NASA SYSTEMS ENGINEERING HANDBOOK

National
Aeronautics and
Space
Administration

SP-6105
June 1995

NASA
Systems Engineering Handbook

by
Robert Shishko, Ph.D.

with contributions by

Robert Aster	**Beth Bain**	**Guy Beutelschies**
Robert G. Chamberlain	**Renee I. Cox**	**Kevin Forsberg, Ph.D.**
Patrick McDuffee	**Harold Mooz**	**Mary Beth Murrill**
Les Pieniazek, Ph.D.	**Lou Polaski**	**Neil Rainwater**
Tom Rowell	**Mark Sluka**	**Ron Wade**

edited by
Randy Cassingham

graphics by
Stephen Brewster
John Matlock

Contents

List of Figures

List of Sidebars

Foreword

In an attempt to demonstrate the potential dangers of relying on purely "cookbook" logical thinking, the mathematician/philosopher Carl Hempel posed a paradox. If we want to prove the hypothesis "All ravens are black," we can look for many ravens and determine if they all meet our criteria. Hempel suggested changing the hypothesis to its logical contrapositive (a rewording with identical meaning) would be easier. The new hypothesis becomes: "All nonblack things are nonravens." This transformation, supported by the laws of logical thinking, makes it much easier to test, but unfortunately is ridiculous. Hempel's raven paradox points out the importance of common sense and proper background exploration, even to subjects as intricate as systems engineering.

In 1989, when the initial work on the NASA Systems Engineering Handbook was started, there were many who were concerned about the dangers of a document that purported to teach a generic NASA approach to systems engineering. Like Hempel's raven, there were concerns over the potential of producing a "cookbook" which offered the illusion of logic while ignoring experience and common sense. From the tremendous response to the initial (September 1992) draft of the handbook (in terms of both requests for copies and praise for the product), it seems early concerns were largely unfounded and that there is a strong need for this handbook.

The document you are holding represents what was deemed best in the original draft and updates information necessary in light of recommendations and changes within NASA. This handbook represents some of the best thinking from across NASA. Many experts influenced its outcome, and consideration was given to each idea and criticism. It truly represents a NASA-wide product and one which furnishes a good overview of NASA systems engineering.

The handbook is intended to be an educational guide written from a NASA perspective. Individuals who take systems engineering courses are the primary audience for this work. Working professionals who require a guidebook to NASA systems engineering represent a secondary audience.

It was discovered during the review of the draft document that interest in this work goes far beyond NASA. Requests for translating this work have come from international sources, and we have been told that the draft handbook is being used in university courses on the subject. All of this may help explain why copies of the original draft handbook have been in short supply.

The main purposes of the NASA Systems Engineering Handbook are to provide: 1) useful information to system engineers and project managers, 2) a generic description of NASA systems engineering which can be supported by center-specific documents, 3) a common language and perspective of the systems engineering process, and 4) a reference work which is consistent with NMI 7120.4/NHB 7120.5. The handbook approaches systems engineering from a systems perspective, starting at mission needs and conceptual studies through operations and disposal.

While it would be impossible to thank all of the people directly involved, it is essential to note the efforts of Dr. Robert Shishko of the Jet Propulsion Laboratory. Bob was largely responsible for ensuring the completion of this effort. His technical expertise and nonstop determination were critical factors to ensure the success of this project.

Mihaly Csikzenthmihali defined an optimal experience as one where there is "a sense of exhilaration, a deep sense of enjoyment that is long cherished and becomes a landmark in memory of what life should be like." I am not quite sure if the experience which produced this handbook can be described exactly this way, yet the sentiment seems reasonably close.

— **Dr. Edward J. Hoffman**
Program Manager, NASA Headquarters
Spring 1995

Foreword to the September 1992 Draft

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

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

It was thought that the best way to reflect the totality of the NASA systems engineering process and to aid in developing the needed training was to prepare a top level (Level 0) document that would contain a broad definition of systems engineering, a broad process outline, and typical tools and procedures. In general, we wanted a top level overview of NASA systems engineering. To this document would be appended each Center's unique systems engineering manual. The group was well aware of the diversity each Center may have, but agreed that this approach would be quite acceptable.

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

During the development of the manual we decided to put in much more than may be appropriate for a Level 0 document with the idea that we could always refine the document later. It was more important to capture the knowledge when we could in order to better position ourselves for later dissemination. If there is any criticism, it may be the level of detail contained in the manual, but this detail is necessary for young engineers. The present document does appear to serve as a good instructional guide, although it does go well beyond its original intent.

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

— Francis T. Hoban
Program Manager, NASA Headquarters

Preface

This handbook was written to bring the fundamental concepts and techniques of systems engineering to NASA personnel in a way that recognizes the nature of NASA systems and the NASA environment. The authors readily acknowledge that this goal will not be easily realized. One reason is that not everyone agrees on what systems engineering is, nor on how to do it. There are legitimate differences of opinion on basic definitions, content, and techniques. Systems engineering itself is a broad subject, with many different aspects. This initial handbook does not (and cannot) cover all of them.

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

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

By design, the handbook covers some topics that are also taught in Project Management/Program Control (PM/PC) courses, reflecting the unavoidable connectedness of these three domains. The material on the NASA project life cycle is drawn from the work of the NASA-wide Systems Engineering Working Group (SEWG), which met periodically in 1991 and 1992, and its successor, the Systems Engineering Process Improvement Task (SEPIT) team, which met in 1993 and 1994. This handbook's project life cycle is identical to that promulgated in the SEPIT report, *NASA Systems Engineering Process for Programs and Projects,* JSC-49040. The SEPIT project life cycle is intentionally consistent with that in NMI 7120.4/NHB 7120.5 (*Management of Major System Programs and Projects*), but provides more detail on its systems engineering aspects.

This handbook consists of five core chapters: (1) systems engineering's intellectual process, (2) the NASA project life cycle, (3) management issues in systems engineering, (4) systems analysis and modeling issues, and (5) engineering specialty integration. These core chapters are supplemented by appendices, which can be expanded to accommodate any number of templates and examples to illustrate topics in the core chapters. The handbook makes extensive use of sidebars to define, refine, illustrate, and extend concepts in the core chapters without diverting the reader from the main argument. There are no footnotes; sidebars are used instead. The structure of the handbook also allows for additional sections and chapters to be added at a later date.

Finally, the handbook should be considered only a starting point. Both NASA as a systems engineering organization, and systems engineering as a discipline, are undergoing rapid evolution. Over the next five years, many changes will no doubt occur, and some are already in progress. NASA, for instance, is moving toward implementation of the standards in the International Standards Organization (ISO) 9000 family, which will affect many aspects of systems engineering. In systems engineering as a discipline, efforts are underway to merge existing systems engineering standards into a common *American National Standard on the Engineering of Systems,* and then ultimately into an international standard. These factors should be kept in mind when using this handbook.

Acknowledgements

I would like to thank Mr. Frank Hoban and Dr. Ed Hoffman, NASA Headquarters/Code FT, for their steadfast financial and intellectual support for this effort, and Dr. Shahid Habib, NASA Headquarters/Code QW for additional financial support to the NASA-wide Systems Engineering Working Group and Systems Engineering Process Improvement Task team. I would also like to acknowledge the participation of many other individuals, both in and out of NASA, who served as contributing authors and reviewers or who shared their systems engineering ideas and material with me during this effort. Special acknowledgements are due to Robert G. Chamberlain and Donna Shirley, NASA/Jet Propulsion Laboratory, who helped set the direction for the handbook. I, however, accept all responsibility for its content.

As contributing authors:
Mr. Robert Aster, NASA/Jet Propulsion Laboratory
Ms. Beth Bain, Lockheed Missiles and Space Company
Mr. Guy Beutelschies, NASA/Jet Propulsion Laboratory
Mr. Robert G. Chamberlain, NASA/Jet Propulsion Laboratory
Ms. Renee I. Cox, NASA/Marshall Space Flight Center
Dr. Kevin Forsberg, Center for Systems Management
Mr. Patrick McDuffee, NASA/Marshall Space Flight Center
Mr. Harold Mooz, Center for Systems Management
Ms. Mary Beth Murrill, NASA/Jet Propulsion Laboratory
Dr. Les Pieniazek, Lockheed Engineering and Sciences Co.
Mr. Lou Polaski, Center for Systems Management
Mr. Neil Rainwater, NASA/Marshall Space Flight Center
Mr. Tom Rowell, NASA/Marshall Space Flight Center
Mr. Mark Sluka, Lockheed Engineering and Sciences Co.
Mr. Ron Wade, Center for Systems Management

As SEPIT team members:
Ms. Beth Bain, Lockheed Missiles and Space Co.
Mr. Randy Fleming, Lockheed Missiles and Space Co.
Mr. Tony Fragomeni, NASA/Goddard Space Flight Center
Dr. Shahid Habib, NASA Headquarters/Code QW
Mr. Henning Krome, NASA/Marshall Space Flight Center
Mr. William C. Morgan, NASA/Johnson Space Center
Dr. Les Pieniazek, Lockheed Engineering and Sciences Co.
Dr. Mike Ryschkewitsch, NASA/Goddard Space Flight Center
Mr. Gerry Sadler, NASA/Lewis Research Center
Mr. Mark Sluka, Lockheed Engineering and Sciences Co.
Mr. Dick Smart, Sverdrup Technology, Inc.
Dr. James Wade, NASA/Johnson Space Center
Mr. Milam Walters, NASA/Langley Research Center
Mr. Don Woodruff, NASA/Marshall Space Flight Center

As general contributors:
Mr. Dave Austin, NASA Headquarters/Code DSS
Mr. Phillip R. Barela, NASA/Jet Propulsion Laboratory
Mr. J.W. Bott, NASA/Jet Propulsion Laboratory
Dr. Steven L. Comford, NASA/Jet Propulsion Laboratory
Ms. Sandra Dawson, NASA/Jet Propulsion Laboratory
Dr. James W. Doane, NASA/Jet Propulsion Laboratory
Mr. William Edminston, NASA/Jet Propulsion Laboratory
Mr. Charles C. Gonzales, NASA/Jet Propulsion Laboratory
Dr. Jairus Hihn, NASA/Jet Propulsion Laboratory
Dr. Ed Jorgenson, NASA/Jet Propulsion Laboratory
Mr. Richard V. Morris, NASA/Jet Propulsion Laboratory
Mr. Tom Weber, Rockwell International/Space Systems

As reviewers and commenters:
Mr. Robert C. Baumann, NASA/Goddard Space Flight Center
Mr. Chris Carl, NASA/Jet Propulsion Laboratory
Dr. David S.C. Chu, Assistant Secretary of Defense/Program Analysis and Evaluation
Mr. M.J. Cork, NASA/Jet Propulsion Laboratory
Dr. Frank Fogle, NASA/Marshall Space Flight Center
Mr. John L. Gasery, Jr., NASA/Stennis Space Center
Mr. Don Hedgepeth, NASA/Langley Research Center
Mr. Jim Hines, Rockwell International/Space Systems
Dr. Jerry Lake, Defense Systems Management College
Mr. Jim Lloyd, NASA Headquarters/Code QS
Dr. Brian Mar, Department of Civil Engineering, University of Washington
Dr. Ralph F. Miles, Jr., Center for Safety and System Management, University of Southern California
Mr. Bernard G. Morais, Synergistic Applications, Inc.
Mr. Ron Moyer, NASA Headquarters/Code QW
Mr. Raymond L. Nieder, NASA/Johnson Space Center
Mr. Leo Perez, NASA Headquarters/Code QP
Mr. David Pine, NASA Headquarters/Code B
Mr. Glen D. Ritter, NASA/Marshall Space Flight Center
Dr. Arnold Ruskin, NASA/Jet Propulsion Laboratory and University of California at Los Angeles
Ms. Donna Shirley, NASA/Jet Propulsion Laboratory
Mr. Don Sova, NASA Headquarters/Code AE
Mr. Lanny Taliaferro, NASA/Marshall Space Flight Center

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

1 Introduction

1.1 Purpose

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

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

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

1.2 Scope and Depth

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

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

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

Selected Systems Engineering Reading

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

Fundamentals of Systems Engineering
Systems Engineering and Analysis (2nd ed.), B.S. Blanchard and W.J. Fabrycky
Systems Engineering, Andrew P. Sage
An Introduction to Systems Engineering, J.E. Armstrong and Andrew P. Sage

Management Issues in Systems Engineering
Systems Engineering, EIA/IS-632
IEEE Trial-Use Standard for Application and Management of the Systems Engineering Process, IEEE Std 1220-1994
Systems Engineering Management Guide, Defense Systems Management College
System Engineering Management, B.S. Blanchard
Systems Engineering Methods, Harold Chestnut
Systems Concepts, Ralph Miles, Jr. (editor)
Successful Systems Engineering for Engineers and Managers, Norman B. Reilly

Systems Analysis and Modeling
Systems Engineering Tools, Harold Chestnut
Systems Analysis for Engineers and Managers, R. de Neufville and J.H. Stafford
Cost Considerations in Systems Analysis, Gene H. Fisher

Space Systems Design and Operations
Space Vehicle Design, Michael D. Griffin and James R. French
Space Mission Analysis and Design (2nd ed.), Wiley J. Larson and James R. Wertz (editors)
Design of Geosynchronous Spacecraft, Brij N. Agrawal
Spacecraft Systems Engineering, Peter W. Fortescue and John P.W. Stark (editors)
Cost-Effective Space Mission Operations, Daryl Boden and Wiley J. Larson (editors)
Reducing Space Mission Cost, Wiley J. Larson and James R. Wertz (editors)

2 Fundamentals of Systems Engineering

2.1 Systems, Supersystems, and Subsystems

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

A Hierarchical System Terminology

The following hierarchical sequence of terms for successively finer resolution was adopted by the NASA-wide Systems Engineering Working Group (SEWG) and its successor, the Systems Engineering Process Improvement Task (SEPIT) team:

System
Segment
Element
Subsystem
Assembly
Subassembly
Part

Particular projects may need a different sequence of layers — an instrument may not need as many layers, while a broad initiative may need to distinguish more layers. Projects should establish their own terminology. The word *system* is also used within NASA generically, as defined in the text. In this handbook, "system" is generally used in its generic form.

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

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

The NASA management instruction for the acquisition of "major" systems (NMI 7120.4) defines a *program* as "a related series of undertakings that continue over a period of time (normally years), which are designed to pursue, or are in support of, a focused scientific or technical goal, and which are characterized by: design, development, and operations of systems." Programs are managed by NASA Headquarters, and may encompass several projects.

In the NASA context, a *project* encompasses the design, development, and operation of one or more systems, and is generally managed by a NASA field center.

Headquarters' management concerns include not only the engineering of the systems, but all of the other activities required to achieve the desired end. These other activities include explaining the value of programs and projects to Congress and enlisting international cooperation. The term *mission* is often used for a program/pro-

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

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

ject's purpose; its connotations of fervor make it particularly suitable for such political activities, where the emotional content of the term is a desirable factor. In everyday conversation, the terms "project," "mission," and "system" are often used interchangeably; while imprecise, this rarely causes difficulty.

2.2 Definition of Systems Engineering

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

Systems Engineering per EIA/IS-632

Systems engineering is "an interdisciplinary approach encompassing the entire technical effort to evolve and verify an integrated and life-cycle balanced set of system people, product, and process solutions that satisfy customer needs. Systems engineering encompasses (a) the technical efforts related to the development, manufacturing, verification, deployment, operations, support, disposal of, and user training for, system products and processes; (b) the definition and management of the system configuration; (c) the translation of the system definition into work breakdown structures; and (d) development of information for management decision making."

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

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

2.3 Objective of Systems Engineering

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

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

Cost

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

Effectiveness

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

Cost-Effectiveness

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

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

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

Design trade studies, an important part of the systems engineering process, often attempt to find designs that provide a better combination of the various dimensions of cost and effectiveness. When the starting point for a design trade study is inside the envelope, there are alternatives that reduce costs without decreasing any aspect of effectiveness, or increase some aspect of effectiveness without decreasing others and without increasing costs. Then, the system manager's or system engineer's decision is easy. Other than in the sizing of subsystems, such "win-win" design trades are uncommon, but by no means rare. When the alternatives in a design trade study, however, require trading cost for effectiveness, or even one dimension of effectiveness for another at the same cost, the decisions become harder.

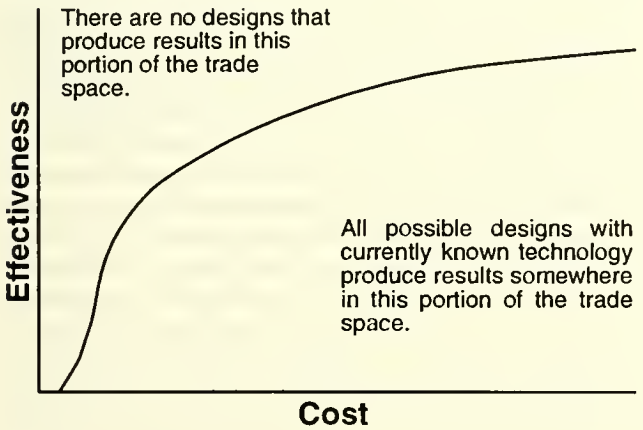


Figure 1 — The Enveloping Surface of Non-dominated Designs.

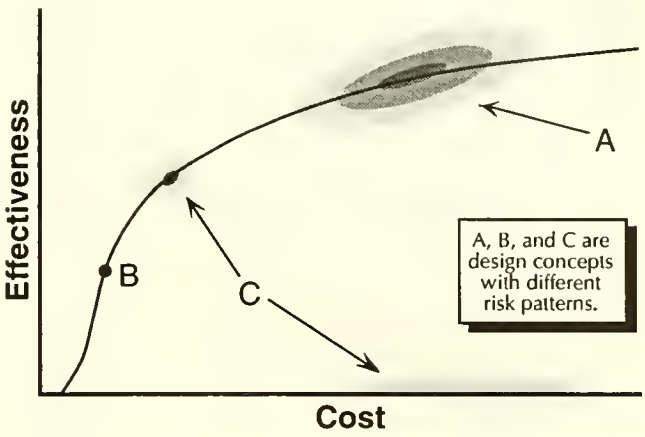


Figure 2 — Estimates of Outcomes to be Obtained from Several Design Concepts Including Uncertainty.

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

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

of its effectiveness from the fact that it was a "first"; a mission to Mars that misses its launch window has to wait about two years for another opportunity — a clear schedule constraint. *Risk* results from uncertainties in realized effectiveness, costs, timeliness, and budgets.

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

The System Engineer's Dilemma

At each cost-effective solution:

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

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

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

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

2.4 Disciplines Related to Systems Engineering

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

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

Systems engineering differs from what might be called *design* engineering in that systems engineering deals with the relationships of the thing being designed to its supersystem (environment) and subsystems, rather than with the internal details of how it is to accomplish its objectives. The systems viewpoint is broad, rather than deep: it encompasses the system functionally from end to end *and* temporally from conception to disposal.

System engineers must also rely on contributions from the *specialty engineering* disciplines, in addition to the traditional design disciplines, for functional expertise and specialized analytic methods. These specialty engineering areas typically include reliability, maintainability, logistics, test, production, transportation, human factors, quality assurance, and safety engineering. Specialty engineers contribute throughout the systems engineering process; part of the system engineer's job is to see that these functions are coherently integrated into the project at the right times and that they address the relevant issues. One of the objectives for Chapter 6 is to develop an understanding how these specialty engineers contribute to the objective of systems engineering.

In both systems *analysis* and systems *engineering*, the amounts and kinds of resources to be made available for the creation of the system are assumed to be among the

decisions to be made. Systems engineering concentrates on the creation of hardware and software architectures and on the development and management of the interfaces between subsystems, while relying on systems analysis to construct the mathematical models and analyze the data to evaluate alternative designs and to perform the actual design trade studies. Systems analysis often requires the use of tools from operations research, economics, or other *decision sciences*, and systems analysis curricula generally include extensive study of such topics as probability, statistics, decision theory, queueing theory, game theory, linear and non-linear programming, and so on. In practice, many system engineers' academic background is richer in the engineering disciplines than in the decision sciences. As a consequence, the system engineer is often a consumer of systems analysis products, rather than a producer of them. One of the major objectives for Chapter 5 is to develop an understanding and appreciation of the state of that art.

