecraft. They meet again, review their progress, and decide on the next course of action.

After a year or so of joint effort and two or three such iterations, a spacecraft and a payload will emerge that are satisfactory to both groups, the scientific community, the division director, the program manager, the program scientist and to senior NASA management. The division director and the program scientists are now ready to select the actual scientists, and their instruments, for the mission.

SELECTION OF THE SCIENTIFIC EXPERIMENTS

The associate administrator for the Office of Space Science and Applications selects the scientists who do research in space. The division director, using an ancient procedure established in 1960, is in charge of all the activities associated with the selection process. People sometimes ask why the experiments are selected by an official at NASA Headquarters rather than by one at the NASA Center that will manage the project. Others ask, why not use the instruments selected by the Payload Working Group for the trial payload and avoid all the time and energy that goes into the NASA selection process? Why NASA Headquarters, why not the National Academy of Sciences Space Science Board? These are good questions, and in some cases, the answer is easy: the particular method has been tried and found not to work; in others, the answer is not obvious and some explanation is necessary.

History shows that the nation needs a vigorous broad-based space science program that involves many academic scientists. Academic scientists are a fertile source of new ideas, and their involvement rapidly disseminates the knowledge and experience gained in the space program to the next generation of scientists and engineers. In addition, the participation of academic scientists and their graduate students helps assure a continuing supply of space scientists and aerospace engineers. Academic scientists also form a strong, vociferous lobby for the NASA space science program.

History also shows that NASA needs competent, creative scientists at its Centers to help conceive and design new missions and to work with the academic scientists who participate in NASA's missions.

The academic scientists and the NASA scientists at the Centers fiercely compete for the right to conduct investigations on NASA missions. If an official at the Center responsible for the mission selected the principal investigators, then the academic scientists would feel that the Center scientists had an unfair advantage. The NASA scientists would be more familiar with the mission and therefore able to prepare better proposals. In addition, they would be colleagues of the Center people handling the selection. If the Space Science Board, made up entirely of non-NASA scientists, handled the selection, then the NASA scientists would feel that academic scientists had an unfair advantage. By mutual agreement between NASA and the Academy, NASA scientists cannot serve on the Board because they would be providing advice to themselves.

NASA procedures were formulated to reduce the fears of these two groups of scientists and to encourage them to participate in NASA's space science program. NASA provides a competitive process that assures equal access to NASA's space science missions for all scientists, whether they are at universities, NASA Centers or in industry, and whether they are domestic or foreign scientists. Administrative scientists at NASA Headquarters, who are no longer conducting research and hence have no conflict of interest, conduct the selection process.

The selection process proceeds through three stages. The first stage, the creation of a trial payload and the design of the spacecraft, was discussed above. Next NASA issues an Announcement of Flight Opportunity (AFO) to scientists to inform them that NASA intends to proceed with the mission and invites them to submit a proposal to conduct experiments during the mission. After the proposals are submitted, they are evaluated, and a final selection is made by NASA Headquarters.

THE ANNOUNCEMENT OF FLIGHT OPPORTUNITY

As soon as the division director is reasonably sure that the mission will be approved by NASA senior management and by Congress, he or she will issue an AFO. The AFO specifies the objectives of the mission and invites scientists to propose investigations. It gives the ground rules for the proposals and the deadline for their submission.

The AFO is a very important document. Several (sometimes 100 or more) teams of scientists will spend several months preparing their proposals. Each team consists of scientists, engineers and financial analysts who use the information in the AFO to prepare the scientific, technical and financial parts of their proposals. Their written proposal is the final and generally the only opportunity they have to persuade NASA to select their experiment. (Sometimes competing scientists are invited to brief the reviewers.) NASA bases its selection on the written proposal. Once the procedure is completed and the experiments are selected, it is almost impossible for a dissatisfied scientist to overturn the decision. Once the selections are made and contracts awarded, the principal investigator's team is legally obligated to produce the instrument, conduct the experiment and publish the results. NASA is legally obligated to provide funds and space on the spacecraft and to conduct flight operations and provide data to the investigator.

Careful preparation of the AFO is essential. Large amounts of time and energy are required to prepare and evaluate the proposals. If the information in the AFO is inadequate or wrong, experimenters may be discouraged from competing, or experimenters with instruments not suitable for the spacecraft may be selected, which can lead to costly overruns or schedule slips.

The preliminary systems engineering done by the Technical Working Group and the Payload Working Group plays a crucial role in the preparation of the AFO. The AFO contains a description of the trial payload and the spacecraft generated by the two working groups. The AFO specifies the subsystems planned for the spacecraft in sufficient detail so that the proposers can design their instruments to function in harmony with subsystems. The AFO must specify any special requirements for the instruments such as the need to keep electromagnetic interference, nuclear radiation levels or outgassing below specified levels. The thermal characteristics of the spacecraft are described, and the thermal specifications that the instruments must meet are included.

The AFO specifies the date the proposals must be returned and in some cases limits the number of pages of a proposal to avoid getting lengthy proposals loaded with extraneous information.

EVALUATING THE PROPOSALS

The scientists send their proposals to the division director at NASA Headquarters who is responsible for the mission. After receipt of all proposals, the division director forms two groups to assist in the evaluation. The first group, chaired by the program scientist, consists of scientists who are peers of those proposing experiments and who will evaluate the scientific and technical merits of the proposals and assign them a priority for inclusion in the mission. This group of scientists must be free of any legal conflict of interest with respect to any of the proposals, which is the reason why they cannot be chosen until all the proposals are in. The second group consists of engineers at the project management Center similar in membership to the Technical Working Group (in many cases it will be the Technical Working Group). This group will examine all the proposals to see if the instruments proposed are compatible with the spacecraft and judge whether the proposer has the team and the facilities required to carry out the investigation.

As soon as the division director has the proposals, copies are sent to both groups. After the two groups complete their work, they send the results of their evaluation to the division director. If an otherwise high priority investigation is incompatible with the spacecraft, the division director may ask the project team to conduct a short study to determine whether the instrument or the spacecraft can be modified to make the two compatible and, if so, to prepare an estimate of the costs involved.

After receiving the evaluation made by the scientific working group and the project team, the division director and the chief scientists prepare a list of the principal investigators who they think are the best qualified to accomplish the objectives of the mission. Their selection is based on, and must be consistent with, the evaluations of the scientists and the project team. The division director is free to choose between two competing proposals that have been given the same priority by the scientists but is not free to pick a proposal that was given a lower priority. In other words, the division director must select a principal investigator whose proposal was placed in Category I by the scientific working group rather than pick an investigator whose proposal was placed in Category II, even though the Category II experiment might be cheaper or easier to integrate with the spacecraft. The instruments of the principal investigators selected must be certified compatible with the spacecraft or the division director must have the results of a study that shows that the instrument or the mission can be modified to make the instrument compatible. Since each of the investigators selected has proposed a specific instrument, in the process of selecting the investigators the division director has also selected the suite of instruments that will make up the payload for the mission.

After completing the list of principal investigators and the justification for their selection, the division director takes the recommendations to the members of the Space Science Steering Committee for their review and recommendation.

THE ROLE OF THE SPACE SCIENCE STEERING COMMITTEE

The Space Science Steering Committee is composed of the directors and the deputies of each of the program divisions in the Office of Space and Applications. Traditionally, if the director is an engineer, the deputy is a scientist and vice versa. Thus the Space Science Steering Committee consists of roughly equal numbers of scientists and engineers and is capable of reviewing the merits of investigators, the selection procedure, and all other technical and managerial aspects of the mission. It is chaired by the chief scientists in Office of Space Science and Applications and reports directly to the associate administrator for that Office.

The Space Science Steering Committee reviews the investigations that have been selected and the process by which they were selected. It reviews the investigations for their scientific and technical merit and for their compatibility with the spacecraft. If there are any objections or reservations raised by anyone about the payload, the Space Science Steering Committee reviews those objections. Normally the investigators chosen by the division director are accepted; however, if a member of the Steering Committee objects to a selection or questions the selection process, then the Committee may send the division director back to prepare a different version of the payload.

The Space Science Steering Committee serves as the court of final review for a payload. By its acceptance of the principal investigators and their instruments, it certifies that, up to this stage, the user requirements have been properly incorporated into the systems engineering process for the mission. After the members of the Committee complete their review, the chairperson sends their recommendations to the associate administrator of the Office of Space Science and Applications who approves the investigators. After approval of the investigators by the associate administrator, the only way to change an investigator or an instrument is to appeal over the head of the associate administrator, to the deputy administrator or the administrator of NASA. Only once in the past 30 years has the decision of an associate administrator been reversed. In that case, NASA modified its selection procedure to facilitate the selection of investigators for the Apollo-Soyuz Mission. The chairperson of the Space Science Board objected to the change; NASA redid its selection and followed the normal procedure.

THE ASSOCIATE ADMINISTRATOR'S APPROVAL

After the associate administrator approves the principal investigators, each of them is sent a letter to inform them of their selection and to give them any guidelines or qualifications that come from the selection process. For instance, only a part of the investigator's proposal may have been approved or the investigator may have agreed to provide environmental data to other investigators on the mission to aid them in the interpretation of their data. Funding for the mission may be limited; the associate administrator may direct each investigator to control costs very carefully and request that some aspect of the investigation be modified or excluded if it becomes apparent that the costs will exceed the funds allocated for the investigation. If the interest in the mission is high and the funds are limited or the resources of the spacecraft, such as the weight, power and telemetry, are very constrained, the associate administrator may give provisional approval to one or more investigators pending an analysis by the project to determine if the resources are available.

The associate administrator's letter to a principal investigator is an informal contract between the associate administrator and the principal investigator that obligates the investigator to devote the time and energy required to accomplish the objectives of the investigation. It obligates the associate administrator to proceed with the mission and provide the resources and assistance that the principal investigator will need.

At the same time the letters are sent to the principal investigators, the associate administrator also sends a letter to the director of the Center responsible for managing the project. This letter notifies the director of the investigators selected and the qualifications or guidelines that have been given. The letter is accompanied by the authorization and transfer of funds that enable the project team to negotiate contracts with and fund the work of the principal investigators. This contract should provide for the support of the principal investigator and specify the work to be done during design, manufacture, preflight testing, operations, analysis of the data and publication of the results. The funding for data analysis is normally carried in a separate line item in the Space Science budget and is transferred to the Center through a separate channel at a later date. Regardless of how the funding for the operational phase is handled, the associate administrator should require that the project team provide for data analysis and publication of the results in these contracts with the principal investigators. The incorporation of the user requirements into the systems engineering process will not be complete unless all phases of the mission are considered, including data analysis, interpretation and publication of the results.

The Space Science Steering Committee's review and the associate administrator's approval of the principal investigators complete those phases of the mission that are led by the division director at NASA Headquarters. Once the investigators have been selected, the focus of the work shifts from Headquarters to the Center, where the project manager and the project scientist take over the technical and scientific leadership of the mission. They are responsible for the final steps in the incorporation of the users requirements into the systems engineering process.

ASSESSMENT OF THE PRINCIPAL INVESTIGATORS

When the associate administrator for the Office of Space Science and Applications selects the principal investigators and authorizes the Center to negotiate contracts with them, the responsibility for working with the scientists is transferred from the division director and the program scientists at Headquarters to the project manager and the project scientists at the Center. Receipt of the letter triggers an intensive assessment by the project manager of each investigator and of the status of each instrument. This assessment should be completed prior to the beginning of preliminary design activity.

The assessment is conducted by a team appointed by the project manager. The team consists of several engineers from the Center. A key member of the project manager's review team is the project scientist, who, among other tasks, serves as the communication link between the investigators and the project team.

THE ROLE OF THE PROJECT SCIENTIST

The Center director, with concurrence of the Office of Space Science and Applications associate administrator, appoints the mission's project scientist. This project scientist has a powerful role during a scientific mission, quite different from that of the project manager and, at this stage, equally important. If the project scientist and the project manager have a conflict they cannot resolve and that may affect the mission's scientific outcome, the project scientist is expected to carry the case to Center management and, if it is a good case, to prevail.

The project scientist should have as vested an interest in the scientific success of the mission as the one who conceived the mission or as an investigator on the mission. As an experienced space scientist and person who has conducted investigations in space, the project scientist should understand what information the project needs from the principal investigator in order to conduct the mission and should be able to accurately communicate those requirements, and the reasons for them, to the scientists. The project scientist should understand the technical requirements submitted by the principal investigators and be able to communicate them to the project. In addition, the project scientist should be able to judge which of the requirements of the principal investigator are mandatory and which are only highly desirable so that the resources of the project are not squandered. Conversely, the project scientist should be able to sort out the highly desirable from the mandatory requirements of the project manager so that unnecessary constraints, reporting requirements or reviews are not placed on the principal investigators. Clearly, the project scientist must have the confidence of the project manager and the investigators on the mission in order to succeed. The assessment of the principal investigators provides an excellent opportunity for the project scientist to become a reliable representative of the scientists to the project team and of the project team to the scientists.

People ask, why all this concern about the communication channel between the project and the investigators? Why can't the project manager deal with the investigators just as one would with the person responsible for any other subsystem on the spacecraft? Early experience in space science showed that a project manager who was not a scientist, or who did not have a strong competent project scientist working with him or her, usually got into one of two kinds of trouble. Either the project manager regarded the scientists as all powerful and gave in to all their whims, thereby driving the costs of the mission out of sight, or the project manager regarded the scientists as overly bright children and overrode their legitimate requests, thus causing their instruments to fail or forcing the scientists to complain to Center management or NASA Headquarters and try to get the project manager replaced.

FACT FINDING

The initial assessment of each principal investigator by the project team is the most important part of the incorporation of the user requirements into the systems engineering process of a mission. The primary purpose of the assessment is to determine the technical requirements of the instruments and their compatibility with each other and with the spacecraft and the operational equipment. In addition, it provides the project manager with the first opportunity to determine the experience and capability of each principal investigator and of the team, and to assess whether the investigator's institution can and will provide the support that will be needed.

The assessment begins with "fact finding," a systematic effort by the review team to collect information about the investigators. The team conducts its review at the investigator's institution, rather than bringing the investigator and the team to the Center. A visit to the institution enables the review team to not only examine the laboratory model of the instrument, but also to review the calculations and test results that support the design. The team can review the facilities that will be available to investigators to develop, test and calibrate the flight instruments. If the investigator plans to have most of the work done by a contractor, then the review team conducts a similar review at the contractor's plant.

The review should cover all the elements that are required by the investigator to complete the objectives of the experiment. By "all the elements," I mean all the pieces of hardware, all the facilities, all the testing gear that will be required, and all the work and the people that will be required to enable the investigator to design, build, test and fly the instrument. In addition, the review should identify all the computers, all the programs and all the software that the investigator will require to analyze the data and publish the results. The review should cover the entire mission, from design and development, to testing and calibration, to placement of the published results and of the data in the archives. The plans, scheduled actions and funding requirements as a function of time are key elements to be reviewed. The impact of project requirements on the investigator or the instrument should be covered in the review. Throughout the review, its two-way nature must be emphasized. The purpose of the review is to determine what the investigator requires of the project and to inform the investigator of the requirements of the project.

The review begins with information and data collection by the team. The team must collect information on the technical resources on the spacecraft that the instrument requires such as weight, telemetry, band width, volume, power, commands and thermal control.

The team must collect data on the engineering constraints imposed by the instrument on the spacecraft, including but not limited to:

Location of the instrument
Look angle and field of view
Pointing and stabilization required
Operational requirements
Special treatment during testing, launch, and operations
Limitations on vibration and shock
Limitations on stray electromagnetic fields
Limitations on material surrounding the instrument
Limitations on outgassing.

The team needs to know the facilities that will be required by the instrument and their availability, either at the investigator's institution, the contractor, or at the field center or its contractors, including but not limited to:

Vacuum chambers
Shock and vibration tables
Solar simulators
Computers
Special test and calibration facilities
Special data handling and analysis facilities.

The team must collect information and plans for the funding, manpower and management capability that will be required by the investigator at the host institution and by the project team to monitor the work of the investigator.

Obviously, not all of this data will be available at this first review. However, where information is not available, the need should be established and the project manager and the principal investigator must formulate a mutually acceptable plan as to who will generate the information and on what schedule.

This initial data gathering phase provides an excellent opportunity for the project manager and the systems engineers to assess the capability of the principal investigator and the team. NASA policy makes the principal investigator, responsible for all phases of the investigation, beginning with the design of the instrument, continuing through to the delivery of a calibrated, tested and flight worthy instrument, and culminating in the publication of the results. During the review, the principal investigator should demonstrate understanding and the ability to discharge this responsibility and should be able to describe how to conduct the day-by-day work of the team. The principal investigator should state whether the day-by-day work of the team will be under the investigator's direction or whether a manager will be appointed to direct the work. If a manager is appointed, do the principal investigator, the manager and the project manager all understand the limits of the authority of the manager? What decisions can be made by the manager and which ones must go to the investigator? Has the investigator delegated sufficient authority to the manager so that decisions can be made and the work can be kept on schedule? How does the principal investigator plan to oversee the work of the manager? Does the investigator plan to attend certain key reviews to see how things are going? Will the manager give weekly reports?

The project manager and the principal investigator should agree on which reviews the investigator will attend and which can be delegated to the manager. They also need to agree on how they will resolve disputes that will arise between the principal investigator's manager and the project manager.

If the principal investigator plans to handle the day-to-day operations, another set of questions needs to be asked. Is the investigator prepared and able to spend the time and energy to handle the daily work? Is the investigator prepared to travel to the Center or to a contractor when reviews must be held and decisions need to be made? Is the investigator prepared to give up other research during the development of the instrument?

Appointing a good project manager is generally better for the investigator and the team. The project manager can concentrate on the daily activity of managing the team and the investigator can focus on meeting the requirements that will be levied by the project manager and the team.

The review team needs to ask other questions. Is the investigator's team adequate for the task? Have they planned their work and laid out a sensible schedule? Are they cooperative and forthright about the status of their instrument? Are the kinds of engineers and technicians that will be needed either on the investigator's team or at the contractor? Has the investigator done a good job estimating the costs as a function of time? Has a reserve been allowed for unforeseen problems, and if so, have criteria and a schedule been laid out for its use? Any weakness in planning or management at this stage, if not corrected, will inevitably result in more serious problems later in the project.

The analysis of the strengths and weaknesses of a principal investigator's team serves an important function in the incorporation of the user requirements into the systems engineering process. If an investigator has a competent team and adequate facilities and equipment, the project manager can reduce the monitoring requirements for that investigator. The investigator can reduce the time allocated for testing and integration and may waive certain tests. On the other hand, if the investigator has a weak team or inadequate facilities, then the project manager has to lay out a project plan and a schedule that takes this weakness into account. Additional money must be set aside to cover overruns. Provisions for additional monitoring must be made and additional time for testing and integration must be allowed. An engineer from the project may be assigned to aid the investigator. The investigator is placed on the list of the project's "Top Ten Problems," thereby alerting the Center management and Headquarters of the problem. Any management or technical problems unearthed in this initial assessment should be treated just as thoroughly and just as promptly as the failure of any subsystem would be treated later in the schedule. Prompt action at this stage will prevent many hardware problems from arising later when there is less time and less money to resolve them.

The review of each principal investigator culminates in the negotiation of a contract between the Center and principal investigator, whereby the investigator is to produce a flight instrument using funds provided by the Center. At the conclusion of the assessment process, a principal investigator will have two contracts: one with the associate administrator of the Office for Space Science and Applications to accomplish the objectives of the experiment proposed, and the other with the project management center to produce an instrument that is ready for flight. A principal investigator who thinks that a Center decision will jeopardize the investigation has the right to appeal the decision directly to the associate administrator of the Office for Space Science and Applications. This appeal channel is rarely, if ever, used.

THE SYSTEMS ENGINEERING PROCESS

Once the review team has completed its fact finding and its assessment of the investigator's capability, the systems engineers are ready to complete the conventional systems analysis of the system. The information the review team has collected enables them to incorporate the user requirements into that process.

By this time, all the broad boundaries of the mission are established; the investigators have been selected, a preliminary design of the spacecraft is available, the transportation system is specified, the total cost of the mission has been set (or a ceiling placed on the total cost) and a preliminary launch date scheduled.

If there is no hard fast launch date, then the launch schedule may become a variable in the systems analysis and shifted forward or back to reduce costs or improve the scientific return of the mission. If it is a planetary mission, however, the launch date is not a variable but is rigorously set by planetary dynamics; the role of the systems engineer is to identify the decisions that must be made and the actions that must be taken to assure the sanctity of that launch date.

In the case of a high priority scientific mission, such as Viking or the Hubble Space Telescope, the scientific objectives may be the primary constraint. The systems engineer can adjust the launch vehicle, the launch date and the total cost to meet the scientific objectives.

For most missions, however, the primary constraints will be technical and financial. The launch vehicle may be specified; there may be a cap on the funding, certain subsystems may be specified and in many cases the spacecraft itself will be specified. In such highly constrained missions, the only variables the systems engineer has to work with are the number and complexity of the scientific instruments that can be accommodated. For such highly constrained missions, the associate administrator of the Office of Space Science and Applications will usually select a core payload that is certain to be accommodated and then add one or more investigations to be included if the systems analysis shows they can be accommodated. In this highly constrained case, the systems engineer takes the requirements of the core payload and the existing constraints and,working closely with the project scientist and the principal investigators, makes a number of tradeoff studies to determine the maximum number of investigations that can be accommodated and the maximum amount of scientific information that can be collected.

The objective of the systems engineering effort at this stage is to plan the entire mission, establish the specifications for the instruments and the spacecraft, lay out a schedule for all the activities of the mission, establish milestones for completion of major activities, schedule the testing and integration work, set a launch date, estimate the cost and lay out a funding plan for the entire mission. The systems engineers identify any technical conflicts that exist between instruments or between an instrument and the spacecraft. Where they find conflicts, they identify the options available to the project to solve them, conduct tradeoffs between the options and recommend the option that they think will produce the greatest scientific return for the lowest cost.

As the systems engineers conduct their analyses, there is a continuous iteration process that takes place throughout the project and among the investigators. Different locations of the instruments on the spacecraft are studied and discussed with the investigators to determine which are best. Tradeoffs may have to be made between the value of adding an investigation and adding more power or more telemetry bandwidth for the core payload. In rare instances, the systems analysis may show that additional resources are available on the spacecraft; then tradeoffs are made to determine how to allocate the resources among the investigators to better accomplish the scientific objectives.

Many complicated tradeoffs are made at this stage in a project. As an example, systems engineers working closely with the project scientist and the investigators may conduct tradeoffs to determine how much data processing should be done on board by each instrument, thereby increasing the weight and power required by the instruments but reducing the complexity of, and the weight and power required by, the communications system of the spacecraft.

Mutually acceptable schedules for the use of common ground facilities such as shake tables, vacuum chambers and calibration equipment are worked out between the project, the investigators and the persons responsible for those facilities. A detailed schedule of all the tests, calibration runs and flight operations is established with each investigator. These schedules, as emphasized repeatedly in this paper, should carry through flight operations and data analysis. Only by doing this can a systems engineer be sure that all the requirements of the scientists have been incorporated into the mission plan. By forcing the occasionally unwilling investigator to sit down and think through the entire experiment, the systems engineer may bring to the surface a major technical problem or an inadequate cost estimate.

Once the entire mission is laid out, the investigators accommodated, their expenses estimated and a launch date established, the systems engineer must estimate how much and what kind of resources need to be reserved for unanticipated problems. Extra slack time must be placed in the schedule to accomplish unanticipated work. The systems engineer must reserve some weight, power and communications capability for shortages that will inevitably arise. Funds to cover overruns must be reserved and a schedule by which the funds are to be released must be prepared. If there is no schedule for the release of reserve funds, then they may all be used up in the early months of the project, leaving nothing for the major problems that will occur later.

The project manager and the overseers at the Center and Headquarters should examine any deviation by an investigator from the planned use of the reserves with the same care they would examine an instrument that is not meeting its design specifications or its milestones. Such a deviation in the rate of use of reserves may identify a weakness in the investigator's team or in the design of the instrument early in the development cycle. If the project manager takes prompt action when an unexpected use of the reserves is first seen, technical or schedule problems that may occur later in the development phase can be eliminated or reduced.

At this time, the project manager establishes another important policy—how the information about the reserves will be treated. The project manager can choose to operate somewhere between two extremes: "everything on the table" or "hold all the cards close to the chest." In the first extreme, everybody in the project is informed, including all the subsystem managers, all the principal investigators and the contractors, exactly what the reserves are, who is holding them and the schedule for their use. At the other extreme, the project manager treats the reserves as highly classified information known only to the project manager and possibly some of the senior management. Both extremes have worked. The choice largely depends on the experience and personality of the project manager and NASA's current management philosophy. A new, insecure or weak project manager may want to keep this information confidential to help control the project. A more confident project manager may choose to operate an open system. If a project manager chooses to operate an open system, there must be a willingness to accept a high level of acrimony in the project. A principal investigator fighting a weight problem or overrunning the budget will eye a compatriot's reserve and scheme to get it. On the other hand, by operating in an open manner the project manager may create a more healthy climate of trust between the investigators and the project team and thereby discover problems earlier than if all the reserves are kept secret. Sharing knowledge of the problems and the reserve being maintained can help a project manager promote teamwork on the project, raise the morale, and encourage the investigators to carefully manage their reserves. On the other hand, if NASA's current policy is to pull all identifiable reserves into a Headquarters reserve to be held by the comptroller, then project managers will instinctively bury any financial reserves somewhere in the project.

Ultimately, the user requirements will be assimilated into the systems engineering process, the preliminary designs will be completed, the schedules established, and the rate of expenditure established. When this is done, the project is ready for its first major design review, the preliminary design review.

PRELIMINARY DESIGN REVIEW

The Preliminary Design Review (PDR) ends the preliminary design work, and completes the incorporation of user requirements into the systems engineering process. All aspects of the mission and all future activities required to accomplish the mission should be planned by this time.

The choice of a chairperson for the PDR depends upon the complexity, cost and national interest in the project. The division director may chair the PDR of a routine, small scientific project. The associate administrator for the Office of Space Science and Applications will chair the PDR of a larger, more complex mission. The administrator or deputy administrator of NASA may chair the PDR of a large, complex, costly, highly visible mission such as the Hubble Telescope, or Earth Observing System. The chairperson should be someone who thrives on crowds and controversy and has a vast curiosity about the mission and a penchant for uncovering unforeseen or concealed problems. The chairperson should use the PDR to identfy and resolve any issues that the project team or the investigators may have overlooked or may be trying to avoid.

The good chairperson goes around the room after the discussion of a controversial item and questions the key people involved to see if they all understand and agree on the project's plan. The chairperson of the PDR cannot be a "shrinking violet" or an introvert (at least not during a PDR).

The project manager conducts the review. Attendance from Headquarters includes, but is not be limited to: the associate administrator for the Office of Space Science and Applications or a designee, the division director, the program manager, the program scientist, the financial analyst, the NASA comptroller or the designee, and the associate administrators for the Offices of Space Flight and Operations or their designees. Someone from the Office of International Affairs attends if there are foreign investigators or if it is a joint mission with another country. Attendance from the Center will include the director, the financial analysts, representatives of the engineering disciplines and the systems engineers. All the principal investigators attend. Senior people from the major contractors also attend. If the PDR is for an applications mission, then senior people from the agency who will use the system will attend.

The chairperson expects the project manager to present a clear, concise statement of the overall objectives of the mission. If there are other nonscientific objectives for the mission—if, for instance, one of the objectives is to test a new subsystem, a new spacecraft or a new tracking system—then the project manager is expected to clearly specify the relationship and priorities between those other objectives and the scientific objectives. The chairperson should make sure that all objectives are clear, understood and agreed to by the attendees.

The project manager should present a complete schedule, extending from the PDR through the Critical Design Review, on through development, testing and calibration of the instruments and continue on to launch operations, data analysis and publication or use of the results. Slack time should be clearly shown. Even though detailed plans for operation and data analysis may not be complete at this time, the systems engineering process should have produced a list of the facilities required and a schedule for their use. Very often, the examination of the mission's schedule at the PDR will uncover potential conflicts for the use of facilities or an underestimate of the cost of some phase of the mission.

The chairperson reviews the status of each instrument. Ideally, the review of an instrument will consist of two parts, a presentation by the principal investigator followed by the project scientist's assessment of the status of the instrument. The principal investigator should describe the experiment, its objectives and how they relate to the objectives of the mission. The principal investigator should describe the instrument, show the schedule and slack times, and present a cost breakdown and a funding schedule. The investigator should identify any issues with the project manager, including any foreseeable technical and procurement problems, and list the top four or five problems. The project scientist should then give the project's view of the status of the instrument and should state whether the project agrees with the status as presented by the investigator. The project scientist should present any concerns the project has about the principal investigator, the team, the institution or the contractor.

This review by the project scientist at the PDR should not lead to a confrontation between the principal investigator and the project scientist or the project manager; through earlier discussions, each should be aware of what the other intends to say; each should be aware of the concerns of the other and at the review they should present a jointly developed plan to solve the problems that exist. The project manager and the principal investigator should understand and accept the actions that the other intends to take to resolve the problems. If the investigator has only a tentative approval to fly on the mission, then the actions and milestones should be specified that will lead to final acceptance or rejection.

The project manager or the manager's designee should review the status of the other elements of the mission, their schedules and problems. If the cost or configuration of a subsystem is being determined by a requirement of a particular investigation, that fact should be presented so that senior management and the principal investigator can decide whether the particular aspect of the investigation merits the additional cost or complexity.

The project team should present an overall assessment of the instruments and their interaction with each other and with the subsystems on the spacecraft. The project manager may elect to divide the experiments into two groups: one group consisting of those investigations in which the design of the instrument is on schedule, within budget, and the investigator is not in need of careful monitoring; the second group consisting of those instruments that have major problems, that will require careful monitoring and perhaps even a backup instrument.