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

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

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

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

2.5 The Doctrine of Successive Refinement

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

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

It should be realized, however, that this spiral represents neither the project life cycle, which encompasses the

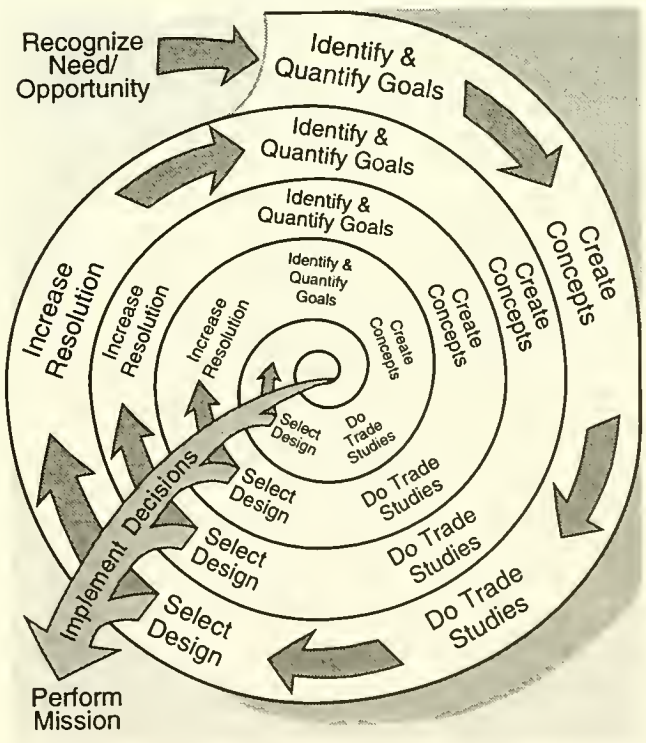


Figure 3 — The *Doctrine of Successive Refinement.*

system from inception through disposal, nor the product development process by which the system design is developed and implemented, which occurs in Phases C and D (see Chapter 3) of the project life cycle. Rather, as the intellectual process of systems engineering, it is inevitably reflected in both of them.

Figure 3 is really a double helix — each *create concepts* step at the level of design engineering initiates a capabilities definition spiral moving in the opposite direction. The concepts can never be created from whole cloth. Rather, they result from the synthesis of potential capabilities offered by the continually changing state of technology. This process of design concept development by the integration of lower-level elements is a part of the systems engineering process. In fact, there is always a danger that the top-down process cannot keep up with the bottom-up process.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

At each turn of the spiral, higher-level goals are analyzed. The analysis should identify the subordinate enabling goals in a way that makes them traceable to the next higher level. As the systems engineering process continues, these are documented as *functional* requirements (what must be done to achieve the next-higher-level goals) and as *performance* requirements (quantitative descriptions of how well the functional requirements must be done). A clear *operations concept* often helps to focus the requirements analysis so that both functional and performance requirements are ultimately related to the original need or opportunity. In later turns of the spiral, further elaborations may become documented as detailed functional and performance specifications.

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

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

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

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

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

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

Whether system models are used or not, the design concepts are developed, modified, reassessed, and compared against competing alternatives in a closed-loop process that seeks the best choices for further development. System and subsystem sizes are often determined during the trade studies. The end result is the determination of

bounds on the relative cost-effectivenesses of the design alternatives, measured in terms of the quantified system goals. (Only bounds, rather than final values, are possible because determination of the final details of the design is intentionally deferred. The bounds, in turn, may be derived from the probability density functions.) Increasing detail associated with the continually improving resolution reduces the spread between upper and lower bounds as the process proceeds.

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

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

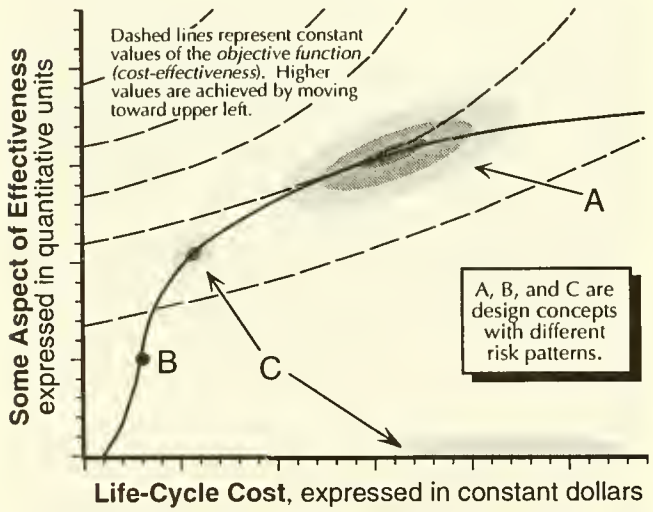


Figure 4 — A Quantitative *Objective Function*, Dependent on Life-Cycle Cost and All Aspects of Effectiveness.

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

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

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

Simple Interfaces are Preferred

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

Increase the Resolution of the Design. One of the first issues to be addressed is how the system should be subdivided into subsystems. (Once that has been done, the focus changes and the *subsystems* become *systems* — from the point of view of a system engineer. The partitioning process stops when the subsystems are simple enough to be managed holistically.) As noted by Morris, "the divi-

be managed holistically.) As noted by Morris, "the division of program activities to minimize the number and complexity of interfaces has a strong influence on the overall program cost and the ability of the program to meet schedules."

Charles Leising and Arnold Ruskin have (separately) pointed out that partitioning is more art than science, but that there are guidelines available: To make interfaces clean and simple, similar functions, designs and technologies should be grouped. Each portion of work should be verifiable. Pieces should map conveniently onto the organizational structure. Some of the functions that are needed throughout the design (such as *electrical power*) or throughout the organization (such as *purchasing*) can be centralized. Standardization — of such things as parts lists or reporting formats — is often desirable. The accounting system should follow (not lead) the system architecture. In terms of breadth, partitioning should be done essentially all at once. As with system design choices, alternative partitioning plans should be considered and compared before implementation.

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

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

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

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

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

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

3 The Project Life Cycle for Major NASA Systems

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

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

Decomposing the project life cycle into phases organizes the entire process into more manageable pieces. The project life cycle should provide managers with incremental visibility into the progress being made at points in time that fit with the management and budgetary enviroments. NASA documents governing the acquisition of major systems (NMI 7120.4 and NHB 7120.5) define the phases of the project life cycle as:

- Pre-Phase A — Advanced Studies ("find a suitable project")
- Phase A — Preliminary Analysis ("make sure the project is worthwhile")
- Phase B — Definition ("define the project and establish a preliminary design")
- Phase C — Design ("complete the system design")
- Phase D — Development ("build, integrate, and verify the system, and prepare for operations")
- Phase E — Operations ("operate the system and dispose of it properly").

Phase A efforts are conducted by NASA field centers; such efforts may rely, however, on pre-Phase A in-house and contracted advanced studies. The majority of Phase B efforts are normally accomplished by industry under NASA contract, but NASA field centers typically conduct parallel in-house studies in order to validate the contracted effort and remain an informed buyer. NASA usually chooses to contract with industry for Phases C and D, and often does so for Phase E. Phase C is normally combined with Phase D, but when large production quantities are planned, these are treated separately.

Alternatives to the project phases described above can easily be found in industry and elsewhere in government. In general, the engineering development life cycle is dependent on the technical nature of what's being developed, and the project life cycle may need to be tailored accordingly. Barry W. Boehm described how several contemporary software development processes work; in some of these processes, the development and construction activities proceed in parallel, so that attempting to separate the associated phases on a time line is undesirable. Boehm describes a spiral, which reflects the doctrine of successive refinement depicted in Figure 3, but Boehm's spiral describes the software product development process in particular. His discussion applies as well to the development of hardware products as it does to software. Other examples of alternative processes are the *rapid prototyping* and *rapid development* approaches. Selection of a product development process paradigm must be a case-dependent decision, based on the system engineer's judgment and experience.

Sometimes, it is appropriate to perform some long-lead-time activities ahead of the time they would normally be done. Long-lead-time activities might consist of technology developments, prototype construction and testing, or even fabrication of difficult components. Doing things out of their usual sequence increases risk in that those activities could wind up having been either unnecessary or improperly specified. On the other hand, overall risk can sometimes be reduced by removal of such activities from the critical path.

Figure 5 (foldout, next page) details the resulting management and major systems engineering products and control gates that characterize the phases in NMI 7120.4 and NHB 7120.5. Sections 3.1 to 3.6 contain narrative descriptions of the purposes, major activities, products, and control gates of the NASA project life cycle phases. Section 3.7 provides a more concentrated discussion of the role of systems engineering in the process. Section 3.8 describes the NASA budget cycle within which program/project managers and system engineers must operate.

3.1 Pre-Phase A — Advanced Studies

The purpose of this activity, which is usually performed more or less continually by "Advanced Projects" groups, is to uncover, invent, create, concoct and/or devise

Pre-Phase A — Advanced Studies

Purpose: To produce a broad spectrum of ideas and alternatives for missions from which new programs/projects can be selected.

Major Activities *and their Products*:

Identify *missions consistent with charter*
Identify and involve users
Perform *preliminary evaluations of possible missions*
Prepare *program/project proposals*, which include:

- Mission justification and objectives
- Possible operations concepts
- Possible system architectures
- Cost, schedule, and risk estimates.

Develop *master plans* for existing program areas

Information Baselined:

(nothing)

Control Gates:

Mission Concept Review
Informal proposal reviews

a broad spectrum of ideas and alternatives for missions from which new projects (programs) can be selected. Typically, this activity consists of loosely structured examinations of new ideas, usually without central control and mostly oriented toward small studies. Its major product is a stream of suggested projects, based on the identification of needs and the discovery of opportunities that are potentially consistent with NASA's mission, capabilities, priorities, and resources.

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

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

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

Advanced studies may extend for several years, and may be a sequence of papers that are only loosely connected. These studies typically focus on establishing mission goals and formulating top-level system requirements and operations concepts. Conceptual designs are often offered to demonstrate feasibility and support programmatic estimates. The emphasis is on establishing feasibility and desirability rather than optimality. Analyses and designs are accordingly limited in both depth and number of options.

3.2 Phase A — Preliminary Analysis

The purpose of this phase is to further examine the feasibility and desirability of a suggested new major system before seeking significant funding. According to NHB 7120.5, the major products of this phase are a formal *Mission Needs Statement* (MNS) and one or more credible, feasible designs and operations concepts. John Hodge describes this phase as "a structured version of the previous phase."

Phase A — Preliminary Analysis

Purpose: To determine the feasibility and desirability of a suggested new major system and its compatibility with NASA's strategic plans.

Major Activities *and their Products*:

Prepare *Mission Needs Statement*
Develop top-level *requirements*
Develop corresponding *evaluation criteria/metrics*
Identify alternative *operations and logistics concepts*
Identify *project constraints and system boundaries*
Consider *alternative design concepts*; including:
feasibility and risk studies, cost and schedule estimates, and advanced technology requirements
Demonstrate that *credible, feasible design(s)* exist
Acquire *systems engineering tools and models*
Initiate *environmental impact studies*
Prepare *Project Definition Plan* for Phase B

Information Baselined:

(nothing)

Control Gates:

Mission Definition Review
Preliminary Non-Advocate Review
Preliminary Program/Project Approval Review

In Phase A, a larger team, often associated with an ad hoc program or project office, readdresses the mission concept to ensure that the project justification and practicality are sufficient to warrant a place in NASA's budget. The team's effort focuses on analyzing mission requirements and establishing a mission architecture. Activities

NASA PROGRAM/PROJECT LIFE CYCLE

Program/Project

Project Phases	FORMULATION			IMPLEMENTATION		
	Pre-Phase A: Advanced Studies	Phase A: Preliminary Analysis	Phase B: Definition	Phase C: Design	Phase D: Development	Phase E: Operations

Agency Reviews

Pre-NAR ▼ — NAR ▼
Pre-PPAR ▼ — PPAR ▼

Independent Annual Reviews
Quarterly Status Reviews
Contractor Metrics Reviews
Cancellation Reviews - if required
Independent Readiness Review - as required

Administrator / CFO/Comptroller

Approve MNS ▲ — Approve PCA ▲ — Re-approve MNS ▲
IAR Reports — *IAR Reports* — *Multiple Unit Production Authorization (if applicable)* — *IAR Reports*

Program AA

Phase A Approval ▲ — AO
Preliminary IPS
Preliminary Program Plan
▲ Instrument Selection Decision
Preliminary PCA
Draft PCA
IPS
Final Program Plan
NEPA Record of Decision
Reaffirmed MNS
Annually Validated PCA
Approved PCA
Annually Validated PCA
Semi-Annual PSR
Annually Validated PCA
Semi-Annual PSR

Program Manager

Risk Management Plan
Integrated Program Master Schedule
Contractor Selection Decision ▲

Project Manager

MNS
Project Definition Plan
Phase B/C/D RFP
Prelim. Project Plan
System Spec
PDR Baseline
Project Plan
Quarterly CMR
Quarterly QSR Reports
CDR Specs/Baseline
Quarterly CMR
Quarterly QSR Reports
(including IRR Reports - as required)
Operational System •

Non-Advocates

Prelim. NAR Report
NAR Report

Technical

Major Reviews	Mission Feasibility	Mission Definition	System Definition	Preliminary Design	Final Design	Fabrication & Integration	Preparation For Deployment	Deployment & Operational Verification	Mission Operations	Disposal
	MCR ▼	MDR ▼	SRR ▽ SDR ▼	PDR ▼	CDR ▼	SAR ▼	FRR ▼	ORR ▼	DR ▼	
Technical Management Products	- Study Plan - Follow-on Plan	- System Engineering Managment Plan - Information Management Plan - Eng. Master Plan/Master Schedule - Risk Management Plan	- Program/Project Management Plan - Development Plans - Statement of Work - System Safety Plan - Configuration Management Plan - Documentation Tree	- Work Breakdown Structure - Technical Perf. Measurement Plan - Contamination Control Plan - EEE Parts Management Plan - Parts Control Plan - Environments Control Plan - Integ. Log. Support Program Plan - EMI/EMC Control Plan - Produc./Manufacturability Prog. Pln - Reliability Program Plan - Quality Assurance Plan - Applicable Standards	- Manufacturing Plan - Tech. Performance Measures Report - Materials and Processes Control Plan - Integrated Logistics Support Plan	- Operations Plan	- Certification of Flight/Launch Readiness			
Technical Products	- Mission Goals and Objectives - Concept/Design Evaluation Criteria - Mission Concepts - Operations Concepts - Life Cycle Cost Estimates - Feasibility Assessment	- Mission Need Statement - Functional Mission Concept - Preliminary System Specification - Science Requirements - Trade & Analysis Results - Technology Development Plan - PD/NSC-2S Databook	- System Concept & Architecture - System Specification - Interface Requirements - Environmental Specification - Human System Standards - Concept/Design Evaluation Criteria - Development Test Plans - Engineering Units - Hardware/Software List - Risk Analyses - Development Test Results - Technology Development Reqs.	- Design-To Specifications - Vendor H/W & S/W Specification - Disposal Requirements - Specification Tree - Drawing Tree/Eng. Drawing List - Interface Control Document - Payload to Carrier Integration Plan - Verification Plans - Verification Requirements Matrix - Environmental Impact Statement	- Build-To Specifications - Manuf. Processes Requirements - Design Disclosure - Operational Limits & Constraints - Integrated Schematics - Spares Provisioning List - Qualification Items - Launch Operations Plan - Transition to Operations Plan - Disposal Plans - Acceptance Plans - Acceptance Criteria - Verif. Requirements & Specifications - Verification Procedures - Integration & Assembly Plan - Instrum. Program and Command List	- Operations Procedures - Training Plan - User Manuals - In-Flight Checkout Plans - Computer Resource Integrated Support Document - Verification Data - Waivers	- H/W & S/W End Items - Operations Data - Launch Facility C/O Results - Go/No-Go Criteria	- Operational Evaluations Results - Problem/Failure Reports - Technical Manuals & Data - Trained Personnel	- Mission Products - Sequential Production - Replacements & Upgrades	- Disposed or Decommissioned Items

Key:
▼ Major Review / Interim Review
▲ Action
Document
- Product or Document(s)

AA Associate Administrator
AO Announcement of Opportunity
CDR Critical Design Review
CFO Chief Financial Officer
CMR Contractor Metrics Report
DR Decommissioning Review
FRR Flight Readiness Review
IAR Independent Annual Review
IPS Integrated Program/Project Summary
IRR Independent Readiness Review
MCR Mission Concept Review
MDR Mission Definition Review
MNS Mission Need Statement
MR Mission Review
NAR Non-Advocate Review
NEPA National Environmental Policy Act
NSC National Security Council
DRR Operational Readiness Review
PCA Program Commitment Agreement
PD Presidential Directive
PDR Preliminary Design Review
PPAR Program/Project Approval Review
PSR Project Status Reports
QSR Quarterly Status Reviews
RFP Request for Proposal
SAR System Acceptance Review
SDR System Definition Review
SRR System Requirements Review

become formal, and the emphasis shifts toward establishing optimality rather than feasibility. The effort addresses more depth and considers many alternatives. Goals and objectives are solidified, and the project develops more definition in the system requirements, top-level system architecture, and operations concept. Conceptual designs are developed and exhibit more engineering detail than in advanced studies. Technical risks are identified in more detail and technology development needs become focused.

The *Mission Needs Statement* is not shown in the sidebar as being baselined, as it is not under configuration control by the project. It may be under configuration control at the program level, as may the program requirements documents and the *Preliminary Program Plan.*

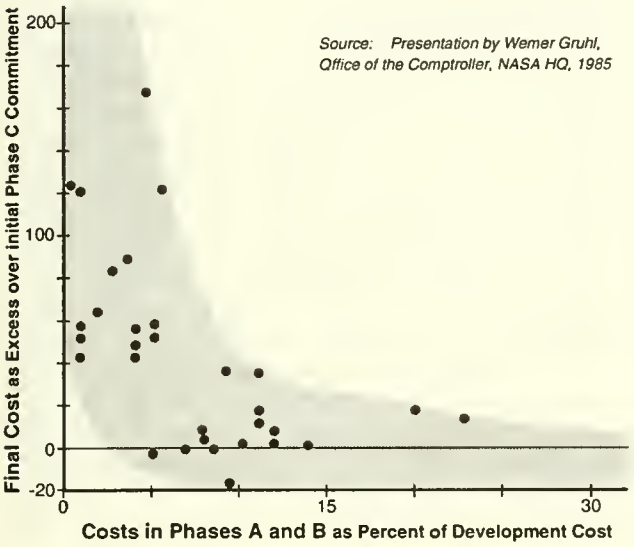


Figure 6 — Overruns are Very Likely if Phases A and B are Underfunded.

3.3 Phase B — Definition

The purpose of this phase is to establish an initial project baseline, which (according to NHB 7120.5) includes "a formal flowdown of the project-level performance requirements to a complete set of system and subsystem design specifications for both flight and ground elements" and "corresponding preliminary designs." The technical requirements should be sufficiently detailed to establish firm schedule and cost estimates for the project.

Actually, "the" Phase B baseline consists of a collection of evolving baselines covering technical and business aspects of the project: system (and subsystem) requirements and specifications, designs, verification and operations plans, and so on in the technical portion of the baseline, and schedules, cost projections, and managment plans in the business portion. Establishment of baselines implies the implementation of configuration management procedures. (See Section 4.7.)

Phase B — Definition

Purpose: To define the project in enough detail to establish an initial baseline capable of meeting mission needs.

Major Activities *and their Products*:

Prepare a *Systems Engineering Management Plan*
Prepare a *Risk Management Plan*
Initiate *configuration management*
Prepare *engineering specialty program plans*
Develop *system-level cost-effectiveness model*
Restate mission needs as *functional requirements*
Identify *science payloads*
Establish the initial *system requirements* and *verification requirements matrix*
Perform and archive *trade studies*
Select a baseline *design solution* and a *concept of operations*
Define *internal and external interface requirements*
(Repeat the process of successive refinement to get "design-to" *specifications and drawings, verifications plans,* and *interface documents* to lower levels as appropriate)
Define the *work breakdown structure*
Define *verification approach and policies*
Identify *integrated logistics support requirements*
Establish *technical resource estimates* and firm *life-cycle cost estimates*
Develop *statement(s) of work*
Initiate *advanced technology developments*
Revise and publish a *Project Plan*
Reaffirm the *Mission Needs Statement*
Prepare a *Program Commitment Agreement*

Information Baselined:

System requirements and verification requirements matrix
System architecture and work breakdown structure
Concept of operations
"Design-to" specifications at all levels
Project plans, including schedule, resources, acquisition strategies, and risk management

Control Gates:

Non-Advocate Review
Program/Project Approval Review
System Requirements Review(s)
System Definition Review
System-level Preliminary Design Review
Lower-level Preliminary Design Reviews
Safety review(s)

A *Credible, Feasible* Design

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

Early in Phase B, the effort focuses on allocating functions to particular items of hardware, software, personnel, etc. System functional and performance requirements along with architectures and designs become firm as system trades and subsystem trades iterate back and forth in the effort to seek out more cost-effective designs. (Trade studies should precede — rather than follow — system design decisions. Chamberlain, Fox, and Duquette describe a decentralized process for ensuring that such trades lead efficiently to an optimum system design.) Major products to this point include an accepted "functional" baseline and preliminary "design-to" baseline for the system and its major end items. The effort also produces various engineering and management plans to prepare for managing the project's downstream processes, such as verification and operations, and for implementing engineering specialty programs.

Along the way to these products, projects are subjected to a *Non-Advocate Review,* or NAR. This activity seeks to assess the state of project definition in terms of its clarity of objectives and the thoroughness of technical and management plans, technical documentation, alternatives explored, and trade studies performed. The NAR also seeks to evaluate the cost and schedule estimates, and the contingency reserve in these estimates. The timing of this review is often driven by the Federal budget cycle, which requires at least 16 months between NASA's budget preparation for submission to the President's Office of Management and Budget, and the Congressional funding for a new project start. (See Section 3.8.) There is thus a natural tension between the desire to have maturity in the project at the time of the NAR and the desire to progress efficiently to final design and development.

Later in Phase B, the effort shifts to establishing a functionally complete design solution (i.e., a "design-to" baseline) that meets mission goals and objectives. Trade studies continue. Interfaces among the major end items are defined. Engineering test items may be developed and used to derive data for further design work, and project risks are reduced by successful technology developments and demonstrations. Phase B culminates in a series of *preliminary design reviews* (PDRs), containing the system-level PDR and PDRs for lower-level end items as appropriate. The PDRs reflect the successive refinement of requirements into designs. Design issues uncovered in the PDRs should be resolved so that final design can begin with unambiguous "design-to" specifications. From this point on, almost all changes to the baseline are expected to represent successive refinements, not fundamental changes. Prior to baselining, the system architecture, preliminary design, and operations concept must have been validated by enough technical analysis and design work to establish a *credible, feasible* design at a lower level of detail than was sufficient for Phase A.

3.4 Phase C — Design

The purpose of this phase is to establish a complete design ("build-to" baseline) that is ready to fabricate (or code), integrate, and verify. Trade studies continue. Engineering test units more closely resembling actual hardware are built and tested so as to establish confidence that the design will function in the expected environments. Engineering specialty analysis results are integrated into the design, and the manufacturing process and controls are defined and validated. Configuration management continues to track and control design changes as detailed interfaces are defined. At each step in the successive refinement of the final design, corresponding integration and verification activities are planned in greater detail. During this phase, technical parameters, schedules, and budgets are closely tracked to ensure that undesirable trends (such as an unexpected growth in spacecraft mass or increase in its cost) are recognized early enough to take corrective action. (See Section 4.9.)

Phase C culminates in a series of *critical design reviews* (CDRs) containing the system-level CDR and CDRs corresponding to the different levels of the system hierarchy. The CDR is held prior to the start of fabrication/production of end items for hardware and prior to the start of coding of deliverable software products. Typically, the sequence of CDRs reflects the integration process that will occur in the next phase — that is, from lower-level CDRs to the system-level CDR. Projects, however, should tailor the sequencing of the reviews to meet their individual needs. The final product of this phase is a "build-to" baseline in sufficient detail that actual production can proceed.

Phase C — Design

Purpose: To complete the detailed design of the system (and its associated subsystems, including its operations systems).

Major Activities *and their Products*:

Add remaining *lower-level design specifications* to the system architecture
Refine *requirements documents*
Refine *verification plans*
Prepare *interface documents*
(Repeat the process of successive refinement to get "build-to" *specifications and drawings*, *verification plans*, and *interface documents* at all levels)
Augment baselined documents to reflect the growing maturity of the system: *system architecture*, *verification requirements matrix*, *work breakdown structure*, *project plans*
Monitor project progress against project plans
Develop the *system integration plan* and the *system operation plan*
Perform and archive *trade studies*
Complete *manufacturing plan*
Develop the *end-to-end information system design*
Refine *Integrated Logistics Support Plan*
Identify opportunities for pre-planned product improvement
Confirm *science payload selection*

Information Baselined:

All remaining lower-level requirements and designs, including traceability to higher levels
"Build-to" specifications at all levels

Control Gates:

Subsystem (and lower level) Critical Design Reviews
System-level Critical Design Review

3.5 Phase D — Development

The purpose of this phase is to build and verify the system designed in the previous phase, deploy it, and prepare for operations. Activities include fabrication of hardware and coding of software, integration, and verification of the system. Other activities include the initial training of operating personnel and implementation of the Integrated Logistics Support Plan. For flight projects, the focus of activities then shifts to pre-launch integration and launch. For large flight projects, there may be an extended period of orbit insertion, assembly, and initial shake-down operations. The major product is a system that has been shown to be capable of accomplishing the purpose for which it was created.

Phase D — Development

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

Major Activities *and their Products*:

Fabricate (or code) the *parts* (i.e., the lowest-level items in the system architecture)
Integrate those items according to the integration plan and perform verifications, yielding *verified components and subsystems*
(Repeat the process of successive integration to get a *verified system*)
Develop *verification procedures* at all levels
Perform *system qualification verification(s)*
Perform *system acceptance verification(s)*
Monitor project progress against project plans
Archive documentation for *verifications* performed
Audit "as-built" configurations
Document *Lessons Learned*
Prepare *operator's manuals*
Prepare *maintenance manuals*
Train *initial system operators and maintainers*
Finalize and implement *Integrated Logistics Support Plan*
Integrate with launch vehicle(s) and launch, perform orbit insertion, etc., to achieve a *deployed system*
Perform *operational verification(s)*

Information Baselined:

"As-built" and "as-deployed" configuration data
Integrated Logistics Support Plan
Command sequences for end-to-end command and telemetry validation and ground data processing
Operator's manuals
Maintenance manuals

Control Gates:

Test Readiness Reviews (at all levels)
System Acceptance Review
System functional and physical configuration audits
Flight Readiness Review(s)
Operational Readiness Review
Safety reviews

3.6 Phase E — Operations

The purpose of this phase is to meet the initially identified need or to grasp the initially identified opportunity. The products of the phase are the results of the mission. This phase encompasses evolution of the system only insofar as that evolution does not involve major changes to the system architecture; changes of that scope

Phase E — Operations

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

Major Activities *and their Products*:

Train replacement *operators and maintainers*
Conduct *the mission(s)*
Maintain and upgrade the *system*
Dispose of the system and supporting processes
Document *Lessons Learned*

Information Baselined:

Mission outcomes, such as:

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

Operations and maintenance logs
Problem/failure reports

Control Gates:

Regular system operations readiness reviews
System upgrade reviews
Safety reviews
Decommissioning Review

constitute new "needs," and the project life cycle starts over.

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

In addition to uncertainty as to when this part of the phase begins, the activities associated with safely decommissioning and disposing of a system may be long and complex. Consequently, the costs and risks associated with different designs should be considered during the project's earlier phases.

3.7 Role of Systems Engineering in the Project Life Cycle

This section presents two "idealized" descriptions of the systems engineering activities within the project life cycle. The first is the Forsberg and Mooz "vee" chart, which is taught at the NASA program/project management course. The second is the NASA program/project life cycle process flow developed by the NASA-wide Systems Engineering Process Improvement Task team, in 1993/94.

3.7.1 The "Vee" Chart

Forsberg and Mooz describe what they call "the technical aspect of the project cycle" by a vee-shaped chart, starting with user needs on the upper left and ending with a user-validated system on the upper right. Figure 7 provides a summary level overview of those activities. On the left side of the vee, decomposition and definition activities resolve the system architecture, creating the details of the design. Integration and verification flow up and to the right as successively higher levels of subsystems are verified, culminating at the system level. This summary chart follows the basic outline of the vee chart developed by NASA as part of the Software Management and Assurance Program. ("CIs" in the figure refer to the hardware and software *configuration items*, which are controlled by the configuration management system.)

Decomposition and Definition. Although not shown in Figure 7, each box in the vee represents a number of parallel boxes suggesting that there may be many subsystems that make up the system at that level of decomposition. For the top left box, the various parallel boxes represent the alternative design concepts that are initially evaluated.

As product development progresses, a series of baselines is progressively established, each of which is put under formal configuration management at the time it is approved. Among the fundamental purposes of configuration management is to prevent requirements from "creeping."

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

However, there is no prohibition against doing detailed work early in the process. In fact, detailed hardware

and/or software models may be required at the very earliest stages to clarify user needs or to establish credibility for the claim of feasibility. Early application of involved technical and support disciplines is an essential part of this process; this is in fact implementation of *concurrent engineering*.

At each level of the vee, systems engineering activities include off-core processes: system design, advanced technology development, trade studies, risk management, specialty engineering analysis and modeling. This is shown on the chart as an orthagonal process in Figure 7(b). These activities are performed at each level and may be repeated many times within a phase. While many kinds of studies and decisions are associated with the off-core activities, only decisions at the core level are put under configuration management at the various control gates. Off-core activities, analyses, and models are used to substantiate the core decisions and to ensure that the risks have been mitigated or determined to be acceptable. The off-core work is not formally controlled, but the analyses, data and results should be archived to facilitate replication at the appropriate times and levels of detail to support introduction into the baseline.

There can, and should, be sufficient iteration downward to establish feasibility and to identify and quantify risks. Upward iteration with the requirements statements (and with the intermediate products as well) is permitted, but should be kept to a minimum unless the user is still generating (or changing) requirements. In software projects, upward confirmation of solutions with the users is often necessary because user requirements cannot be adequately defined at the inception of the project. Even for software projects, however, iteration with user requirements should be stopped at the PDR, or cost and schedule are likely to get out of control.

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

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

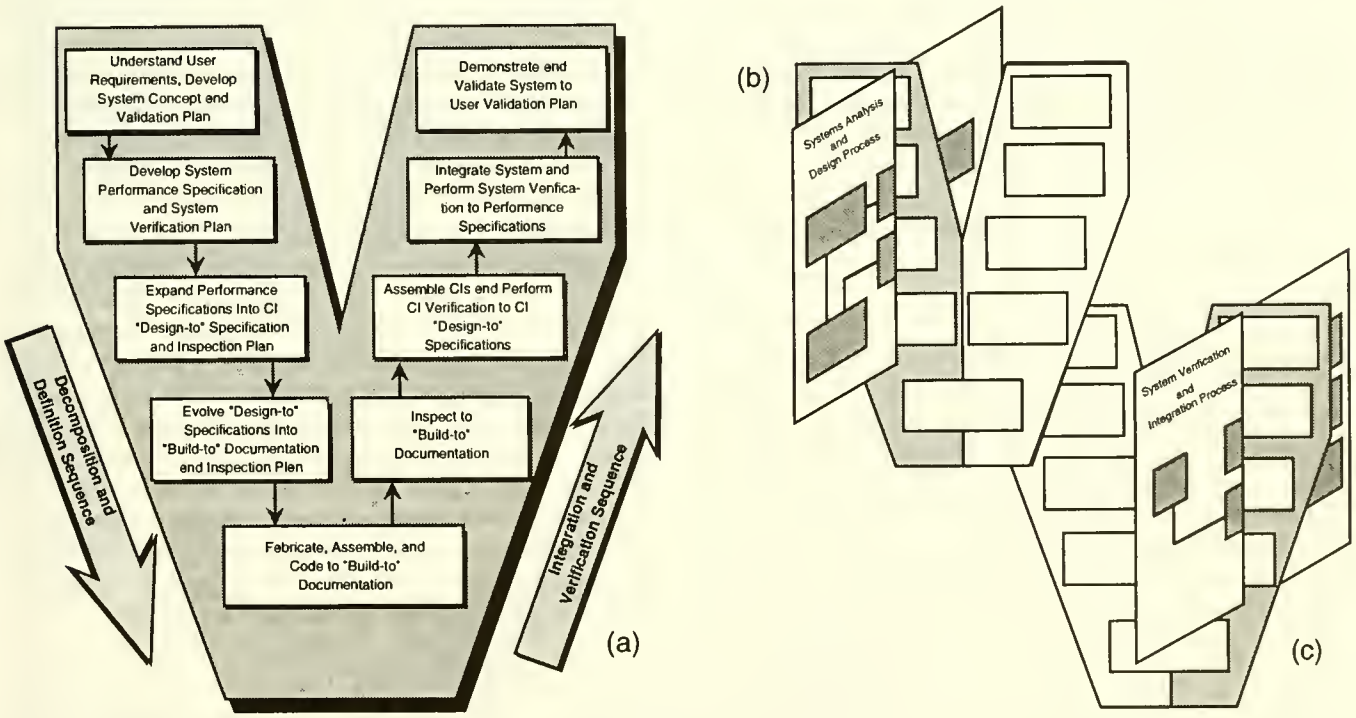


Figure 7 — Overview of the Technical Aspect of the NASA Project Cycle.

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

Even at the highest levels, as user requirements are translated into system requirements, the system verification approach, which will prove that the system does what is required, must be determined. The technical demands of the verification process, represented as an orthagonal process in Figure 7(c), can drive cost and schedule, and may in fact be a discriminator between alternative concepts. For example, if engineering models are to be used for verification or validation, they must be specified and costed, their characteristics must be defined, and their development time must be incorporated into the schedule from the beginning.

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

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

3.7.2 The NASA Program/Project Life Cycle Process Flow

Another idealized description of the technical activities that occur during the NASA project life cycle is illustrated in Figure 8 (foldout, next page). In the figure, the NASA project life cycle is partitioned into ten process flow blocks, which are called *stages* in this handbook. The stages reflect the changing nature of the work that needs to be performed as the system matures. These stages are related both temporally and logically. Successive stages mark increasing system refinement and maturity, and require the products of previous stages as inputs. A transition to a new stage entails a major shift in the nature or extent of technical activities. Control gates assess the wisdom of progressing from one stage to another. (See Section 4.8.3 for success criteria for specific reviews.) From the perspective of the system engineer, who must oversee and monitor the technical progress on the system, Figure 8 provides a more complete description of the actual work needed through the NASA project life cycle.

In practice, the stages do not always occur sequentially. Unfolding events may invalidate or modify goals and assumptions. This may neccessitate revisiting or modifying the results of a previous stage. The end items comprising the system often have different development schedules and constraints. This is especially evident in Phases C and D where some subsystems may be in final design while others are in fabrication and integration.

The products of the technical activities support the systems engineering effort (e.g., requirements and specifications, trade studies, specialty engineering analyses, verification results), and serve as inputs to the various control gates. For a detailed systems engineering product database, database dictionary, and maturity guidelines, see JSC-49040, *NASA Systems Engineering Process for Programs and Projects.*

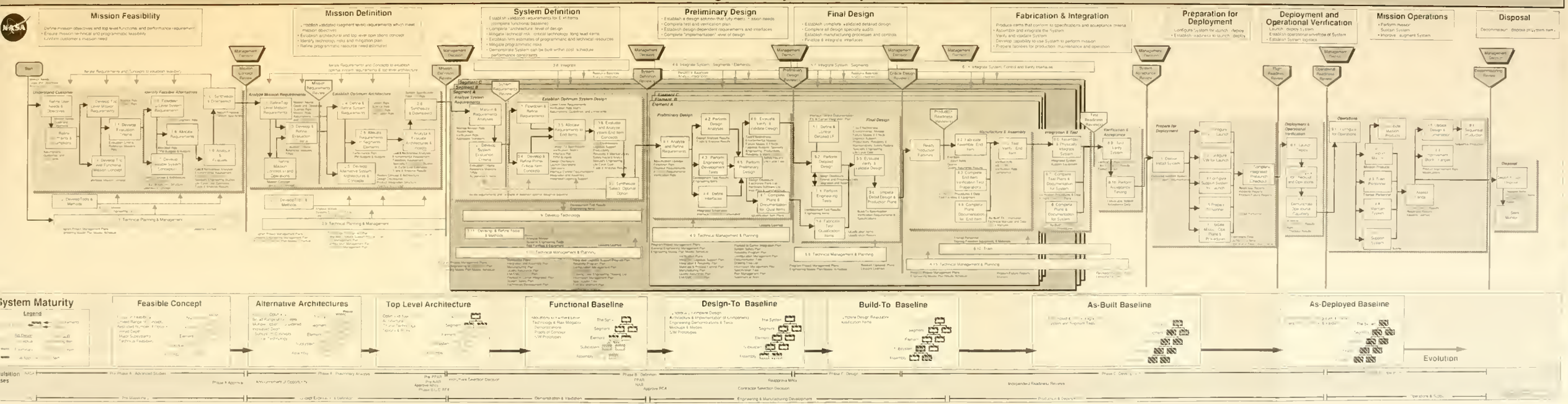
Several topics suggested by Figures 7 and 8 merit special emphasis. These are concurrent engineering, technology insertion, and the distinction between verification and validation.

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

Concurrent engineering is the simultaneous consideration of product and process downstream requirements by multidisciplinary teams. Specialty engineers from all disciplines (reliability, maintainability, human factors, safety, logistics, etc.) whose expertise will eventually be represented in the product have important contributions throughout the system life cycle. The system engineer is responsible for ensuring that these personnel are part of the project team at each stage. In large projects, many integrated *product development teams* (PDTs) may be required. Each of these, in turn, would be represented on a PDT for the next higher level in the project. In small projects, however, a small team is often sufficient as long as the system engineer can augment it as needed with experts in the required technical and business disciplines.

The informational requirements of doing concurrent engineering are demanding. One way concurrent engineering experts believe it can be made less burdensome is by an automated environment. In such an environment, systems engineering, design and analysis tools can easily ex-

The NASA Program/Project Life Cycle Process Flow



Integrated Product Development Teams

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

change data, computing environments are interoperable, and product data are readily accessible and accurate. For more on the characteristics of automated environments, see for example Carter and Baker, *Concurrent Engineering*, 1992.

Technology Insertion. Projects are sometimes initiated with known technology shortfalls, or with areas for which new technology will result in substantial product improvement. Technology development can be done in parallel with the project evolution and inserted as late as the PDR. A parallel approach that is *not* dependent on the development of new technology must be carried unless high risk is acceptable. The technology development activity should be managed by the project manager and system engineer as a critical activity.

Verification vs. Validation. The distinction between verification and validation is significant: *verification* consists of proof of compliance with specifications, and may be determined by test, analysis, demonstration, inspection, etc. (see Section 6.6). *Validation* consists of proof that the system accomplishes (or, more weakly, *can* accomplish) its purpose. It is usually much more difficult (and much more important) to validate a system than to verify it. Strictly speaking, validation can be accomplished only at the system level, while verification must be accomplished throughout the entire system architectural hierarchy.

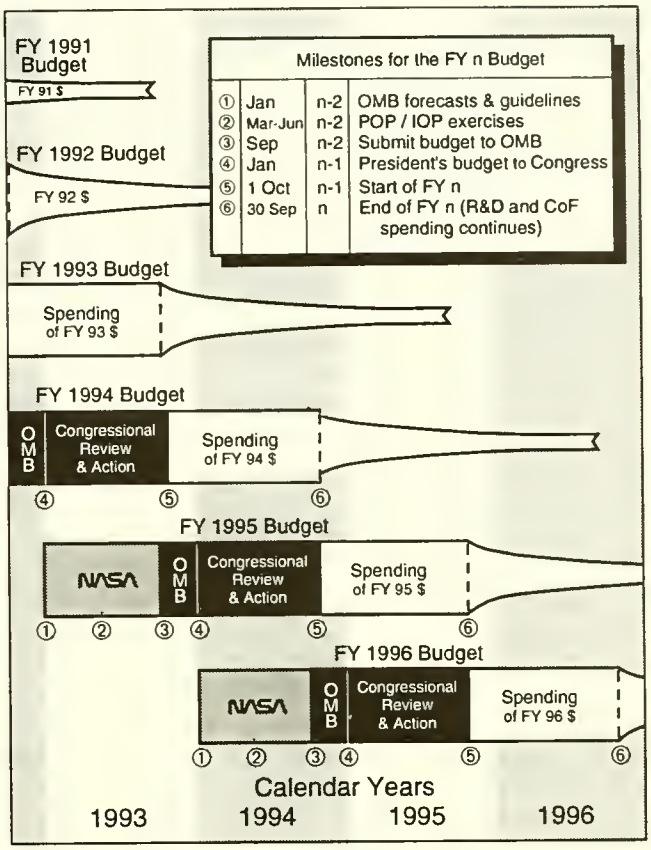


Figure 9 — Typical NASA Budget Cycle.

3.8 Funding: The Budget Cycle

NASA operates with annual funding from Congress. This funding results, however, from a three-year rolling process of budget formulation, budget enactment, and finally, budget execution. A highly simplified representation of the typical budget cycle is shown in Figure 9.