The project manager should review the status of the resources available to the project, the reserves that are being held and the schedule for their release. At the conclusion of the PDR, the project manager should identify the top 10 problems for the overall project and describe plans to resolve them.

At the conclusion of the PDR, all the participants—Headquarters, Center management, the project team, the principal investigators and the subsystem managers—should all understand and accept the status and requirements of the investigations scheduled for the mission. The principal investigators should agree with the status of their experiment as presented, and they should understand and be prepared to accept the requirements and meet the schedules that have been placed on them by the project.

Once the actions that were assigned to the project and the investigators by the PDR have been completed, the requirements of the investigators should be incorporated into the systems engineering process. The project team and the investigators are then ready to proceed with the detailed design and manufacture of the instruments and the spacecraft.

The majority of the systems engineering effort required to incorporate the user requirements should be complete at this time. Normal project management and engineering techniques should be adequate to complete the integration of the investigators into the mission. There will, however, be a continuing need for systems engineers to support the project team. No matter how good and how complete the systems engineering effort has been, and how carefully the PDR is conducted, problems will still be encountered in the instruments or in the subsystems and changes will have to be made. The systems engineer will have to trace the impact of those changes through the system, identify the problems that are created and provide the options for their solution. Inevitably, there will be a shortage of resources available—additional power or weight required—and the systems engineer will have to assess the system to see how the resources can be found and analyze the impact of using those resources. Occasionally, excess resources will become available; the systems engineer will have to examine these extra reserves and determine how they can best be applied to enhance the quality of the mission.

As the work progresses, the engineers will eventually understand the instruments and their spacecraft, their designs will be frozen, all the options will be eliminated and the systems engineer will no longer be needed. Sometime before this stage is reached, the good systems engineers will become bored and will move on to a new system with new challenges.

GLOSSARY

Mission. An effort to increase human knowledge that requires the launch of one or more spacecraft. A mission begins with the initial concept and concludes with the publication of the results.

System. All the tasks and all the equipment, both ground and space based, required to accomplish a mission.

Systems engineering. The systematic planning activity that begins with the mission objectives and the requirements of the scientists and turns them into specifications for hardware and facilities, conducts tradeoff studies between competing subsystems, analyzes the interaction between the subsystems to eliminate unwanted interference, and prepares schedules, cost estimates and funding plans.

Program. The formulation and documentation of a mission prepared by NASA Headquarters and used to obtain authorization and funding from Congress to conduct the mission.

Project. All the equipment produced or purchased by, and all the activity conducted and directed by, a NASA Center to accomplish a mission.

Division director. An individual at NASA Headquarters responsible for a group of related scientific programs.

Program manager. A person, usually an engineer, at NASA Headquarters in charge of a program. A program manager reports to a division director.

Program scientist. A scientist at NASA Headquarters responsible for formulating the scientific objectives of a program. A program scientist reports to a division director.

Project manager. The person, usually an engineer, at a NASA Center who is responsible for the success of a project. The project manager reports to the senior management of the Center.

Project scientist. The scientist at a NASA Center responsible for accomplishing the scientific objectives of a project. The project scientist reports to the senior management of the Center.

Principal investigator. A scientist, selected by NASA Headquarters, to conduct an experiment during a mission.

Systems Engineering and Integration Processes Involved with Manned Mission Operations

by Eugene F. Kranz and Christopher C. Kraft

The quality of the systems engineering and integration (SE&I) process determines the viability, effectiveness and the survivability of major NASA flight programs. In mission operations, SE&I is the process by which the technical, operational, economic and political aspects of programs are integrated to support the program objectives and requirements consistent with sound engineering, design and operations management principles.

Major flight programs involve operational, cost, and political elements and priorities, international prerogatives, and often poorly focused utilization requirements, in addition to traditional technical trades, technology utilization, and interface definition and control. This combination demands an effective SE&I process that spans and involves all these elements.

SE&I, therefore, is a distributed process that involves the structuring and integrated management of a program within and between the program, project and technical levels, with a life cycle consistent with the program phase. SE&I must anticipate program needs by providing clear technical assessments, trades and alternatives aimed at satisfying the program objectives and requirements.

This paper will describe the key principles and processes used within mission operations, emphasizing the pre-mission preparation activities most useful for describing the principles of an effective SE&I process.

Early Development of Mission Operations

The development of mission operations capabilities for manned space flight involved a rapid evolution from the traditional method of aircraft flight test operations used during the early Mercury program to the mature and structured process used for Apollo. The flight experience of the Mercury program revealed the need for a deeper knowledge of spacecraft systems by flight operations teams. It further indicated a need for systems documentation tailored to the operator's real-time task. By the completion of Mercury, a systems handbook had been developed as an "on-console," real-time document for flight systems data. Direct communication was established between the operating team and the manufacturer so that any additional systems data needed during the course of the mission could be obtained. This communication also provided a means for getting engineering judgment on operational trades, whenever time permitted. The flight rules became the focus of operational policies.

The Gemini program required the development of the trajectory capabilities needed for rendezvous and docking, as well as a guided reentry capability. These requirements established the linkage between trajectory; guidance, navigation and control (GNC) systems; and propulsive consumables. The Gemini extravehicular activity (EVA) increased awareness of the relationship between crew, the task and the working environment.

During Apollo, science became the final mission component supported by the operations teams. The Apollo operations team worked in an integrated fashion on all issues involving flight systems, flight design, science and manned operations.

It was during the Skylab program that the first formal and broad-scale application of the mission operations (SE&I) process emerged to support the early flight system hardware and software design. During the Skylab design reviews, many of the review item discrepancies (RIDs) revealed the need for much closer relations between systems design and operational utilization.

The multiple Skylab systems elements, combined with the broad spectrum of scientific objectives and the complexity of manned and unmanned flight, required an early and effective relationship between flight systems designer, scientist-user and mission operations. A Johnson Space Center (JSC) operations team and a Marshall Space Flight Center (MSFC) engineering team joined to conduct a series of systems operations compatibility assessment reviews (SOCARs). During these and all subsequent reviews, the Skylab systems and software handbooks produced by mission operations were used as the baseline reference documentation for the SOCAR. These documents were also used by the JSC and MSFC teams for the flight phase of the program. Skylab real-time operations demonstrated the effectiveness of this relationship between the JSC and MSFC teams.

The mission operations team supported the design and development phase of the Space Shuttle program at the program and project levels and helped develop operational workarounds for flight systems and software deficiencies that could not be corrected before the flight test phase of the program.

MISSION OPERATIONS STRUCTURE

The Mission Operations Directorate (MOD) at the Johnson Space Center is highly integrated and structured around the principal skills needed for mission preparation, planning, training, reconfiguration, facility development, facility operations and real-time flight operations.

Each mission operations element consists of a single functional discipline, e.g., mission design, flight systems, reconfiguration, training, etc. Usually each organizational element is structured to provide dedicated support to either the Shuttle or Space Station. This is believed to be the best way for assuring accountability in individuals and management, avoiding conflicting priorities and providing leadership focus. The only exception is a Flight Design and Dynamics Division (FDDD), which provides integrated flight design for the Shuttle and all programs using Shuttle services.

Each division is responsible for integration within its work area and provides mission operations representation to the project–level boards. Program-level boards are generally supported through the Flight Director Office, by the Operations Division and by the FDDD. Integration between programs is accomplished by the MOD assistant directors for the Shuttle and for the Space Station.

In addition to the internal integration process, each division generally has a horizontal integration responsibility that identifies, collects and documents the capabilities and constraints imposed by other elements. This integration process frequently incorporates participants external to mission operations (for example, participants from the program and the project), as well as the flight system contractor and the payload user. In most cases, this is accomplished by mission operations directed panels that are chartered by the program.

INTRODUCTION TO MISSION OPERATIONS SE&I

This paper will discuss three mission operations functions that are illustrative of the key principles of operations SE&I and of the processes and products involved.

- The flight systems process was selected to illustrate the role of the systems product line in developing the depth and cross-disciplinary skills needed for SE&I and providing the foundation for dialogue between participating elements.
- FDDD was selected to illustrate the need for a structured process to assure that SE&I provides complete and accurate

results that consistently support program needs.

- The flight director's role in mission operations was selected to illustrate the complexity of the risk/gain tradeoffs involved in the development of the flight techniques and flight rules process as well as the absolute importance of the leadership role in developing the technical, operational, and political trades.

Flight Systems Division SE&I

The early Mercury program employed a mixture of operations and engineering personnel to support the real-time operations. Later, flight experience established the need for a full-time systems operations team. The need for an integrated compilation of flight system data usable by the crew and ground team for real-time operations led to early versions of the systems handbooks that are the foundation for today's handbooks. Rudimentary integrated schematics were used for Gemini, but with the Apollo program came more complex inflight computing capability. Consequently, the schematics were expanded to define the computer interfaces and used significantly more of the vehicle design and performance data base within the schematic notes.

As mentioned earlier, the schematics were used for the first time to support the Skylab critical design reviews and the SOCAR. During these reviews, program and project management recognized that the systems operations teams and the systems handbooks were an SE&I asset. The modularity of the Skylab elements, along with the integrated nature of the systems, established the pre-mission role for the systems handbooks to support the flight system design review process as an integrated activity. The usefulness of the handbooks in addressing integrated systems issues was thus formally established. For the Apollo Soyuz Test Program (ASTP), and the Shuttle and Spacelab programs, the preliminary version of the mission operations schematics were completed prior to the flight system critical design review (CDR) and were used as the foundation for the mission operations assessments.

The Systems Handbook Today

Mission operations schematics are developed by the controllers to a common set of internal drafting standards and conventions and use the design engineering drawings, vendor schematics and software source code. For the Shuttle, operations personnel were required to develop Houston Aerospace Language/Shuttle software language skills as a job requirement. Permanent, prime contractor, in-house and in-plant support assures the flow of the raw design data and provides the communications conduit between the systems operations personnel and the prime contractor design engineers so they can address questions as they arise. After the STS-51L accident, all handbook schematics were expanded to provide direct traceability to design drawings by title, drawing number, revision and date.

The systems controllers who develop the schematics derive significant training from using design data and translating this data into an operationally useful format. The schematic development and the integration of data from supporting systems and subsystems provides independent validation of the system design intent. In particular, it identifies issues where the integrated design may have compromised the program intent. The drawing configuration control process requires verification by section and branch chiefs and final approval by the division chief. Formal reviews are conducted before major handbook releases. As a result, the operator and the supervisory chain derive a training benefit from the systems handbook process.

The systems handbooks are used by crews, flight directors, training instructors and mission operations payload support personnel. They are a formal portion of training documentation and are carried in the Shuttle flight data file. The schematics support airborne system troubleshooting and provide a common base for the crew and the ground to discuss suspected problems and follow-on actions. They provide the basis for MOD discussion with the contractor engineering team and with the mission support team.

Flight Procedures. The development of the systems handbook provided the foundation for the development of flight procedures. Three basic categories of flight procedures are developed: the operations checklists, the pocket checklists and the malfunction procedures.

The operations checklist procedures allow the crew and ground systems operations to accomplish a planned activity and are normally developed as blocks of integrated systems activities; for example, aligning the inertial measurement unit. Procedures development requires intimate familiarity with the system; its interfaces, controls, and displays; and with the intended task to be accomplished. Operations checklist procedures cross all systems and technical disciplines, and as a result of their development, provide another level of systems integration and design validation. Procedures associated with an Orbital Maneuvering Subsystem burn, for example, involve loading the maneuver targets into the computer, selecting and configuring engines for the burn, activating the correct digital autopilot, selecting displays, and specifying of data to be recorded.

Pocket checklists are emergency procedures based on the operations checklist. The term "pocket" is used because the checklists must be readily available for critical mission phases and are sized to be carried by the crew in the pockets of their flight suits.

The pocket checklist procedures define the steps to be taken when an unplanned event occurs. These procedures address critical failures and are flight-phase unique. They require knowledge of system performance limits, crew capabilities, failure modes, and crew and ground response times. The emergency procedures therefore provide a bridge from operations checklist procedures into options that allow the crew to continue the current flight phase with modification, to reconfigure to recover capabilities, or to utilize an alternate capability. Figure 1 is a typical procedure used during powered flight for a main B undervolt condition.

The final type of flight procedures developed by the controllers are the malfunction procedures (MALS), which are used when time is available to troubleshoot, locate and define the boundaries of problems that occur inflight. To solve the problem, the crew and ground use the full range of instrumentation available and any visual or external cues available. The procedures are developed in a logical format using a series of "if," "and," and "or" statements. Warning notes are provided, as well as permissive steps when ground and crew consultation is required prior to continuing the procedural sequence. These procedures have allowed the correct isolation of the majority of inflight problems for the Shuttle program.

A final category of flight procedures concern payload operations and involve multiple flight elements.

Flight Systems Organizations. Since Gemini, the MOD flight systems organizations have been structured to address a complete space system. Examples include command service module, lunar module and Shuttle. Each section within an organization has responsibility for an assigned system, with its subsystems, software, instrumentation, display, crew controls, command controls, procedures, mechanical, power, cooling, and thermal and consumable interfaces. During the Skylab program, each organization also had to know about inflight maintenance and support logistics.

The systems organizations of the MOD participate in flight systems design via formal membership on the working groups,

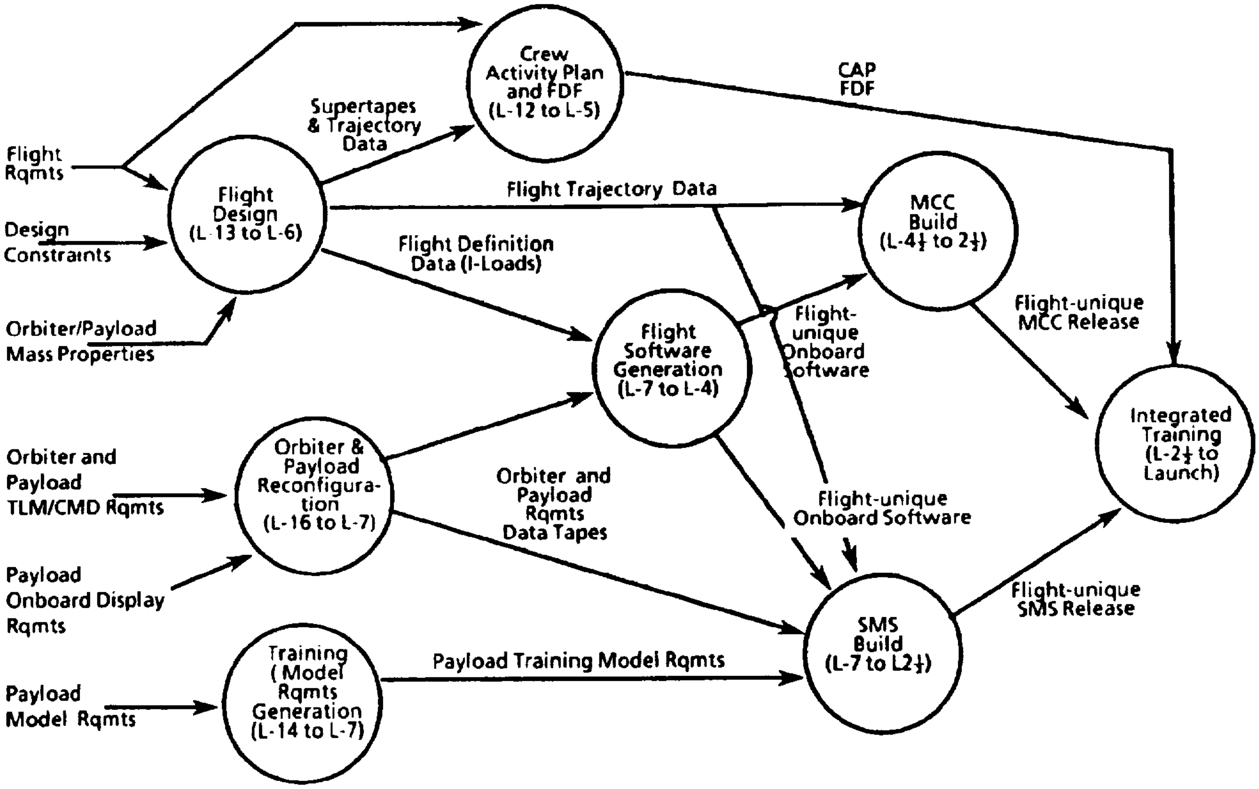

Crew Activity Plan and FDF (L-12 to L-5)
CAP FDF
Supertapes & Trajectory Data
Flight Rqmts
Flight Design (L-13 to L-6)
Design Constraints
Flight Trajectory Data
MCC Build (L-4½ to 2½)
Flight Definition Data (I-Loads)
Orbiter/Payload Mass Properties
Flight Software Generation (L-7 to L-4)
Flight-unique Onboard Software
Flight-unique MCC Release
Integrated Training (L-2½ to Launch)
Orbiter and Payload TLM/CMD Rqmts
Orbiter & Payload Reconfiguration (L-16 to L-7)
Orbiter and Payload Rqmts Data Tapes
Flight-unique Onboard Software
Payload Onboard Display Rqmts
SMS Build (L-7 to L2½)
Flight-unique SMS Release
Training (Model Rqmts Generation (L-14 to L-7)
Payload Training Model Rqmts
Payload Model Rqmts

Figure 1 MOD Production Process Overview (L-Time in Months)

panels and boards established by the program office. During the early design phase, they establish the data base for the development of schematics and procedures for the flight controllers. Because of this, direct contractor liaison is maintained within the MOD systems organization and in-plant.

Development of the mission product line by the systems flight controllers increases their skills and knowledge. In addition, the product line focuses the operations assessments of overall flight system architecture and provides the foundation for subsequent steps. Finally, as a recognized product, it is used by several groups in support of their individual responsibilities. Program SE&I products typically must exhibit the same characteristics—they must pass the value-added test.

The systems operations contribution to the early design and eventual operation of the flight system has been essential in assuring safe, effective and functional system capability for space flight. The perspective of the systems operator provides the cross-disciplinary assessment needed to assure effective overall systems engineering and integration. This perspective is the cornerstone of the real-time capability of the manned spaceflight operations team.

Flight Design Division SE&I

The flight design process involves the integration of payload and engineering requirements with mission objectives to form an integrated mission design. The flight design must satisfy both Shuttle system design and payload design constraints while considering the additional constraints imposed in consideration of safe mission conduct and mission success.

The flight design process is a critical node in the Shuttle mission preparation process. In addition to flight design, the process provides initialization data for the ground

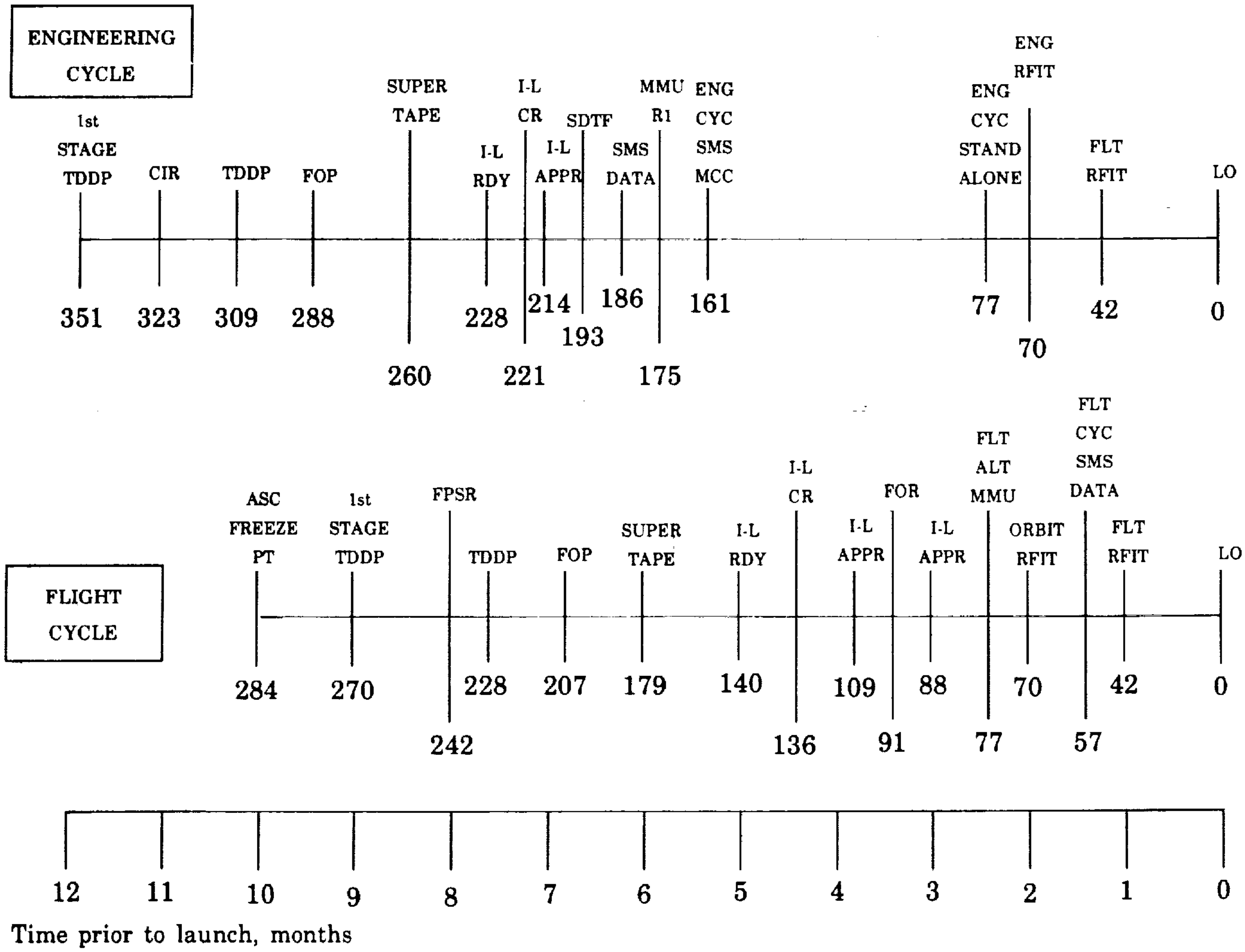


Figure 2 Flight Design Template

facilities, Shuttle primary and backup software, flight and payload planning, and real-time decision support products.

Within the flight design and dynamics discipline there are three mission phase analysis and design work areas—ascent, orbit and descent—and one functional area—real-time operations. The FDD, working in coordination with other mission operations elements, establishes and integrates the propulsive and non-propulsive consumables, abort propellant dump analysis, and manipulator requirements and analysis into the overall flight design. The overall integration of activities supporting a mission is provided by a flight design manager.

The flight design process acquires a vast amount of input data from a wide variety of sources. The input data for the early phase of the program is typical specification data, but during the operational phase of the program it becomes highly flight specific and frequently component specific. A good example would be constraints for engine throttling related to a specific Space Shuttle Main Engine turbo pump.

Flight Design Cycles. The flight design process has three principal cycles designed to satisfy the requirements and lead times of the many users. The conceptual flight profile cycle provides the program office with data for making commitments to the payload customers and assessing the overall suitability of the operations flight design approach.

The engineering cycle supports the initialization of the engineering and test facilities as well as the initial shuttle mission simulator (SMS) training load. The flight cycle supports MCC and SMS initialization for final training and operations, Kennedy Space Center (KSC) Launch Processing System checkout and launch support, Goddard Space Flight Center network support, and range safety. The flight design cycles are under review to determine if a single cycle could be used to satisfy all user requirements. This latter objective requires significant standardization within the program, improved and timely provision of payload specific data and significant training standardization.

Flight Design Documentation. The flight design process is the last of the mission operations processes to be documented as a structured flow from the conceptual phase through the delivery of the launch-loads used for flight. The full documentation of these processes is now contained in 22 volumes of flight design handbooks. Documentation was undertaken to serve four distinct objectives: (1) document the corporate memory of this process before it is lost; (2) establish an error- and omission-free process, necessary because of the critical nature and use of the flight design products; (3) support the design of an integrated computing system as an aid to support the flight design process; and (4) assure consistent design and rationale between similar missions.

The two years after the STS-51L accident were used to safe the flight design system, document the process and initiate a multiyear plan for code conversion, consolidation, documentation and configuration control of all applications software. Process flow charts were developed for every activity involved in the flight design analysis and production activity.

The flight design handbooks developed during recent years have documented the flight design SE&I process and, to a great extent, represent the structure and relationships that must exist to incorporate integrated trajectory design into any space program. These documents are invaluable examples of the structure and approach needed for further space exploration activity. They also provide a good textbook for personnel involved in SE&I management to describe the relation between trajectory, systems, software and objective data. In addition, they define input/output requirements, integration nodes, audit points and interfaces to external elements for data acquisition and transfer.

An Illustration of the Flight Design Process. The integration of the constraints imposed by the flight system, environment, payload and operations in the determination of the launch window will be used to illustrate one aspect of the flight design process.

The launch window is the time period that the Shuttle should launch to achieve precise program requirements. This activity is described in the flight design handbook via three processes that satisfy Shuttle and payload requirements. These processes are further combined and iterated to develop the integrated launch window. This initial step of the process provides input data for subsequent planning involving deorbit opportunities, sequence of events, pointing, thermal assessments and so forth.

The constraints imposed in launch window determination represent the broad range of considerations faced by the flight designer in this task. Where practicable, priorities are established to assist the flight designer. The actual development of the launch window analyses is governed by a 27-page procedure within the flight design handbook.

Flight design is an essential element for space flight. The documentation of this process captured what was in the minds of the talented and imaginative individuals working in this field, and provided the definitive text for future flight design work for space exploration.

For the Space Station Freedom program, MOD has developed process flow charts for all functions that describe the input/output activities within mission operations and between mission operations and the Level II program elements, MSFC, KSC, GSFC and international partners. These flow charts described interfaces, product exchanges and work templates. They were used to define the roles and mission boundaries needed for sustained and effective relationships between participants. Documentation of the SE&I process is absolutely essential to clear and effective role and responsibility definition, and is a primary step in minimizing jurisdictional battles between SE&I elements.

Flight Directors SE&I

The mission operations SE&I process uses the Flight Director Office to provide the top level, multidisciplinary integration, risk/gain assessment and validation of the integrated mission preparation.

Flight directors are selected from the ranks of MOD personnel. Selection is based on leadership, technical abilities, stability and judgment as established by their performance during flight operations. They are already intimately familiar with the operating disciplines, interfaces, flight and ground systems capabilities, crew capabilities and the mission risk/gain process. The challenge for the flight directors is acquiring and maintaining the clear perspective needed for multidisciplinary technical, operational and political trades and leading the many diverse elements to operationally correct risk/gain decisions.

The lead flight director is central to the process for the assigned missions.

Pre-CDR Support. Support to a program from the Flight Director Office is initiated between the preliminary design reviews (PDRs) and CDRs. This phase is characterized by major tradeoffs between program requirements, flight system design, crew and ground and customer roles, schedule and cost. During this period the flight director, supported by all mission operations elements, refines the operating concepts and leads the operational trades involving autonomy, fault tolerance, crew and ground functions, and flight design and payload supportability. As flight system design becomes more focused during this period, the program costs and the real world design trades converge and program tradeoffs must be implemented. As a result, the mission operations integration process is initiated to provide the program and project managers with a clear understanding of available options. The options are generally provided by in the form of operations compatibility studies, similar to the SOCARs described previously, or in the form of an integrated mission design assessment.

CDR Support. The CDR support to the program from the mission operations team is significantly different because of the availability of the mission operations flight systems handbooks and the increased knowledge of the team. The operations team has acquired significant experience in working with the program and project as a member of the change control board (CCB) and through the CCB processes. The CDR represents a milestone for reassessing the design and is frequently the first time that the maturity of the software begins to approach the maturity of the hardware.

The principal contribution from mission operations during this time is in the detailed operational suitability assessments. These assessments concern the mission suitability of the flight system design and involve program requirements, hardware and software design, mission design, and crew and ground capabilities. Through these assessments the preliminary risk/gain trades and fault down options are established, operating philosophies are defined and mission options ascertained. Within mission operations, the CDR is not a discrete process. It is considered one of the many milestones of a process characterized by an increasing involvement by operations personnel in the change boards and control mechanisms established by the program. The involvement extends to the flight preparation period, which has two distinct processes and products representative of the flight director's role in the mission operations SE&I. These processes involve flight techniques and flight rules.

Flight Techniques. The initial flight techniques process was developed, and since Apollo, has been chartered by the Level II program. The process was established to address the growing complexity of the interaction between flight software, flight system and flight objectives. This process provided the technical focus for the operations, engineering and contractor teams to address the use of the as-built flight system, the software, and the crew and ground capabilities in accomplishing flight objectives. During Apollo, the ground system, flight procedures and flight software were the only elements that could be readily changed within cost and schedule considerations. The flight techniques process, assisted by Draper Laboratories and the operational vehicle and software developers, established virtually all of the navigation capabilities for Apollo. They developed the technique for the Apollo 12 pinpoint landing and were a principal contributor to the Apollo 13 return.

The product line of the techniques process is initially the series of detailed meeting minutes, which provide the basis for flight procedures and the rationale for the majority of the flight rules and mission design constraints. The flight techniques process provides the integration of the knowledge base available on the flight system to drive flight designs, procedures and flight rules.

Flight Rules. Flight rules are the fundamental risk/gain policy document for mission conduct. The "flight rules outline preplanned decisions to minimize the amount of real-time rationalization required when non-nominal situations occur from the start of the terminal countdown through crew egress."

The most complex, difficult and critical of the integration processes provided by the Flight Director Office is flight rules development. Flight rules used today trace their beginnings to aircraft flight tests. Rudimentary guidelines were provided for the flight test pilots relative to test conditions, and go-no-go criteria were provided for test continuation or termination. Similarly, during Mercury the rules for selected systems failures were also a simple set of go-no-go criteria involving powered flight abort and mission continuation or termination. Rules also addressed the control center, network and flight instrumentation requirements. Today's flight rules involve sophisticated risk/gain trades across redundant systems, multiple mission phases, engineering and payload objectives, and crew and controller capabilities. They also reflect and tradeoff the payload objectives, crew adaptation and flight system survivability in defining mission duration for off-normal conditions. Additionally, they clearly define the responsibilities of key personnel implementing flight operations.

While the rules are infinitely more complex, the principle of the rules remains the same; that is, "to establish the risk versus gain trades" before the mission, utilizing the full range of operational, program and engineering judgment available in the premission environment.

To assure complete visibility to all tradeoffs involved in the flight rules, rule rationale, techniques data and Systems Operations Data Book (SODB), references are contained in the published rules. The SODB and its variants were developed during

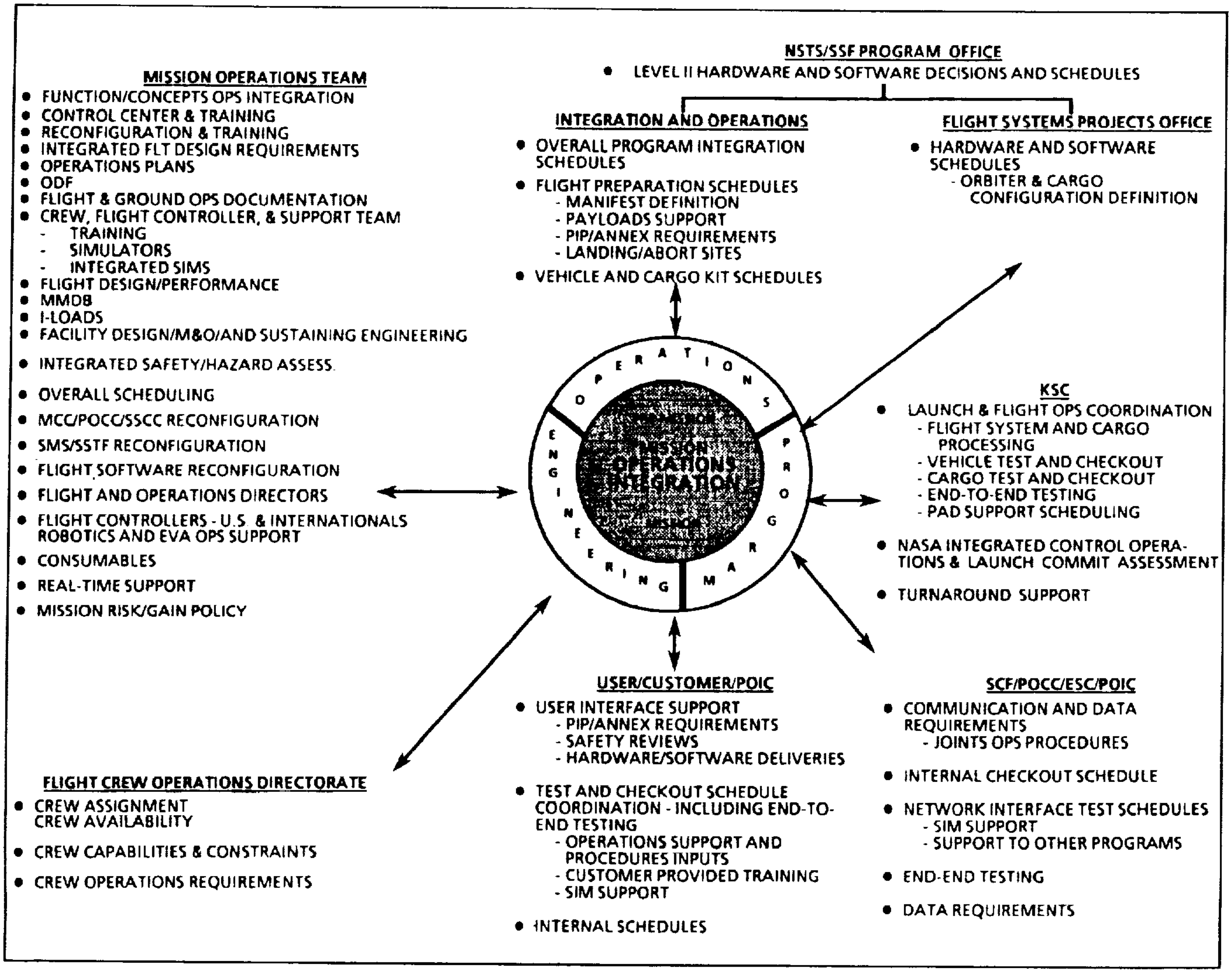

MISSION OPERATIONS TEAM
• FUNCTION/CONCEPTS OPS INTEGRATION
• CONTROL CENTER & TRAINING
• RECONFIGURATION & TRAINING
• INTEGRATED FLT DESIGN REQUIREMENTS
• OPERATIONS PLANS
• ODF
• FLIGHT & GROUND OPS DOCUMENTATION
• CREW, FLIGHT CONTROLLER, & SUPPORT TEAM
- TRAINING
- SIMULATORS
- INTEGRATED SIMS
• FLIGHT DESIGN/PERFORMANCE
• MMDB
• I-LOADS
• FACILITY DESIGN/M&O/AND SUSTAINING ENGINEERING
• INTEGRATED SAFETY/HAZARD ASSESS.
• OVERALL SCHEDULING
• MCC/POCC/SSCC RECONFIGURATION
• SMS/SSTF RECONFIGURATION
• FLIGHT SOFTWARE RECONFIGURATION
• FLIGHT AND OPERATIONS DIRECTORS
• FLIGHT CONTROLLERS - U.S. & INTERNATIONALS ROBOTICS AND EVA OPS SUPPORT
• CONSUMABLES
• REAL-TIME SUPPORT
• MISSION RISK/GAIN POLICY
NSTS/SSF PROGRAM OFFICE
• LEVEL II HARDWARE AND SOFTWARE DECISIONS AND SCHEDULES
INTEGRATION AND OPERATIONS
• OVERALL PROGRAM INTEGRATION SCHEDULES
• FLIGHT PREPARATION SCHEDULES
- MANIFEST DEFINITION
- PAYLOADS SUPPORT
- PIP/ANNEX REQUIREMENTS
- LANDING/ABORT SITES
• VEHICLE AND CARGO KIT SCHEDULES
FLIGHT SYSTEMS PROJECTS OFFICE
• HARDWARE AND SOFTWARE SCHEDULES
- ORBITER & CARGO CONFIGURATION DEFINITION
OPERATIONS
PROGRAM
ENGINEERING
MISSION OPERATIONS INTEGRATION
KSC
• LAUNCH & FLIGHT OPS COORDINATION
- FLIGHT SYSTEM AND CARGO PROCESSING
- VEHICLE TEST AND CHECKOUT
- CARGO TEST AND CHECKOUT
- END-TO-END TESTING
- PAD SUPPORT SCHEDULING
• NASA INTEGRATED CONTROL OPERATIONS & LAUNCH COMMIT ASSESSMENT
• TURNAROUND SUPPORT
FLIGHT CREW OPERATIONS DIRECTORATE
• CREW ASSIGNMENT CREW AVAILABILITY
• CREW CAPABILITIES & CONSTRAINTS
• CREW OPERATIONS REQUIREMENTS
USER/CUSTOMER/POIC
• USER INTERFACE SUPPORT
- PIP/ANNEX REQUIREMENTS
- SAFETY REVIEWS
- HARDWARE/SOFTWARE DELIVERIES
• TEST AND CHECKOUT SCHEDULE COORDINATION - INCLUDING END-TO-END TESTING
- OPERATIONS SUPPORT AND PROCEDURES INPUTS
- CUSTOMER PROVIDED TRAINING
- SIM SUPPORT
• INTERNAL SCHEDULES
SCF/POCC/ESC/POIC
• COMMUNICATION AND DATA REQUIREMENTS
- JOINTS OPS PROCEDURES
• INTERNAL CHECKOUT SCHEDULE
• NETWORK INTERFACE TEST SCHEDULES
- SIM SUPPORT
- SUPPORT TO OTHER PROGRAMS
• END-END TESTING
• DATA REQUIREMENTS

Figure 3 Mission Operations Integration

Gemini by mission operators with support by the prime contractor for the purpose of documenting the performance capabilities and limitations of the flight system. Since Apollo, the SODB has been maintained by the prime contractor, with mission operations as the primary user.

The leadership function provided by the flight director, using the flight techniques and flight rules process, provides the focus for the integration of flight-specific work within mission operations.

The rules and rationale section in the all-flights document is almost 900 pages. The flight-specific annex published for each mission is about 70 pages. It is provided to address the flight-unique objective and payload risk/gain trades for each specific mission, flight objective and payload element.

Flight directors, like program and project managers, depend on a matrix structure of organizations to accomplish their responsibilities. The flight directors are consistently successful because their roles are well defined, and because the integration techniques are facilitated by the MOD organization structure as well as by clearly defined product line and support processes. These characteristics must exist to successfully cope with the complex issues imposed by all mission elements.

PRINCIPAL REQUIREMENTS OF AN EFFECTIVE SE&I PROCESS

The mission operations elements, processes, and products are oriented to the singular objective of safe and successful manned flight operations. The spacecraft on the drawing board, like the ship in a harbor, is a safe ship, but that is not what spacecraft and ships are for. The mission operations job is to take the spacecraft from the harbor of the drawing board into space, accomplish a mission and then safely return the spacecraft to Earth.

In recognition of this responsibility, the mission operations processes are structured to assure effective policy, objective, system and operations integration. Within this framework, complex risk/gain trades are conducted and validated at all levels, culminating in a completely independent and dynamic assessment and stress test during the integrated training process.

The mission operations process can illustrate the principles necessary to a successful SE&I. It is believed that these principles are useful to other SE&I elements that have the responsibility for NASA flight programs at the project and program level.

1. *SE&I must have necessary roles and missions that are clearly defined by the program and implemented by the project and technical organizations.*

SE&I is necessary because the integration processes needed to address the technical, operational, political and economic aspects of major programs are complex.

The value-added principle is the basic test that should be used in determining role and mission assignments.

SE&I by its nature will be controversial and participating elements may stonewall the process. When this occurs, the program, project or technical manager must quickly and personally address the issue, establish a program position and demand the support required.

2. *SE&I must utilize the existing capabilities of organizations.*

SE&I is the "integration" of the technical, operational, economic and political aspects needed to support a major program. The broad range of work, skills required and complexity of issues virtually precludes the development of a single SE&I organization for a major program. SE&I responsibility must be distributed to be successful.

3. *SE&I elements must recognize and accept that major and complex programs will involve technical, operational, political and economic needs.*

Major programs must address and support the needs of the various constituencies involved in establishing the program and must consider all of the economic issues involved in program development and operations. This recognition is essential if NASA and its contractors are to develop a more flexible and responsive approach to program management.

4. *SE&I must have a process-based structure and a defined product line and life cycle.*

The complexity of SE&I requires a structured process to assure all interfaces are addressed, proper responsibilities assigned, and SE&I is effectively mechanized. SE&I requires a solid grasp of all the elements to be brought together, where the elements logically come from, where they fit in the sequence, what the end product is and what the alternatives are.

SE&I can be accomplished by a few gifted people for a limited time, but without structured processes, SE&I will become inefficient, outputs will not meet schedule commitments, "more integration resources will be needed, and the downward spiral will begin." SE&I is not provided by massive application of resources. It comes about by structured processes that clearly establish the roles and responsibilities of the supporting elements and use them effectively.

The SE&I process definition is also used to establish the product line of participating elements and define input/output requirements. This product line must be phased to the life cycle of the program.

5. *SE&I leadership must exist within all elements of the SE&I process structure and must be clearly recognized and accepted by the assigned individuals and their organizations.*

Accepting an SE&I leadership role is to recognize and accept conflict, particularly in the project and technical organizations. Organizations assigned an SE&I role must recognize and accept the technical, operational, political and economic implications of the SE&I role. SE&I must address the needs of the program, which must supersede the needs of individuals and organizations.

SE&I within NASA's flight programs is a constantly evolving and complex process involving many conflicting requirements that must be brought together to support program needs throughout the program's life cycle. An SE&I process that is effectively structured with distributed responsibilities will support program needs and recognize many of the prerogatives of the existing NASA elements. Each complex program, however, will have some elements that do not fit neatly into the existing NASA infrastructure because of economic, political or other considerations. SE&I will always be controversial, in structure and in implementation.

SYSTEMS ENGINEERING CONSIDERATIONS FOR OPERATIONAL SUPPORT SYSTEMS

by Robert O. Aller

Operations support as considered here is the infrastructure of people, procedures, facilities and systems that provide NASA with the capability to conduct space missions. This infrastructure involves most of the Centers but is concentrated principally at the Johnson Space Center, the Kennedy Space Center, the Goddard Space Flight Center, and the Jet Propulsion Laboratory. It includes mission training and planning, launch and recovery, mission control, tracking, communications, data retrieval and data processing.

Operations support of NASA's space flight systems during the 1960s and the 1970s was associated with operations characterized as Research and Development. Flight programs were a single flight of limited duration or a series of flights to obtain specific data or to demonstrate an operational capability. This required operational support systems to be reactive and responsive to relatively short duration programs.

In the past ten years, this has continued with some notable exceptions. With advances in space and data technologies, the demonstrated capabilities and advantages of space operations and the increased cost and complexity of space systems has led to longer duration and repetitive flight programs. Systems engineering of operational support systems must accommodate this evolution and the increasing operational nature of NASA.

The need for systems engineering is critical to NASA in its preparations for conducting operations in the late 1990s and into the next decade. The planning and implementation of the operational support systems for this era are under way. Proper systems engineering is vital to the development of each new system, as well as to a "total systems engineering" of the functionality and interfaces of the entire operational system.

Implementation, integration and transition of these major changes to the Agency's operational capacity require significant management attention. To assure NASA's future in research, development and operations, this system must be implemented successfully and designed to minimize NASA's operational costs.

TOTAL SYSTEMS ENGINEERING

The need for incorporation of systems engineering concepts and discipline is much broader for operations support systems than the hardware and software systems for which it is normally considered. As noted, operations support is an infrastructure of people, procedures, facilities and systems. Although systems engineering is routinely applied to each new system, the major problems often occur between systems and frequently among people, procedures and facilities. A disciplined systems engineering approach formulating each of these elements in the establishment of the "system" cannot be overemphasized. NASA has learned many times that good system contractors do not necessarily nurture good operational personnel and technicians nor do they necessarily develop usable maintenance procedures. Experience has also shown that facilities not adequately analyzed in conjunction with the planned utilization of the facilities require constant modification to meet operational needs. In considering support capability, each of the infrastructure elements requires analysis and carefully managed selection and attention.

An organizational tier of system analysis from the whole to each element can be applied in a macro sense to assure consideration of both technical and nontechnical systems. A macro analysis of the system involves many considerations; two nontechnical areas that have often caused problems are inadequately skilled personnel and underdesigned facilities.

The nature of operations support requires a spectrum of talents and skill levels. Most newly developed systems have not properly analyzed the experience and skill mix needed nor the number of personnel required, which varies from skilled flight controllers to maintenance and repair technicians. Too often a process to analyze the system operation and system maintenance and repair requirements is not properly developed in advance, resulting in an operations team that is undersized and underskilled.

A second issue is simply undersizing facilities. While managers operate on the "nature abhors a vacuum" principle and insist that each square foot of a new facility needs clear functional definition, too often new facilities are found to be inadequately sized even before they are put into operation. This is particularly true with new operational systems. Facilities should be designed to accommodate the unforeseen. Quite often the unforeseen is a result of an incomplete analysis of the operational and system requirements prior to facility design, but also new requirements will emerge. A contributing difficulty is NASA's facility approval process, which is instituted before a reliable utilization analysis is available. It is prudent to provide capacity for some growth to accommodate new requirements.

Another nontechnical factor that is of increasing importance to NASA is life cycle costing (LCC). NASA has not traditionally incorporated LCC as a critical selection, design or engineering process. The elements critical to LCC have all been managed and considered, but an LCC process has not been established within NASA or by NASA's contractors as a routine process. LCC was used as a contract selection factor by NASA for the first time in 1988 with the selection of the second Tracking and Data Relay Satellite System (TDRSS) Ground Terminal. It is rare that a contractor has an established technique to trade and iterate design cost against operations costs. LCC needs to be a driving discipline to assure that the costs of operating the increasingly more sophisticated flight systems can be controlled. The flight systems of today are projected for 15-20 years of operation. This demands that the operational support systems be analyzed and designed to minimize LCC, or the cost of operations will increasingly erode NASA's resources for new development capacity. NASA and its contractors should establish more sophisticated models of development, operations and maintenance costs that will provide more reliable data for conducting operations cost trades against alternative system designs.

SYSTEMS ENGINEERING AND OPERATIONAL SUPPORT SYSTEMS

Systems engineering for operational support systems follows the traditional disciplines applied to the development of major flight systems. Operational support requirements need to be translated into performance parameters and configurations through multiple iterations to optimize system design. The purview of systems engineering includes requirements definitions and verification, system analysis and design, integration planning, requirements control, configuration control and testing.

While similar to the design and development of major flight systems, the emphasis of the systems engineer for operational support systems is generally to provide generic support to an aggregate of flight programs and the increasing necessity to provide systems with extended operational usefulness. This operational longevity can be attained by systems capable of accommodating change while continuing to provide service. The Deep Space Network operated by Jet Propulsion Laboratory and the Goddard Space Flight Network are excellent examples of major systems that have provided space flight program support with tracking and data retrieval service for 30 years, all of the while undergoing changes to provide support for increasingly complex missions.

In addition to providing generic support to many users, a vital characteristic of support systems is operability. The focus in the vehicle development community is principally directed toward designing a system that optimizes performance; the operations community's focus is directed more toward an effective and efficient operation of the system. Operability emphasizes ease of operation, resistance to system problems and failures, maintainability, reparability, simplicity, efficiency, capacity for growth and modification, and accommodation of users.

These two features, multiple program support and system operability, are key to assuring the proper systems engineering of operations support systems. They are historically the most difficult to sustain as cost and schedule pressures frequently tend to compromise the system's range of utility and operability.

REQUIREMENTS, EVALUATION, VERIFICATION AND CONTROL

Operations systems development is generally driven by new, expanded or improved support service required by new flight programs or expanded program objectives. The systems engineer needs to challenge user requirements to assure the "real" needs are not sacrificed at the expense of low priority, highly demanding requirements. Occasionally, requirements are driven by the fact that new technology is available and not that it is essential (or even desirable) for effective operation. The systems engineer must consider the broad base of program users and not provide a narrow focus of support that overly complicates or ignores operations of the aggregate of users.

While sharply defining real needs, it is equally critical to consider the potential to provide for future capacity. In the information age, the computer (including software), communications, and electronics industries have developed new technologies and capabilities often before a flight program's support requirements are established. The incorporation of these new services needs careful examination and scrutiny; when these new services clearly enable future or expanded programs, however, the operational community should provide them to enhance future operations. An example of capability beyond defined need was clearly incorporated in the TDRSS program in 1975. The TDRSS provides capacity and data rates that will meet the requirements of the 1990s and well into the next century. It has also enhanced flight control concepts by greatly increasing the capability to access and control spacecraft. If phasing in of added capabilities can be accommodated, it will permit smoothing of resources and help the budgeting process.

Another important consideration of the systems engineer in the evaluation of support requirements is the impact these services will impose on the user. The goal is always to limit the interface restrictions imposed on the user program. Two of NASA's major operating systems have caused major constraints in their use. The Shuttle Program has imposed major safety and integration complications on deployed payloads and the TDRSS program has imposed scheduling and radio frequency interface constraints that have been restrictive to some users. Some of these constraints with both the Shuttle and the TDRSS were intrinsic to their operational concepts, but some were avoidable, had operability and utilization been more completely evaluated.

When developing systems such as the Shuttle and the TDRSS that represent a major departure in operating concepts and expansion of the operational envelope, the systems engineer needs to broaden analysis to the entire mission or spectrum of missions to better define and limit the major complications to system operations and utilization.

NASA's experience with both of these programs has clearly indicated much more effort is required to operationally understand the implications of their use. This experience should be understood and applied in the development of the Space Station, the Earth Observation System, and their associated support systems in consideration of their broad utilization objectives.

Requirements verification and control is generally practiced with all new developments, but control can be difficult to sustain throughout an extended development of an operational support system and its operational life. Unfortunately, the nature of flight programs is to evolve operational support requirements and occasionally to transfer capabilities planned for the flight system as requirements to the ground support systems. Careful monitoring and control of these requirements is essential, particularly in the development of software support systems. Requirement changes will constantly occur, however, and an efficient process to identify, approve and control requirements is vital. Clear and precise interface definition is necessary to enable this control. A detailed knowledge of the flight programs that intend to use the support system, as well as an understanding of other related support systems (operational support systems rarely provide the total functional support services), is required for effective requirements control by the systems engineer. Interface definition and control are essential to maintaining requirements control.

SYSTEM ARCHITECTURE AND SOFTWARE DESIGN

For those operational systems that contain standard computers and specialized software, which are a majority of the ground systems, a special subset of systems engineering must be performed to obtain the optimum hardware and software combination. The selection of the wrong hardware may result in software needs that are difficult and expensive to develop. Similarly, less expensive hardware solutions may be possible when the full range of software abilities is considered. (The designer must always bear in mind, however, the probable need for system expansion, which may make the selection of a more complex hardware element the prudent choice since software modifications are generally less costly than computer replacement.) This analysis of system architecture may involve the estimation of size, complexity and structure of the software needed for a series of mainframe computers.

Management and the systems engineer must realize the definition, design and implementation of major software packages require the same systems management disciplines and controls as do hardware components. Because software code can be easily erased or changed, it does not follow that changes should be considered any more lightly than they are for hardware. The flexibility associated with software is its greatest asset, but if not well managed and controlled, it becomes its greatest problem. Although software design has made astonishing progress over the years, software development remains a significant problem to most major systems. The inability of management to accurately predict software costs, delivery schedules and performance has consistently been a severe problem in the development of major operational systems.

LONG-RANGE REQUIREMENTS

An area often inadequately considered in the design of a support system is its capacity for future modification and upgrade as new technology becomes available and as requirements change over time. Many systems must continue to provide services while undergoing these modifications. Proper consideration for redundancy and capacity can greatly alleviate future expense and complications. Making assumptions regarding future support requirements can lead to a system design that reasonably accommodates alternative future growth requirements. Designs that fail to gracefully accommodate change are limited and will lead to a dead end.

While the Deep Space Network and the Goddard Space Flight Network have effectively accommodated change, the initial design of the TDRSS ground station failed to properly consider the long-term need to modernize and upgrade. This required extensive redesign and change at significant cost. A focus on the current needs may result in limited system utility, and pressures to implement the least cost system may constrain future expansion and ultimately, be the least cost effective.

The development of new features or major changes to operating systems is frequently implemented with new contractors. Generally, if NASA and the systems engineer did not specifically assure that the original contractor provided adequate hooks, the new contractor's implementation will be difficult and costly. The term "transition phase" is applied by NASA to the period when an online system is undergoing change while continuing to provide support services. This is a delicate and challenging problem to the systems engineer and critical in the selection of an appropriate design. It is important that transition be planned in conjunction with the design process and not after the design is established.

In considering long-range requirements for operational systems, the type of system, the importance of support, and accessibility are major factors. These factors were central to NASA's decision and ability to sustain the Deep Space Network (DSN) and the Goddard Space Tracking Network over their extended lifetimes while undergoing numerous modifications and changes. The continuous availability of these sites has been possible because of the redundancy within each ground station, a configuration of multiple sites (redundancy among the ground stations), and their accessibility. The recent major rebuilding of the 240-ft. DSN antenna reflectors prior to the Uranus Encounter was feasible because each antenna was sequentially modified, and alternate antenna systems were available at each DSN location to provide continuous tracking support. Redundancy within the system—provided because of the critical nature of tracking and communications support—and ground station accessibility have been critical to NASA's ability to continuously operate these networks while modernizing their capabilities.

When considering system changes, space-based operational support systems present a different challenge. Two major factors influence the consideration to change—accessibility and cost. Cost is directly related to the lack of direct access. Accessibility is difficult at best and impractical for most. The Hubble Space Telescope is accessible at great expense by using the Shuttle but the TDRSS satellites are presently inaccessible. The systems engineering of space-born support systems must consider the criticality of the service to be provided, the longevity of the service (providing adequate redundancy and projected service requirements), and the lack of ready access to the system. Satellites can of course be replaced by an upgraded satellite; systems that use multiple satellites at multiple locations, however, such as TDRSS, require identical satellite configurations to provide orbital coverage as an effective operational system. Spacecraft replacements are normally planned to sustain the system through its projected life with no ground interface and no service changes to the system.

When new services become necessary, they are expensive and require an extended period to implement. A space-based system that consists of several satellites, such as TDRSS, requires a change to the services of each satellite in orbit to provide an effective orbital service to the user. This is consistent with the practice of upgrading all ground station locations to the same service configuration; the accessibility makes the upgrade of space systems more costly and requires a much longer time.

NASA is now planning to modify the TDRSS with a higher data rate KA band service. The system and budget planning for this upgrade was begun in earnest in about 1985, and it is anticipated the satellite fleet will not be in orbit until early in the next century, a 15- to 20-year period. The TDRSS will have been operating for 20 years or more by that time. A similar projection will mean the replacement system, Advanced TDRSS, will likely be operating to the year 2020 and perhaps beyond. It is clear this system will be as challenging as the original, with new problems replacing those resolved with TDRSS. The transition of replacing the TDRSS systems presents a significant new challenge not faced with initiating the original service. Providing systems engineering for the Advanced TDRSS to remain viable 20 to 30 years in the future will tax any manager. Systems can no longer be replaced frequently or modified to meet individual program desires. Careful and complete system analysis and forward-thinking engineering are essential to the establishment of durable, effective support systems.

ASSURING OPERABILITY

To succeed in developing a support system that meets the goals of operability—ease of operation, failure resistance, maintainability, efficiency, expandability and accommodation to users—requires continuous effort and emphasis by the systems engineer. An oversight and regular review from the operator's viewpoint will contribute to success. Both the government and the contractors should provide an operational position within their program management structure that is responsible for maximizing the system's operability. Developments that continuously focus on the ultimate operation are consistently superior in performance and in total costs.

The need for NASA to be alert to systems engineering is more prevalent now than ever before in NASA's history. The implementation of new operating systems is planned throughout the 1990s to prepare the agency for managing the operations of complex, long duration and extremely high data rate programs. The quantity of data the agency will be processing and managing in the later part of the decade was unimaginable in the 1960s and the 1970s. This data will be generated by programs that will be launched in a period when NASA will already be operating and supporting a complex array of flight vehicles. New ground systems, with evolving capabilities and changing interfaces, will come into operation almost continuously throughout this period. The complex nature of interaction among these systems demands a visibility and overarching control that can only be accomplished through a systems engineering network. Management and coordination of the individual systems is required to assure total system functionality, interface definition, requirements control and the optimization of each system.

NASA has done an excellent job for the past 30 years in providing an operations infrastructure that has met the demands of exploring space. The next 30 years of space operations are equally exciting but represent a far greater challenge. The quality of the systems engineering of the operations support team is critical to both the success of the nation's civil space flight programs and to sustaining a viable operational role within NASA.

POLITICAL AND INSTITUTIONAL FACTORS AFFECTING SYSTEMS ENGINEERING

by John F. Yardley

Most systems engineering courses and textbooks discuss only the engineering aspects of the subject and are silent about the non-technical world's influence on the planned project. This approach, although entirely satisfactory for many engineering programs, including smaller NASA programs, leaves out a significant element affecting large NASA programs. Some traditionalists believe these nontechnical aspects should not even be considered in the systems engineering process. However, if we take the broad view that systems engineering should take into account all significant requirements in order to produce the proper end-product, then it should include consideration of those outside non-technical parties who can levy requirements on NASA programs. This paper identifies these elements, discusses their viewpoints and probable influence, and reviews some past case histories as illustrations of these problems. It also presents some suggestions for working with these non-technical groups, which ma; better achieve overall optimum systems engineering and integration (SE&I) solutions.

THE NON-TECHNICAL GROUPS

There are many outside parties that provide inputs to NASA program requirements.

The public at large can have a profound influence on whether large sums are appropriated for NASA's major programs. They respond to NASA triumphs and disasters and are sensitive to NASA's role in projecting the American image around the world. Their influence is exercised by letters to Congress and the White House, by public appearances (interviews and speeches, for example), and through public opinion polls regarding the space program. All of these methods influence both the executive and legislative branches of our government.

The President and his staff are very important to NASA's programs. They must make a positive decision to include money for specific NASA programs in the budget request before it is even considered by Congress. In these times of large government deficits, which makes starting new programs very difficult, NASA is pressured to cut back requirements and save money. This pressure even results in the stretch-out and cancellation of some ongoing projects. Sometimes in negotiations with the Office of Management and Budget, NASA is asked to choose between programs.

The Congress is one of the most significant groups that has a major impact on NASA's requirements. In addition to representing their constituents' opinions, members feel it is their duty to closely watch the details of NASA's large programs. In the last several decades, they have acquired the technical staff needed to exercise this detailed oversight. As a result, they are in a position to demand program requirement changes, and they have the appropriation muscle to back up their demands.