NASA starts developing its budget each January with economic forecasts and general guidelines being provided by the Executive Branch's Office of Management and Budget (OMB). In early May, NASA conducts its Program Operating Plan (POP) and Institutional Operating Plan (IOP) exercises in preparation for submittal of a preliminary NASA budget to the OMB. A final NASA budget is submitted to the OMB in September for incorporation into the President's budget transmittal to Congress, which generally occurs in January. This proposed budget is then subjected to Congressional review and approval, culminating in the passage of bills authorizing NASA to obligate funds in accordance with Congressional stipulations and appropriating those funds. The Congressional

process generally lasts through the summer. In recent years, however, final bills have often been delayed past the start of the fiscal year on October 1. In those years, NASA has operated on continuing resolutions by Congress.

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

4 Management Issues in Systems Engineering

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

4.1 Harmony of Goals, Work Products, and Organizations

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

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

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

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

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

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

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

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

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

Each project should be managed in accordance with a project life cycle that is carefully tailored to the project's risks. While the project manager concentrates on managing the overall project life cycle, the project-level or lead system engineer concentrates on managing its technical aspect (see Figure 7 or 8). This requires that the system engineer perform or cause to be performed the necessary multiple layers of decomposition, definition, integration, verification and validation of the system, while orchestrating and incorporating the appropriate concurrent engineering. Each one of these systems engineering functions requires application of technical analysis skills and techniques.

The techniques used in systems engineering management include work breakdown structures, network scheduling, risk management, requirements traceability and reviews, baselines, configuration management, data management, specialty engineering program planning, definition and readiness reviews, audits, design certification, and status reporting and assessment.

The Project Plan defines how the project will be managed to achieve its goals and objectives 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. The SEMP also communicates how the systems engineering management techniques noted above should be applied.

4.2.1 Role of the SEMP

The SEMP is the rule book that describes to all participants how the project will be technically managed. The responsible NASA field 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 field center should have its SEMP developed before attempting to prepare an initial cost estimate, since activities that incur cost, such as technical risk reduction, need to be identified and described beforehand. The contractor should have its SEMP developed during the proposal process (prior to costing and pricing) because the SEMP describes the technical content of the project, the potentially costly risk management activities, and the verification and validation techniques to be used, all of which must be included in the preparation of project cost estimates.

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

4.2.2 Contents of the SEMP

Since the SEMP describes the project's technical management approach, which is driven by the type of project, the phase in the project life 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 Project Planning and Control. This section should identify organizational responsibilities and authority for systems engineering management, including 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/project reviews; and control of documentation.

This section should describe:

- The role of the project office
- The role of the user
- The role of the Contracting Office Technical Representative (COTR)
- The role of systems engineering
- The role of design engineering
- The role of specialty engineering
- Applicable standards
- Applicable procedures and training
- Baseline control process

- Change control process
- Interface control process
- Control of contracted (or subcontracted) engineering
- Data control process
- Make-or-buy control process
- Parts, materials, and process control
- Quality control
- Safety control
- Contamination control
- Electromagnetic interference and electromagnetic compatibility (EMI/EMC)
- Technical performance measurement process
- 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
- Requirements allocation 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
- Specification and drawing structure
- Configuration management process
- Data management process
- Use of mathematical models
- Use of simulations
- 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, as needed, 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
- Maintainability
- Quality assurance
- Integrated logistics
- Human engineering
- Safety
- Producibility
- Survivability/vulnerability
- Environmental assessment
- Launch approval.

4.2.3 Development of the SEMP

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

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

4.2.4 Managing the Systems Engineering Process: Summary

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

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 life cycle by involving the required specialty engineering disciplines. The SEMP is the guiding document for these activities.

SEMP Lessons Learned from DoD Experience

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

4.3 The Work Breakdown Structure

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

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

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

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

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

This section provides some techniques for developing a WBS, and points out some mistakes to avoid. Appendix B.2 provides an example of a WBS for an airborne telescope that follows the principles of product-based WBS development.

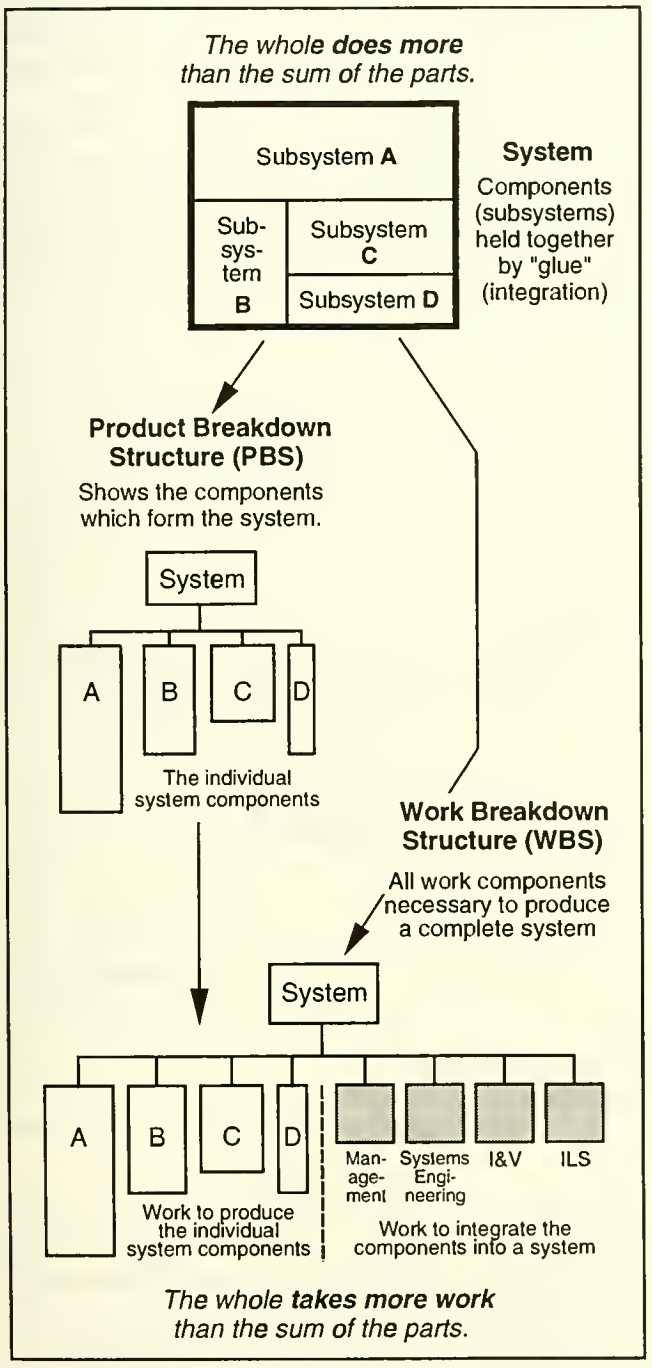


Figure 10 — The Relationship Between a System, a Product Breakdown Structure, and a Work Breakdown Structure.

4.3.1 Role of the WBS

A product-based WBS is the organizing structure for:

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

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

4.3.2 Techniques for Developing the WBS

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

The PBS and associated WBS can then be developed level by level from the top down. In this approach, a project-level system engineer finalizes the PBS at the pro-

ject level, and provides a draft PBS for the next lower level. The WBS is then derived by adding appropriate services such as management and systems engineering to that lower level. This process is repeated recursively until a WBS exists down to the desired cost account level.

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

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

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

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

4.3.3 Common Errors in Developing a WBS

There are three common errors found in WBSs:

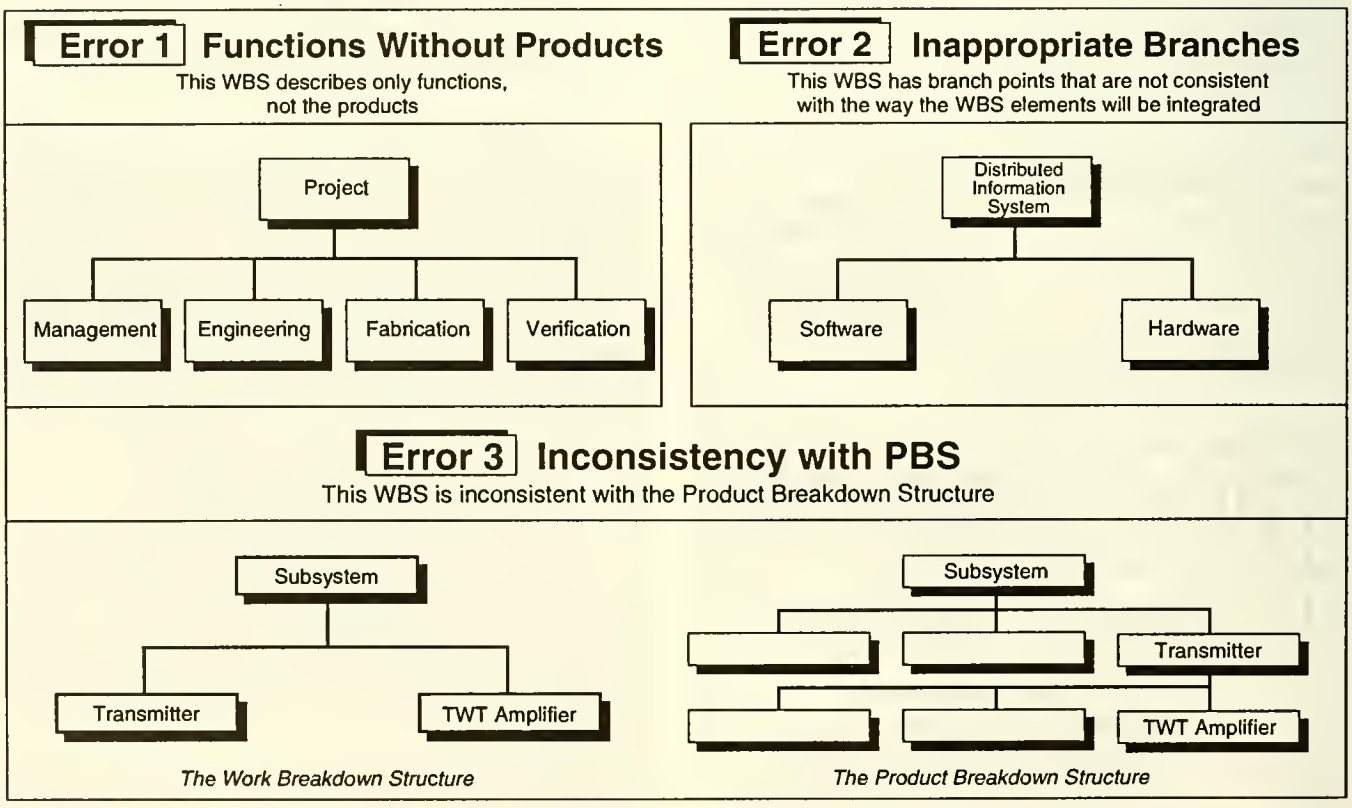


Figure 11 — Examples of WBS Development Errors.

- *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 11. Each one prevents the WBS from successfully performing its roles in project planning and organizing. These errors are avoided by using the WBS development techniques described above.

4.4 Scheduling

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

4.4.1 Role of Scheduling

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

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

4.4.2 Network Schedule Data and Graphical Formats

Network schedule data consist of:

- Activities
- Dependencies between activities (e.g., where an activity depends upon another activity for a receivable)
- Products or milestones that occur as a result of one or more activities
- Duration of each activity.

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

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

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

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

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

4.4.3 Establishing a Network Schedule

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

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

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

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

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.

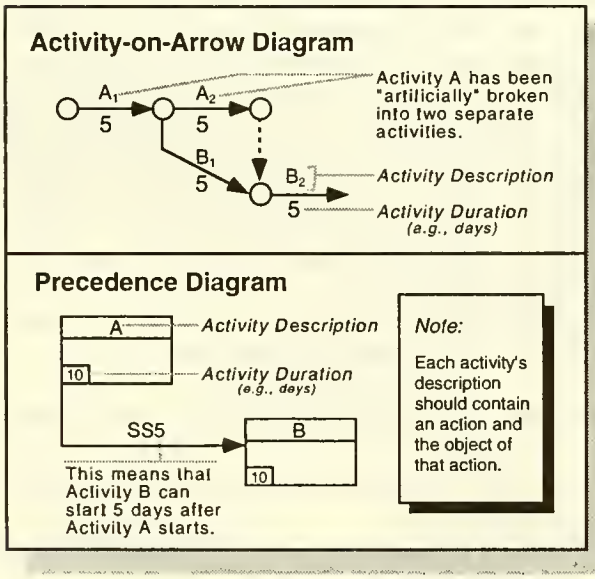


Figure 12 — Activity-on-Arrow and Precedence Diagrams for Network Schedules.

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

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

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

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

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

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

4.4.4 Reporting Techniques

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

4.4.5 Resource Leveling

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

4.5 Budgeting and Resource Planning

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

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

Desirable Features in Gantt Charts

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

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

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

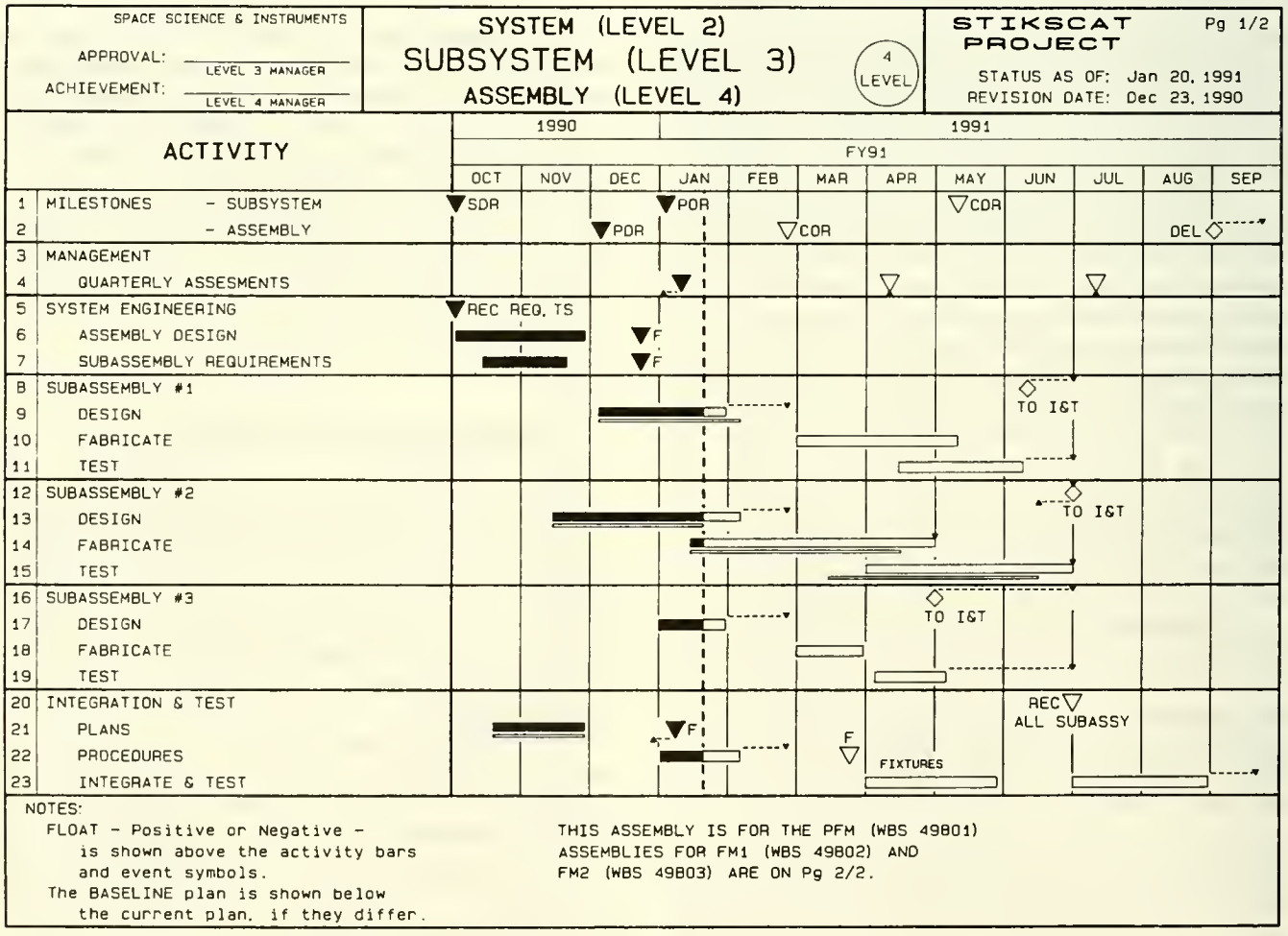


Figure 13 — An Example of a Gantt Chart.

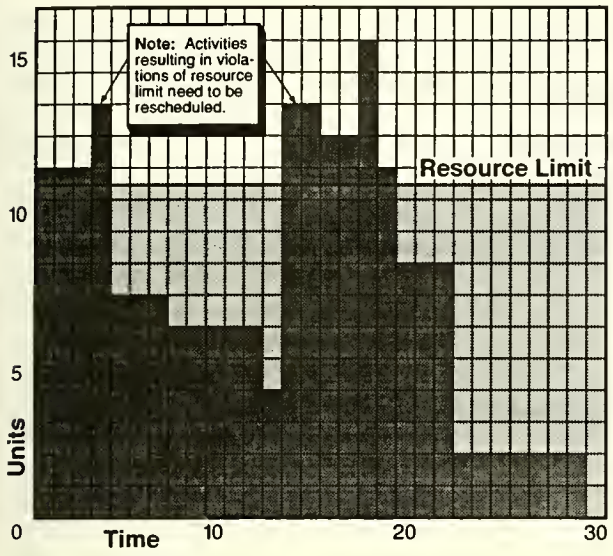


Figure 14 — An Example of an Unleveled Resource Profile.

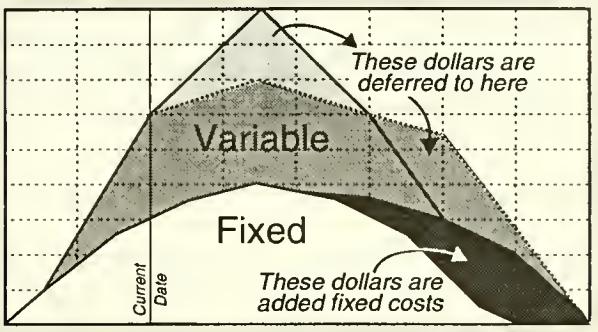
- Direct labor costs
- Overhead costs
- Other direct costs (travel, data processing, etc.)
- Subcontract costs
- Material costs
- General and administrative costs
- Cost of money (i.e., interest payments, if applicable)
- Fee (if applicable)
- Contingency.

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

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

Assessing the Effect of Schedule Slippage

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



To quickly assess the effect of a simple schedule slippage:

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

4.6 Risk Management

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

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

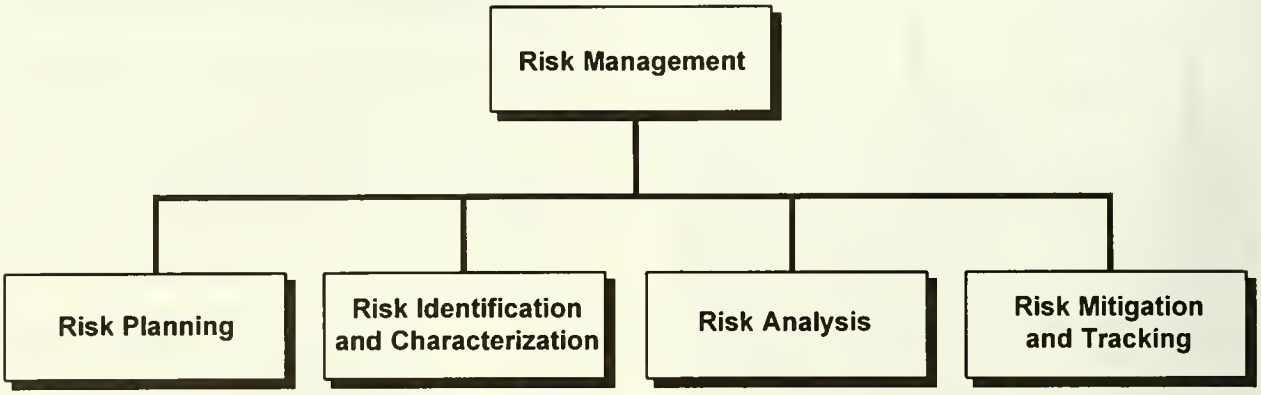


Figure 15 — Risk Management Structure Diagram.