The Department of Defense (DoD) and other national security agencies often get involved in NASA's programs because they have agreed to participate in a joint development or because they plan to use the end-product. They are involved in monitoring NASA's projects from a national security viewpoint, and they sometimes require changes in NASA programs if they see potential security problems. DoD is always included as a major player in any high-level White House space study or committee.

Some NASA partisans feel that certain DoD offices take a biased view and try to reduce the NASA program so DoD can play a larger role in space study.

Other executive departments substantially involved in NASA program matters include the State Department, the Commerce Department, the Transportation Department, and the Office of Management and Budget.

Government agencies and national commissions that fact-find, study and advise the executive and legislative branches upon request include the General Accounting Office, the Office of Technology Assessment, the National Academy of Sciences, the National Academy of Engineering, the National Research Council, the National Commission on the Challenger Accident, the Advisory Committee on the Future of the U.S. Space Program, and a number of other ad hoc committees.

International cooperation agreements often involve political considerations, and the foreign parties usually desire a part of the job that interfaces with many of the mainstream elements. If these agreements are not structured with the interface problems in mind, they can have major effects on systems engineering.

Scientific specialist groups feel they could more wisely spend the money appropriated for the large NASA manned space programs on their own research or on unmanned scientific space programs. This group sometimes works through "associations" seeking to plead their case in the media.

Local communities near NASA centers often inject themselves into the process of dividing the program work between Centers. The actual division of work can have a substantial effect on the efficiency of the collective NASA effort and can make the systems engineering effort much more difficult than a distribution based on technical merits. The political realities usually result in a "technically non-optimum" work split.

EXAMPLES FROM THE PAST

History provides examples of political and institutional influences that illustrate how these factors affect NASA's programs. After the first Sputnik launch, the basic thrust to start the space agency, as well as to initiate the Mercury Program, came mostly from Congress, with lukewarm support from the Eisenhower administration. NASA's founding organizations, the National Advisory Committee for Aeronautics (NACA), was used as a technical staff; decisive actions were primarily political in nature.

During the sixties, the Kennedy Administration's decision to land astronauts on the Moon and return them safely was political; namely, to catch up with the Russians and get back U.S. world technological leadership. NASA provided a large part of the technical staff work, which consisted of preliminary analyses and estimated success probabilities.

In the case of the Space Shuttle start decision, interaction increased between systems engineering and the non-technical world. Richard Nixon had become President in early 1969, just a few months before the lunar landing. He requested the National Space Council to study and report on the options for the next phase of space flight and the long-term future. NASA was heavily involved in this year-long study. The report recommended that development of a Space Station and a fully reusable Space Shuttle be undertaken in parallel as the next step in manned space flight and as the precursor of later lunar colonies and manned Mars expeditions. At this point, a political decision was made to continue study of the Space Shuttle but to defer the Space Station. Work then proceeded on the Shuttle with Phase A contracts and then Phase B contracts. It soon became apparent that the Shuttle development cost was more than double the original preliminary estimates used in earlier decision making. Much interaction ensued between

NASA, the Office of Management and Budget, and Congress, with NASA trying to get the added funding commitment. When this was not forthcoming, the program management exhorted the projects to reduce cost without changing the basic concept.

After more work confirmed that the cost ceiling could not be achieved with the two-stage fully reusable Shuttle, it was finally decided by NASA management that the concept had to be changed in order to stay within funding limitations imposed by the Administration. Phase B contracts were extended, a major realignment of contractor teams was required, and the current Space Shuttle configuration (solid first stage, parallel burn) emerged. After the Apollo program and its blank check atmosphere, NASA was not used to this limited funding approach.

This process left much to be desired from many points of view. It delayed the program, caused a lot of wasted effort, and contractors formed teams and wasted a lot of their discretionary funds (estimated at $100 million). No one is to blame for this, since everyone was feeling their way in a new environment. A better process, however, would have been very worthwhile.

In contrast to the Shuttle, the Space Station did have strong support from President Reagan. This support was not for short-term political gain but rather because President Reagan believed it was in the best long-term interest of the country, despite the fact that most of the President's cabinet members and his close advisors were against starting the space station (Hans Mark's book).

The fragmented nature of the final Space Station hardware split between Centers resulted from an intense tug of war for appropriate shares of the program between the NASA Centers and their supporting political communities. Some NASA Centers felt that much of this struggle was for their very survival. Others in NASA felt this type of work distribution was necessary for broad Congressional support. While the final system is probably workable, it certainly is not considered optimum from a technical or efficiency viewpoint.

MINIMIZING DISRUPTION FROM POLITICAL AND INSTITUTIONAL SOURCES

We have identified many of the outside sources of SE&I requirements and have given some examples to illustrate how important these inputs can be. Although most of these examples involve major program changes, many smaller requirements are questioned and changed. Now we will discuss methods of dealing with these inputs efficiently, minimizing disruption and avoiding adversarial relationships with these outside organizations.

Good two-way communication between NASA and these outside groups is one of the major keys to negotiating proper agreements on these external requirements. In order to properly deal with these outside inputs, we need to know what new requirements they are considering before these requirements are placed on NASA as irreversible demands. If we wait until then, it is very probable that we will develop adversarial relationships with the requester who has "gone public" and will be embarrassed to lose the argument. This will make the requestor very difficult to deal with during subsequent negotiations.

This means NASA must be organized and managed in a manner that facilitates communication of both internal and external pertinent information.

Most of these outside inputs are discussed at lower levels during interface or coordination meetings as "what if's." They rarely first surface at the NASA decision level in the program office or the SE&I management. This means that the lower-level NASA people interfacing with outside organizations must be trained to recognize these potential inputs at the beginning, and the overall NASA organization must have good communications at all levels so these issues can get to the appropriate level early, a strategy can be developed, special analyses can be performed, and contacts to discuss the issues can be planned.

When preparing the material for discussion with the requester, NASA must be very careful to consider the requestor's point of view objectively and not just from the NASA parochial viewpoint of pure engineering ease, i.e., the "invented here" syndrome or the "bad for the Center" rationale. NASA must remember it is not the user or the owner but rather the implementor of someone else's requirements. When presenting the material, NASA must be careful to avoid patronizing the requester. If the requestor senses a patronizing attitude, the relationship rapidly becomes adversarial.

It is also important for NASA to advise and sell the appropriate outside groups on any requirement changes they feel are necessary before the action has been taken beyond the point of reasonable return. This is particularly true when NASA wants to relax requirements that were important to outside groups once the program was begun. Many examples exist where Congress finds out after the fact that the program can no longer meet the planned launch rate or some other fundamental requirement, and the original "NASA promise" must be broken. This has a very negative effect on rapport with Congress, the scientific community or any other major stakeholder. It is therefore important to level with these outside groups as quickly as possible after deciding to revise a basic requirement.

NASA must also develop harmonious relationships with the pertinent outside groups and individuals. This can be done, among other ways, using a network of committees or scheduled small meetings among selected individuals. The important thing is to plan for relationships and have the meetings regularly. These meetings should be used to bring the groups up to date, to permit them to ask questions and critique the activity, to smoke out impending requirements, changes or additions, and to develop rapport. While doing these things, it is very important for NASA individuals to come across as open, forthright, and on top of their jobs. If the outside participants sense ulterior motives that are not discussed, or evasiveness and bluffing, trust cannot develop. In fact, many of these groups currently have a "corporate memory," which includes perceptions of many NASA Center biases. These must be overcome by careful and fair negotiations, bending over backward to diffuse any biased reputation.

NASA Centers have tended to think of many of these non-technical meetings as NASA Headquarters' responsibility (and a big, time-wasting nuisance), believing the Center's only role should be the engineering and management of the program. For NASA to do the most efficient and effective job, this concept must be changed. Whereas NASA Headquarters should participate in many of these contacts, the Center people who best know the subject and have prepared the material should present it. This is also an excellent training mechanism. The younger Center people will rapidly develop a much broader view of the outside world from interacting with NASA. Working with the centers in this manner, Headquarters also facilitates better internal communications.

Interfacing with Congress presents some special problems, particularly when NASA is trying to sell them a new program. There are laws prohibiting government employees from lobbying, and the line between lobbying and briefing on the merits of a new program is somewhat blurred. NASA must use its legislative and legal offices to help the program people properly interpret the law. In all probability, NASA will not be able to communicate with Congress on critical subjects in the manner and with the frequency they desire.

An alternative to direct NASA communication with Congress is for NASA to work with its contractors and keep them informed. The contractors are not bound by any laws against lobbying and can communicate more freely with Congress. The contractors will contact the appropriate Representatives and their staffs with their own messages, in any case. It is not necessary for NASA to direct them to lobby (this being illegal), but NASA should inform them of its position so that if the contractors do contact Congress, they have the correct information.

On some past programs, all of the prime contractors informally worked together to keep Congress informed. One technique that has been popular with Congress is an "Information Notebook" on a given NASA program. This notebook is kept in the Congressional member's office for easy reference and is updated monthly, providing a useful monthly resource for informal discussions.

NATIONAL STRATEGIC PLANNING FOR SPACE

After the Apollo program and President Kennedy's clear mandate to land astronauts on the Moon and return in the sixties, the U.S. space program suffered from a lack of clear national goals and a strategic plan to achieve them. In the Apollo era, all of the diverse forces involved coalesced behind President Kennedy because they wanted to beat our superpower adversary, the U.S.S.R., in the technological war. Since that time, we have been unable to generate such a unifying environment. If this could be done, and a framework for future space activity could be agreed on in the form of a strategic plan, the problems of interfacing with the outside groups would be much easier.

As of this writing, the Bush administration has outlined a long-range plan for exploration that includes colonizing the Moon a and a manned exploration of Mars, which could form the framework for a good strategic plan. However, it must be accepted by these outside parties and backed with appropriations by Congress before any plan can realistically be made. During this period of a growing national deficit, tensions in the Middle East, and the bail-out of the savings and loan industry, such an ambitious plan will be difficult to accomplish.

SUMMARY AND CONCLUSIONS

External groups have a significant impact on NASA's programs. Ten groups affecting NASA are identified, and examples are given for some of the them. Methods of dealing with these external inputs are discussed, the most important being good and open two-way communications and an objective attitude on the part of the NASA participants. The importance of planning ahead, of developing rapport with these groups, and of effective use of NASA contractors is covered. The need for an overall strategic plan for the U.S. space program is stressed.

In order to obtain the broadest range of opinions on the political and institutional factors that affect systems engineering, the writer requested thoughts from a number of senior individuals who have been involved in the interfaces between NASA and the outside world.

In any subject as complex as this one, there are always some differences of opinion. The viewpoints expressed above are those of the writer and sometimes agree with the majority, and at other times do not. To provide the reader with another viewpoint, an additional paper by David Wensley is reproduced in its entirety in the appendix to this chapter. Mr. Wensley examines the subject through the eyes of a prime Space Station contractor executive.

The author concludes that NASA does not pay sufficient attention to the impact of political and institutional factors in conducting its business and is being hurt by this attitude. NASA should therefore focus on working with these outside groups, adjust NASA policies and organizations to facilitate interfacing with them, and train NASA personnel to conduct themselves appropriately in this environment.

POLITICAL AND INSTITUTIONAL FACTORS AFFECTING SYSTEMS ENGINEERING: AN INDUSTRY PERSPECTIVE

by David Wensley

The "nominal" or "idealized" systems engineering process must take into consideration the political and institutional factors that have become prevalent in the government funded and, to a certain extent, the privately funded civil space activity. Attempts to ignore these influences may result in delay and frustration of the systems engineering process.

NASA programs are currently growing larger in scope, longer in duration and fewer in number. The increasing number of participants includes NASA Centers, other U.S. agencies, international agencies and contractors. NASA programs are also characterized by higher public visibility, and are more costly and more politically sensitive.

In this environment, the Congressional committees that appropriate and authorize budgets will demand more justification for expenditures, more political return from the investments and more oversight of ongoing activities.

POLITICAL FACTORS

Space projects have always been an instrument of domestic politics and a tool of political influence in international relations. As the scope and importance of these projects increases, we can expect more political influence on the systems engineering process.

The political influence may take any of several forms:

- Geographical distribution of funds to gain political support.
- Creation of international partnerships.
- Insertion of technical requirements to satisfy strategic national goals.
- Increased Congressional and Administration involvement in the technical decision-making processes.
- Funding constraints used as a mechanism of technical and political control.

An effective project management and systems engineering process must deal constructively with these influences. They may affect program content, allocation of responsibilities, schedules, interface definitions, optimization and trade-off criteria, and technical decisions. They may even affect mission definition, and they most certainly will affect funding availability versus time. Effective management must provide for flexibility to react to these influences without undue penalties on performance, cost or schedule. A constructive and cooperative relationship between the legislators and program management can minimize the impact of these interactions on planned efforts.

Many examples of the influences noted above can be cited in the Space Station Freedom program, including:

- Legislated use of a Flight Telerobotic Servicer to advance U.S. robotic technology.
- Allocation of responsibilities to international partners.
- Political influence on the work distribution between NASA Centers.
- Increased complexity of interfaces and management processes resulting from distributed responsibilities.
- Funding constraints (fencing) in budget authorization bills.
- Oversight committees and hearings to critique technical progress and to influence resolution of technical issues.

The systems engineering process must stand the tests of external review and critique. The assumption that technical management and decision making is part of an immune internal process is, unfortunately, unrealistic. Techniques for effectively managing the external factors include:

- Open communication between project management and stakeholders to understand needs and develop trust.
- Realistic planning to support schedule and cost commitments.
- Disciplined control of requirements to avoid unwarranted cost and schedule growth.
- Effective use of risk management techniques to minimize iterations on design and testing.
- Cost-effectiveness and life-cycle cost analysis to substantiate trade decisions.
- Early emphasis on operations, maintenance and logistical support to avoid unpredicted support costs.
- Early constructive resolution of responsibility conflicts between NASA Centers and between NASA and international partners.

These features are characteristic of traditional management and represent the expectations of legislators and budget authorities. Deviations from these norms, especially if uncovered through Congressional or media probing, can be disruptive and potentially dangerous to the stability and continuity of a program. The systems engineering process can significantly reduce these risks by staying on track and by making summary data available to project managers to use in open dialogue with legislators.

Program changes are unavoidable, and systems engineering and project management must be equipped with the analytical tools to respond effectively to these changes. The ability to re-prioritize and reschedule activities rapidly and with reasonable accuracy is essential, especially in response to funding adjustments emanating from the annual budgetary process. More often than not, these events are unanticipated and result in traumatic and costly adjustments. A preplanned strategy for deferral of less critical elements, retaining the systems engineering effort to establish interface requirements and essential design definitions, can minimize such effects.

INSTITUTIONAL FACTORS

Numerous institutional factors will affect the systems engineering process, principally those inherent in NASA and the participating Centers. Examples include:

- Accepted standards, design criteria, and specifications.
- Design, management and operational preferences of the Center functional divisions.
- Availability and preference for use of Center test facilities.
- The organization and management structure adopted for the program.
- Traditional practices such as use of committees, panels, boards, documentation formats and integration processes.
- Use of support contractors to supplement NASA staff.
- NASA and Center policies and priorities that may influence, for example, technology selections, responsibility issues and requirements decisions.

The above considerations can have a major impact on systems engineering requirements derivations, trade studies, architecture and design selections, test plans and operational concepts. They will also affect the schedule and effort required to evolve the design baseline, to resolve integration issues and to establish interface agreements. The potential magnitude of these effects dictates early planning for their accommodation in the systems engineering process. It is virtually pointless to embark on a systems engineering process that ignores these considerations. The institutional characteristics have evolved over time and are the product of many successes and failures. It is unlikely that personnel assigned to new projects will adopt practices that violate tradition. Contractor personnel should be prepared to adapt to customer preferences, but customer (NASA) personnel should be prepared to consider new alternatives as part of a continuous improvement process.

THE SEARCH FOR IMPROVEMENT

Increased budget pressures and heightened concern for foreign competition create a demand for NASA to seek new methods of achieving quality and reducing costs. Industry is similarly under pressure in these areas and is rapidly adopting techniques such as Total Quality Management (TQM) principles. NASA is beginning to apply TQM criteria in new procurements and has started to look for TQM opportunities within its organizational structure. Conversion to these principles represents a major cultural change and, in many respects, is contrary to recent trends within NASA. TQM teachings emphasize reduction in top-down management direction, preferring increased delegation and empowerment of the lower tier personnel. Since the Challenger accident, the tendency within NASA has been to increase management and technical oversight. In the Space Station Freedom program, for example, many layers of management and technical oversight exist within the Level II and Level III organizations above the prime contractors and their subcontractor teams. Although contractors are generally committed to cost and schedule objectives, their progress is often controlled by the efficiency and speed of the NASA management and systems engineering processes and integration. If the involved participants agree that improvement is essential to create an environment of credibility and trust at the political level, recognition of these relationships can lead to constructive changes.

Measurement of performance is essential in the search for improvement. Both NASA and contractors must be measured as elements of a closed-loop process that affects the efficiency and quality of our space activities. The identification of improvement candidates should focus on the inanimate process, not on the organizations or people. This allows the people to conduct constructive problem identification and resolution without personal implications.

CONCLUSION

NASA stands at a crossroads. The opportunities for space exploration and the exploitation of space attributes and resources have never been better. Public acceptance of space projects and reliance on space technology as a means to resolve worldwide environmental and resource issues have never been higher. Yet NASA lacks credibility with the legislators of this country who are eager to voice criticism of NASA's planning and implementation of space projects. Their depth of penetration into NASA's technical activities is increasing. Not only is the continuity of NASA funding at risk, the scope of NASA's responsibilities is also threatened. Transfer of responsibilities to other agencies and even the creation of new agencies is topical conversation. Resolution of this dilemma requires more than a willingness to communicate and to negotiate differences; it requires a change in the NASA management culture that recognizes the degree of maturity of the space industry. The mystery of discovery and the complexity of space technology is no longer an adequate defense for cost or schedule overruns. Critics demand performance that meets expectations. NASA has the opportunity to lead the family of federal agencies in demonstrating fiscal responsibility combined with technical achievements. Systems engineering will be a major contributor to this success by providing the guidance for timely decisions leading to effective project management.

OPTIMIZATION IN THE SYSTEMS ENGINEERING PROCESS

Loren A. Lemmerman

The essential elements of the design process consist of the mission definition phase that provides the system requirements, the conceptual design, the preliminary design and finally the detailed design (Figure 1). Mission definition is performed largely by operations analysts in conjunction with the customer. The result of their study is handed off to the systems engineers for documentation as the systems requirements. The document that provides these requirements is the basis for the further design work of the design engineers at the Lockheed-Georgia Company.

The design phase actually begins with conceptual design, which is generally conducted by a small group of engineers using multidisciplinary design programs. Because of the complexity of the design problem, the analyses are relatively simple and generally dependent on parametric analyses of the configuration. The result of this phase is a baseline configuration from which preliminary design may be initiated.

Preliminary design is far more complicated, both because the analysis techniques are more complex, and also because these techniques require specialized knowledge. The objective of this step is to refine the design estimates made during conceptual design and to add additional detail to the description of the configuration. At the conclusion of this phase, the aircraft is defined well enough so that a company can comfortably bid the cost of producing it.

Detail design is largely mechanical in nature, and normally occurs after receipt of an order for production. This is not an area of concentration in this presentation, however.

To provide a basis for amplification of the conceptual design process, look at Figure 2. The function of the conceptual design process is to conduct a multidisciplinary analysis of an aircraft to produce values of parameters that describe an aircraft. These parameters are top level descriptions that leave most of the actual configuration details undefined.

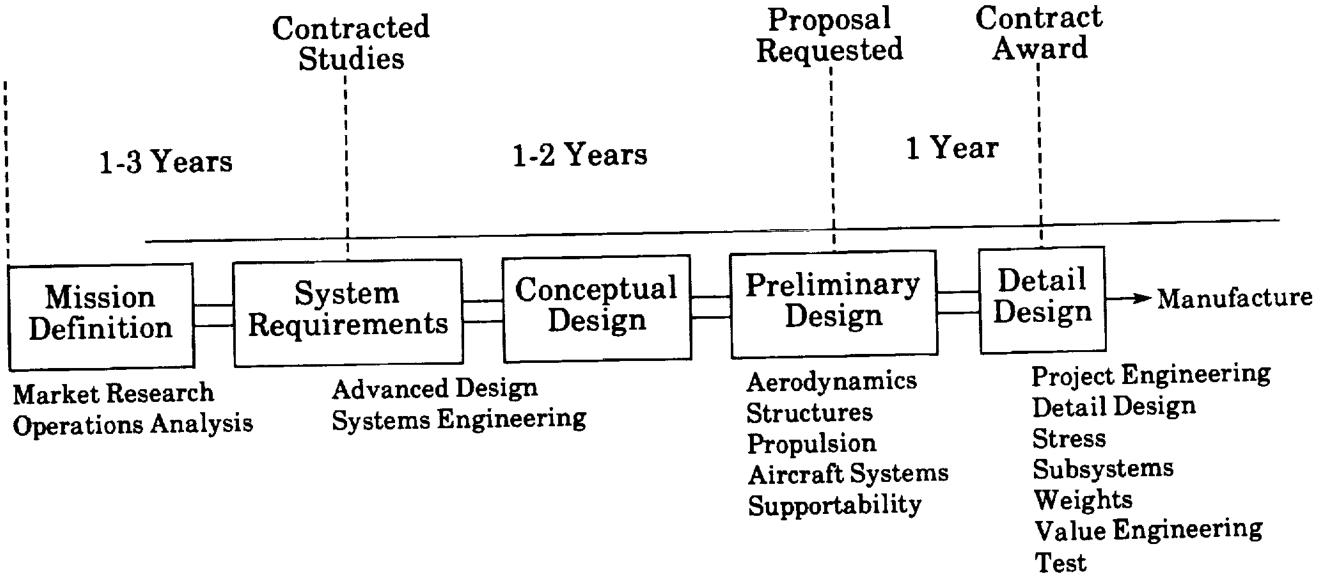


Figure 1 Essential Elements of the Design Process

However, implicit in this process is the trading of factors that relate to the performance of the configuration. The trades I mean are typified by the thinness of a wing desired by an aerodynamicist versus the thickness of a wing as desired by a structural analyst.

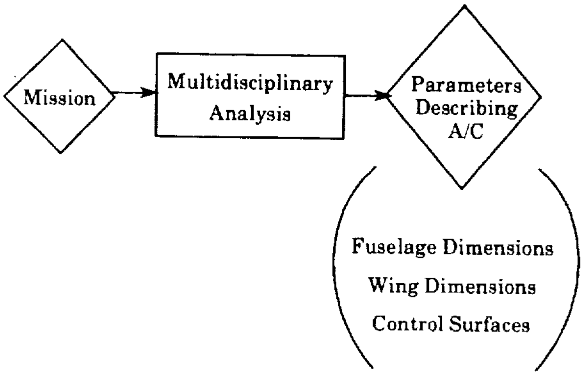


Figure 2 Conceptual Design

Typical parameters defined at this stage are fuselage length and width, wing area, sweep, aspect ratio and, to a limited extent, control surface.

In former times, conceptual design was manually directed and highly iterative. The process consisted of guessing an initial configuration, analyzing that configuration, and then systematically varying each of several design parameters to examine a design space within which manual optimization could take place. Normally the number of parameters examined did not exceed four, because of the human limitations in absorbing more variations than that. There were several disadvantages to the former approach. This process was time consuming, fallible and tedious. It was time consuming because the answer depended on many executions of a computer code. It was fallible because the choice of the parameter variation to be examined was entirely at the discretion of the designer. Thus, the quality of the answers was directly dependent on the skill of that designer. In addition, no one could be sure that a large enough design space had been investigated to ensure that a true optimum had been found. This old procedure was also tedious. All data had to be manipulated manually. Although this did provide useful insight to the designer, the cost was a further delay. Dozens of computer runs had to be scanned, the results judged for correctness, and the results plotted on carpet plots. Many hours of talented labor were consumed performing menial tasks.

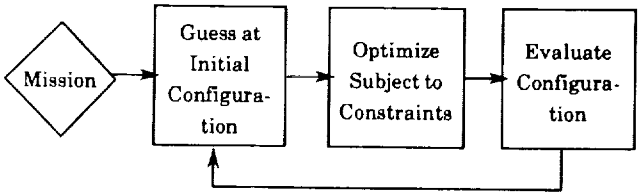

Mission
Guess at Initial Configuration
Optimize Subject to Constraints
Evaluate Configuration

Figure 3 Preliminary Design

The former process was basically eliminated at Lockheed-Georgia several years ago, in favor of the approach shown here, based entirely on numerical optimization. The new process is described schematically here (Figure 3). The former process was usually completed in one day. Many of the manual actions have been eliminated. Now, a given study may consume as much time as formerly, but a much larger range of design variables has been included.

PRELIMINARY DESIGN PROCESS (PARTIAL)

The next step in the design process is preliminary design. This is the process, partially illustrated in Figure 4, by which the conceptual design baseline is analyzed in greater depth to confirm the design or provide foundation for changing the design. This process is typified by the more or less simultaneous execution of many detailed design codes in several disciplines. Obviously, the communication during the process is difficult, and the designs proposed by each discipline are frequently inconsistent. Iterative loops, while very common, cannot be represented because

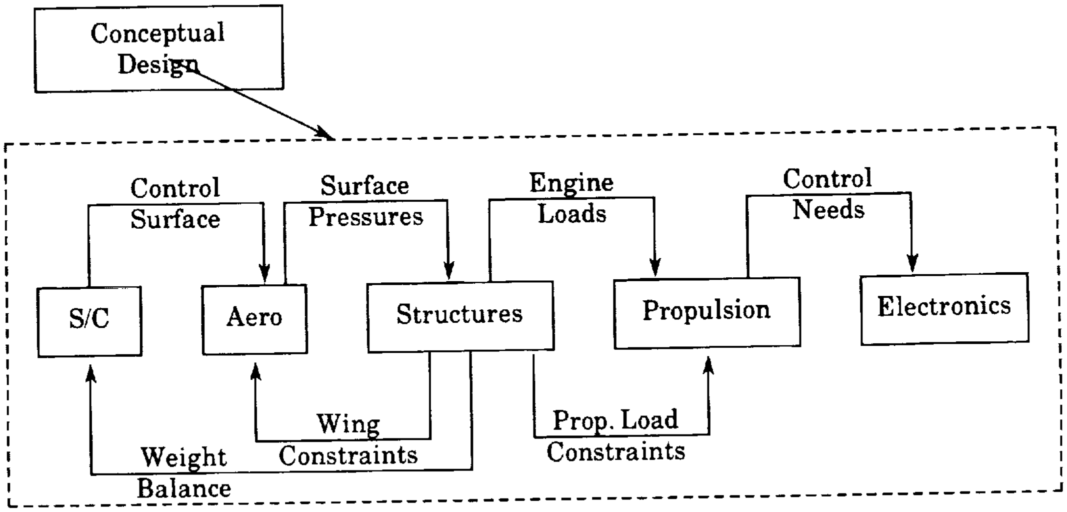


Figure 4 Preliminary Design Process

of the indeterminate sequence of such iteration.

As an example of the type of analysis conducted in this phase, consider aerodynamics for a moment. The codes frequently applied in this phase consist of full potential subsonic or transonic codes for configuration analysis, full potential codes for direct design, and Navier-Stokes codes for highly complex viscous flow analyses. As a result of the aerodynamic analysis done during this phase of design, the wing external contours are fully defined and more reliable estimates of the vehicle performance are available. Similar refinements and definition are added by each of the participating disciplines.

The deficiencies of the current approach are immediately obvious. First and foremost, the result is a suboptimal configuration. Even though optimization may be used within isolated analyses, the difficulty of communication in real-time and the lack of available tradeoff criteria mean that no global, rigorous optimization occurs.

I have already alluded to the use of optimization on individual analyses in this phase. Here are some examples of such optimizations. The aerodynamics discipline has been very active in developing optimization techniques for the design of wings in transonic flow, largely based on FLO codes. These methods provide a wing shape, starting with a specification of a desirable pressure distribution. Using such methods, the wing contour and twist distribution may be calculated directly.

Subsonic optimization techniques have generally been limited to the design of high lift systems. In this case, the optimal location of a slotted trailing edge flap can be found by optimizing on the axial force for the system and by using paneling methods for calculating the flap system pressure distribution.

Structural optimization has been done for minimizing structural weight, given loading conditions. In this case, the structure is modeled using finite element techniques, with element geometries such as thicknesses or cross sectional areas taken as design variables. Another example of structural optimization is in the design of composite panels. The objective is to determine the ply orientation to respond to specific loading conditions.

If I were to summarize the preliminary design optimization work currently being done at Lockheed-Georgia, I would have to say that its use is relatively new, that it has been very well accepted, and that its use is certainly increasing. But this may eventually become a severe problem for us, since the optimization is being applied to subprocesses within design. Worse yet, it is being applied

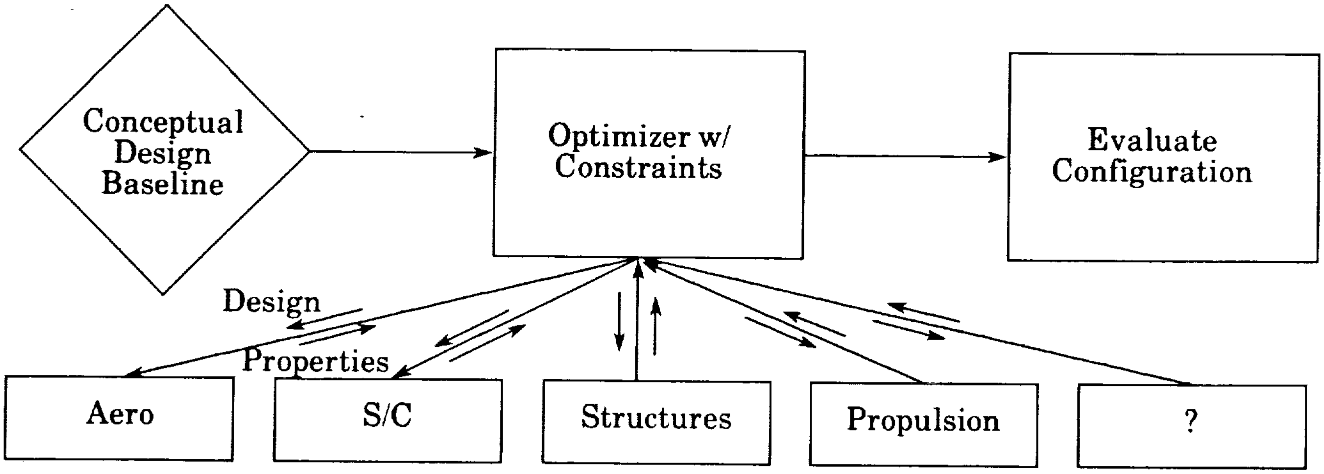


Figure 5 Proposed Preliminary Design Process

to old design philosophies. The result has to be suboptimal designs.