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

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

Risk

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

The first is planning the risk management program, which should be documented in a *risk management program plan.* That plan, which elaborates on the SEMP, 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 methodologies, processes, and tools to be used for risk identification and characterization, risk analysis, and risk mitigation and tracking
- A description of the role of risk management with respect to reliability analyses, formal reviews, and status reporting and assessment
- Documentation requirements for each risk management product and action.

The level of risk management activities should be consistent with the project's overall risk policy established in conjunction with its NASA Headquarters program office. At present, formal guidelines for the classification of projects with respect to overall risk policy do not exist; such guidelines exist only for NASA payloads. These are promulgated in NMI 8010.1A, *Classification of NASA Payloads, Attachment A,* which is reproduced as Appendix B.3.

With the addition of data tables containing the *results* of the risk management activities, the risk management program plan grows into the project's Risk Management Plan (RMP). These data tables should contain the project's identified significant risks. For each such risk, these data tables should also contain the relevant characterization and analysis results, and descriptions of the related mitigation and tracking plans (including any descope options and/or required technology developments). A sample RMP outline is shown as Appendix B.4.

The technical portion of risk management begins with 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 their likelihood of occurrence (e.g., high, medium, or low), and separately, according to the severity of their 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, as Figure 16 suggests. The primary methods used in this process are qualitative; hence in systems engineering literature, this step is sometimes called *qualitative risk assessment.* The output of this step is a list of significant risks (by phase) to be given specific management attention.

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

Risk mitigation is the formulation, selection, and execution of strategies designed to economically reduce risk. When a specific risk is believed to be intolerable, risk analysis and mitigation are often performed iteratively, so that the effects of alternative mitigation strategies can be actively explored before one is chosen. Tracking the effectivity of these strategies is closely allied with risk mitigation. Risk mitigation is often a challenge because

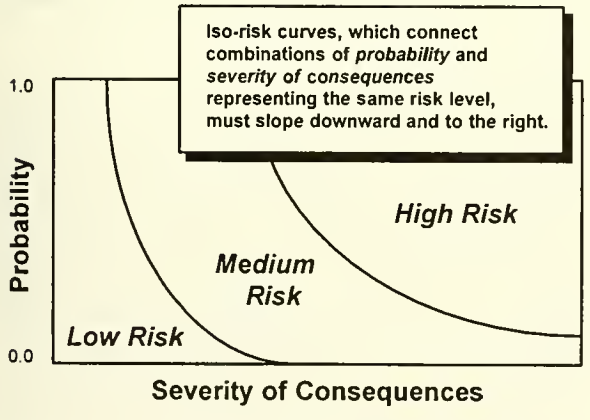


Figure 16 — Characterizing Risks by Likelihood and Severity.

Table 1 — Techniques of Risk Management.

Risk Identification and Characterization	Risk Analysis	Risk Mitigation and Tracking
Expert interviews	Decision analysis	Watchlists/milestones
Independent assessment (cost, schedule and technical)	Probabalistic Risk Assessment (PRA)	Contingency planning/ descope planning/par-allel development
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/ Fault Trees		

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 system engineer need to understand the system-wide effects of various strategies in order to make a rational allocation of resources.

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

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

4.6.1 Types of Risks

There are several ways to describe the various types of risk a project manager/system engineer faces. Traditionally, project managers and system engineers have attempted to divide risks into three or four broad categories — namely, cost, schedule, technical, and, sometimes, safety (and/or hazard) risks. More recently, others have entered the lexicon, including the categories of organizational, management, acquisition, supportability, political, and programmatic risks. These newer categories reflect

the expanded set of concerns of project managers and system engineers who must operate in the current NASA environment. Some of these newer categories also represent supersets of other categories. For example, the Defense Systems Management College (DSMC) Systems Engineering Management Guide wraps "funding, schedule, contract relations, and political risks" into the broader category of programmatic risks. While these terms are useful in informal discussions, there appears to be no formal taxonomy free of ambiguities. One reason, mentioned above, is that often one type of risk can be exchanged for another. A second reason is that some of these categories move together, as for example, cost risk and political risk (e.g., the risk of project cancellation).

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

An example of the first kind of uncertainty occurs in the unpredictability of the spares upmass requirement for alternative Space Station *Alpha* 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 *Alpha* 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 is the probability estimator's perceived state of information. Consequently, subjectivists find the distinction between the two kinds of uncertainty of little or no practical significance. The implication of the subjectivist's view for risk management is that, even with little or no data, the system engineer's subjective probability estimates form a valid basis for risk decision making.

4.6.2 Risk Identification and Characterization Techniques

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

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

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

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

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

Lessons Learned. A review of the lessons learned files, data, and reports from previous similar projects can produce insights and information for risk identification on a new project. For technical risk identification, as an example, it makes sense to examine previous projects of similar function, architecture, or technological approach. The lessons learned from the *Infrared Astronomical Satellite* (IRAS) project might be useful to the *Space Infrared Telescope Facility* (SIRTF) project, even though the latter's degree of complexity is significantly greater. The key to applying this technique is in recognizing what aspects are analogous in two projects, and what data are relevant to the new project. Even if the 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, Digraphs, and Fault Trees. Failure Modes, Effects, and Criticality Analysis (FMECA), Failure Modes and Effects Analysis (FMEA), digraphs, and fault trees 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 there is less emphasis on the severity classification portion of the analysis.

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 in Section 6.2, if quantitative probability estimates are needed.

4.6.3 Risk Analysis Techniques

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

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

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

In brief, decision analysis is a technique that allows:

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

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

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

Doing a PRA is itself a major effort, requiring a number of specialized skills other than those provided by reliability engineers and human factors engineers. PRAs also require large amounts of system design data at the component level, and operational procedures data. For additional information on PRAs, the system engineer can reference the *PRA Procedures Guide* (1983) by the American Nuclear Society and Institute of Electrical and Electronic Engineers (IEEE).

Probabilistic Network Schedules. Probabilistic network schedules, such as PERT (Program Evaluation and Review Technique), permit the duration of each activity to be treated as a random variable. By supplying PERT with the minimum, maximum, and most likely duration for each activity, a probability distribution can be computed for project completion time. This can then be used to determine, for example, the chances that a project (or any set of tasks in the network) will be completed by a given date. In this probabilistic setting, however, a unique critical path may not exist. Some practitioners have also cited difficulties in obtaining meaningful input data for probabilistic network schedules. A simpler alternative to a full probabilistic network schedule is to perform a Monte Carlo simulation of activity durations along the project's critical path. (See Section 5.4.2.)

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

Probabilistic Risk Assessment Pitfalls

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

$$R = \sum_{s} P_s C_s$$

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

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

4.6.4 Risk Mitigation and Tracking Techniques

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

with a co-participant, (3) take preventive action to avoid or reduce the risk, and (4) plan for contingent action.

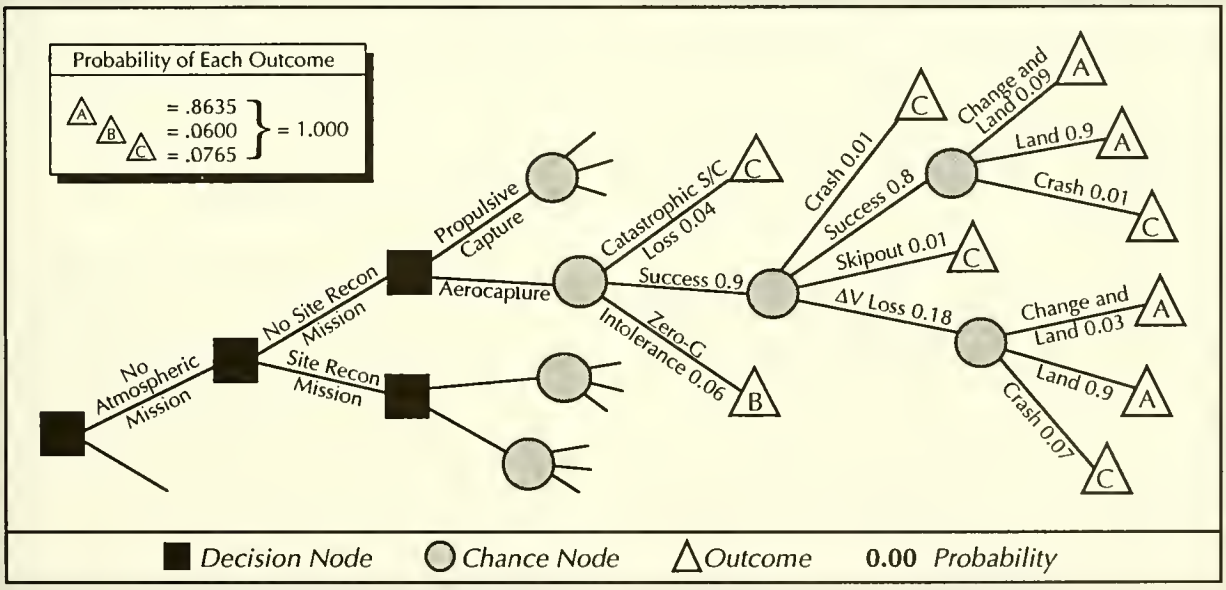
The first response is to accept a specific risk consciously. (This response can be accompanied by further risk information gathering and assessments.) Second, a risk can sometimes be shared with a co-participant — that is, with a international partner or a contractor. In this situation, the goal is to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

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

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

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 subsequently selected for the manned mission. 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.

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

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

Contingency Planning, Descope Planning, and Parallel Development. These techniques are generally used in conjunction with a watchlist. The focus 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, these techniques and resources act as a form of project insurance. (The term *contingency* here should not be confused with the use within NASA of the same term for project-held reserves.)

Critical Items/Issues Lists. A Critical Items/Issues List (CIL) is similar to a watchlist, and has been extensively used on the Shuttle program to track items with significant system safety consequences. An example is shown as Appendix B.5.

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

4.6.5 Risk Management: Summary

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

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

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

4.7 Configuration Management

Configuration management is the discipline of identifying and formalizing the functional and physical characteristics of a configuration item at discrete points in the product evolution for the purpose of maintaining the integrity of the product system and controlling changes to the *baseline*. The baseline for a project contains all of the technical requirements and related cost and schedule requirements that are sufficiently mature to be accepted and placed under change control by the NASA project manager. The project baseline consists of two parts: the technical baseline and the business baseline. The system engineer is responsible for managing the technical baseline and ensuring that it is consistent with the costs and schedules in the business baseline. Typically, the project control office manages the business baseline.

Configuration 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 life cycle), and to

change the baseline requirements only by a formal configuration control process. The buyer might be a NASA program office or an external funding agency. For example, the buyer for the GOES project is NOAA, and the seller is the NASA GOES project office. Configuration 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. Configuration management is established through program/project requirements documentation and, where applicable, through the contract Statement of Work.

Configuration management is essential to conduct 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 since it manages the project's configuration documentation. (See Section 4.9.2 on Technical Performance Measures.) The project's approach to configuration management and the methods to be used should be documented in the project's Configuration Management Plan. A sample outline for this plan is illustrated in Appendix B.6. The plan should be tailored to each project's specific needs and resources, and kept current for the entire project life cycle.

4.7.1 Baseline Evolution

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

- Functional and performance requirements (or specifications) for hardware, software, information items, and processes
- Interface requirements
- Specialty engineering requirements
- Verification requirements
- Data packages, documentation, and drawing trees
- Applicable engineering standards.

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 System Requirements Review (SRR)
- "Design-to" baseline at Preliminary Design Review (PDR)
- "Build-to" (or "code-to") baseline at the Critical Design Review (CDR)
- "As-built" (or "as-coded") baseline at the System Acceptance Review (SAR)
- "As-deployed" baseline at Operational Readiness Review (ORR).

The evolution of the five baselines is illustrated in Figure 17. As discussed in Section 3.7.1, only decisions made along the core of the "vee" in Figure 7 are put under configuration control and included in the approved baseline. Systems analysis, risk management, and development test activities (off the core of the vee) must begin early and continue throughout the decomposition process of the project life cycle to prove that the core-level decisions are sound. These early detailed studies and tests must be documented and retained in the project archives, but they are not part of the technical baseline.

4.7.2 Techniques of Configuration Management

The techniques of configuration management include configuration (or baseline) identification, configura-

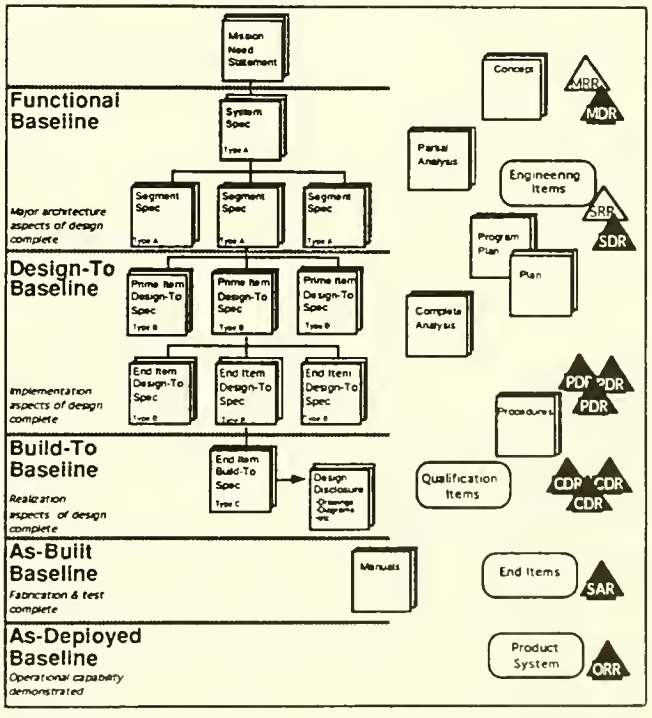


Figure 17 — Evolution of the Technical Baseline.

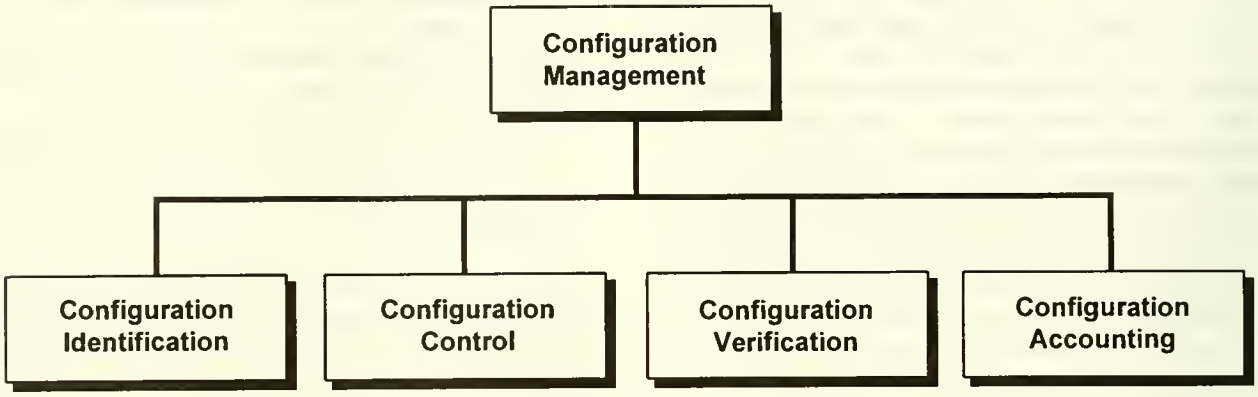


Figure 18 — Configuration Management Structure Diagram.

tion control, configuration verification, and configuration accounting (see Figure 18).

Configuration Identification. Configuration identification of a baseline is accomplished by creating and formally releasing documentation that describes the baseline to be used, and how changes to that baseline will be accounted for, controlled, and released. Such documentation includes requirements (product, process, and material), specifications, drawings, and code listings. Configuration documentation is not formally considered part of the technical baseline until approved by control gate action of the buyer.

An important part of configuration identification is the physical identification of individual configuration items using part numbers, serial numbers, lot numbers, version numbers, document control numbers, etc.

Configuration Control. Configuration control is the process of controlling changes to any approved baseline by formal action of a configuration control board (CCB). This area of configuration management is usually the most visible to the system engineer. In large programs/projects, configuration control is accomplished by a hierarchy of configuration control boards, reflecting multiple levels of control. Each configuration control board has its own areas of control and responsibilities, which are specified in the Configuration Management Plan.

Typically, a configuration control board meets to consider change requests to the business or technical baseline of the program/project. The program/project manager is usually the board chair, who is the sole decision maker. The configuration manager acts as the board secretary, who skillfully guides the process and records the official events of the process. In a configuration 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 program/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?

Configuration Control Board Conduct

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

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

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

Decision: The CCB members discuss the Change Request (CR) and formulate a decision. Project manager agrees or overrides. The secretary prepares a CCB directive, which records and directs the CR's disposition.

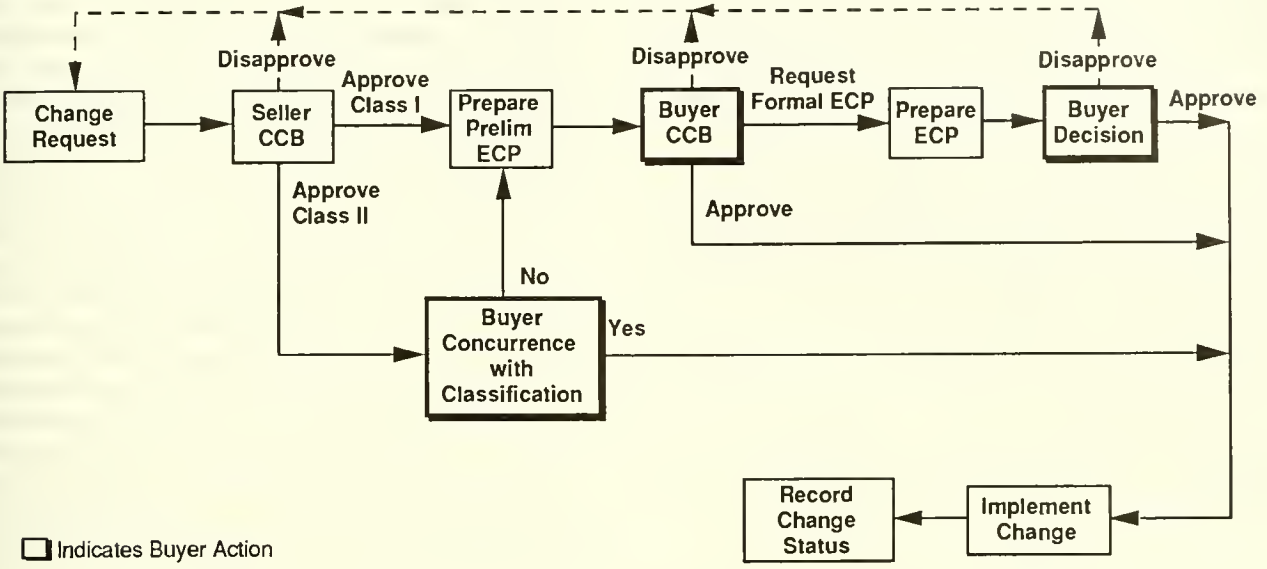


Figure 19 — Contract Change Control Process.

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

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

The system engineer should also ensure that the active approved baseline is communicated in a timely manner to all those relying on it. This communication keeps project teams apprised as to the distinction between what is frozen under formal change control and what can still be decided without configuration control board approval.

Configuration control is essential at both the contractor and NASA field 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 19. 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. Class 2 changes are dispositioned by the contractor's CCB and do not require the NASA project manager's approval.

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 effective configuration control on every project.

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

Approved changes on a development project that has only one deliverable obviously are only applicable to that one deliverable item. However, for projects that have multiple deliverables of "identical" design, changes may become effective on the second or subsequent production articles. In such a situation, the configuration 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 or-

biters 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.

Configuration Verification. Configuration verification is the process of verifying that resulting products (e.g., hardware and software items) conform to the intentions of the designers and to the standards established by preceding approved baselines, and that baseline documentation is current and accurate. Configuration verification is accomplished by two types of control gate activity: audits and technical reviews. (See Section 4.8.4 for additional information on two important examples: the Physical Configuration Audit and the Design Certification Review.) Each of these serves to review and challenge the data presented for conformance to the previously approved baseline.

Configuration Accounting. Configuration accounting (sometimes called configuration status accounting) is the task of maintaining, correlating, releasing, reporting, and storing configuration data. Essentially a data management function, configuration accounting ensures that official baseline data is retained, available, and distribution-controlled for project use. It also performs the important function of tracking the status of each change from inception through implementation. A project's change status system should be capable of identifying each change by its unique change identification number (e.g., ECRs, CRs, RIDs, waivers, deviations, modification kits) and report its current status.