The preliminary design process is clearly another candidate for improvement by optimization. The technical challenge of this problem is much greater that that of the conceptual design process, but the potential payoff is also much larger. The challenge comes, in part, from the large number of individuals and computer programs normally invoked at this design state, and the current dearth of technology available to solve the very different problems thus posed.

One possible way to apply optimization in the preliminary design process is shown here. The fundamental idea is that candidate design parameters flow downward to the individual analysis modules and the result of the analysis flows back up to the optimizer.

Obviously, such a system is far from reality. The technical challenges outweigh those of optimization itself. The analysis methods normally used in preliminary design are state-of-the-art methods that are time consuming, user-sensitive and modeling sensitive. Because of this, not only will new optimization techniques be needed, but so will entirely new operational procedures. For example, optimization now is executed mostly as a black box program. The analysis points provided by support codes are considered to be correct and not subject to code sensitivities. In the preliminary design process illustrated here, the former approach clearly will not work. The new process must include a method for disciplinary engineers to examine the analysis code results as they are being generated to ensure that the optimized results are valid. When such an optimization method is available, however, I submit that the problem is far from finished. This is so because people inevitably are the designers, and the design techniques, whether through optimization or not, must take the human element into consideration.

SYSTEMS ENGINEERING - A DEFINITION

To expand on this theme, let me begin be giving you my orientation. I am in the Systems Engineering Department at Lockheed-Georgia. This gives a reasonable definition of what Systems Engineering means to us: a discipline that coordinates the engineering activities within large organizations to help produce a superior, cost-effective, timely product. By its very definition, it is a process of dealing with people in a large design operation. As such, our interest is not in the internal working of design codes, but rather in how individuals use given design codes to produce designs, and then how those individuals transmit their information to other designers in the organization.

Let me present the four main tasks of the Systems Engineering operation. They involve the management of trade studies, requirements, interfaces and technical risk. Another way to express these four tasks is Communication, Communication, Communication, Communication.

Decisions *are* the design process. By its very nature, design requires definition of some configuration from an infinity of possibilities. The best design is some compromise of many and widely varying constraints. Many times the choices to be made are aesthetic, or subjective, or not amenable to computer analysis. In these situations, and sometimes even in well-defined engineering choices, trade studies must be performed that are outside the domain of the optimization process.

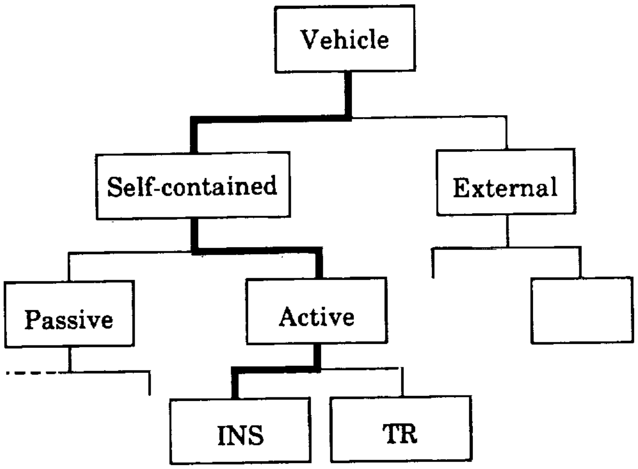


Figure 6 Hierarchy of Decisions to Select a Navigation System for an Airplane

The illustration above (Figure 6) is a simple representation of the decisions that might be made to select a navigation system for an airplane. These choices are displayed as a hierarchy, beginning with the top level vehicle considerations, and then working downward to finer levels of detail. Systems Engineering is responsible for generating such a trade tree to illustrate the decisions to be made, defining the design groups to be involved, coordinating the studies needed, and documenting the result.

Some of the decisions illustrated in this trade tree are supported by optimized methods. For example, the vehicle may be initially sized with optimization, and components may also be designed with optimized methods. Nonetheless, when design decisions are to be made, there is a high likelihood that not all the decisions will have been supported through optimization. The point is, optimization methods are embedded in the total design process, and this must be taken into account in the development of these optimization methods.

This last feature is what I am trying to illustrate in Figure 7. Some decisions of the design process will be made within the optimization process. Some will not. But those that do not must have information available from the optimization to assist the manual decision-making process. This is true whether the outside decision is being made concurrently with the optimization or whether it lags the optimization by days, weeks or months.

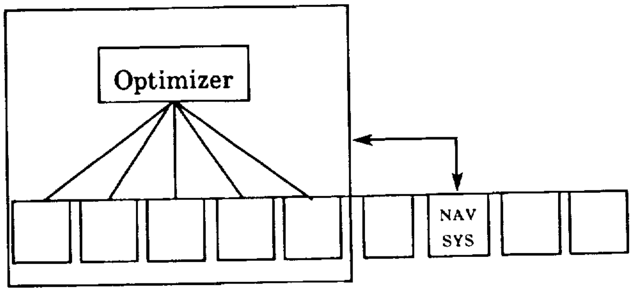


Figure 7 Trade Studies with Optimization

The implication is that information more comprehensive than just the final optimized configuration must be provided and stored. Possible information needs include sensitivities around the optimal point and the optimization history. In addition, it will be necessary to provide a way to interrupt the optimization process as it is occurring to input new information to the optimization process and to influence, on the fly, the outcome.

REQUIREMENTS FLOWDOWN

Let me provide one more example, that of requirements flowdown. This is another example of the communication involved in the design process. In this case, the objective is to communicate to each individual designer the importance of design in meeting the top level performance requirements. This is done by analyzing the top level system requirements and assigning or allocating these top level requirements to the next lower level to determine the drivers in the system. This process is repeated to successively lower levels until the final objective is accomplished. That is, the question "What is each individual's contribution to the total system performance?" is answered at the lowest logical level.

A specific performance might be maintenance manhours per flight hour, or it might be minimum range requirements. Whatever the requirement, this process allocates it to the lowest level of the configuration, maintains the traceability to the top level requirement and assures that the total system requirement will be met.

The question is, "What is a proper allocation?" If a top level requirement is rippled to the lowest level, which functional area should contribute what proportion to the final performance? If we rely on a optimization process that merely gives a final answer, we are blind. This is another case of not all functions being included in the optimization process. For these "outside" functions, we have no sensitivity information upon which to base realistic allocations. The actual situation might be as illustrated here, where the cost of attaining a given level of performance varies greatly from one discipline to another. I have used cost as the measure, but I could have used any measure of merit. For the illustration I have given, the

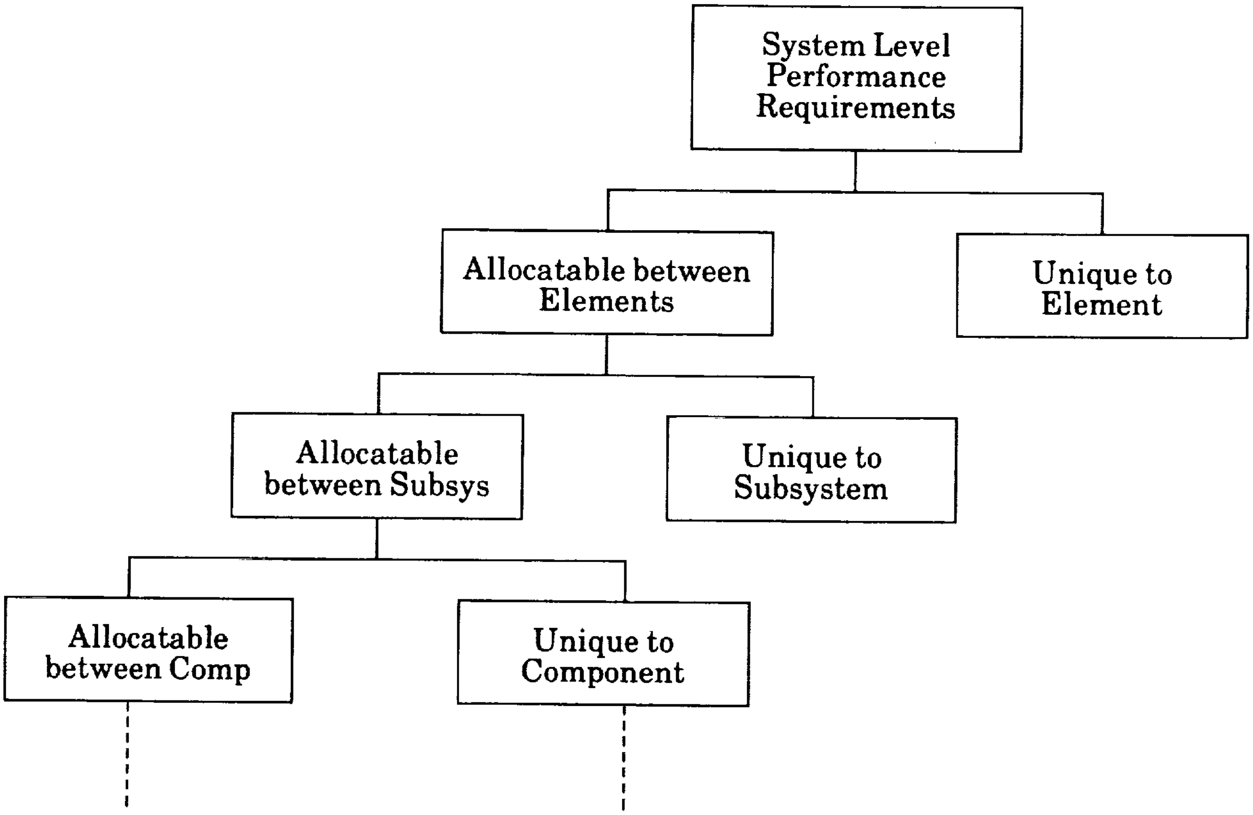

System Level Performance Requirements
Allocatable between Elements
Unique to Element
Allocatable between Subsys
Unique to Subsystem
Allocatable between Comp
Unique to Component

Figure 8 Requirements Flowdown

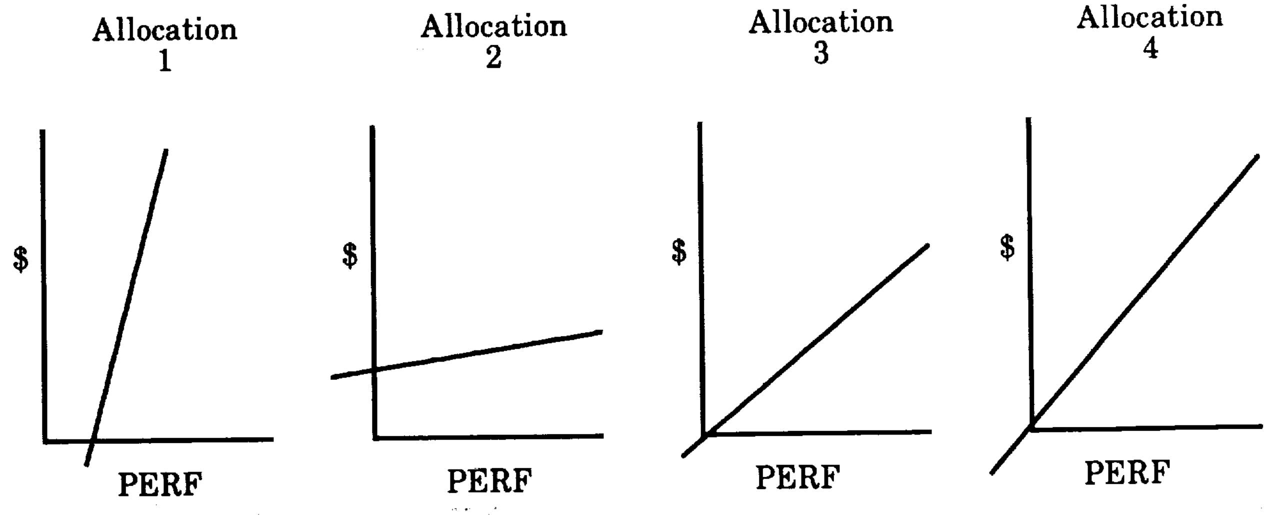


Figure 9 Optimize Allocations

optimal allocation of the requirement is that which simultaneously attains the top level system performance and minimizes the cost. In the future, our optimization processes must provide visibility for such data.

I have attempted to illustrate that optimization has a role in our design process, both today and in the future. The benefits are well known already, but I believe that we are only seeing the proverbial tip of the iceberg.

Optimization must, however, continue to be sold and this selling is best done by consistent good performance. For this good performance to occur, the future approaches must be clearly thought out so that the optimization methods solve the problems that actually occur during design. The visibility of the design process must be maintained as further developments are proposed. Careful attention must be given to the management of data in the optimization process, both for technical reasons and for administrative purposes. Finally, to satisfy program needs, provisions must be included to give data to support program decisions, and to communicate with design processes outside of the optimization process.

If we fail to adequately consider all of these needs, the future acceptance of optimization will be impeded. We simply cannot allow that to happen. Optimization is too important.

THE INITIAL FLIGHT ANOMALIES OF SKYLAB 1

By the NASA Investigation Board

At approximately 63 seconds into the flight of Skylab 1 on May 14, 1973, an anomaly occurred which resulted in the complete loss of the meteoroid shield around the orbital workshop. This was followed by the loss of one of the two solar array systems on the workshop and a failure of the interstage adapter to separate from the S-II stage of the Saturn V launch vehicle. The investigation reported herein identified the most probable cause of this flight anomaly to be the breakup and loss of the meteoroid shield due to aerodynamic loads that were not accounted for in its design. The breakup of the meteoroid shield, in turn, broke the tie downs that secured one of the solar array systems to the workshop. Complete loss of this solar array system occurred at 593 seconds when the exhaust plume of the S-II stage retro-rockets impacted the partially deployed solar array system. Falling debris from the meteoroid shield also damaged the S-II interstage adapter ordnance system in such a manner as to preclude separation.

Of several possible failure modes of the meteoroid shield that were identified, the most probable in this particular flight was internal pressurization of its auxiliary tunnel which acted to force the forward end of the meteoroid shield away from the shell of the workshop and into the supersonic air stream. The pressurization of the auxiliary tunnel was due to the existence of several openings in the aft region of the tunnel. Another possible failure mode was the separation of the leading edge of the meteoroid shield from the shell of the workshop (particularly in the region of the folded ordnance panel) of sufficient extent to admit ram air pressures under the shield.

The venting analysis for the auxiliary tunnel was predicated on a completely sealed aft end; the openings in the tunnel thus resulted from a failure of communications among aerodynamics, structural design, and manufacturing personnel. The failure to recognize the design deficiencies of the meteoroid shield through six years of analysis, design and test was due, in part, to a presumption that the shield would be "tight to the tank" and "structurally integral with the S-IVB tank" as set forth in the design criteria. In practice, the meteoroid shield was a large, flexible, limp system that proved difficult to rig to the tank and to obtain the close fit that was presumed by the design. These design deficiencies of the meteoroid shield, as well as the failure to communicate within the project the critical nature of its proper venting, must therefore be attributed to an absence of sound engineering judgment and alert engineering leadership concerning this particular system over a considerable period of time.

The overall management system used for Skylab was essentially the the same as that developed in the Apollo program. This system was fully operational for Skylab; no conflicts or inconsistencies were found in the records of the management reviews. Nonetheless, the significance of the aerodynamic loads on the meteoroid shield during launch were not revealed by the extensive review process. Possibly contributing to this oversight was the basic view of the meteoroid shield as a piece of structure, rather than as a complex system involving several different technical disciplines. Complex, multidisciplinary systems such as the meteoroid shield should have a designated project engineer who is responsible for all aspects of analysis, design, fabrication, test and assembly.

The Board found no evidence that the design deficiencies of the meteoroid shield were the result of, or were masked by, the content and processes of the management systems that were used for Skylab. On the contrary, the rigor, detail, and thoroughness of the systems are doubtless necessary for a program of this magnitude. At the same time, as a cautionary note for the future, it is emphasized that management must always be alert to the potential hazards of its systems and take care that an attention to rigor, detail and thoroughness does not inject an undue emphasis on formalism, documentation, and visibility in detail. Such an emphasis can submerge the concerned individual and depress the role of the intuitive engineer or analyst. It will always be of importance to achieve a cross-fertilization and broadened experience of engineers in analysis, design, test or operations. Positive steps must always be taken to assure that engineers become familiar with actual hardware, develop an intuitive understanding of computer-developed results, and make productive use of flight data in this learning process. The experienced chief engineer, who can spend most of the time in a subtle integration of all elements of the system under purview, free of administrative and managerial duties, can also be a major asset to an engineering organization.

THE SKYLAB PROGRAM

Skylab missions have several distinct goals: to conduct Earth resources observations, advance scientific knowledge of the sun and stars, study the effects of weightlessness on living organisms, particularly human, and study and understand methods for the processing of materials in the absence of gravity. The Skylab mission utilizes the astronaut as an engineer and as a research scientist, and provides an opportunity for assessing potential human capabilities for future space missions.

Skylab uses the knowledge, experience and technical systems developed during the Apollo program along with specialized equipment necessary to meet the program objectives.

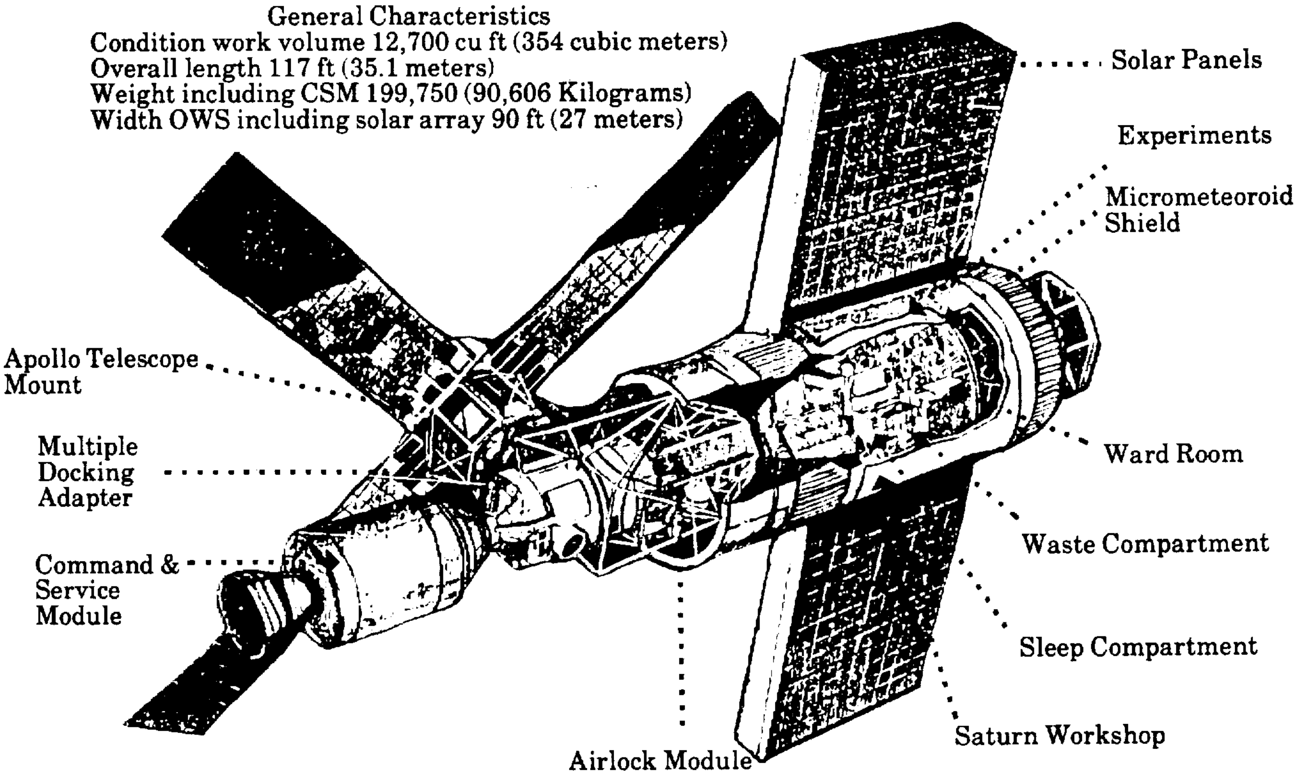
Figure 1 shows the Skylab in orbit. Its largest element is the orbital workshop, a cylindrical container 48 feet long and 22 feet in diameter weighing some 78,000 pounds. The basic structure of the orbital workshop is the upper stage, or S-IVB stage, of the S-IB and S-V rockets which served as the Apollo program launch vehicle. The orbital workshop has no engines, except attitude control thrusters, and has been modified internally to provide a large orbiting space laboratory and living quarters for the crew. The Skylab 1 (SL-1) space vehicle included a payload consisting of four major units—orbital workshop, airlock module, multiple docking adapter, Apollo telescope mount—and a two-stage Saturn-V (S-IC and S-II) launch vehicle as depicted in Figure 2. To provide meteoroid protection and thermal control, an external meteoroid shield was added to cover the orbital workshop habitable volume. A solar array system (SAS) was attached to the orbital workshop to provide electrical power.

The original concept called for a "wet workshop." In this concept, a specially constructed S-IVB stage was to be launched "wet" as a propulsive stage on the S-IB launch system filled with propellants. The empty hydrogen tank would then be purged and filled with a life-supporting atmosphere. A major redirection of Skylab was made on July 22, 1969, six days after the Apollo 11 lunar landing. As a result of the successful lunar landing, S-V launch vehicles became available to the Skylab program. Consequently, it became feasible to completely equip the S-IVB on the ground for immediate occupancy and use by a crew after it was in orbit. Thus it would not carry fuel and earned the name of "dry workshop."

The nominal Skylab mission called for the launch of the unmanned S-V vehicle and workshop payload SL-1 into a near-circular (235 nautical miles) orbit inclined 50 degrees to the equator. About 24 hours after the first launch, the manned Skylab 2 (SL-2) launch would take place using a command service module payload atop the S-IB vehicle. After the command service module rendezvous and docking with the orbiting cluster, the crew enters and activates the workshop; Skylab is then ready for its first operational period of 28 days. At the end of this period, the crew returns to Earth with the command service module, and the Skylab continues in an unmanned quiescent mode for some 60 days. The second three-person crew is launched with a second S-IB, this time for a second 56-day period in orbit after which they will return to Earth. The total Skylab mission activities cover a period of roughly eight months, with about 140 days of manned operation.

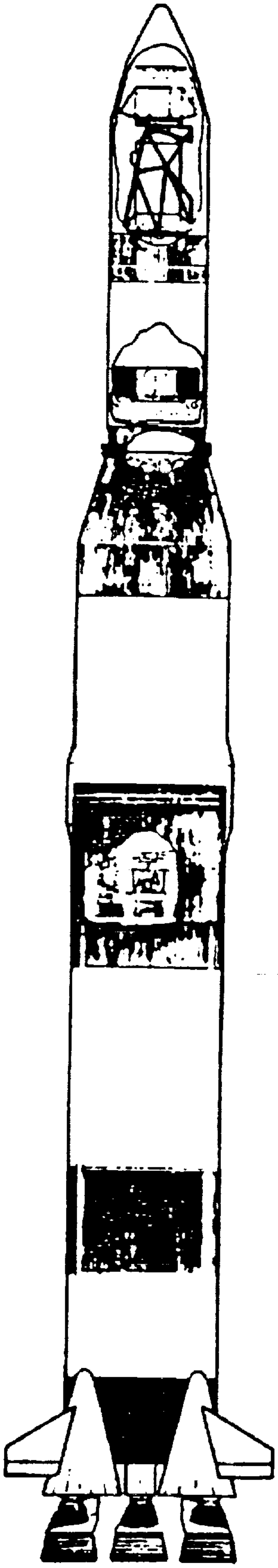
THE FLIGHT OF SKYLAB 1

Skylab 1 was launched at 1730:00 (range time, R=0) on May 14, 1973, from Complex 39 A, Kennedy Space Center. At this time, the Cape Kennedy launch area was experiencing cloudy conditions with warm temperatures and gentle surface winds. Total sky cover consisted of scattered cumulus at 2,400 feet, scattered stratocumulus at 5,000 feet, broken altocumulus at 12,000 feet, and cirrus at 23,000 feet. During ascent, the vehicle passed through the cloud layers but no lightning was observed in the area. Upper area wind conditions were being compared to



General Characteristics
Condition work volume 12,700 cu ft (354 cubic meters)
Overall length 117 ft (35.1 meters)
Weight including CSM 199,750 (90,606 Kilograms)
Width OWS including solar array 90 ft (27 meters)

Figure 1 Skylab Cluster



PS Payload Shroud
Diameter 6.6 meters (21.7 feet)
Length 16.8 meters
Weight 11,794 kilograms (26,000 lbs.)

ATM Apollo Telescope Mount
Wi Width 3.3 meters
Length 4.4 meters
Weight 11.181 kilograms (24,650 lbs.)

MDA Multiple Docking Adapter
Diameter 3 meters (1o feet)
Length 5.2 meters (17.3 feet)
Weight 6,260 kilograms (13,800 lbs.)

AM Airlock Module
Diameter STS 3 meters (10 feet)
Diameter FAS 6.6 meters (21.7feet)
Length 5.3 meters (17.5 feet)
Weight 22,226 kilograms (49,000 lbs.)

IU Instrument Unit
Diameter 6.6 meters (21.7 feet)
Length 0.9 meter (3 feet)
Weight 2,064 kilograms (4,550 lbs.)

OWS Orbital Workshop
Diameter 6.6 meters (21.7 feet)
Length 14.6 meters (48.5 feet)
Weight 35,380 kilograms (78,000 lbs.)

S-II Second Stage
Diameter 10 meters (33 feet)
Length 24.8 meters (81.5 feet)
Weight 488,074 kilograms (1,076,000 lbs.) fueled
35,403 kilograms (78,050 lbs.) dry
Engines J-2 (5)
Propellants: Liquid Oxygen 333,837 liters (88,200 gallons)
Liquid Hydrogen 1,030,655 liters (272,300 gallons)
Thrust 5,150,000 Newtons (1,150,000 lbs.)
Interstage Approx. 5,171 kilograms (11,400) lbs.)

S-IC First Stage
Diameter 10 meters (33 feet)
Length 42 meters (138 feet)
Weight 2,245,320 kilograms (4,950,000 lbs.) fueled
130,410 kilograms (287,500 lbs.) dry
Engines F-1 (5)
Propellants: Liquid Oxygen 1,318,315 liters (348,300 gallons)
RP-1 (Kerosene) 814,910 liters (215,300 gallons)
Thrust 31,356,856 Newtons (7,723,726 lbs.)

Figure 2 SL-1 vehicle

most other Saturn-V flights. The flight environment was quite favorable.

The automatic countdown proceeded normally with Guidance Reference Release occurring at R-17.0 seconds and orbit insertion occurring at R+599.0 seconds. The orbital workshop solar array deployment was commanded on time; however, real-time data indicated that the system did not deploy fully.

The solar array system (SAS) on the orbital workshop consists of two large beams enclosing three major sections of solar cell assemblies within each. During ascent, the sections are folded like an accordion inside the beams which in turn are stowed against the workshop. The meteoroid shield is a lightweight structure wrapped around the converted S-IVB stage orbital workshop and is exposed to the flight environment. The two hinged solar array system wings are secured to the orbital workshop by tie downs above and below the meteoroid shield. Seals attached to the solar array system perimeter actually press against the shield to form an airtight cavity prior to launch. Once in orbit, the solar array system beams are first deployed out 90 degrees. The meteoroid shield is deployed later to a distance of about five inches from the orbital workshop wall. After the ordnance release is fired, meteoroid shield deployment is effected by torsion rods and swing links spaced around the structure fore and aft. The rods are torqued prior to launch and simply unwind in orbit to move the meteoroid shield away from the tank, Detection of pertinent conditions associated with the meteoroid shield and solar array system is afforded by measuring various parameters by telemetered instrumentation.

When the orbital workshop solar array system was commanded to deploy, telemetered data indicated that events did not occur as planned. The flight data was analyzed by flight operations personnel to reveal the possible source of the problem. At about R+60 seconds, the S-II telemetry reflected power increased slightly. At about 63 seconds, numerous measurements indicated the apparent early deployment and loss of the meteoroid shield. At this time, the vehicle was at about 28,600 feet altitude and at a velocity of about Mach 1.

At this time, vehicle dynamic measurements such as vibration, acceleration, attitude error, and acoustics indicated strong disturbances. Measurements which are normally relatively static at this time, such as torsion rod strain gauges, tension strap breakwires, temperatures, and solar array system position indicators, indicated a loss of the meteoroid shield and unlatch of the SAS-2 wing. Further preliminary evaluation revealed abnormal vehicle accelerations, vibrations, and solar array system temperature and voltage anomalies at about R+593 seconds. Temperature data loss and sudden voltage drops indicated that the SAS-2 wing was separated from the orbital workshop at this time. Other data later in the flight indicated the SAS-1 wing did not fully deploy when commanded to do so. Although not apparently associated with the 63-second and 593-second anomalies, the S-II stage range safety receiver signal strengths showed several drops throughout the flight beginning at about R+260 seconds.

63-SECOND ANOMALY: LOSS OF METEROID SHIELD

The Investigation Board evaluated the telemetry data in order to explain the various anomalies that occurred on Skylab 1. The first anomalous indication was an increase in S-II telemetry reflected power from a steady 1.5 W beginning at R+59.80 seconds. At this time the telemetry forward power remained steady at 58.13 W. By 61.04 seconds, the reflected power had reached 1.75 W, and by 80.38 seconds, the reflected power had stabilized at about 2.0 W. This abnormal increase in power might be indicative of a vehicle physical configuration change which altered the antenna ground plane characteristic.

Shortly after the telemetry reflected power increase, the meteoroid shield torsion rod 7 forward (measurement G7036) indicated a slight change toward the deployed condition. This occurred at R+60.12 seconds, and at 61.78 seconds the vehicle roll rate decreased slightly from a normal value of 1.1 degrees per second clockwise looking forward. The next torsion rod 7 forward sample at about 62.52 seconds revealed a further relaxation. The increase in telemetry reflected power and the movement of torsion rod 7 forward tend to indicate meteoroid shield lifting between positions I and II.

Between R+62.75 and 63.31 seconds, several vehicle dynamic measurements indicated a significant disturbance. A sensor on the orbital workshop film vault showed an abnormal vibration at 62.75 seconds followed by disturbances sensed by X and Y accelerometer pickups in the instrument unit, the pitch, yaw, and longitudinal accelerometers, and the pitch, yaw, and roll rate gyros. At 62.78 seconds, the roll rate gyro sensed a sudden clockwise roll rate resulting in a peak amplitude of 3.0 degrees per second clockwise 62.94 seconds. A sensor at the instrument unit upper mounting showed a maximum peak-to-peak shock of 17.2 g's at 63.17 seconds. In addition, the S-II engine actuators experienced pressure fluctuations caused by vehicle movement against the inertia of the non-thrusting engine nozzles.

The data indicate that the most probable sequence of meteoroid shield failure was initial structural failure of the meteoroid shield between the SAS-2 wing and the main tunnel (between positions I and II). The initial failure propagation from this area appears likely since the wardroom window thermocouple indication (C7013) remained normal at 62.94 seconds after SAS-2 indicated unlatched at 62.90 seconds and after the K7010 and K7011 tension strap measurements failed.

593-SECOND ANOMALY

As a consequence of the meteoroid shield failure at approximately 63 seconds, the SAS-2 wing was unlatched and partially deployed as evidenced by minor variations in the main solar array system electrical voltages and SAS-2 temperatures. Full deployment was prevented due to the aerodynamic forces and accelerations during the remainder of powered flight.

At the completion of the S-II phase of flight, the four 35,000-pound thrust retro-rockets fired for approximately two seconds commencing at R+591.10 seconds followed by spacecraft separation at 591.2 seconds. The effect of retro-rocket plume impingement was observed almost immediately on the SAS-2 temperature and on vehicle body rates.

At 593.4 seconds the wing imparted momentum to the vehicle, probably by hitting and breaking the 90 degree fully deployed stops, and at 593.9 imparted a final kick as it tore completely free at the hinge link. In-orbit photographs show clearly the hinge separation plane and the various wires which were torn loose at the interface.

INTERSTAGE SECOND PLANE SEPARATION ANOMALY

Post-flight analysis revealed unexpectedly high temperatures and pressures in the S-II engine compartment following ignition and continued high after interstage separation command. The unusually high temperatures from S-II ignition and until the S-II interstage separation signal are considered by Marshall Space Flight Center (MSFC) to be caused by a change in the engine heat shield skirts introduced on this flight, and therefore do not indicate a problem. However, the increasing temperatures after the time of normal S-II interstage separation are indicative of an abnormal condition. More detailed investigation based on performance evaluation and axial acceleration time history revealed that the interstage had not been jettisoned; however, due to the vehicle performance characteristics and performance margin, the desired orbit was achieved.

Data analysis confirms that the primary ordnance command was properly issued at R+189.9 seconds. The backup command was issued 100 milliseconds later but the exploding bridge wire circuit discharge was characteristic of an open circuit consistent with separation of the interstage disconnect by a minimum of 0.25 inch.

The linear shaped charge is mounted circumferentially around the S-II interstage. When fired by the primary command, the charge cuts the tension straps (in the direction of position II to position I) allowing the skirt to drop away. Normal propagation time of the linear shaped charge is approximately four milliseconds. Assuming a failure to propagate completely around the structure, analyses were made by appropriate contractor and government personnel to determine what area must remain intact in order to retain the skirt and what area must have been cut to allow rotation of the skirt sufficient to disconnect the connector panel. The various analyses isolate the region of failure to an arc extending from approximately $\Theta = 100$ degrees to as much as $\Theta = 200$ degrees.

This ordnance installation was different from prior Saturn flights. Previously, a single fire command from the instrumentation unit was issued which simultaneously detonated the linear shaped charge from both ends allowing the charge to propagate from both directions. On this flight, in an attempt to provide redundant firing commands, the detonators at each end of the linear shaped charge were separately connected to two command channels spaced 100 milliseconds apart due to the characteristics of the airborne equipment. As a result of the partial cutting of the interstage, it rotated sufficiently to separate the electrical connector prior to issuing the backup command.

A review of the history of manufacturing, acceptance, checkout, qualification and flight environment revealed no basic cause for failure. The most probable cause is secondary damage as a result of the meteoroid shield failure, attributed to falling debris as evidenced by the various shock and acoustic disturbances occurring in the 63-second time period.

The redundant mode of ordnance operation of all prior Saturn flights in which both ends of the linear shaped charge are fired at once from a single command would probably have prevented the failure, depending on the extent of damage experienced by the linear shaped charge.

FORWARD INTERSTAGE INTERNAL PRESSURE ANOMALY

Flight data indicated a deviation of the S-II forward interstage pressure from analytical values commencing at approximately 63 seconds. Inasmuch as the deviation from the analytical curve of the internal pressure versus time appeared to be coincident with the meteoroid shield failure, it was postulated that a portion of the shield had punctured the forward interstage. On this basis, it was possible to correlate the flight data with either an assumed 2.0 square foot hole in the conical section or an assumed 0.75 square foot hole in the cylindrical section.

RANGE SAFETY RECEIVER ANOMALY

During the S-II portion of the flight, the signal strength indications from both range safety receivers showed drops in level. From liftoff through R+259 seconds, both receivers maintained relatively stable values above range requirements. At R+259.57 seconds, receiver 2 signal strength began to drop and between this time and 522.1 seconds, both receivers indicated various degrees of signal strength shift. These signal strength shifts dropped below the 12 db safety margins required by Air Force Eastern

Test Range Manual 127-1. At R+327.81 seconds the receiver 2 signal strength dropped briefly below its threshold sensitivity. At this instant this receiver probably would not have responded to any range safety commands. Receiver 1 was, however, capable of receiving commands. At R+521.16, receiver 2 strength again dropped briefly to its threshold sensitivity. None of these drops could be correlated to ground system performance.

Analysis indicates that the most probable cause of the S-II receiver signal strength dropout was a variable phase shift within the vehicle's hybrid coupler due to the changing aspect angle produced by the moving vehicle and the fixed transmitting site. Because the decrease in receiver signal strength occurred with only one receiver at a time, range safety commands could have been received continuously throughout power flight. During two of these drops, however, the planned redundancy of range safety receivers was not available.

During this investigation, it was revealed that the Wallops Island and Bermuda ground stations did not continuously record ground transmitter power levels. The Board considers that such continuous recordings would be of value.

THE METEOROID SHIELD DESIGN

Although fairly simple in concept, the meteoroid shield had to provide such a variety of functions that it was, in fact, a quite complicated device. It was, foremost, a very lightly built cylindrical structure 270 inches in diameter (in the deployed condition) by 265 inches long.

In brief, the meteoroid shield is formed of a set of sixteen curved sheets of 2014 T6 aluminum panels, 0.025 inches thick, assembled at flanges and other fittings to form the cylinder shown. The forward and aft ends were reinforced with curved 7075 T6 angles.

Various special details were included in the assembly in order to hold it in place, deploy it in orbit, and provide access to the orbital workshop interior during prelaunch activities. The principal means of holding the shield in place in orbit (and to a lesser extent during powered flight) was a set of tension straps under the main tunnel. These straps were bonded to the orbital workshop wall and fitted with a hinge on each end to take the butterfly hinge that attaches to the adjacent meteoroid shield panel. These butterfly hinges were designed to rotate so as to lie against the sides of the main tunnel which enclosed the tension straps and various cable runs on the orbital workshop.

Clockwise from the tension straps and butterfly hinge, the next special feature is the auxiliary tunnel. This tunnel extends in an arch between panels of the thin meteoroid shield. The 28 titanium frames of this tunnel provide a very springy section in the relatively rigid hoop provided by the rest of the shield. The auxiliary tunnel also encloses a smaller tunnel covering the wiring for the thruster attitude control system. Farther around, in position I, there are two curved rectangular smaller panels, included to provide access to the orbital workshop.

Between positions I and IV, the two halves of the meteoroid shield overlap and are joined by a series of 14 trunnion bolts and straps. These trunnion bolts were used to adjust the tension with which the shield was held against the orbital workshop. Adjusting the bolts in the trunnion assemblies was a major aspect in positioning and tightening the meteoroid shield against the orbital workshop (rigging).

In order to provide the extra 30 inches of perimeter required when the meteoroid shield was deployed, a foldout panel assembly is included in the panel adjacent to the trunnions. The only remaining distinctive features of the meteoroid shield are the panels located over the scientific airlock and wardroom window at position III. The meteoroid shield is completed at the butterfly hinges and tension straps at position I.

Deployment Provisions

The deployment of the 265-inch-long meteoroid shield was accomplished by providing two folding panel sections on each side of a contained explosive pyrotechnic chain which extended axially for the full length of the shield except for short end reinforcements. When the ordnance strip is fired and separates the fold-over panel, the segments are released and the shield is deployed. After release of this folded panel, a number of swing arms are used to displace the shield away from the orbital workshop wall and hold it there. A rotational force is applied to these swing arms by a total of sixteen torsion rods suitably spaced around the ends of the meteoroid shield. When the meteoroid shield is stowed for launch, there is a larger twist in the torsion rods than after deployment. The links on one side of the ordnance chain swing in a direction opposite to those on the other side. The butterfly hinges on each side of the main tunnel permit the radial displacement of the shield at the location of the tension straps.

The meteoroid shield should therefore be regarded as a very limp system, which depends on being stretched tight around the orbital workshop to withstand the aerodynamic, vibratory, flutter and thrust loads at launch. After deployment, it needs very little strength to serve its primary objective as a meteoroid shield.

The Auxiliary Tunnel

The auxiliary tunnel extends from the forward skirt, down the full length of the meteoroid shield shield, and below the meteoroid shield by about 57 inches. Venting of this tunnel was provided through an outlet of 10 square inches under the corrugations of the tunnel cover at the aft end of the forward fairing. The tunnel was intended to be sealed at the aft end by a rubber boot assembly in both the stowed and deployed position. Note that the tunnel is displaced some 5 or 6 inches circumferentially upon deployment of the shield.

The main structural members of the auxiliary tunnel are titanium, arch-shaped, frame springs. These frames provide the structural tie between two meteoroid shield panels and provide both regulation of the pre-loading of the meteoroid shield to the orbital workshop and act as a flexible relief for diametrical changes resulting from thermal and pressure changes of the orbital workshop.

The tunnel also serves to protect the thrust attitude control system cables located in a small channel-shaped cover permanently attached to the orbital workshop. A segmented and corrugated outer skin form an aerodynamic fairing for the complete system and seals between forward and aft fairings.

Thermal Control

Although the primary purpose of the meteoroid shield is that of providing protection of the orbital workshop from meteoroids, it also plays a significant role in the thermal control system. Much of the overall thermal design was accomplished passively by painting the outer surfaces of the meteoroid shield black except for a large white cross-shaped pattern on the Earth side during flight. The entire surface of the orbital workshop wall was covered with gold foil. The overall choice of finishes biased the thermal design toward the cold side, it being easier to vernier control by heating rather than cooling.

Friction between the Meteoroid Shield and Orbital Workshop Wall

To provide a uniform tension throughout the meteoroid shield upon assembly and rigging for flight, and to permit transfer of the trunnion bolt tension into the frames of the auxiliary tunnel, it was necessary to minimize friction between the shield and the external surface of the orbital workshop. This was accomplished by applying a Teflon coating to the entire inner surface of the meteoroid shield assembly. Special care was also taken to assure that all fastening rivets be either flush with or below the Teflon surface of the shield. In addition to considerations of friction, the elimination of rivet head protrusions was important in not damaging the rather delicate gold surface used to provide the proper emissivity of the outer orbital workshop wall surfaces as mentioned above. This was a vapor-deposited gold surface applied to a Kapton backing and bonded to the outer workshop wall with an adhesive.

Panel Details

The sixteen panels comprising the meteoroid shield were formed of 0.025 inch thick aluminum stock fitted with doublers and angles to permit their assembly. In each of these panel joints, 96 holes of 1/8-inch diameter were drilled to vent any air trapped under the meteoroid shield skin. The special panel joint is required next to the SAS-1 wing because of the unavailability of sufficiently wide panel stock for the panel under SAS-1. It was a strap of metal of this special joint that became embedded in the SAS-1 cover and prevented automatic deployment of SAS-1 in orbit. It is, perhaps, of passing interest to note the longer length of exposed bolts in this particular joint.

Around the top of the panels is located an angle and a neoprene rubber rain or weather seal. This seal was not intended to be an aerodynamic seal and could not be expected

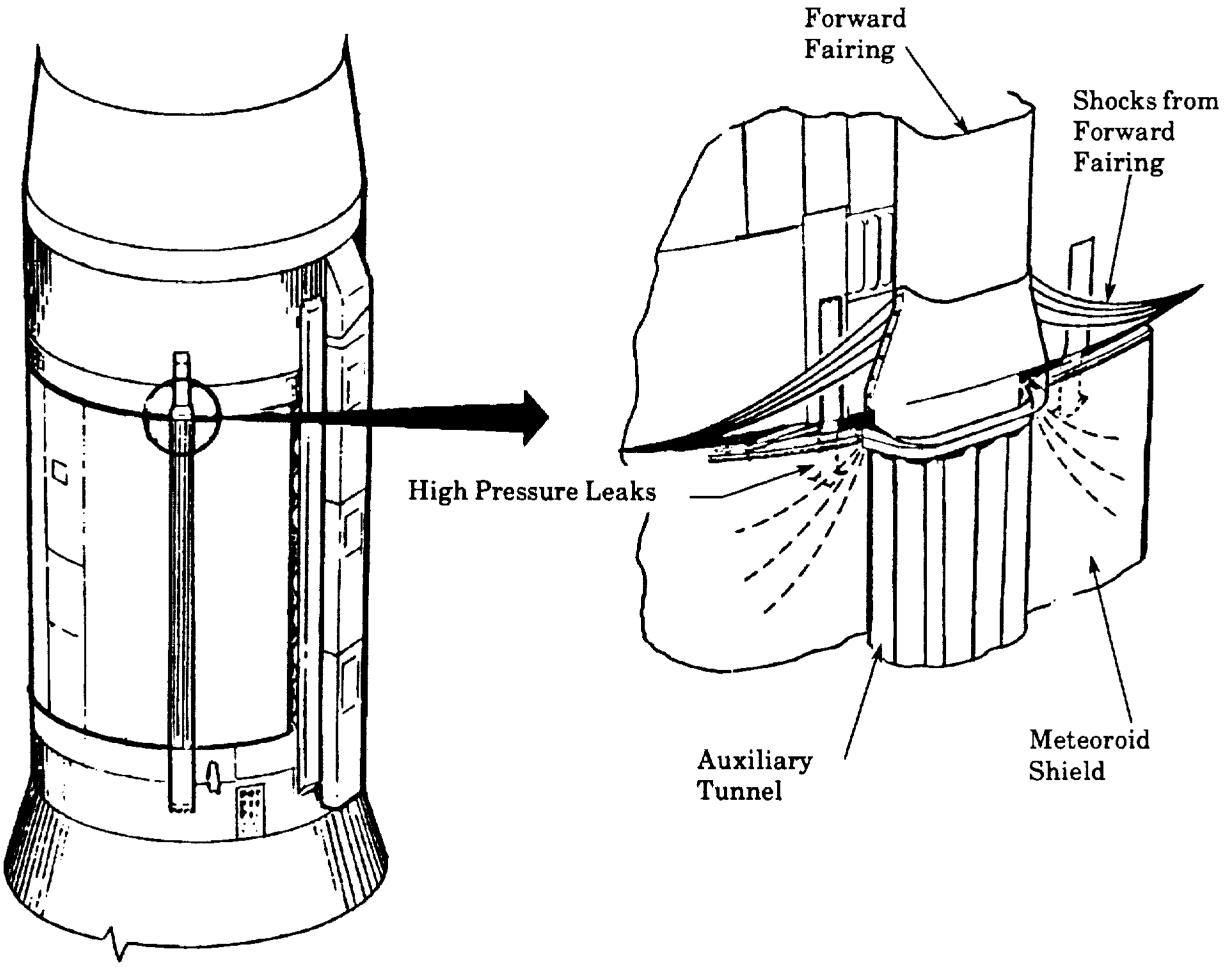


Figure 3 Compressibility waves from the forward auxiliary tunnel fairing

to accommodate significant relative deflections between the orbital workshop and meteoroid shield surfaces. To provide meteoroid protection at the two ends of the meteoroid shield, small strips of thin stainless steel "fingers" were squeezed down between the orbital workshop and the meteoroid shield when stowed. The thrust load of the shield, which weighs some 1200 pounds, is transferred to the forward flange of the aft skirt through a group of twelve thrust blocks.

SUMMARY AND RECOMMENDATIONS

The preceding analysis and discussion of possible failure modes of the meteoroid shield have identified at least two ways that it could fail in flight. Although the most probable cause of the present failure was the lifting of the shield from the orbital workshop tank by excessive pressures in the auxiliary tunnel, other failure modes could have occurred in other regions of flight or under more severe flight environments that were encountered by Skylab 1.

Among these other modes of potential failure, which could combine in various ways under varying conditions of flight, are excessive pressures under the forward edge of the shield, or inadequate venting of the folded ordnance panel. The inherently light spring force of the auxiliary tunnel frames, the crushing loads on these frames in flight, the inherent longitudinal flexibility of the shield assembly, the forces applied by the swing links to deploy the shield, the possible breathing of the shield panels as cavities are vented, the noncylindrical nature of the underlying pressurized tank, and the uncertain tension loads applied to the shield in rigging for flight all contribute to a lack of rigidity of the shield and a weakness of its structural integrity with the underlying tank structure.

A simple and straightforward solution to these inherent problems of the present shield design is therefore not likely. A fundamentally different design concept seems in order. One solution is, of course, to simply omit the meteoroid shield, suitably coat the orbital workshop for thermal control and accept the meteoroid protection afforded by the orbital workshop tank walls. Although the Board has not conducted an analysis, meteoroid flux levels are now known to be considerably lower than those used in the original calculations. A new analysis, based on these flux levels, may show acceptable protection.

Should some additional meteoroid protection be required, the Board is attracted to the concept of a fixed, nondeployable shield. Although the inherent weight advantages of a separable bumper are not available in this approach, the mission of Skylab could probably be satisfied in this manner. One concept would be to bond an additional layer of metal skin to the surface of the tank with a layer of nonventing foam between the orbital workshop tank and the external skin. The problem being statistical in nature, the entire shell of the orbital workshop would not have to be covered.

POSTULATED SEQUENCE OF THE MOST PROBABLE FAILURE MODE

The availability of flight data from the instrumentation on the meteoroid shield and the vehicle disturbances, the design features of the meteoroid shield, the solar array system photographs taken in orbit, descriptions by the astronauts, and other information permit the following postulation of the probable sequence of events associated with the meteoroid shield failure.

In Figure 4, sketches and details of salient events are correlated to the roll rate data around the 63 second anomaly period. The events are designated on the figures by times which are consistent with the available data.

60.12 Seconds - Meteoroid shield liftoff and local inflation in the vicinity of the auxiliary tunnel was indicated by a small shift in position of the torsion rod on the forward edge just to the left of the tunnel.

61.78 Seconds - Air entered the forward fairing opening, raised the pressure under the shield and high mass flows escaped through the adjacent holes in the butterfly hinge. This flow produced reactive force causing a gradual decrease in roll rate between 61.78 seconds and 62.74 seconds.

62.74 to 62.79 Seconds - Burst pressure under the auxiliary tunnel and adjacent meteoroid shield caused a large tangential load on the forward section of the butterfly hinge, causing the whole hinge to unzip. Fly around inspection indicated that the failure of the butterfly hinge occurred at the hinge line adjacent to the main tunnel.

The butterfly hinge was now completely broken. Aerodynamic drag on the meteoroid shield including the bulky auxiliary tunnel produced tension in the shield and pulled on the vehicle so as to roll it in the direction shown, that is, opposite to that noted earlier. The large area and mass of this metal flag induced a more rapid change in roll rate than the earlier jetting through the butterfly hinge. This process terminated as the meteoroid shield started to wrap around and lift the SAS-2 wing.

62.79 to 62.90 Seconds - During this interval the shield was wrapping around the SAS-2 wing producing a negative roll torque in the vehicle. At about 62.85 seconds the SAS-2 tie-downs were broken.

62.90 Seconds - Upon release of the SAS-2, the tension in the shield was transferred to the trunnions, causing failure of the trunnion straps. Upon separation of this section of the shield, the negative roll torque ended.

62.90 to 62.95 Seconds - In this interval, the remaining section of the meteoroid shield began unwinding, introducing a large positive roll torque.

63.17 Seconds - A large shock was detected by the instrument unit upper mounting ring vibration sensor due to the impact of the separated section of the meteoroid shield upon the conical adapter between the orbital workshop and the SAS-1 stage.

63.7 Seconds - The meteoroid shield continued to unwind and whip until 63.7 seconds when it reached SAS-1 wing. As the meteoroid shield began to wrap around the SAS-1 wing, a negative roll torque resulted. The meteoroid shield then ripped apart from top to bottom at the longitudinal joint adjacent to SAS-1, pulling a portion of the joint assembly over the SAS-1 wing as the meteoroid shield section departed. From this point on the vehicle showed normal response to its roll control system.

POSSIBLE IMPACT OF COSTS AND SCHEDULES ON THE METEOROID SHIELD

The origin of Skylab in late 1966—as an extension of the use of Apollo hardware for experiments in Earth orbit—imposed an initial environment of limited funding and strong schedule pressures on the program. Skylab, then designated the Apollo Applications Program (AAP), was to fit in among the Apollo flights under schedules imposed by the mainline Apollo program. Funding was provided out of the Apollo program and thus the needs of Skylab competed with those of the higher priority Apollo program.

The situation changed in mid-1969 when Skylab became a major line item in its own right and was to use a Saturn-V launch vehicle with a dedicated, dry, orbital workshop. From that point on, increased funding and new flight schedules were established for Skylab. Nonetheless, the original concept of the meteoroid shield was retained when the orbital workshop changed from Saturn-IB propulsion stage to a dry workshop launched by a Saturn-V. The Board was therefore interested in determining the extent, if any, that either the initial limitation of funds and time, or any subsequent limitations, determined the design or thoroughness of development of the meteoroid shield. This inquiry was limited to the possible effect of funding and schedule of the meteoroid shield as designed and flown on Skylab 1 and did not consider whether meteoroid protection could have or should have been provided in some other way had the program not evolved as it did.

In the Board's review of the evolution of the meteoroid shield from initial design concept, through testing and development, to final assembly for flight, particular attention was devoted to any impacts arising from limitation of funds or time. Extensive discussions were also held with management personnel of MDAC-W, MSFC, JSC, and NASA Headquarters on this matter. In no instance could the Board find any evidence that the design or testing of the meteoroid shield was

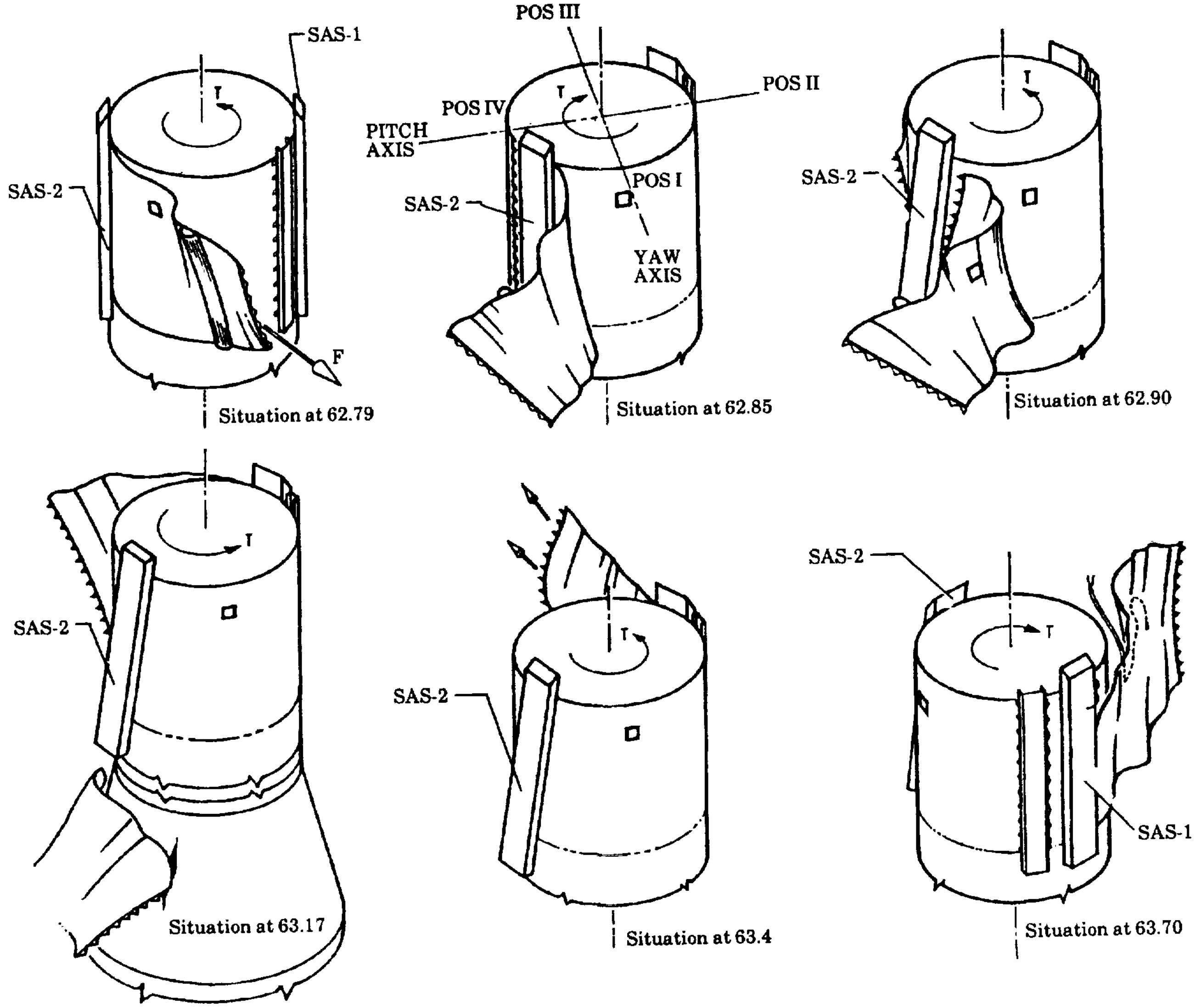


Figure 4 — Postulated Sequence Failure Mode

compromised by lack of funds or time. Program personnel, both government and contractor, had full confidence in the basic concept of the meteoroid shield and thus saw no need to alter the design when the change to a dry, Saturn-V launched orbital workshop occurred. Given the concept that the shield was to be maintained tight to the orbital workshop tank, and thus structurally integrated with the well-established S-IVB structure, the emphasis of testing given to ordnance reliability and shield deployment was considered proper. Neither the records of Skylab nor the memories of key personnel revealed any tests or analyses of the meteoroid shield that were considered desirable at the time and which were precluded by lack of funds or time.

THE SKYLAB MANAGEMENT SYSTEM

The management system utilized for the Skylab program was derived directly from that which had been developed and used in the Apollo program. As such, it included a series of formal reviews and certifications at progressive points in the program life cycle that are intended to provide visibility to contractor and NASA management on program status, problems and their resolution. The selected review points and their primary purpose are set forth in Skylab Program Directive No. 11A, which is summarized as follows:

Preliminary Requirements Review (PRR). "To verify by formal review the suitability of the conceptual configuration and to establish the requirements and action necessary to achieve a design baseline."

Preliminary Design Review (PDR). "To verify by formal review the suitability of the baseline design of the Contract End Item."

Critical Design Review (CDR). "To verify by formal review the suitability of the design of a Contract End Item when the design is essentially complete."

Configuration Inspection (CI). "To certify that the configuration for the Contract End Item as being offered for delivery is in conformance with the baseline established at the CDR."

Certification of Flight Worthiness (COFW). "To certify that each flight stage module and experiment is a complete and qualified item of hardware prior to shipment."