The Role of the Configuration Manager. The configuration manager is responsible for the application of these techniques. In doing so, the configuration manager performs the following functions:

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

4.7.3 Data Management

For any project, proper data management is essential for successful configuration management. Before a project team can produce a tangible product, it must produce descriptions of the system using words, drawings, schematics, and numbers (i.e., symbolic information). 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 project 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 a non-trivial statement, since poor data management practices (e.g., allowing someone to borrow the only copy of a document or drawing) can allow controlled information to become lost. Also, the material must be retained for the life of the program/project (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 database must be developed and maintained to show the parentage of any requirement. The database must also be able to display all children derived from a given requirement. Finally, traceability must be provided to reports that document trade study results and other decisions that played a key role in the flowdown of requirements. The data management function therefore encompasses managing and archiving supporting analyses and trade study data, and keeping them convenient for configuration management and general project use.

4.8 Reviews, Audits, and Control Gates

The intent and policy for reviews, audits, and control gates should be developed during Phase A and defined in the Program/Project Plan. The specific implementation of these activities should be consistent with the types of reviews and audits described in this section, and with the NASA Program/Project Life Cycle chart (see Figure 5) and the NASA Program/Project Life Cycle Process Flow chart (see Figure 8). However, the timing of reviews, audits, and control gates should be tailored to each specific project.

4.8.1 Purpose and Definitions

The purpose of a *review* is to furnish the forum and process to provide NASA management and their contractors assurance that the most satisfactory approach, plan or

design has been selected, that a configuration item has been produced to meet the specified requirements, or that a configuration item is ready. Reviews (technical or management) are scheduled to communicate an approach, demonstrate an ability to meet requirements, or establish status. Reviews help to develop a better understanding among task or project participants, open communication channels, alert participants and management to problems, and open avenues for solutions.

The purpose of an *audit* is to provide NASA management and its contractors a thorough examination of adherence to program/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 life 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 Program/Project Plan.

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.

4.8.2 General Principles for Reviews

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

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

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 and development 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 life 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-discipline expertise to identify any design shortfalls or recommend design improvements. The review audience should also include non-project specialists in the area under review, and specialists in production/fabrication, testing, quality assurance, reliability, and safety. Some reviews may also require the presence of both the contractor's and NASA's contracting officers.

Prior to and during the review, board members and review attendees may submit requests for action or engineering change requests (ECRs) that document a concern,

deficiency, or recommended improvement in the presented approach, plan, or design. Following the review, these are screened by the review board to consolidate them, and to ensure that the chair and cognizant manager(s) understand the intent of the requests. It is the responsibility of the review board to ensure that adequate closure responses for each of the action requests are obtained.

Post Review Report. The review board chair has the responsibility to develop, where necessary, a consensus of the findings of the board, including an assessment of the risks associated with problem areas, and develop recommendations for action. The chair submits, 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 field 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 life of a project, it is advisable to select extra board members and rotate active assignments to cover needs.

4.8.3 Major Control Gates

This section describes the purpose, timing, objectives, success criteria, and results of the major control gates in the NASA project life cycle. This information is intended to provide guidance to project managers and system engineers, and to illustrate the progressive maturation of review activities and systems engineering products. The checklists provided below aid in the preparation of specific review entry and exit criteria, but do not take their place. To minimize extra work, review material should be keyed to project documentation.

Mission Concept Review.

Purpose — The Mission Concept Review (MCR) affirms the mission need, and examines the proposed mission's objectives and the concept for meeting those objectives. It is an internal review that usually occurs at the cognizant NASA field center.

Timing — Near the completion of a mission feasibility study.

Objectives — The objectives of the review are to:

- Demonstrate that mission objectives are complete and understandable
- Confirm that the mission concepts demonstrate technical and programmatic feasibility of meeting the mission objectives
- Confirm that the customer's mission need is clear and achievable
- Ensure that prioritized evaluation criteria are provided for subsequent mission analysis.

Criteria for Successful Completion — The following items compose a checklist to aid in determining readiness of MCR product preparation:

- Are the mission objectives clearly defined and stated? Are they unambiguous and internally consistent?
- Will satisfaction of the preliminary set of requirements provide a system which will meet mission objectives?
- Is the mission feasible? Has there been a solution identified which is technically feasible? Is the rough cost estimate within an acceptable cost range?
- Have the concept evaluation criteria to be used in candidate system evaluation been identified and prioritized?
- Has the need for the mission been clearly identified?
- Are the cost and schedule estimates credible?
- Was a technology search done to identify existing assets or products that could satisfy the mission or parts of the mission?

Results of Review — A successful MCR supports the determination that the proposed mission meets the customer need, and has sufficient quality and merit to support a field center management decision to propose further study to the cognizant NASA Program Associate Administrator (PAA) as a candidate Phase A effort.

Mission Definition Review.

Purpose — The Mission Definition Review (MDR) examines the functional and performance requirements defined for the system and the preliminary program/project plan, and assures that the requirements and the selected architecture/design will satisfy the mission.

Timing — Near the completion of the mission definition stage.

Objectives — The objectives of the review are to:

- Establish that the allocation of the functional system requirements is optimal for mission satisfaction with respect to requirements trades and evaluation criteria that were internally established at MCR
- Validate that system requirements meet mission objectives
- Identify technology risks and the plans to mitigate those risks
- Present refined cost, schedule, and personnel resource estimates.

Criteria for Successful Completion — The following items compose a checklist to aid in determining readiness of MDR product preparation:

- Do the defined system requirements meet the mission objectives expressed at the start of the program/project?
- Are the system-level requirements complete, consistent, and verifiable? Have preliminary allocations been made to lower levels?
- Have the requirements trades converged on an optimal set of system requirements? Do the trades address program/project cost and schedule constraints as well as mission technical needs? Do the trades cover a broad spectrum of options? Have the trades identified for this set of activities been completed? Have the remaining trades been identified to select the final system design?
- Are the upper levels of the system PBS completely defined?
- Are the decisions made as a result of the trades consistent with the evaluation criteria established at the MCR?
- Has an optimal final design converged to a few alternatives?
- Have technology risks been identified and have mitigation plans been developed?

Results of Review — A successful MDR supports the decision to further develop the system architecture/design and any technology needed to accomplish the mission. The results reinforce the mission's merit and provide a basis for the system acquisition strategy.

System Definition Review.

Purpose — The System Definition Review (SDR) examines the proposed system architecture/design and the flowdown to all functional elements of the system.

Timing — Near the completion of the system definition stage. It represents the culmination of efforts in system requirements analysis and allocation.

Objectives — The objectives of the SDR are to:

- Demonstrate that the architecture/design is acceptable, that requirements allocation is complete, and that a system that fulfills the mission objectives can be built within the constraints posed
- Ensure that a verification concept and preliminary verification program are defined
- Establish end item acceptance criteria
- Ensure that adequate detailed information exists to support initiation of further development or acquisition efforts.

Criteria for Successful Completion — The following items compose a checklist to aid in determining readiness of SDR project preparation:

- Will the top-level system design selected meet the system requirements, satisfy the mission objectives, and address operational needs?
- Can the top-level system design selected be built within cost constraints and in a timely manner? Are the cost and schedule estimates valid in view of the system requirements and selected architecture?
- Have all the system-level requirements been allocated to one or more lower levels?
- Have the major design issues for the elements and subsystems been identified? Have major risk areas been identified with mitigation plans?
- Have plans to control the development and design process been completed?
- Is a development verification/test plan in place to provide data for making informed design decisions?
- Is the minimum end item product performance documented in the acceptance criteria?
- Is there sufficient information to support proposal efforts? Is there a complete validated set of requirements with sufficient system definition to support the cost and schedule estimates?

Results of Review — As a result of successful completion of the SDR, the system and its operation are well enough understood to warrant design and acquisition of the end items. Approved specifications for the system, its segments, and preliminary specifications for the design of appropriate functional elements may be released. A configuration management plan is established to control

design and requirement changes. Plans to control and integrate the expanded technical process are in place.

Preliminary Design Review. The Preliminary Design Review (PDR) is not a single review but a number of reviews that includes the system PDR and PDRs conducted on specific Configuration Items (CIs).

Purpose — The PDR demonstrates that the preliminary design meets all system requirements with acceptable risk. It shows that the correct design option has been selected, interfaces identified, and verification methods have been satisfactorily described. It also establishes the basis for proceeding with detailed design.

Timing — After completing a full functional implementation.

Objectives — The objectives of the PDR are to:

- Ensure that all system requirements have been allocated, the requirements are complete, and the flowdown is adequate to verify system performance
- Show that the proposed design is expected to meet the functional and performance requirements at the CI level
- Show sufficient maturity in the proposed design approach to proceed to final design
- Show that the design is verifiable and that the risks have been identified, characterized, and mitigated where appropriate.

Criteria for Successful Completion — The following items compose a checklist to aid in determining readiness of PDR product preparation:

- Can the proposed preliminary design be expected to meet all the requirements within the planned cost and schedule?
- Have all external interfaces been identified?
- Have all the system and segment requirements been allocated down to the CI level?
- Are all CI "design-to" specifications complete and ready for formal approval and release?
- Has an acceptable operations concept been developed?
- Does the proposed design satisfy requirements critical to human safety and mission success?
- Do the human factors considerations of the proposed design support the intended end users' ability to operate the system and perform the mission effectively?
- Have the production, verification, operations, and other specialty engineering organizations reviewed the design?
- Is the proposed design producible? Have long lead items been considered?
- Do the specialty engineering program plans and design specifications provide sufficient guidance, constraints, and system requirements for the design engineers to execute the design?
- Is the reliability analysis based on a sound methodology, and does it allow for realistic logistics planning and life-cycle cost analysis?
- Are sufficient project reserves and schedule slack available to proceed further?

Results of Review — As a result of successful completion of the PDR, the "design-to" baseline is approved. It also authorizes the project to proceed to final design.

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

Purpose — The CDR discloses the complete system design in full detail, ascertains that technical problems and design anomalies have been resolved, and ensures that the design maturity justifies the decision to initiate fabrication/manufacturing, integration, and verification of mission hardware and software.

Timing — Near the completion of the final design stage.

Objectives — The objectives of the CDR are to:

- Ensure that the "build-to" baseline contains detailed hardware and software specifications that can meet functional and performance requirements
- Ensure that the design has been satisfactorily audited by production, verification, operations, and other specialty engineering organizations
- Ensure that the production processes and controls are sufficient to proceed to the fabrication stage
- Establish that planned Quality Assurance (QA) activities will establish perceptive verification and screening processes for producing a quality product
- Verify that the final design fulfills the specifications established at PDR.

Criteria for Successful Completion — The following items compose a checklist to aid in determining readiness of CDR product preparation:

- Can the proposed final design be expected to meet all the requirements within the planned cost and schedule?
- Is the design complete? Are drawings ready to begin production? Is software product definition sufficiently mature to start coding?
- Is the "build-to" baseline sufficiently traceable to assure that no orphan requirements exist?
- Do the design qualification results from software prototyping and engineering item testing, simulation, and analysis support the conclusion that the system will meet requirements?
- Are all internal interfaces completely defined and compatible? Are external interfaces current?
- Are integrated safety analyses complete? Do they show that identified hazards have been controlled, or have those remaining risks which cannot be controlled been waived by the appropriate authority?
- Are production plans in place and reasonable?
- Are there adequate quality checks in the production process?
- Are the logistics support analyses adequate to identify integrated logistics support resource requirements?
- Are comprehensive system integration and verification plans complete?

Results of Review — As a result of successful completion of the CDR, the "build-to" baseline, production, and verification plans are approved. Approved drawings are released and authorized for fabrication. It also authorizes coding of deliverable software (according to the "build-to" baseline and coding standards presented in the review), and system qualification testing and integration. All open issues should be resolved with closure actions and schedules.

System Acceptance Review.

Purpose — The System Acceptance Review (SAR) examines the system, its end items and documentation, and test data and analyses that support verification. It also ensures that the system has sufficient technical maturity to authorize its shipment to and installation at the launch site or the intended operational facility.

Timing — Near the completion of the system fabrication and integration stage.

Objectives — The objectives of the SAR are to:

- Establish that the system is ready to be delivered and accepted under DD-250
- Ensure that the system meets acceptance criteria that were established at SDR
- Establish that the system meets requirements and will function properly in the expected operational environments as reflected in the test data, demonstrations, and analyses
- Establish an understanding of the capabilities and operational constraints of the "as-built" system, and that the documentation delivered with the system is complete and current.

Criteria for Successful Completion — The following items compose a checklist to aid in determining readiness of SAR product preparation:

- Are tests and analyses complete? Do they indicate that the system will function properly in the expected operational environments?
- Does the system meet the criteria described in the acceptance plans?
- Is the system ready to be delivered (flight items to the launch site and non-flight items to the intended operational facility for installation)?
- Is the system documentation complete and accurate?
- Is it clear what is being bought?

Results of Review — As a result of successful completion of the SAR, the system is accepted by the buyer, and authorization is given to ship the hardware to the launch site or operational facility, and to install software and hardware for operational use.

Flight Readiness Review.

Purpose — The Flight Readiness Review (FRR) examines tests, demonstrations, analyses, and audits that determine the system's readiness for a safe and successful launch and for subsequent flight operations. It also ensures that all flight and ground hardware, software, personnel, and procedures are operationally ready.

Timing — After the system has been configured for launch.

Objectives — The objectives of the FRR are to:

- Receive certification that flight operations can safely proceed with acceptable risk
- Confirm that the system and support elements are properly configured and ready for launch
- Establish that all interfaces are compatible and function as expected

- Establish that the system state supports a launch "go" decision based on go/no-go criteria.

Criteria for Successful Completion — The following items compose a checklist to aid in determining readiness of FRR product preparation:

- Is the launch vehicle ready for launch?
- Is the space vehicle hardware ready for safe launch and subsequent flight with a high probability for achieving mission success?
- Are all flight and ground software elements ready to support launch and flight operations?
- Are all interfaces checked out and found to be functional?
- Have all open items and waivers been examined and found to be acceptable?
- Are the launch and recovery environmental factors within constraints?

Results of Review — As a result of successful FRR completion, technical and procedural maturity exists for system launch and flight authorization, and in some cases initiation of system operations.

Operational Readiness Review.

Purpose — The Operational Readiness Review (ORR) examines the actual system characteristics and the procedures used in its operation, and ensures that all flight and ground hardware, software, personnel, procedures, and user documentation reflect the deployed state of the system accurately.

Timing — When the system and its operational and support equipment and personnel are ready to undertake the mission.

Objectives — The objectives of the ORR are to:

- Establish that the system is ready to transition into an operational mode through examination of available ground and flight test results, analyses, and operational demonstrations
- Confirm that the system is operationally and logistically supported in a satisfactory manner considering all modes of operation and support (normal, contingency, and unplanned)
- Establish that operational documentation is complete and represents the system configuration and its planned modes of operation
- Establish that the training function is in place and has demonstrated capability to support all aspects of system maintenance, preparation, operation, and recovery.

Criteria for Successful Completion — The following items compose a checklist to aid in determining readiness of ORR product preparation:

- Are the system hardware, software, personnel, and procedures in place to support operation?
- Have all anomalies detected during prelaunch, launch, and orbital flight been resolved, documented, and incorporated into existing operational support data?
- Are the changes necessary to transition the system from flight test to an operational configuration ready to be made?
- Are all waivers closed?
- Are the resources in place, or financially planned and approved to support the system during its operational lifetime?

Results of Review — As a result of successful ORR completion, the system is ready to assume normal operations and any potential hazards due to launch or flight operations have been resolved through use of redundant design or changes in operational procedures.

Decommissioning Review.

Purpose — The Decommissioning Review (DR) confirms that the reasons for decommissioning are valid and appropriate, and examines the current system status and plans for disposal.

Timing — When major items within the system are no longer needed to complete the mission.

Objectives — The objectives of the DR are to:

- Establish that the state of the mission and/or system requires decommissioning/disposal. Possibilities include no further mission need, broken/degraded system elements, or phase out of existing system assets due to a pending upgrade
- Demonstrate that the plans for decommissioning, disposal, and any transition are correct, current and appropriate for current environmental constraints and system configuration
- Establish that resources are in place to support disposal plans
- Ensure that archival plans have been completed for essential mission and project data.

Criteria for Successful Completion — The following items compose a checklist to aid in determining readiness of DR product preparation:

- Are reasons for decommissioning/disposal well documented?
- Is the disposal plan completed and compliant with local, state, and federal environmental regulations?
- Does the disposal plan address the disposition of existing hardware, software, facilities, and processes?
- Have disposal risks been addressed?
- Have data archival plans been defined?
- Are sufficient resources available to complete the disposal plan?
- Is a personnel transition plan in place?

Results of Review — A successful DR completion assures that the decommissioning and disposal of system items and processes are appropriate and effective.

4.8.4 Interim Reviews

Interim reviews are driven by programmatic and/or NASA Headquarters milestones that are not necessarily supported by the major reviews. They are often multiple review processes that provide important information for major NASA reviews, programmatic decisions, and commitments. Program/project tailoring dictates the need for and scheduling of these reviews.

Requirements Reviews. Prior to the PDR, the mission and system requirements must be thoroughly analyzed, allocated, and validated to assure that the project can effectively understand and satisfy the mission need. Specifically, these interim requirements reviews confirm whether:

- The proposed project supports a specific NASA program deficiency
- In-house or industry-initiated efforts should be employed in the program realization
- The proposed requirements meet objectives
- The requirements will lead to a reasonable solution
- The conceptual approach and architecture are credibly feasible and affordable.

These issues, as well as requirements ambiguities, are resolved or resolution actions are assigned. Interim requirements reviews alleviate the risk of excess design and analysis burdens too far into the life cycle.

Safety Reviews. Safety reviews are conducted to ensure compliance with NHB 1700.1B, *NASA Safety Policy and Requirements Document*, and are approved by the program/project manager at the recommendation of the system safety manager. Their purpose, objectives, and general schedule are contained in appropriate safety management plans. Safety reviews address possible hazards associated with system assembly, test, operation, and support. Special consideration is given to possible operational and environmental hazards related to the use of nuclear and other toxic materials. (See Section 6.8.) Early reviews with field center safety personnel should be held to identify and understand any problems areas, and to specify the requirements to control them.

Software Reviews. Software reviews are scheduled by the program/project manager for the purpose of ensuring that software specifications and associated products are well understood by both program/project and user personnel. Throughout the development cycle, the pedigree, maturity, limitations, and schedules of delivered preproduction items, as well as the Computer Software Configuration Items (CSCI), are of critical importance to the project's engineering, operations, and verification organizations.

Readiness Reviews. Readiness reviews are conducted prior to commencement of major events that commit and expose critical program/project resources to risk. These reviews define the risk environment and address the capability to satisfactorily operate in that environment.

Mission Requirements Review.

Purpose — The Mission Requirements Review (MRR) examines and substantiates top-level requirements analysis products and assesses their readiness for external review.

Timing — Occurs (as required) following the maturation of the mission requirements in the mission definition stage.

Objectives — The objectives of the review are to:

- Confirm that the mission concept satisfies the customer's needs
- Confirm that the mission requirements support identification of external and long-lead support requirements (e.g., DoD, international, facility resources)
- Determine the adequacy of the analysis products to support development of the preliminary Phase B approval package.

Criteria for Successful Completion — The following items compose a checklist to aid in determining readiness of MRR product preparation:

- Are the top-level mission requirements sufficiently defined to describe objectives in measurable parameters? Are assumptions and constraints defined and quantified?
- Is the mission and operations concept adequate to support preliminary program/project documentation development, including the Engineering Master Plan/Schedule, Phase B Project Definition Plan, technology assessment, initial Phase B/C/D resource requirements, and acquisition strategy development?
- Are evaluation criteria sufficiently defined?
- Are measures of effectiveness established?
- Are development and life-cycle cost estimates realistic?
- Have specific requirements been identified that are high risk/high cost drivers, and have options been described to relieve or mitigate them?

Results of Review — Successful completion of the MRR provides confidence to submit information for the Preliminary Non-Advocate Review and subsequent submission of the Mission Needs Statement for approval.

System Requirements Review.

Purpose — The System Requirements Review (SRR) demonstrates that the product development team understands the mission (i.e., project-level) and system-level requirements.

Timing — Occurs (as required) following the formation of the team.

Objectives — The objectives of the review are to:

- Confirm that the system-level requirements meet the mission objectives
- Confirm that the system-level specifications of the system are sufficient to meet the project objectives.

Criteria for Successful Completion — The following items compose a checklist to aid in determining readiness of SRR project preparation:

- Are the allocations contained in the system specifications sufficient to meet mission objectives?
- Are the evaluation criteria established and realistic?
- Are measures of effectiveness established and realistic?
- Are cost estimates established and realistic?
- Has a system verification concept been identified?
- Are appropriate plans being initiated to support projected system development milestones?
- Have the technology development issues been identified along with approaches to their solution?

Results of Review — Successful completion of the SRR freezes program/project requirements and leads to a formal decision by the cognizant Program Associate Administrator (PAA) to proceed with proposal request preparations for project implementation.

System Safety Review.

Purpose — System Safety Review(s) (SSR) provides early identification of safety hazards, and ensures that measures to eliminate, reduce, or control the risk associated with the hazard are identified and executed in a timely, cost-effective manner.

Timing — Occurs (as needed) in multiple phases of the project cycle.

Objectives — The objectives of the reviews are to:

- Identify those items considered as critical from a safety viewpoint
- Assess alternatives and recommendations to mitigate or eliminate risks and hazards
- Ensure that mitigation/elimination methods can be verified.

Criteria for Successful Completion — The following items comprise a checklist to aid in determining readiness of SSR product preparation:

- Have the risks been identified, characterized, and quantified if needed?
- Have design/procedural options been analyzed, and quantified if needed to mitigate significant risks?
- Have verification methods been identified for candidate options?

Result of Review — A successful SSR results in the identification of hazards and their causes in the proposed design and operational modes, and specific means of eliminating, reducing, or controlling the hazards. The methods of safety verification will also be identified prior to PDR. At CDR, a safety baseline is developed.

Software Specification Review.

Purpose — The Software Specification Review (SoSR) ensures that the software specification set is sufficiently mature to support preliminary design efforts.

Timing — Occurs shortly after the start of preliminary design.

Objectives — The review objectives are to:

- Verify that all software requirements from the system specification have been allocated to CSCIs and documented in the appropriate software specifications
- Verify that a complete set of functional, performance, interface, and verification requirements for each CSCI has been developed
- Ensure that the software requirement set is both complete and understandable.

Criteria for Successful Completion — The following items comprise a checklist to aid in determining the readiness of SoSR product preparation:

- Are functional CSCI descriptions complete and clear?
- Are the software requirements traceable to the system specification?
- Are CSCI performance requirements complete and unambiguous? Are execution time and storage requirements realistic?
- Is control and data flow between CSCIs defined?
- Are all software-to-software and software-to-hardware interfaces defined?
- Are the mission requirements of the system and associated operational and support environments defined? Are milestone schedules and special delivery requirements negotiated and complete?
- Are the CSCI specifications complete with respect to design constraints, standards, quality assurance, testability, and delivery preparation?

Results of Review — Successful completion of the SoSR results in release of the software specifications based upon their development requirements and guidelines, and the start of preliminary design activities.

Test Readiness Review.

Purpose — The Test Readiness Review (TRR) ensures that the test article hardware/software, test facility, ground support personnel, and test procedures are ready for testing, and data acquisition, reduction, and control.

Timing — Held prior to the start of a formal test. The TRR establishes a decision point to proceed with planned verification (qualification and/or acceptance) testing of CIs, subsystems, and/or systems.

Objectives — The objectives of the review are to:

- Confirm that in-place test plans meet verification requirements and specifications
- Confirm that sufficient resources are allocated to the test effort
- Examine detailed test procedures for completeness and safety during test operations
- Determine that critical test personnel are test- and safety-certified
- Confirm that test support software is adequate, pertinent, and verified.

Criteria for Successful Completion — The following items comprise a checklist to aid in determining the readiness of TRR product preparation:

- Have the test cases been reviewed and analyzed for expected results? Are results consistent with test plans and objectives?
- Have the test procedures been "dry run"? Do they indicate satisfactory operation?
- Have test personnel received training in test operations and safety procedures? Are they certified?
- Are resources available to adequately support the planned tests as well as contingencies, including failed hardware replacement?
- Has the test support software been demonstrated to handle test configuration assignments, and data acquisition, reduction, control, and archiving?

Results of Review — A successful TRR signifies that test and safety engineers have certified that preparations are complete, and that the project manager has authorized formal test initiation.

Production Readiness Review.

Purpose — The Production Readiness Review (ProRR) ensures that production plans, facilities, and personnel are in place and ready to begin production.

Timing — After design certification and prior to the start of production.

Objectives — The objectives of the review are to:

- Ascertain that all significant production engineering problems encountered during development are resolved
- Ensure that the design documentation is adequate to support manufacturing/fabrication
- Ensure that production plans and preparations are adequate to begin manufacturing/fabrication
- Establish that adequate resources have been allocated to support end item production.

Criteria for Successful Completion — The following items comprise a checklist to aid in determining the readiness of ProRR product preparation:

- Is the design certified? Have incomplete design elements been identified?
- Have risks been identified and characterized, and mitigation efforts defined?
- Has the bill of materials been reviewed and critical parts been identified?
- Have delivery schedules been verified?
- Have alternative sources been identified?
- Have adequate spares been planned and budgeted?
- Are the facilities and tools sufficient for end item production? Are special tools and test equipment specified in proper quantities?
- Are personnel qualified?
- Are drawings certified?
- Is production engineering and planning mature for cost-effective production?
- Are production processes and methods consistent with quality requirements? Are they compliant with occupational safety, environmental, and energy conservation regulations?

Results of Review — A successful ProRR results in certification of production readiness by the project manager and involved specialty engineering organizations. All open issues should be resolved with closure actions and schedules.

Design Certification Review.

Purpose — The Design Certification Review (DCR) ensures that the qualification verifications demonstrated design compliance with functional and performance requirements.

Timing — Follows the system CDR, and after qualification tests and all modifications needed to implement qualification-caused corrective actions have been completed.

Objectives — The objectives of the review are to:

- Confirm that the verification results met functional and performance requirements, and that test plans and procedures were executed correctly in the specified environments
- Certify that traceability between test article and production article is correct, including name, identification number, and current listing of all waivers
- Identify any incremental tests required or conducted due to design or requirements changes made since test initiation, and resolve issues regarding their results.

Criteria for Successful Completion — The following items comprise a checklist to aid in determining the readiness of DCR product preparation:

- Are the pedigrees of the test articles directly traceable to the production units?
- Is the verification plan used for this article current and approved?
- Do the test procedures and environments used comply with those specified in the plan?
- Are there any changes in the test article configuration or design resulting from the as-run tests? Do they require design or specification changes, and/or retests?
- Have design and specification documents been audited?
- Do the verification results satisfy functional and performance requirements?
- Do the verification, design, and specification documentation correlate?

Results of Review — As a result of a successful DCR, the end item design is approved for production. All open issues should be resolved with closure actions and schedules.

Functional and Physical Configuration Audits. The Physical Configuration Audit (also known as a configuration inspection) 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 verifies that the acceptance test results are consistent with the test requirements previously approved at the PDR and CDR. It ensures that the test results indicate performance requirements were met, and test plans and procedures were executed correctly. It should also document differences between the test unit and production unit, including any waivers.

4.9 Status Reporting and Assessment

An important part of systems engineering planning is determining what is needed in time, resources, and people to realize the system that meets the desired goals and objectives. Planning functions, such as WBS preparation,

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

These processes together form the feedback loop depicted in Figure 20. This loop takes place on a continual basis throughout the project life cycle.

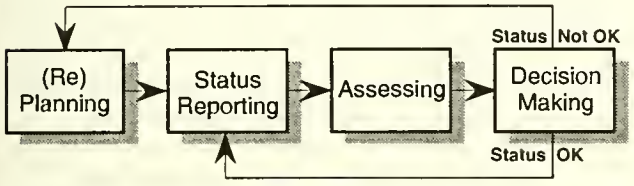


Figure 20 — Planning and Status Reporting Feedback Loop.

This loop is applicable at each level of the project hierarchy. Planning data, status reporting data, and assessments flow up the hierarchy with appropriate aggregation at each level; decisions cause actions to be taken down the hierarchy. Managers at each level determine (consistent with policies established at the next higher level of the project hierarchy) how often, and in what form, reporting data and assessments should be made. In establishing these status reporting and assessment requirements, some principles of good practice are:

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

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

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

4.9.1 Cost and Schedule Control Measures

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

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

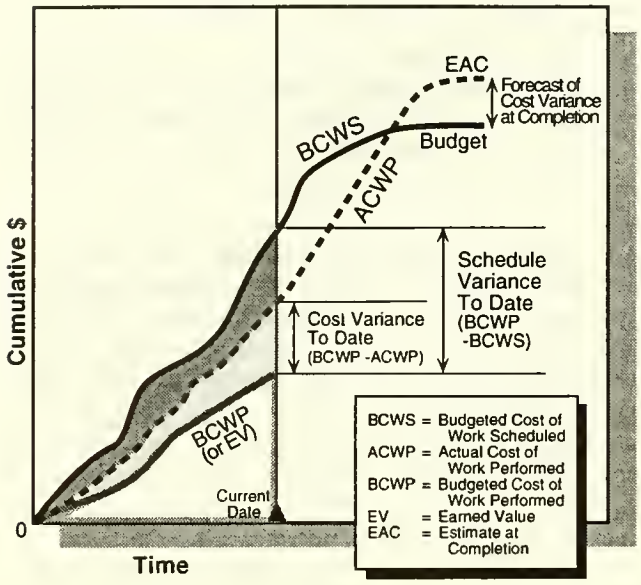


Figure 21 — Cost and Schedule Variances.

Assessment Methods. The traditional method of cost and schedule control is to compare 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 21 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*.

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 life cycle. (See sidebar on computing EAC.)

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

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

The first action taken to control an excessive negative variance is to have the cognizant manager or system 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, fire, or other calamity.

Computing the Estimate at Completion

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

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

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

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

4.9.2 Technical Performance Measures

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

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

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

In selecting TPMs, the system engineer should focus on those that can be objectively measured during the project life 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

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. (See sidebar on Learning Curve Theory.)

exercised using demonstrated values. As the project life cycle proceeds through Phases C and D, the measurement of TPMs should become increasingly more accurate because of the availability of more "actual" data about the system.

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

Assessment Methods. The traditional method of assessing a TPM is to establish a time-phased *planned profile* for it, and then to compare 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 specification). The planned profile method is the technical performance measurement counterpart to the Earned Value method for cost and schedule control described earlier.

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

Depending on which method is chosen, the system engineer's role is to propose reasonable planned profiles or margin requirements for approval by the cognizant manager. The value of either of these methods is that they allow management by exception — that is, only deviations from planned profiles or margins below requirements signal potential future problems requiring replanning. If this occurs, then new cost, schedule, and/or technical changes should be proposed. Technical changes may imply some new planned profiles. This is illustrated for a hypothetical TPM in Figure 22(a). In this example, a significant dem-

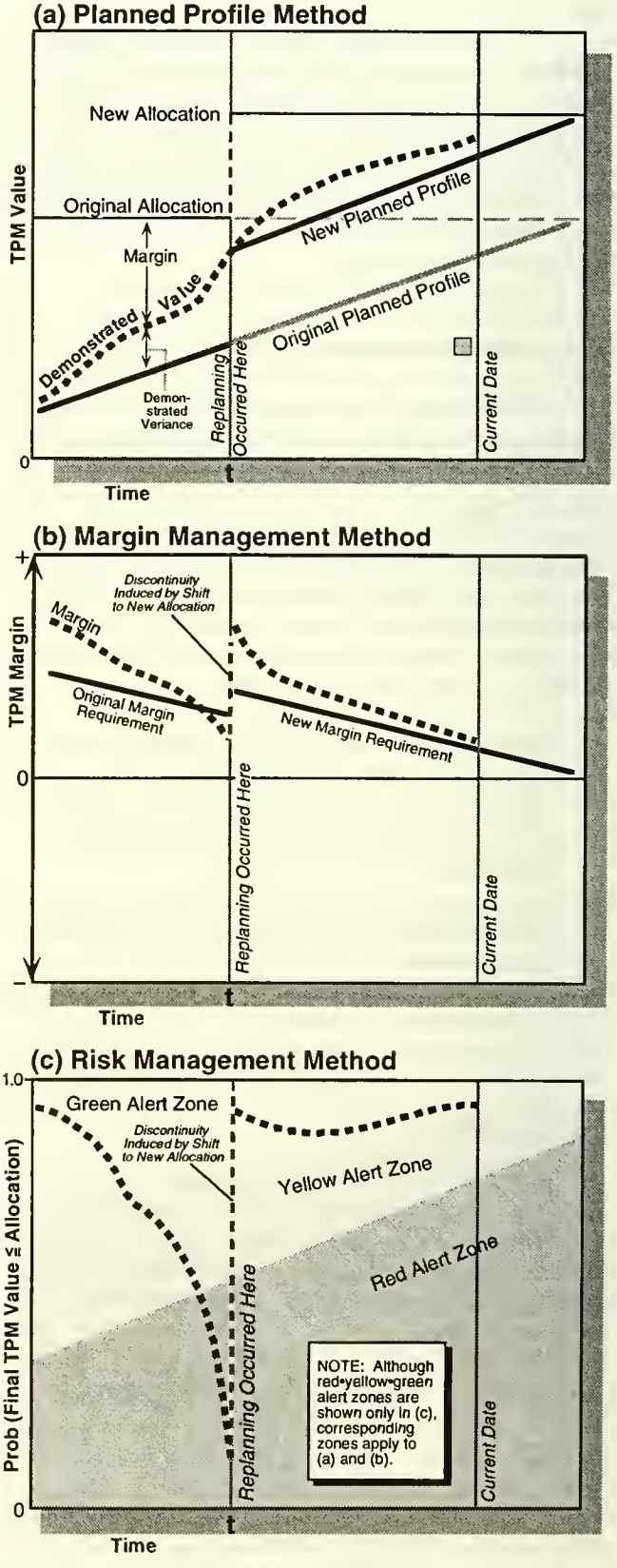


Figure 22 — Three TPM Assessment Methods.

onstrated 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 22(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 22(c). The replanning at time *t* occurred when the probability of the final TPM value being less than the allocation fell precipitously into the red alert zone. The new higher allocation for the TPM resulted in a substantial improvement in that probability.

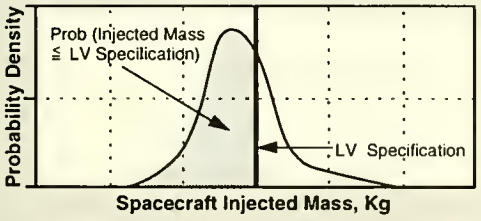
The risk management method requires an estimate of the probability distribution for the final TPM value. (See sidebar on tracking spacecraft mass.) Early in the TPM tracking program, when the demonstrated value is based on indirect means of estimation, this distribution typically has a larger statistical variance than later, when it is based on measured data, such as 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 22(c)) despite its growth. The three methods represent different ways to assess TPMs and communicate that information to management, but whichever is chosen, the pattern of success or failure should be the same for all three.

Relationship of TPM Tracking Program to the SEMP. The SEMP is the usual document for describing the project's TPM tracking program. This description should include a master list of those TPMs to be tracked, and the measurement and assessment methods to be employed. If analytical methods and models are used to measure certain high-level TPMs, then these need to be identified. The reporting frequency and timing of assessments should be specified as well. In determining these, the system engineer must balance the project's needs for accurate, timely, and effective TPM tracking against the cost of the TPM

An Example of the Risk Management Method for Tracking Spacecraft Mass

During Phases C and D, a spacecraft's injected mass can be considered an uncertain quantity. Estimates of each subsystem's and each instrument's mass are, however, made periodically by the design engineers. These estimates change and become more accurate as actual parts and components are built and integrated into subsystems and instruments. Injected mass can also change during Phases C and D as the quantity of propellant is fine-tuned to meet the mission design requirements. Thus at each point during development, 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 20(c).

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

tracking program. The TPM tracking program plan, which elaborates on the SEMP, should specify each TPM's allocation, time-phased planned profile or margin requirement, and alert zones, as appropriate to the selected assessment method.

4.9.3 Systems Engineering Process Metrics

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

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

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

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

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

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

Table 2 — Systems Engineering Process Metrics.

Function	Systems Engineering Process Metric	Category
Requirements development and management	Requirements identified vs. completed vs. approved	S
	Requirements volatility	Q
	Trade studies planned vs. completed	S
	Requirements approved per systems engineering hour	P
Design and Development	Specifications planned vs. completed	S
	Processing of ECRs/ECOs	Q
	Engineering drawings planned vs. released	S
Verification and validation (V&V)	V&V plans identified vs. approved	S
	V&V procedures planned vs. completed	S
	Functional requirements approved vs. verified	S
	V&V plans approved per systems engineering hour	P
	Processing of Problem/Failure 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-related

newly identified requirements, or as the number of changes to already-approved requirements. As another example, Engineering Change Request (ECR) processing could be tracked by comparing cumulative ECRs opened versus cumulative ECRs closed, or by plotting the age profile of open ECRs, or by examining the number of ECRs opened last month versus the total number open. The system engineer should apply his/her own judgment in picking the status reporting and assessment method.

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 successfully completed project of the same type. The latter provides a benchmark against which the system engineer can judge his/her own efforts.

5 Systems Analysis and Modeling Issues

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

Systems Analysis

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

5.1 The Trade Study Process

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

Figure 23 shows the trade study process in simplest terms, beginning with the step of *defining the system's*

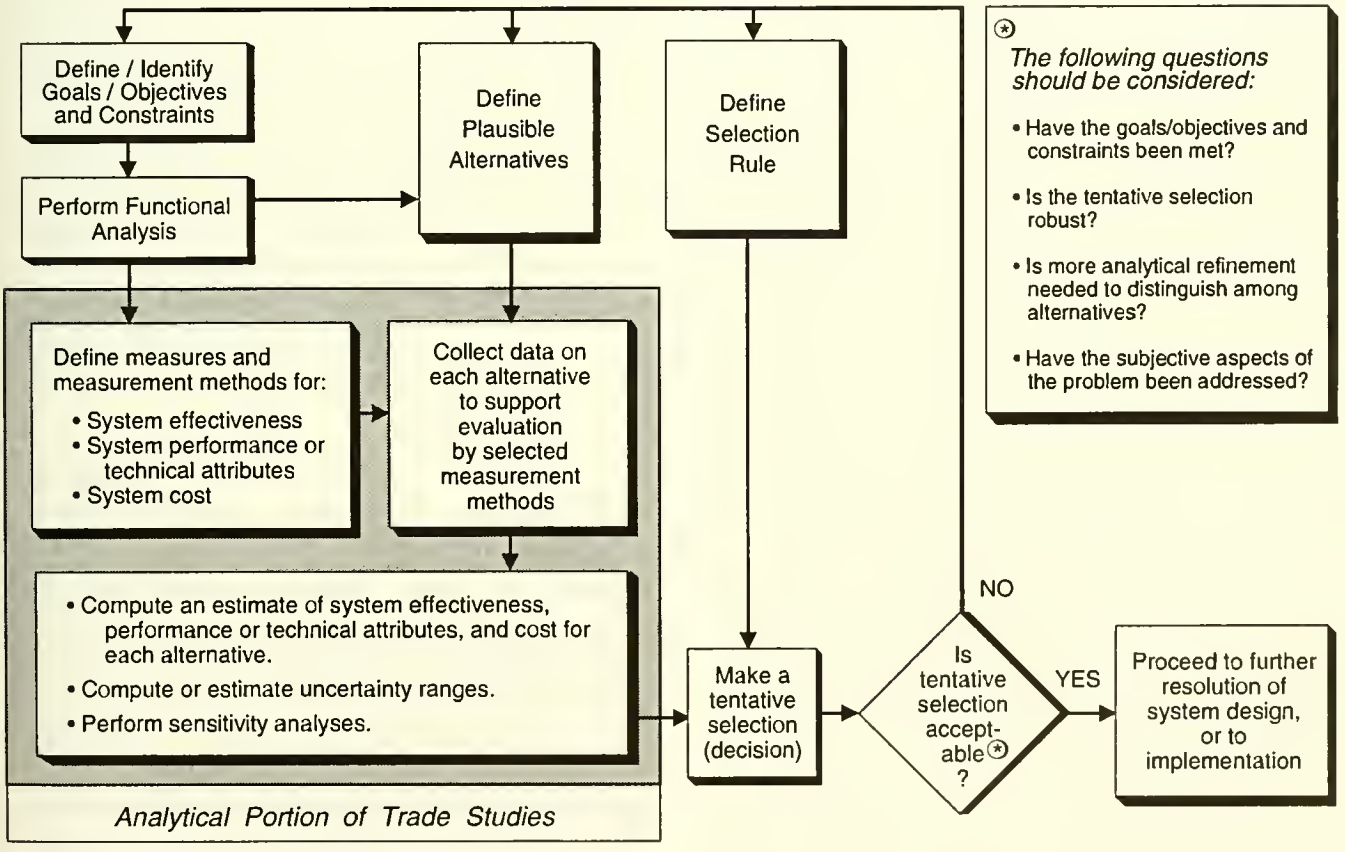


Figure 23 — The Trade Study Process.