Design Certification Review (DCR). "To examine the design of the total mission complex for proof of design and development maturity."

Flight Readiness Review (FRR). "A consolidated review of the hardware, operational and support elements to assess their readiness to begin the mission."

The primary thrust of these key program milestones is thus a formal review and certification of equipment design or program status; the primary purpose being served is to provide visibility into these matters to senior NASA and contractor program management. As noted in the Skylab Program Directive, the organization and conduct of the review is a major responsibility of a senior program or management official. For each review, specific objectives are to be satisfied, in conformance with preestablished criteria and supported by specified documentation. The reviews are thus highly structured and formal in nature, with a major emphasis on design details, status of various items and thoroughness of documentation. Several hundred specialists, subsystem engineers and schedule managers are generally in attendance.

The material presented in these reviews is, of course, developed over a period of time in many lower-level reviews and in monthly progress reports dealing with various systems and subsystems. In addition, several other major reviews peculiar to Skylab were conducted, including the following:

- Cluster System Review of December 1967
- Mathew's Subsystem Review Team of August 1970-July 1971

- Critical Mechanisms Review of March 1971
- Systems Operations Compatibility Assessment Review of October 1971-June 1972
- Structural/Mechanical Subsystem Reviews of July 1971-May 1972
- Hardware Integrity Review of March 1973
- MSFC Center Director's Program Reviews

There was thus no shortage of reviews. In order to determine the consideration given to the meteoroid shield throughout the program, the Board examined the minutes, presentation material, action items, and closeout of data of each of these reviews and progress reports. In every case, complete records and documentation were available for inspection. In no case did the Board uncover any conflict or inconsistency in the record. All reviews appeared to be in complete conformance to Program Directive 11A and were attended by personnel appropriate to th? subject matter under consideration. The system was fully operational.

And yet, a major omission occurred throughout this process—consideration of aerodynamic loads on the meteoroid shield during the launch phase of the mission. Throughout this six year period of progressive reviews and certifications the principal attention devoted to the meteoroid shield was that of achieving a satisfactory deployment in orbit and containment of the ordnance used to initiate the deployment. As noted in the preceding section on possible failure modes, design attention was also given to the strength of the hinges, trunnion straps and bolts, to the crushing pressures on the frames of the auxiliary tunnel, to flutter and to the venting of both the auxiliary tunnel and the several panels of the shield. But never did the matter of aerodynamic loads on the shield or aeroelastic interactions between the shield and its external pressure environment during launch receive the attention and understanding during the design and review process which in retrospect it deserved.

This omission, serious as it was, is not surprising. From the beginning, a basic design concept and requirement was that the shield be tight to the tank. As clearly stated in much of the early documentation, the meteoroid shield was to be structurally integral with the S-IVB tank—a piece of structure that was well proven in many previous flights. The auxiliary tunnel frames, the controlled torque on the trunnion bolts and the rigging procedure itself were all specifically intended to keep the shield tight against the tank. The question of whether the shield would stay there under the dynamics of flight through the atmosphere was simply not considered in any coordinated manner—at least insofar as the Board could determine by this concentrated investigation.

Possibly contributing to this oversight was the basic view of the meteoroid shield as a piece of structure. Organizationally, responsibility for the meteoroid shield at MDAC-W was established to develop it as one of the several structural subsystems, along with such items as spacecraft structure and penetrations, pressure vessels, scientific airlocks, protective covers and finishes. Neither the government, (MSFC), or the contractor, (MDAC-W), had a full-time subsystem engineer assigned to the meteoroid shield. While it is recognized that one cannot have a full-time engineer on every piece of equipment, it is nonetheless possible that the complex interactions and integration of aerodynamics, structure, rigging procedures, ordnance, deployment mechanisms, and thermal requirements of the meteoroid shield would have been enhanced by such an arrangement. Clearly, a serious failure of communications among aerodynamics, structures, manufacturing and assembly personnel, and a breakdown of a systems engineering approach to the shield, existed over a considerable period of time. Further, the extensive management review and certification process itself, in its primary purpose of providing visibility of program status to management, did not identify these faults.

Further insight into this treatment of the meteoroid shield as one of several structural subsystems is obtained by a comparison of a listing of the design reviews conducted on both the meteoroid shield and the solar array system. At MDAC-W, the solar array system was considered a major subsystem and was placed under the direction of a full-time project engineer.

The Board is impressed with the thoroughness, rigor and formalism of the management review system developed by Apollo and used by Skylab. Great discipline is imposed upon everyone by this system and it has served very well. In a large program as geographically dispersed and intrinsically complex as Skylab, such visibility of program status and problems is a management necessity. We therefore have no wish to alter this management system in any basic manner. But all systems created by humans have their potential flaws and inherent hazards. Such inherent flaws and weaknesses must be understood by those who operate the system if it is not to become their master. We therefore wish to identify some of those potential flaws as they have occurred to us in this investigation, not to find fault or to identify a specific cause of this particular flight failure but to use this experience to further strengthen the management processes of large and complex endeavors.

As previously noted, the management system developed by NASA for manned space flight places large emphasis on rigor, detail and thoroughness. In hand with this emphasis comes formalism, extensive documentation, and visibility in detail to senior management. While nearly perfect, such a system can submerge the concerned individual and depress the role of the intuitive engineer or analyst. It may not allow full play for the intuitive judgment or past experience of the individual. An emphasis on a management system, can, in itself, serve to separate the people engaged in the program from the real world of hardware. To counteract these potential hazards and flaws, we offer the following suggestions.

- Deployable systems or structures that have to move, or that involve other mechanisms, devices, or components in their operation, should not be considered as a piece of structure or be the basic responsibility of a structures organization.
- A complex, multi-disciplinary system such as the meteoroid shield should possibly have a designated project engineer who is responsible for overseeing all aspects of analysis, design, fabrication, test and assembly.
- Management must always strive to counteract the natural tendency of engineers to believe that a drawing is the real world. First-hand experience with how hardware behaves and can fail is of the essence to design engineers. Possibly, some design engineers should be required to spend time in testing, operations, or failure analysis. Such experience may not contribute to cleverness or sophistication of analysis, but something equally valuable—actual experience—may be added to the design group. An unfamiliarity with hardware, first hand, makes it difficult to conceptualize a living, breathing, piece of hardware from an analysis or a drawing.
- The, extensive use of the computer for complex analyses can serve to remove the analyst from the real world. One should, therefore, require a simplified or supporting analysis that provides an understandable rationale for the phenomena under consideration before accepting the results of a computer analysis.
- The emphasis on "visibility to management" in the review process should not be extended to the point that one can be led to believe the job is completed, or the design is satisfactory, when such visibility

is provided. A major emphasis on status, on design details, or on documentation can detract from a productive examination of "how does it work" or "what do you think."

- Today's organizations seldom include the old-fashioned chief engineer who, relatively devoid of administrative or managerial duties, brings total experience and spends most of the time in the subtle integration of all elements of the system under purview. Perhaps we should more actively seek and utilize these talented individuals in an engineering organization.

SIGNIFICANT FINDINGS

1) The launch anomaly that occurred at approximately 63 seconds after lift-off was a failure of the meteoroid shield of the orbital workshop.
2) The SAS-2 wing tie downs were broken by the action of the meteoroid shield at 63 seconds. Subsequent loss of the SAS-2 wing was caused by retro-rocket plume impingement on the partially deployed wing at 593 seconds.
3) The failure of the S-II interstage adapter to separate in flight was probably due to damage to the ordnance separation device by falling debris from the meteoroid shield.
4) The most probable cause of the failure of the meteoroid shield was internal pressurization of its auxiliary tunnel. This internal pressurization acted to force the forward end of the tunnel and meteoroid shield away from the orbital workshop and into the supersonic air stream. The resulting forces tore the meteoroid shield from the orbital workshop.
5) The pressurization of the auxiliary tunnel resulted from the admission of high pressure air into the tunnel through several openings in the aft end. These openings were: (1) an imperfect fit of the tunnel with the aft fairing; (2) an open boot seal between the tunnel and tank surface; and (3) open stringers on the aft skirt under the tunnel.
6) The venting analysis for the tunnel was predicated on a completely sealed aft end. The openings in the aft end of the tunnel thus resulted from a failure to communicate this critical design feature among aerodynamics, structural design, and manufacturing personnel.
7) Other marginal aspects of the design of the meteoroid shield which, when taken together, could also result in failure during launch are:
   a) The proximity of the meteoroid shield forward reinforcing angle to the air stream
   b) The existence of gaps between the orbital workshop and the forward ends of the meteoroid shield
   c) The light spring force of the auxiliary tunnel frames
   d) The aerodynamic crushing loads on the auxiliary tunnel frames in flight
   e) The action of the torsion-bar actuated swing links applying an outward radial force to the meteoroid shield
   f) The inherent longitudinal flexibility of the shield assembly
   g) The nonuniform expansion of the orbital workshop tank when pressurized
   h) The inherent difficulty in rigging for flight and associated uncertain tension loads in the shield.
8) The failure to recognize many of these marginal design features through six years of analysis, design and test was due, in part, to a presumption that the meteoroid shield would be "tight to the tank" and "structurally integral with the S-IVB tank" as set forth in the design criteria.
9) Organizationally, the meteoroid shield was treated as a structural subsystem. The absence of a designated project engineer for the shield contributed to the

lack of effective integration of the various structural, aerodynamic, aeroelastic, test fabrication, and assembly aspects of the meteoroid shield system.

10) The overall management system used for Skylab was essentially the same as that developed in the Apollo program. This system was fully operational for Skylab; no conflicts or inconsistencies were found in the records of the management reviews. Nonetheless, the significarce of the aerodynamic loads on the meteoroid shield during launch was not revealed by the extensive review process.

11) No evidence was found to indicate that the design, development and testing of the meteoroid shield were compromised by limitations of funds or time. The quality of workmanship applied to the meteoroid shield was adequate for its intended purpose.

12) Given the basic view that the meteoroid shield was to be completely in contact with and perform as structurally integral with the S-IVB tank, the testing emphasis on ordnance performance and shield deployment was appropriate.

13) Engineering and management personnel on Skylab, on the part of both contractor and government, were available from the prior Saturn development and were highly experienced and adequate in number.

14) The failure to recognize these design deficiencies of the meteoroid shield, as well as to communicate within the project the critical nature of its proper venting, must therefore be attributed to an absence of sound engineering judgment and alert engineering leadership concerning this particular system over a considerable period of time.

CORRECTIVE ACTIONS

1) If the backup orbital workshop or a similar spacecraft is to be flown in the future, a possible course of action is to omit the meteoroid shield, suitably coat the orbital workshop for thermal control, and accept the meteoroid protection afforded by the orbital workshop tank walls. If, on the other hand, additional protection should be necessary, the Board is attracted to the concept of a fixed, nondeployable shield.

2) To reduce the probability of separation failures such as occurred at the S-II interstage Second Separation Plane, both linear shaped charges should be detonated simultaneously from both ends. In addition, all other similar ordnance applications should be reviewed for a similar failure mode.

3) "Structural" systems that have to move or deploy, or that involve other mechanisms, equipment or components for their operation, should not be the exclusive responsibility of a structures organization.

4) Complex, multi-disciplinary systems such as the meteoroid shield should have a designated project engineer who is responsible for all aspects of analysis, design, fabrication, test and assembly.

OBSERVATIONS ON THE MANAGEMENT SYSTEM

The Board found no evidence that the design deficiencies of the meteoroid shield were the result of, or were masked by, the content and processes of the management system that were used for Skylab. On the contrary, the rigor, detail, and thoroughness of the system are doubtless necessary for a program of this magnitude. At the same time, as a cautionary note for the future, it is emphasized that management must always be alert to the potential hazards of its systems and take care that an attention to rigor, detail and thoroughness does not inject an undue emphasis on formalism, documentation, and visibility in detail. Such an emphasis can submerge the concerned individual and depress the role of the intuitive engineer or analyst. It will always be of importance to achieve a cross-fertilization and broadened experience of engineers in analysis, design, test or operations. Positive steps must always be taken to assure that engineers become familiar with actual hardware, develop an intuitive understanding of computer-developed results, and make productive use of flight data in this learning process. The experienced chief engineer, whose time can be spent in the subtle integration of all elements of the system under review, free of administrative and managerial duties, can also be a major asset to an engineering organization.

REPORT OF THE SEASAT FAILURE REVIEW BOARD

by the NASA Investigation Board

The Seasat spacecraft failed on October 9, 1978, after satisfactory operation in orbit for 105 days, as a result of a loss of electrical power in the Agena bus that was used as a part of the spacecraft. The loss of power was caused by a massive and progressive short in one of the slip ring assemblies that was used to connect the rotating solar arrays into the power subsystem. The most likely cause of this short was the initiation of an arc between adjacent slip ring brush assemblies. The triggering mechanism of this arc could have been either a wire-to-brush assembly contact, a brush-to-brush contact, or a momentary short caused by a contaminant that bridged internal components of opposite electrical polarity.

The slip ring assembly, as used in the Seasat spacecraft, was connected into the power subsystem in such a way that most of the adjacent brush assemblies were of opposite electrical polarity. This wiring arrangement, together with the congested nature of the design itself, made the Seasat slip ring assembly a unique, first-of-a-kind component that was particularly prone to shorting.

The possibility of slip ring failures resulting from placing opposite electrical polarities on adjacent brush assemblies was known at least as early as the summer of 1977 to other projects within the contractor's organization. Furthermore, failures of slip ring assemblies due to shorting between brushes had been experienced by the prime contractor on the slip ring assemblies used by other programs. That the Seasat organization was not fully aware of these potential failure modes was due to a breakdown in communication within the contractor's organization.

In addition to this small, though fatal, breakdown in communications, the failure to give the slip ring assembly the attention it deserved was due, in large part, to an underlying program policy and a pervasive view that Seasat's Agena bus was a standard, well-proven piece of equipment that had been used on other programs. In actuality, however, three major subsystems—the electrical power subsystem, the attitude control subsystem, and the data subsystem—were substantially modified for use on Seasat's Agena bus. So firmly rooted was this principle of using a "standard Agena bus" that, even after the engineering staffs of both the government and the contractor were well aware of the final uniqueness of their bus, the words, and the associated way of doing business, persisted to the end.

The point of view that the Seasat bus was flight proven, standard equipment proved to have far-reaching consequences. It became program policy to minimize testing and documentation, to qualify components by similarity wherever possible, and to minimize the penetration into the Agena bus by the government. It led to a concentration by project management of the sensors, sensor integration, and the data management system to the near exclusion of the bus subsystems. Important component failures were not reported to project management, a test was waived without proper approval, and compliance with specifications was weak. The component that failed—the slip ring assembly—was never mentioned in the briefing charts for either the Consent to Ship meeting or the Critical Design Review.

The Failure Modes, Effects and Criticality Analysis that was conducted for the electrical power subsystem did not consider shorts as a failure mode and thus did not reveal the presence of single point failure modes in the system or provide a basis for the development of a full complement of safing command sequences that could be used by the flight controllers in responding to anomalies in the power subsystem. A lack of clarity and rigor in the operating requirements and constraints documents for the power subsystem of the bus, together with this lack of safing command sequences, prevented the flight controllers from having all the tools they needed to do their job. The flight controller for the power subsystem was also new to his job at the time of the failure and thus was not sufficiently knowledgeable of the system he was controlling. While no action of the flight controllers contributed to the failure, they did fail to follow the prescribed procedures in response to the information available to them at the time of the failure.

The advantages of using standard, well proven equipment in terms of both cost and mission success are well recognized. But the experience of Seasat illustrates the risks that are associated with the use of equipment that is classified as "standard" or "flight proven." The uncritical acceptance of such classifications by the Seasat engineering staff submerged important differences in both design and application from previously used equipment. It is therefore important that thorough planning be conducted at the start of a project to fully evaluate the heritage of previously used equipment and to establish project plans and procedures that enable the system to be selectively penetrated.

THE SEASAT MISSION AND ITS SPACECRAFT

The Seasat Project was a proof-of-concept mission whose objectives included demonstration of techniques for global monitoring of oceanographic and surface meteorological phenomena and features, provision of oceanographic data for both application and scientific areas, and the determination of key features of an operational ocean dynamics monitoring system.

To fulfill these objectives, the Seasat sensor complement comprised a radar altimeter (ALT), a synthetic aperture radar (SAR), a Seasat-A scatterometer system (SASS), a scanning multichannel microwave radiometer (SMMR), and a visual and infrared radiometer (VIRR). All of these sensors except the SAR operated continuously; telemetry from them, as well as from all engineering subsystems, was sent in real-time when over a ground station and recorded on a tape recorder for later transmission to provide data for a full orbit. SAR data had to be transmitted in real-time, without the use of the onboard recorder, to specially equipped stations because of its high data rate. The normal duty cycle for the SAR was four percent.

The five sensors were integrated into a sensor module that provided mounting, thermal control, power conditioning, telemetry, and command support to the instruments. The second major element of the spacecraft was an Agena bus which provided attitude control, electrical power, telemetry and command functions to the sensor module. In addition to these on-orbit functions, the Agena bus also provided injection stage propulsion and guidance to orbit. The spacecraft was three-axis stabilized with all sensors Earth pointing and is shown in its on-orbit configuration in Figure 1. To provide near global coverage, the spacecraft was injected into a 790 kilometer, near circular orbit with an inclination of 108 degrees and a period of approximately 101 minutes. Design lifetime was one year on orbit, with expendables provided for a three-year life.

The sensors were provided by various NASA Centers. The sensor module, the Agena bus and the integration of the sensors, sensor module and Agena bus into a spacecraft was provided by the Lockheed Missles and Space Company under contract to the Jet Propulsion Laboratory (JPL).

Responsibility for Seasat project management, mission planning and direction, mission operations and experiment data processing resided at JPL. The Goddard Space Flight Center (GSFC) provided network support and spacecraft orbit and attitude determinations; use was therefore made of the existing Spaceflight Tracking and Data Network, the NASA Communications (NASCOM) network, and the Project Operations Control Center that are operated by GSFC.

To place this failure review in a proper perspective, it is noted that the Seasat spacecraft operated in orbit in a generally satisfactory maneuver for over three months and provided a large amount of scientific data. The sensors represented a significant advance in technology and their integration into the sensor module, a large engineering challenge. In addition, Seasat also required the creation of significantly enlarged capabilities in the acquisition and processing of flight data. That the important and significant technical and engineering advancements were achieved is a tribute to the skill and dedication of all who were associated with this program.

The Seasat spacecraft was successfully launched on June 26, 1978, and thus operated for 105 days until the failure occurred on October 9, 1978. During this time in orbit, the spacecraft operation was generally satisfactory with considerable data being obtained from all of the sensors. Three significant anomalies were experienced during the life of Seasat in orbit, one involving sun interference in the attitude control system scan wheels, one caused by a sticking thermostat in a sensor heater circuit, and one in which the spacecraft suffered an abnormally low bus voltage for several orbits. Because of a possible relationship of these latter two anomalies with the failure of October 9, 1978, they were specifically investigated by the Board.

PROGRAM HISTORY AND MANAGEMENT

The Seasat program was conceived and initiated in a period of transition in the philosophy of management of NASA programs following the Apollo program. Apollo, and to varying degrees other NASA flight programs, were characterized by extensive test programs, large formal documentation systems, and comprehensive and frequent technical and management reviews. A large in-house staff was required in order to implement this approach. The high cost of conducting space programs in this mode severely constrained the future uses of space. During the final phases of the Apollo program, NASA management accordingly instituted a policy aimed at reducing the cost space missions. This policy was aggressively pursued by the highest levels of management.

A Low Cost Systems Office was established in Headquarters to oversee a standardization program and to encourage the use of existing hardware. This program included the development of standard components as well as a multimission spacecraft.

A major emphasis was placed on shifting work from in-house to out-of-house in consideration of reducing the NASA manpower base. Design-to-cost techniques and cost benefits of heritage through the use of hardware and software developed for other programs were subjects to be addressed at each step in the approval cycle.

The basic philosophy of the Seasat program was thus established in an environment in which management emphasis was shifting from one of demonstrating a national capability to operate reliably in space to one of reducing the cost of utilizing space. Design-to-cost was a fundamental tenet of the Seasat project definition. A cost estimate of $58.2 million was established as a target cost at the end of the feasibility study phase in mid-1973 and was imposed as a design-to-cost ceiling in December 1973 by NASA management. Any overruns were to be offset by descoping the mission content.

In attempting to define a program which would both satisfy the user community and live within the ceiling cost, the concept of making maximum use of proven existing hardware and software was adopted early in the program planning phase. This in turn provided for a reduction in design and development effort and in the size of the in-house staff needed to monitor the activity.

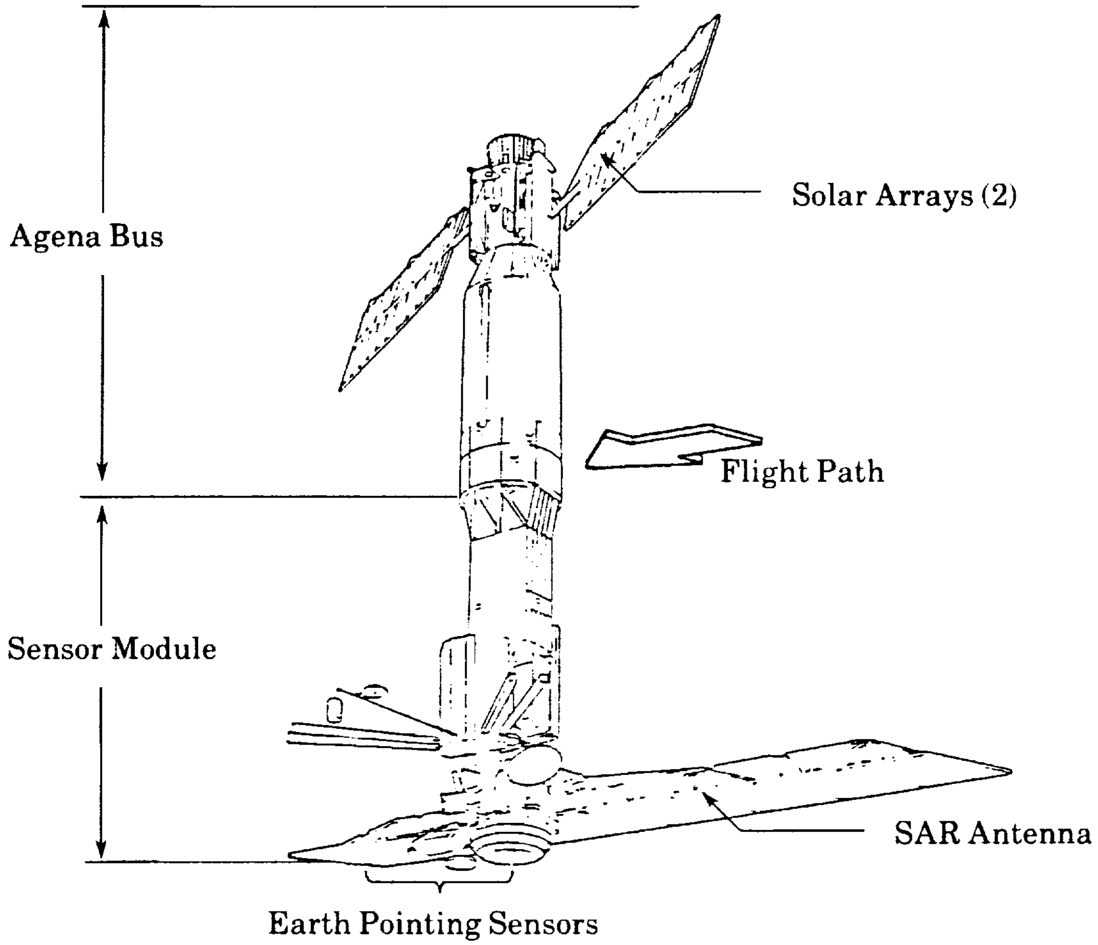


Figure 1 On-Orbit Configuration of the Seasat Spacecraft

These were key elements of the management philosophy which influenced the structure and conduct of the program.

PROGRAM PLANNING

Feasibility Studies (Phase A) - Feasibility for the Seasat mission was established in '73 through three studies conducted by the JPL, GSFC, and the Applied Physics Laboratory of the Johns Hopkins University. These studies were aimed at meeting the set of user requirements generated at a series of meetings held in the first half of 1973 among NASA and representatives of the governmental, commercial, and institutional communities of users of ocean dynamics data. With the user requirements as a basis, the feasibility studies examined the Seasat mission from an overall systems viewpoint, including a review of instrumentation and possible spacecraft (bus) approaches to accommodate the instrumentation.

Subsequent to the submission of the Phase A studies in July 1973, a joint NASA/User Study Task Team was formed to review the Phase A studies, integrate the results, and provide technical and programmatic guidance for more in-depth Definition Phase studies.

As a result of this review, the Task Team recommended a Baseline Mission which included a complement of the five sensor types that actually ended up flying on Seasat.

Based upon cost estimates prepared by the Phase A study participants, the Task Team recommended a target cost of $58.2 million for the Baseline Mission. This included the cost of the spacecraft bus and instruments, the launch vehicles, and tracking and data acquisition. An Alternate Payload Mission of reduced capability, excluding the synthetic aperture radar, was also recommended for further study with a target cost of $43.2 million.

There was some discussion in the Seasat Study Study Task Team Report (October 1973) of the use of an existing bus to minimize cost. The idea, however, was addressed with some skepticism. While it was believed that the use of subsystems with a high degree of inheritance from existing programs was desirable and possible, it was not clear at that time that an existing bus could be adapted economically.

Definition Studies and Preliminary Design (Phase B) - Definition Phase Studies of the Baseline and Alternate Payload Missions recommended by the Seasat Study Task Team were conducted from November 1973 to the summer of 1974. The Wallops Flight Center managed the Definition Phase Study of the Baseline Mission which was conducted by the Applied Physics Laboratory. The JPL, assisted by various aerospace companies familiar with Earth satellite design, conducted the Definition Phase Study of the Alternate Mission.

In December 1973, NASA management adopted the $58.2 million figure recommended by the Task Team as a *not to exceed ceiling* for the Seasat Baseline Mission. The efforts of the Definition Phase Study participants were accordingly intensified to develop the most economical satellite system possible that would best suit the user requirements within the cost ceiling.

GSFC declined to participate in the Definition Phase activity as they had serious doubts as to their ability to structure a full Baseline Mission within the design-to-cost ceiling.

With the stimulus of the design-to-cost ceiling, and management emphasis on the maximum use of existing subsystem hardware, the JPL Definition Phase Group proposed the of idea building a spacecraft system comprising two major elements: a sensor module designed specifically for Seasat, and a spacecraft bus based on an existing, flight proven bus devloped for other Air Force or NASA programs. The JPL viewed the results of the Phase A studies as indicating that the requirements of the sensors could be satisfied by standard support subsystems for attitude control, power, structures, thermal control, etc. On the other hand, the area of greatest uncertainty was seen to be the definition of the sensor's operating capabilities, data requirements and sensor system integration. It was therefore proposed that if a suitable spacecraft bus were available, the design and development effort could be concentrated on the sensors and their integration with a sensor module that could then be mated to the bus via a mechanical/electrical interface.

The JPL entered into four $15,000 study contracts with aerospace companies (Boeing, General Electric, Lockheed, and TRW) that had existing spacecraft designs with capabilities in the range of Seasat requirements to evaluate the concepts that: (1) there are existing buses that could be used, without modification, to supply the necessary support functions for the sensor payload, and (2) new design functions could be incorporated in a separate module along with the sensors and thereby reduce the systems development task to a *sensor system* development task. The studies were conducted from November 15, 1973 to March 30, 1974. The sensors were described to the study contractors as they were developed on December 15, 1973, with updates as appropriate until the end of these studies.

It was concluded as a result of these studies that basic sensor support requirements could be satisfied by the existing spacecraft bus designs studied with "no major changes," although "minor modifications" were acknowledged to be required. It was contemplated, for example, that minor modifications would be required of the attitude control, power, and temperature control subsystems. Telemetry, tracking and command subsystems were reported to be off-the-shelf designs, but required significant modification. It should be noted that the contractor bus studies were concerned almost solely with mission performance requirements. The reports did not sufficiently define the subsystem design or component selections to provide a basis for an adequate penetration of heritage. The JPL Definition Phase Final Report nevertheless concluded that the existing bus approach had significant cost, schedule and risk advantages, and permitted a concentration of development efforts on the sensor system.

Midterm reports in May 1974 of the JPL and the Wallops Flight Center and Applied Physics Laboratory Definition Phase study groups demonstrated that neither the Baseline nor Alternate Payload Mission was achievable within the $58.2 million ceiling. The Wallops Flight Center and Applied Physics Laboratory's estimate for the Baseline Mission, which included an in-house designed spacecraft, was $85.2 million. At this point in time the Wallops Flight Center and the Applied Physics Laboratory adopted the sensor module/existing bus concept that JPL was pursuing. JPL's midterm estimate for the Alternate Payload Mission using the existing bus concept was $65.9 million.

The JPL and the Wallops Flight Center and Applied Physics Laboratory searched for ways to descope the project in order to stay within the cost ceiling. Each group performed a number of iterations wherein sensor performance and sensor combinations were varied in order to decrease the cost and yet meet the basic user requirements.

A final presentation of the JPL and Wallops Flight Center and Applied Physics Laboratory's Definition Phase studies to NASA Headquarters management in August 1974 resulted in a reduced baseline payload at the $58.2 million ceiling which eliminated the microwave radiometer and combined the altimeter and scatterometer into a single instrument, but which retained the synthetic aperture radar, as well as the visual and infrared radiometer.

SPACECRAFT REQUIREMENTS AND DOCUMENTATION

The two primary contractual documents on Seasat were the Satellite Vehicle Specification (Part I and Part II) and the Satellite Vehicle System Test Plan. There were 13 other documents which required JPL approval, but these were primarily implementation and operations type plans; i.e., Data Management Plan, Quality Assurance Plan, etc. One of these plans, the Reliability Assurance Plan, is relevant to this chapter and will be discussed herein.

Part I of the Satellite Vehicle Specification established the performance, design, development, and qualification requirements for the Seasat mission. Part II of the specification established the product configuration and system test acceptance requirements. This specification is similar to a typical Part I, Part II Contract End Item specification used for most NASA programs.

The Satellite Vehicle Systems Test Plan established the test program for assembling, testing, monitoring and operating the Seasat spacecraft from manufacturing through launch. The Satellite Vehicle Systems included all Lockheed and government furnished hardware installed in the Agena bus assembly and the sensor module. The test plan was the controlling test document and subordinate only to the Satellite Vehicle Specification. An evaluation was made regarding this flow of requirements and the interrelationships of Lockheed and JPL relative to control and the visibility of requirements.

Compliance with Requirements - During the Board's review, it was determined that a significant test required by the JPL approved test plan was not conducted. The Satellite Vehicle Test Plan required electronic assemblies to be subjected to eight cycles in thermal environment of which, as a minimum, two cycles should be in a vacuum chamber (acceptance test). The Slip Ring Assembly Component Specification, however, did not require a thermal vacuum test. This noncompliance was not recognized by JPL or Lockheed systems engineering until the present failure investigation was begun. Discussions with Lockheed and JPL personnel revealed that there was not a closed loop system to assure compliance with contractual requirements identified in the test plan.

The fact that a component specification that violated a contractual requirement could be issued is indicative of a lack of checks and balances in the system. Another indication of this lack surfaced in reviewing the qualification requirements. In at least two cases, to be discussed below, qualification requirements noncompliance was not documented. In fact, in the areas where the Board performed an in-depth evaluation, inconsistencies in requirements were noted in many cases. Most inconsistencies were minor; however, the impression left was that both compliance with requirements by Lockheed and the check and balance system at Lockheed and JPL were deficient.

Engineering Memoranda - Environmental derivations, test criteria and detailed test requirements were documented in engineering memoranda (EMs). Lockheed stated that EMs were used to allow early generation of requirements while the spacecraft design was being finalized. A considerable number of EMs were developed during the course of the Seasat program, and it accordingly became very difficult to establish a documentation trail as to how test requirements were established, modified, and satisfied. In fact, two particular incidents were uncovered during detailed evaluation into the qualification status of the electrical power subsystem components that point out the weakness of the EM system.

In one case, the Seasat environmental requirements specified a five minute per axis random vibration level but several components were qualified by similarity to a program that required only a three minute per axis vibration. This five minute per axis requirement was also specified in Part I of the Satellite Vehicle Specification. There was no documented evidence that this noncompliance was acceptable. In the second incident, pyro shock levels for Seasat were not enveloped by the program to which the Seasat slip ring assemblies were "qualified by similarity." While an EM stated that the slip ring assemblies are "not highly sensitive to pyro shock," there was no documentation or analysis to support the stated conclusion.

Because Seasat was a one-of-a-kind vehicle, Lockheed did not summarize the requirements contained in the various EMs into a single baseline document. A baseline document, with change control, would have been a systematic approach to assuring requirements were satisfied and would have provided a feedback mechanism to all parties. The large number of EMs produced in the Seasat program made it very difficult for Lockheed to use the EMs to manage the program and to assure continuity in requirements, as exemplified above, and equally difficult for JPL to effectively penetrate the system.

The Failure Modes, Effects and Criticality Analysis (FMECA) - The FMECA prepared for Seasat utilized the Fault Tree Analysis Technique. In effect, this was a method for studying the factors that could cause an undesired event to occur and inputting these factors into a computer model to which probability data could be applied to determine the most critical and probable sequence of events that could produce the undesirable event.

The Reliability Assurance Program Plan required that a FMECA be performed at the system level. Further evaluation revealed that "critical/new equipment" would also be subjected to an FMECA. Out of the 74 critical items identified on Seasat, only three were judged to require component level FMECAs. These were the command timing unit (CTU), the telemetry sensor unit (TSU) and the synthetic aperture radar (SAR) antenna (supplier performed).

The FMECA for the electrical power subsystem stated that there were "no single point failures" and listed a number of redundancies, including main bus power supply channels, batteries, charge controllers, and others. Electrical shorts were, however, *not* included as possible failure modes; almost all of the effort was directed toward consideration of failure modes that would result in loss of solar array power, and the only slip ring assembly failure mode considered was "slip ring contact failure." The lack of consideration of electrical shorts in effect prevented the FMECA from serving as a tool for directing attention to those portions of the system where electrical shorts could occur and led to the erroneous conclusions that there were no single point failure modes in the electrical power subsystem.

Component Specifications – Component specifications were used on Seasat to define the design, performance, acceptance, and qualification requirements of the major hardware items and subassemblies. Because the program intent was to utilize as much off-the-shelf hardware as possible, many existing specifications were redlined and updated for the Seasat Agena bus. These redlined specifications were then converted into component specifications by the responsible equipment engineers. After April 1976, a program directive established that all component specifications on Seasat required the signature approval of reliability engineering, of space technology, and of the chief systems engineer in addition to the responsible equipment engineer and the program engineer. Two specifications were released prior to April 1976 and never received the full complement of signature approvals. These two specifications were for the Slip Ring Assemblies and the Solar Array Drive Motors. Had the other three engineering organizations reviewed the specifications, quite possibly the Slip Ring Assembly thermal vacuum test deletion may have been prevented and inconsistencies in the qualification requirements may have been avoided. The component specifications were not reviewed and approved by JPL.

Qualification for Flight – The Seasat program used the classical methods of qualifying hardware for flight. These were:

a) Qualification by test to demonstrate the capability of an item to meet specification requirements.
b) Qualification by design similarity whereby an unqualified item is compared with an item qualified by test to determine whether the requirements for both items and their configurations are sufficiently similar to justify not testing the unqualified item.
c) Qualification by engineering analysis, independently or in conjunction with test and/or similarity, to meet a specific qualification in the specifications. The use of engineering analysis alone could not be used to satisfy all qualification requirements.

In September 1976, the Lockheed Seasat project issued a directive creating an Equipment Qualification Review Board for the purpose of reviewing and approving all qualification and design similarity certificates. The primary membership of the board included the program engineering managers, the chief systems engineer, the program reliability engineer, the quality assurance manager, and the applicable space technology manager. This Board met every two weeks to review the status of the qualification program and to determine what additional tasks were required to qualify a given item. Status reports were issued by program reliability engineering which tracked the qualification progress and documented open items.

The qualification cycle concluded with a meeting to review all test data, design similarity statements, engineering analyses, and individual component pedigree packages. Individual Certificates of Qualification were issued stating that the specific component had been qualified to the intended environment and was acceptable for flight. A JPL engineering representative attended these qualification review meetings but was not required to approve the qualification certificate. A JPL reliability representative attended approximately 25 percent of the review meetings.

Review of Build Paper – An evaluation of the Seasat "build" paper was made with primary attention focused on the electrical power subsystem. The review encompassed the electrical harness fabrication and installation, the "pedigree packages" on electrical components and assemblies, nonconformance reports on anomalies encountered in assembly and test, vehicle log books, and the vehicle acceptance summary.

Because the Board's failure analysis eventually identified the slip ring assembly as the component responsible for the Seasat failure, the detailed build paper associated with only this component will be discussed in the next section. However, some brief observations are presented below that deal with other findings made during the course of the investigation.

The nonconformance reports are used by Lockheed to document nonconforming conditions and resultant dispositions and correction actions. In general, the nonconformance report system at Lockheed was found to be acceptable. At the Board's request, Lockheed reviewed, cataloged, and summarized all electrical power subsystem nonconformance reports and made a conscious decision as to the possible effect of the anomaly in contributing to the Seasat failure. None of the nonconformances were judged to be contributory to the failure.

Evaluation of the spacecraft build paper of the electrical power subsystem indicated that the Air Force Plant Representative Office involvement, operating under delegation from JPL, was shallow. Inspection coverage was concentrated at the system level with few in-process mandatory inspection points.

Early negotiations surfaced the fact that the Air Force Plant Representative Office could provide neither the number of personnel nor the required skill levels to perform electronic inspections. As a result of these negotiations, JPL elected to send three JPL inspectors on extended temporary duty to perform 100 percent of the solder joint inspections and electronic component acceptance testing. While it cannot be stated that a more in-depth involvement by the government would have prevented the failure, it is the opinion of the Board that the depth of penetration was inappropriate and a more selective penetration would have been in order rather than a nearly total reliance on system level audits and shakedown inspections for the bus assembly operations.

SLIP RING HERITAGE

Consistent with the basic philosophy of the Seasat program to use, to the maximum extent possible, standard flight-proven equipment, the solar array drive motors and slip ring assemblies for Seasat were adapted from another Lockheed program. At the time of initial contract negotiations, this other Lockheed program had just developed a slip ring assembly and was in the process of performing qualification testing. This slip ring was also being considered for still other Lockheed programs and it was anticipated that the assembly would be a qualified and flight-proven design by the time Seasat was flown. As it turns out, however, the program for which the design was originally developed was canceled after completion of slip ring qualification but prior to flight; however, one other Lockheed program did fly a slip ring assembly of this design shortly before Seasat was launched. While the designs of the slip ring assembly for Seasat and this "previously flown" program were identical, the wiring sequence of the individual rings and brushes was different in the two programs. As noted earlier, the Seasat slip rings were wired such that most of the adjacent power brushes were of opposite DC polarity while the other Lockheed program was wired such that the adjacent power brushes had the same polarity. This difference in how the slip ring assemblies were connected into the electrical power subsystem thus became crucial to the heritage of the Seasat slip ring assembly; when the Seasat slip ring assembly became, in its application, connected in a manner that was different from its sole predecessor it became a unique, first of a kind component.

Two significant problems were noted as a result of random vibration testing of the slip ring assemblies used for the other Lockheed flight program. An isolation failure was found after vibration testing in two adjacent brush/ring circuits. The corrective action was to separate the brushes. Also, when the assembly was opened for this operation, a crack was noted in the brush mounting block at a mounting hole. This block was replaced on the failed unit and a "T" strengthener was added to all identical slip ring assemblies, including the Seasat units, to distribute the mounting loads away from the mounting point.

Failure History – Slip ring assemblies of the design flown by Seasat experienced two nonconformances that provide evidence of two separate failure mode possibilities. One of these was the isolation failure noted above on the other Lockheed flight program that was indicative of a possible failure mode due to contact between adjacent brushes of opposite polarity. Another failure mode identified on one of the Seasat assemblies was caused by shorting of a wire to ground due to cold flow of the Teflon insulation in the region where high stresses were imposed on the wire. This incident will be described later.

Considerable evidence exists in published reports that the sliding friction between brushes and rings will generate debris particles that can accumulate and produce electrical noise or, in some cases, short circuits between adjacent rings and brushes. Lockheed experienced a shorting failure in a slip assembly used in ground tests of a control moment gyro prior to June 1977, which was attributed to accumulation of brush-generated debris and subsequent arcing between adjacent power brushes. Discussion with engineering personnel from TRW, Ball Corporation, and Sperry Flight Systems have indicated that other aerospace contractors have experienced similar slip ring shorts in ground tests. As a result of their experience with slip rings, Sperry initiated an experimental study of the possible effects of debris. While the Board recognizes that there are significant differences between the design and application of the Seasat slip ring assembly and these other units, experience illustrates a third possible failure mode due to shorting caused by contaminants or debris within the assembly.

Seasat Slip Ring History – A portion of the build history of components is assembled by Lockheed into pedigree packages. These packages contain component drawings, a component specification including acceptance and qualification test requirements, nonconformance reports, and some vendor documentation including specified testing and plans test records. Component selection for pedigree packages was determined by the Seasat Program Office and the quality assurance organization at Lockheed. The Seasat slip ring assemblies are documented by such pedigree packages. Relevant component history not contained in the slip ring pedigree packages include vendor assembly and test nonconformance reports (including failure reports), assembly test procedures and records (including brush alignments and pressure checks and brush "run-in" procedures), and relevant vendor and customer correspondence.

The timing of the Seasat contract was such that Lockheed was able to acquire two partially assembled slip ring assemblies when another Lockheed program referred to herein as Program A, was canceled. Program A had initially contracted for 10 assemblies and, at the time of termination, had accepted delivery of one qualification unit, one development unit, and two production units leaving six partially assembled units at the vendor. The Seasat program picked up two of these units and Lockheed Program B picked up the additional units. Reference will be made to Program B in other portions of this report relative to test experience and use of Program B qualification testing as a basis for qualifying the Seasat slip rings by similarity.

Program A personnel were informed by Poly-Scientific in late 1973 that the constraints placed upon the length of the assembly were found to be restrictive and that relief of the specifications would enhance reliability. Program A, however, could not relax the specification. Although the Seasat application was not constrained by length, the program desire to use available off-the-shelf hardware precluded the development of a new unit having increased dimensional tolerances between the rings and brush assemblies with possibly enhanced inherent reliability.

Seasat personnel initiated discussions with Poly-Scientific in late 1975 using the Lockheed Program A specification as a baseline. On February 3, 1976, Poly-Scientific submitted its first written quote for two assemblies to be fabricated and tested per the Program A specification. This initial quote was not acceptable to Lockheed, and the responsible equipment engineer and buyer responded on March 5, 1976, with a Seasat redlined version of the Program A specification. It was in this March 5, 1976, specification that the Program A requirement for 10 cycles of thermal vacuum acceptance testing was deleted. This deletion occurred even though: (1) the majority of the Seasat electronic assemblies and electromechanical assemblies were subjected to a thermal vacuum acceptance test; (2) Seasat reliability and systems engineering personnel, and JPL personnel were unaware of this deletion until the present failure investigation; and (3) the thermal vacuum test was contractually required and a waiver of the requirement was never issued.

Upon pursuing the thermal vacuum deletion further, it was determined from interviews with involved personnel that the test was deleted during verbal negotiations between both the responsible equipment engineer and the buyer at Lockheed, and the vendor in order to reduce unit cost of the slip ring assemblies. The responsible Lockheed program engineer approved the deletion but, at that time, there was no requirement to coordinate specifications with the Seasat program reliability engineer or the chief systems engineer. The fact that a waiver was not issued on this and other contract noncompliances is indicative of a weak compliance system between Lockheed and JPL.

On March 25, 1976, Lockheed issued a formal Request for Quote to Poly-Scientific for two Seasat slip ring assemblies built to the March 5, 1976 specification with a requested delivery date of one year. On May 26, 1976, Lockheed authorized contract go ahead for two slip ring assemblies at a unit price of $8,953.50.

Researching the manufacturing history and fabrication and test anomalies at Poly-Scientific resulted in the following:

a) There were four anomalies noted on slip ring unit 1001. Three were minor and appear to have had no real impact on assembly reliability. The fourth anomaly was a Teflon wire short to an adjacent ground lug. The repair action, approved by Lockheed engineering, was to insulate the ground terminal and repot with ES 222-2 cement. The damaged insulation on the wire was not repaired. This discrepancy report was not included in the vendor's data package and consequently this failure was not contained in the Lockheed pedigree package.

b) Slip Ring Unit 1002 (-Y solar array) had the more significant anomalies noted during fabrication and test. These anomalies are summarized as follows:
   1) 9/20/76 - 80 minute run-in of brushes to rings at 100± 10 rpm. Run-in time should have been for 100 to 115 minutes. This discrepancy was missed and not documented.
   2) 9/23/76 - discrepancy No. 146522 - discolored rings noted after above run-in test. Unit had to be completely disassembled, brushes and rings recleaned, unit reassembled and another run-in performed. The exact run-in time was not recorded nor entered into the log book.
   3) 11/12/76 - discrepancy No. 151887 - excessive noise noted caused by moisture pick-up in the brush material. Corrective action was to run the unit in vacuum at 14.4 rpm for $1\frac{1}{2}$ hours. No vacuum cleanup was performed after this 14.4 rpm run-in test. This run time was not entered into the log book.

c) Review of vendor documentation and subsequent teleconferences with Poly-Scientific personnel revealed the following assembly technique and procedures:
   1) The assembly planning documentation specified that the brushes were to be aligned "in center of the rings." This requirement was verified visually by the inspector, but no dimensional checks were made. Proper alignment of the brushes is dependent, therefore, on the inspector's judgment.
   2) Poly-Scientific stated that the tolerances within the slip ring assembly could allow adjacent brushes to touch. It is noted here that an identical slip ring assembly experienced an isolation failure during acceptance testing which was probably caused by adjacent brushes touching. (Program B hardware).

Both Seasat slip ring assemblies were shipped from Poly-Scientific on February 22, 1977. These units were received and accepted at Lockheed on March 11, 1977, where they remained in storage until required for installation on their respective solar array modules.

In approximately July 1977, Lockheed Program B, which utilized identical slip ring assemblies, made a wiring change external to the slip rings that separated the polarity arrangement of adjacent slip rings. By changing connector pin functions, the power applied to individual rings was changed from a configuration in which adjacent rings were of opposite polarity to one having positive contacts on one end of the slip ring assembly and negative contacts on the opposite end. This wiring change significantly reduced the possibility of internal shorts within the slip ring assembly.

The Seasat chief system engineer was contacted by a system engineer from Program B about this change in wiring in August 1977. The explanation given for the wiring change was a concern that the ascent vibration environment could cause adjacent brushes to make contact and thus produce an electrical short because Program B slip rings had power applied during launch. The chief system engineer discussed this change with the Seasat program engineer and they decided not to make a similar wiring change because Seasat did not see the same launch vibration levels and because Seasat slip rings were not planned to be powered during launch. It is noted that in April 1978, a change in launch relay configuration was made which did apply power to the slip ring assemblies. In retrospect, the decision not to change the wiring sequence for Seasat was a crucial one. When the other program changed its wiring and Seasat did not, Seasat became the first program to fly a 52-brush slip ring assembly with adjacent brushes of opposite polarity. Had there been better visibility to the problems experienced with slip rings by both the vendor and by other organizations within Lockheed, the Seasat engineering managers may have been more sensitive to the failure prone nature of this complicated device and to the importance of the electrical polarity of adjacent brushes. Unfortunately, such visibility, which may only have needed to have been slight to have been effective, was lacking.

Slip Ring Assembly serial number 1002 was installed on the -Y solar array module on August 17, 1977. On August 30, 1977, a nonconformance report was written because the mechanic "lost" an undetermined number of shim washers.

Review of the installation drawing revealed that four number 10 washers were required between the solar array mounting structure and the slip ring assembly. The cover of the assembly is made of thin sheet metal and is prone to bow up during installation operations. Because the mounting bolts go through the cover plate into the threaded holes in the slip ring body, the mechanic had to place the round washers over the bolts between the structure and the cover plate. It was during this operation that the mechanic lost the washers. The S/N 1002 slip ring assembly was removed from the solar array module, the cover plate removed and three washers were found. Because some areas were still obscured, an x-ray of the slip ring was taken. No additional washers were located. A nonconformance report was then written against Slip Ring Assembly 1001 and no washers were found by either visual or x-ray inspection. It is interesting to note two things: (1) there were no downstream electrical functional checks after installation of the slip ring assembly which could have detected missing washers in the slip rings, and (2) it was never conclusively determined if all lost washers were found.

The solar array modules, including the slip ring assemblies, were shipped to the launch site in April 1978. The last reported anomaly on the slip rings was high contact resistance on unit 1002 during interface tests performed when the solar array modules were mated to the vehicle. The resistance reading recorded was 2.38 ohms; the specification value was 2.00 ohms maximum. The engineering disposition in the nonconformance report was "use-as-is" because inflight operation would decrease the contact resistance.

SIGNIFICANT FINDINGS

1) The spacecraft failure that occurred on October 9, 1978, was due to a loss of electrical power in the Agena bus as a result of a massive and progressive electrical short within the slip ring assembly of the -Y solar array.
2) The electrical short was most probably initiated by an arc between adjacent components in the slip ring assembly. Possible triggering mechanisms for this arc are momentary shorts caused by wire-to-brush assembly contact, brush-to-brush contact, or by a contaminant.
3) The congested nature of the slip ring design, coupled with a wiring arrangement for connecting the slip rings into the power subsystem that resulted in most of the adjacent brush assemblies being of opposite polarity, made the Seasat slip ring assembly particularly prone to shorting.
4) The combination of design and wiring sequence used for the Seasat slip ring assemblies made these unique, first-of-a-kind components.
5) The possibility of slip ring failures resulting from placing opposite electrical polarities on adjacent brush assemblies was known at least as early as the summer

1977 to other projects within the prime contractor's organization. That the Seasat organization was not fully aware of these potential failure modes was due to a breakdown in communications within the contractor's organization.

6) The failure to recognize the potential failure modes of the slip ring assembly and to give this critical component the attention it deserved was due, in part, to the underlying program policy and pervasive view that it was an existing component of a well-proven and extensively used standard Agena bus. This program policy further led to a concentration by project management on the sensors and sensor module of the spacecraft to the near exclusion of the bus subsystems. In actuality, many of these subsystems, including the power subsystem, contained components that were neither flight proven nor truly qualified by similarity.
7) Lack of proper attention by both Lockheed and JPL Seasat program engineering to the new and unproven components on the Agena bus resulted in several instances of both noncompliance with contractual, qualification and acceptance requirements and failure to document such noncompliances.
8) The Failure Modes, Effects, and Criticality Analysis that was conducted for the electrical power subsystem did not consider shorts as a failure mode and thus did not reveal the presence of single point failure modes in the subsystem nor provide a basis for the development of a full complement of safing command sequences that could be used by the flight controllers in responding to anomalies.
9) The strong desire on the part of all concerned to initiate the project as soon as possible resulted in inadequate time for an effective Phase B study. As a result, the project office did not have the opportunity to plan the activity thoughtfully and establish the preliminary designs, component evaluations, test plans, and other Phase B project plans before becoming engaged in the actual spacecraft development.

Although unrelated to the failure of the Seasat, certain deficiencies in flight control procedures were present that are worthy of note as a lesson for the future. The flight controllers were not provided with an adequate set of safing command sequences to use in response to anomalies, were not sufficiently familiar with the system they were controlling, received insufficient anomaly training and, during the failure event itself, failed to follow the prescribed procedures in response to the flight data available to them. Compounding these difficulties were the frequent breakdowns of the ground data acquisition and processing system throughout the mission.

It is ironic, and yet typical, of spacecraft failures that the termination of the Seasat flight was caused not by a malfunction of a new or sophisticated device, but by a failure in a very common component of a type that has flown in many spacecraft for many years. It is also ironic, and instructive, that the smallest of events or the slightest of communications could have prevented the failure. Better clarity in an oral communication, a brief memorandum of the right kind at the right time, a failure report coming to the right person, or an alert engineer could have made all the difference.

Basic to the Seasat mission was the concept of using an existing, flight-proven spacecraft bus for the services and housekeeping functions required by the sensors in order to minimize program costs and to permit a concentration of effort on the sensors and their integration into the spacecraft. Thus the use of a "standard Agena bus" as part of the Seasat spacecraft became an enduring tenet of the program. So firmly rooted was this principle in program philosophy that, even after the engineering staffs of both the government and the contractor were well aware of the final uniqueness of their Agena bus, the words, and the associated way of doing business, persisted. They became deceived by their own words.

Consistent with the concept of the "standard Agena bus" was the policy decision to minimize testing and documentation, to qualify components by similarity wherever possible and to minimize the penetration into the Agena bus by the government. As a result, a test was waived without proper approval, important component failures were not reported to project management, compliance with specifications was weak, and flight controllers were inadequately prepared for their task. Significantly, the Seasat slip ring assembly had no applicable flight history at the time of its launch and, in its application to the spacecraft, was a new device.

There can, of course, be no quarrel with the policy of using existing and well proven equipment. The use of such equipment has certainly reduced the costs and contributed to the success of many space missions. But the world of space flight is an unforgiving one and words like "standard," "existing," and "similar to" can be traps for the unwary. The technical risks of using standard equipment can be as high as those present in a new or untried piece of equipment, but the approach, both technical and managerial, must be different. For new equipment, one designs carefully, reviews thoroughly, and tests completely — and that we know how to do. For standard equipment, one should diligently and thoroughly probe the heritage that justifies the classification and identify, component by component and piece by piece, those that are truly standard and those that are not. One should assume that each space vehicle is unique until proven otherwise. Then, for those parts that are standard or well proven, and that are applied in the same way, one can forego design, reviews, testing and extensive documentation. Conversely, components that are different should be treated as new. The policy of limited penetration into Seasat's Agena bus by the government was appropriate, but a limited penetration must be a selective penetration and not a reduced effort everywhere.

This identification of the heritage of previously used equipment, in both design and application, need not require a large staff or a lot of money. But it does take time, both at the start of the project and at the time of the Critical Design Review. And here, responding to strong desires by all concerned to get the project on contract and underway, the Seasat project was denied the advantage of an effective Phase B study. Had there been an effective Phase B study period, preliminary designs would have been completed, component selections better understood, test plans and qualification requirements better established, and possibly, the critical role and inherent complexities of the slip ring assembly might have been more apparent to the Seasat engineering staffs. Whether such a Phase B study period would have precluded the Seasat failure is, of course, uncertain for history does not reveal its alternatives. But such a carefully conducted planning and study period would have minimized the chances for the type of failure that did occur.

The policy of using existing, flight-proven equipment can be both valid and cost effective. But it is the main lesson of Seasat that an uncritical acceptance of such classifications as "standard" can submerge important differences from previously used equipment in both design and in application. It is important, therefore, that thorough planning be conducted at the start of a project to fully evaluate the heritage of such equipment, to identify those that are standard and those that are not, and to establish project plans and procedures that enable the system to be penetrated in a selective manner.

Omit to end

DEFINING SYSTEMS ENGINEERING

by George S. Trimble

Editors' Note

Back on September 27, 1968, a NASA engineer by the name of George S. Trimble wrote to the Chief of the Management Analysis and University Programs Office after the Chief issued a letter to find a universally suitable definition for "systems engineer." The engineer told the manager that the term had no particular meaning at all. "In fact," Trimble claimed, "I may know the guy who thought it up or resurrected it, as the case may be, for modern usage." His seemingly authoritative account follows:

During the war, new management practices were introduced at a great rate, and one of the functions that came to the fore was the business of writing job descriptions and evaluating them. Certain industrial relations experts fell heir to this function, and there was a tendency for them to write very clear job descriptions for all jobs except their own. It soon became obvious that the value of a job, or, more importantly, the money it paid (or even more importantly, its draft-dodging power), was inversely proportional to the ease with which one could describe it. Industrial relations people were able to describe any engineering job in 25 words or less, whereas an industrial relations function might take two or three pages. Miserable to begin with, engineering salaries were futher threatened and so was draft status.

Of course, everyone knows that engineers are very creative. They could see that the industrial relations boys had a good thing going, so they borrowed the approach and improved on it (typical engineering method).

Soon it took five pages to describe the most menial engineering task, and the engineers were saved. It was a simple matter to spend three hours explaining to a job analyst from industrial relations why a 'systems engineering' blueprint file was much more complicated to run than a simple old 'engineering' blueprint file, which was, of course, familiar. The guy from industrial relations never did understand it because the guy who explained it, didn't. It takes a lot of words to explain something you don't understand or that isn't there. Try explaining 'zero' sometime.

A parallel effort with the objective of emphasizing *!!ENGINEERING!!* was carried out with great dispatch by the 'scientists,' all of whom became famous at the close of WWII because a couple of them invented and built the A-bomb, all by themselves, with great secrecy. What they were really doing all that time, of course, wasn't science—it was engineering. When this was discovered, a mixed wave of nausea and terror ran through the brotherhood. It was worse than being caught reading a dirty book in church. Most learned scientists knew that engineers were people who ran around with special hats and oil cans and made steam locomotives go, and who, incidentally, made too much money. Being identified as part of the same crowd was too much for the intellectuals to bear. Scientists had to be working on something more important than 'engineering,' which is supervised by a Ph.D and is therefore high-class and also obvious to those schooled properly, but difficult if not impossible for anybody else to understand.

Since, as we all know, very few, if any, Ph.Ds understand the meaning of plain, ordinary 'engineering,' it follows that 'systems engineering' has given engineering a bad name, and should be avoided for that reason alone.

A third group who helped the cause for systems engineering were the pre-war 'handbook' engineers who discovered creative engineering when they joined up with a wartime industrial engineering group to avoid being drafted. They had always thought that engineering was the *choosing* from a catalog of the proper washer for a quarter-inch bolt. It was difficult for them to use the same name for their new discovery, creative engineering (*designing* a washer for a quarter-inch bolt). The term 'systems engineering' suited well, and groups of people were noising it around by then. It sounded nice and, after all, a quarter-inch bolt is a ***fastening*** system of high complexity. It consists of a bolt with threads (helical inclined plane), a nut of the proper size, hand and thread configuration (bolt interface problem), external shape (wrench interface problem), one or more washers (structures interface problem), and sometimes even a cotter pin (reliability).

Moreover, one could dream of performing systems engineering at increased hierarchical levels by considering at one and the same time not only the quarter-inch bolt, but also the half-inch bolt. Advanced systems engineering.

So much for the history and meaning of systems engineering. You can demonstrate the validity of my story to yourself in several ways. Your letter, for instance, can be clarified by eliminating the word 'systems.' I believe it appears 10 times. Check the universities for courses in systems engineering and find out what they are really teaching. Note also that the term 'systems engineering' does not yet appear in an accredited dictionary. This is because Webster cannot figure it out either. Good luck!