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

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

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

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

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

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

Defining the selection rule is the step of explicitly determining how the outcome variables will be used to make a (tentative) selection of the preferred alternative. As an example, a selection rule may be to choose the alternative with the highest estimated system effectiveness that

Functional Analysis Techniques

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

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

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

costs less than x dollars (with some given probability), meets safety requirements, and possibly meets other political or schedule constraints. Defining the selection rule is essentially deciding how the selection is to be made. This step is independent from the actual measurement of system effectiveness, system performance or technical attributes, and system cost.

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

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

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

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

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

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

In an ideal world, all input values would be precisely known, and models would perfectly predict outcome variables. This not being the case, the system engineer should supplement point estimates of the outcome variables for each alternative with computed or estimated uncertainty ranges. For each uncertain key input, a range of values should be estimated. Using this range of input values, the sensitivity of the outcome variables can be gauged, and their uncertainty ranges calculated. The system engineer may be able to obtain *meaningful probability distributions* for the outcome variables using Monte Carlo simulation (see Section 5.4.2), but when this is not feasible, the system engineer must be content with only ranges and sensitivities.

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

This tentative selection should not be accepted blindly. In most trade studies, there is a need to subject the results to a "reality check" by considering a number of questions. Have the goals, objectives, and constraints truly been met? Is the tentative selection heavily dependent on a particular set of input values to the measurement methods, or does it hold up under a range of reasonable input values? (In the latter case, the tentative selection is

said to be robust.) Are there sufficient data to back up the tentative selection? Are the measurement methods sufficiently discriminating to be sure that the tentative selection is really better than other alternatives? Have the subjective aspects of the problem been fully addressed?

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

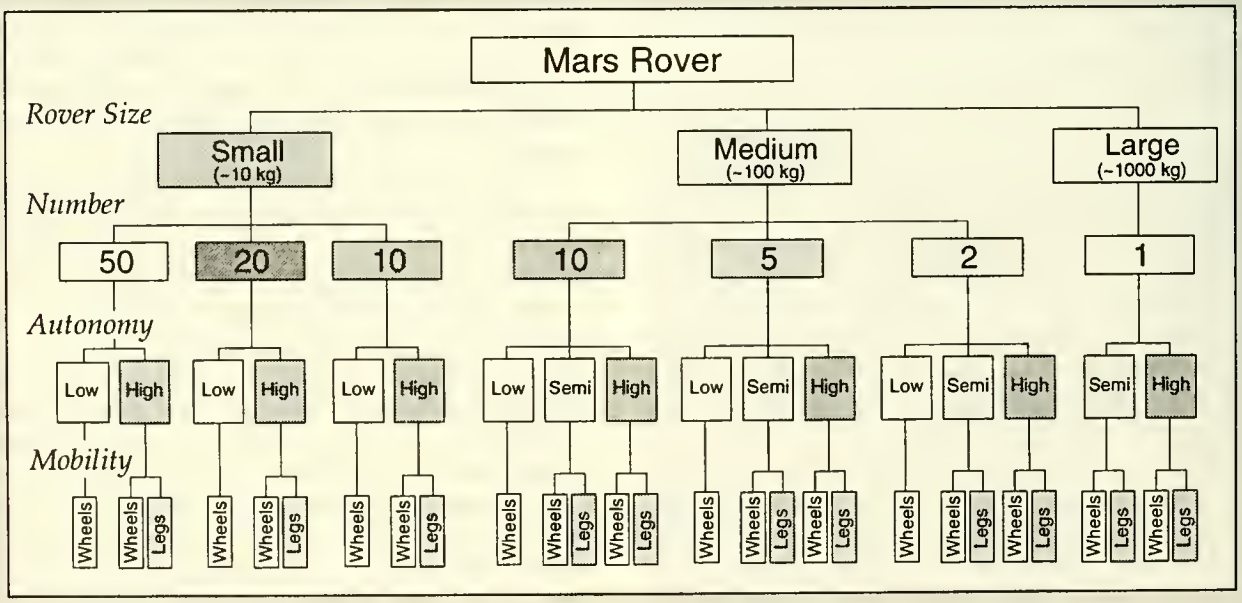
If the reality check is not met, the trade study process returns to one or more earlier steps. This iteration may result in a change in the goals, objectives, and constraints, a new alternative, or a change in the selection rule, based on the new information generated during the trade study. The reality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

5.1.1 Controlling the Trade Study Process

There are a number of mechanisms for controlling the trade study process. The most important one is the Systems Engineering Management Plan (SEMP). The SEMP specifies the major trade studies that are to be performed during each phase of the project life cycle. It

An Example of a Trade Tree for a Mars Rover

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



Trade Study Reports

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

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

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

should also spell out the general contents of trade study reports, which form part of the *decision support packages* (i.e., documentation submitted in conjunction with formal reviews and change requests).

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

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

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

5.1.2 Using Models

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

- An abstraction of reality designed to answer specific questions about the real world
- An imitation, analogue, or representation of a real-world process or structure; or
- A conceptual, mathematical, or physical tool to assist a decision maker.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Both relevance and credibility are crucial to the acceptance of a model for use in trade studies. Relevance is determined by how well a model addresses the substantive cost-effectiveness issues in the trade study. A model's credibility results from the logical consistency of its mathematical relationships, and a history of successful (i.e., cor-

rect) predictions. A history of successful predictions lends credibility to a model, but full validation — proof that the model's prediction is in accord with reality — is very difficult to attain since observational evidence on those predictions is generally very scarce. While it is certainly advantageous to use tried-and-true models that are often left as the legacy of previous projects, this is not always possible. Systems that address new problems often require that new models be developed for their trade studies. In that case, full validation is out of the question, and the system engineer must be content with models that have logical consistency and some limited form of outside, independent corroboration.

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

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

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

5.1.3 Selecting the Selection Rule

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

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

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

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

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

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

Table 3 first divides selection rules according to the importance of uncertainty in the trade study. This division is reflective of two different classes of decision problems — decisions to be made under conditions of certainty, and decisions to be made under conditions of uncertainty. Uncertainty is an inherent part of systems engineering, but the distinction may be best explained by reference to Figure 2, which is repeated here as Figure 24. In the former class, the measures of system effectiveness, performance or tech-

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

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

nical attributes, and system cost for the alternatives in the trade study look like those for alternative B. In the latter class, they look like those for alternative C. When they look like those for alternative A, conditions of uncertainty should apply, but often are not treated that way.

The table further divides each of the above classes of decision problems into two further categories: those that apply when cost and effectiveness measures are scalar quantities, and thus suffice to guide the system engineer to the best alternative, and those that apply when cost and effectiveness cannot be represented as scalar quantities.

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

Another selection rule is to choose the alternative that *maximizes effectiveness for a given level of cost.* This rule is applicable when system effectiveness and system cost can be unambiguously measured, and the appropriate level of cost is known. Since the purpose of the selection rule is to compare and rank the alternatives, practical application requires that each of the alternatives be placed on an equal cost basis. For certain types of trade studies, this

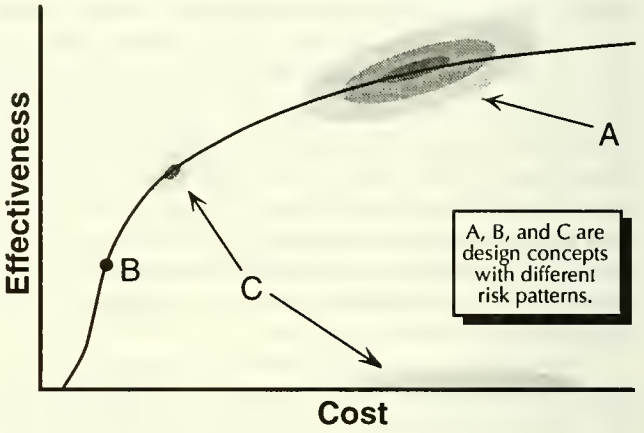


Figure 24 — Results of Design Concepts with Different Risk Patterns.

does not present a problem. For example, changing system size or output, or the number of platforms or instruments, may suffice. In other types of trade studies, this may not be possible.

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

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

Another group of selection rules is needed when cost and/or effectiveness cannot be represented as scalar quantities. To choose the best alternative, a *multi-objective* selection rule is needed. A multi-objective rule seeks to select the alternative that, in some sense, represents the best balance among competing objectives. To accomplish this, each alternative is measured (by some quantitative

method) in terms of how well it achieves each objective. For example, the objectives might be national prestige, upgrade or expansion potential, science data return, low cost, and potential for international partnerships. Each alternative's "scores" against the objectives are then combined in a value function to yield an overall figure of merit for the alternative. The way the scores are combined should reflect the decision maker's preference structure. The alternative that *maximizes the value function* (i.e., with the highest figure of merit) is then selected. *In essence, this selection rule recasts a multi-objective decision problem into one involving a single, measurable objective.*

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

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

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

The Analytic Hierarchy Process

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

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

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

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

Multi-Attribute Utility Theory

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

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

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

studies in which the alternatives cannot be described by a set of continuous mathematical relationships.

Selection Rules When Uncertainty Predominates. When the measures of system effectiveness, performance or technical attributes, and system cost for the alternatives in the trade study look like those for alternative C in Figure 22, the selection of the best alternative may need to be handled differently. This is because of the general propensity of decision makers to show risk-averse behavior when dealing with large variations in cost and/or effectiveness outcomes. In such cases, the *expected value* (i.e., the mean) of some stochastic outcome variable is not a satisfactory point measure of that variable.

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

The basis for the von Neumann-Morgenstern selection rule is a set of mathematical axioms about how individuals should behave when confronted by uncertainty. Practical application of this rule requires an ability to enumerate each "state of nature" (hereafter, simply called "state"), knowledge of the outcome associated with each enumerated state for each alternative, the probabilities for the various states, and a mathematical expression for the decision maker's utility function. This selection rule has also found use in the evaluation of system procurement alternatives. See Section 4.6.3 for a discussion of some related topics, including decision analysis, decision trees, and probabilistic risk assessment.

Another selection rule for this class of decision problem is called the *minimax rule.* To apply it, the system engineer computes a loss function for each enumerated state for each alternative. This rule chooses the alternative that *minimizes the maximum loss.* Practical application re-

quires an ability to enumerate each state and define the loss function. Because of its "worst case" feature, this rule has found some application in military systems.

5.1.4 Trade Study Process: Summary

System architecture and design decisions will be made. The purpose of the trade study process is to ensure that they move the design toward an optimum. The basic steps in that process are:

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

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

5.2 Cost Definition and Modeling

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

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

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

5.2.1 Life-Cycle Cost and Other Cost Measures

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

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

What Costs Should be Counted. The most comprehensive measure of the cost of an alternative is its life-cycle cost. According to NHB 7120.5, a system's life-cycle cost is "the total of the direct, indirect, recurring, nonrecurring, and other related costs incurred, or estimated to be incurred in the design, development, production, operation, maintenance, support, and retirement [of it] over its planned life span." A less formal definition of a system's life-cycle cost is the total cost of acquiring, owning, and disposing of it over its entire lifetime. System life-cycle cost should be estimated and used in the evaluation of alternatives during trade studies. The system engineer should include in the

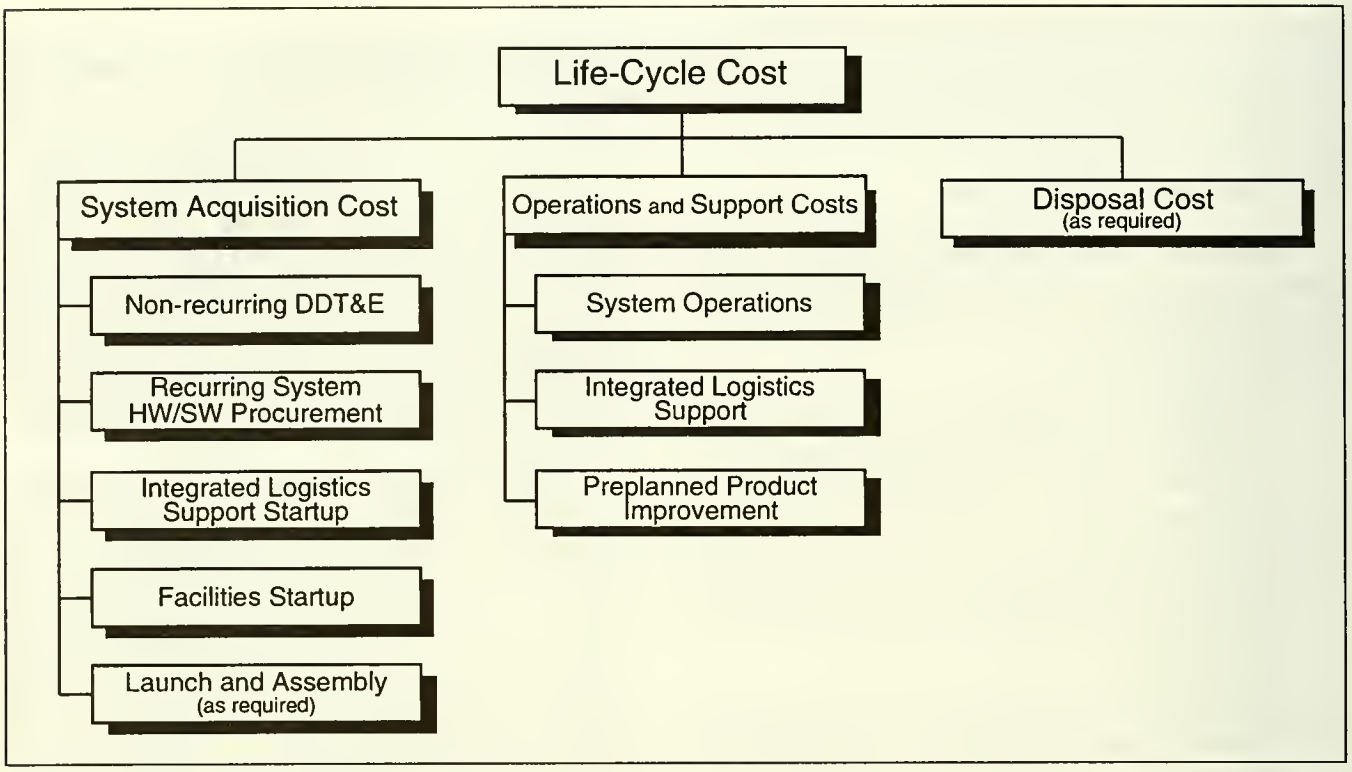


Figure 25 — Life-Cycle Cost Components.

life-cycle cost those resources, like civil service work-years, that may not require explicit project expenditures. A system's life-cycle cost, when properly computed, is the best measure of its cost to NASA.

Life-cycle cost has several components, as shown in Figure 25. Applying the informal definition above, life-cycle cost consists of (a) the costs of acquiring a usable system, (b) the costs of operating and supporting it over its useful life, and (c) the cost of disposing of it at the end of its useful life. The system acquisition cost includes more than the DDT&E and procurement of the hardware and software; it also includes the other start-up costs resulting from the need for initial training of personnel, initial spares, the system's technical documentation, support equipment, facilities, and any launch services needed to place the system at its intended operational site.

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

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

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

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

Calculating Present Discounted Value

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

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

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

Once the yearly costs have been estimated, the choice of the discount rate is crucial to the evaluation since it ultimately affects how much or how little runout costs contribute to the PDV. While calculating the PDV is generally accepted as the way to deal with costs occurring over a period of years, there is much disagreement and confusion over the appropriate discount rate to apply in systems engineering trade studies. The Office of Management and Budget (OMB) has mandated the use of a rate of seven percent for NASA systems when constant dollars (dollars adjusted to the price level as of some fixed point in time) are used in the equation. When nominal dollars (sometimes called then-year, runout or real-year dollars) are used, the OMB-mandated annual rate should be increased by the inflation rate assumed for that year. Either approach yields essentially the same PDV. For details, see OMB Circular A-94, *Guidelines and Discount Rates for Benefit Cost Analysis of Federal Programs,* October 1992.

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

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

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

5.2.2 Controlling Life-Cycle Costs

The project manager/system engineer must ensure that the system life-cycle cost (established at the end of Phase A) is initially compatible with NASA's budget and strategic priorities and that it *demonstratively remains so over the project life cycle.* According to NHB 7120.5, every NASA program/project must:

- Develop and maintain an effective capability to estimate, assess, monitor, and control its life-cycle cost throughout the project life cycle
- Relate life-cycle cost estimates to a well-defined technical baseline, detailed project schedule, and set of cost-estimating assumptions
- Identify the life-cycle costs of alternative levels of system requirements and capability
- Report time-phased acquisition cost and technical parameters to NASA Headquarters.

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

This suggests that efforts to acquire better information about the life-cycle cost of each alternative early in the project life-cycle (Phases A and B) potentially have very high payoffs. The system engineer needs to understand what the principal life-cycle cost drivers are. Some

major questions to consider are: How much does each alternative rely on well-understood technology? Can the system be manufactured using routine processes or are higher precision processes required? What tests are needed to verify and validate each alternative system design, and how costly are they? What reliability levels are needed by each alternative? What environmental and safety requirements must be satisfied?

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

Another way of acquiring better information on the cost of alternatives is for the project to have *independant* cost estimates prepared for comparison purposes.

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

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

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

The Space Station *Alpha* Program provides many examples of such potential trades. As one example, consider the life-cycle cost effect of increasing the mean time between failures (MTBF) of *Alpha's* Orbital Replacement Units (ORUs). This is likely to increase the acquisition cost, and may increase the mass of the station. However, annual maintenance hours and the weight of annual replacement spares will decline. The same station availability may be achieved with fewer on-orbit spares, thus saving precious internal volume used for spares storage. For ORUs external to the station, the amount of extravehicular activity, with its associated logistics support, will also decline. With such complex interactions, it is difficult to know what the optimum point is. At a minimum, the system engineer must have the capability to assess the life-cycle cost of each alternative. (See Appendix B.8 on the operations and operations cost effects of ORU MTBF and Section 6.5 on Integrated Logistics Support.)

5.2.3 Cost Estimating

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

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

Table 4 — Cost Estimating Techniques by Phase.

Technique	Pre-Phase A and Phase A	Phase B	Phase C/D
Parametric Cost Models	Primary	Applicable	May be applicable
Analogy	Applicable	Applicable	May be applicable
Grass-roots	May be applicable	Applicable	Primary

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

Statistical Cost Estimating Relationships: Example and Pitfalls

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

$$\ln C = 3.822 + 1.110 \ln W + 0.436 \ln z$$

where W is the receiver/exciter's weight, and z is the number of receiver boxes; ln is the natural logarithm function. (Source: U.S. Air Force Systems Command-Space Division, *Unmanned Space Vehicle Cost Model, Seventh Edition*, August 1994.) CERs are used extensively in advanced technology systems, and have been challenged on both theoretical and practical grounds. One challenge can be mounted on the basis of the assumption of an unchanging relationship between cost and the independent variables. Others have questioned the validity of CERs based on weight, a common independent variable in many models, in light of advances in electronic packaging and composite materials. Objections to using statistical CERs also include problems of input accuracy, low statistical significance due to limited data points, ignoring the statistical confidence bands, and, lastly, biases in the underlying data.

tance, to obtain an estimate of cost. Similarly, software CERs have been developed as well, relying on judgments about source lines of code and other factors to obtain development costs. (See sidebar on statistical CERs.)

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

The principal advantages of parametric cost models are that the results are reproducible, are more easily documented than other methods, and often can be produced with the least amount of time and effort. This makes a properly constructed performance-based parametric cost model useful in early trade studies.

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

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

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

Table 5 — Some Space Systems Parametric Cost Models.

Model	Source	Application
Unmanned Space Vehicle Cost Model (USCM)*	Air Force Materiel Command/Space and Missile Systems Center	Unmanned Earth-orbiting space vehicles DDT&E, FH, AGE, LOOS**
Programmed Review of Information for Costing and Evaluation (PRICE)	GE/RCA	PRICE/H for electronic and mechanical hardware DDT&E and production, PRICE/S for software
Model for Estimating Space Station Operations Costs (MESSOC)	Space Station Headquarters Support Office	All mature operations costs for Earth-orbiting space stations
Software Costing Tool (SCT)*	JPL	NASA manned and unmanned flight and ground software development costs
Multi-variable Instrument Cost Model (MICM)*	GSFC (Code 152.0)	Cost of developing and building prototype instruments
Small Satellite Cost Model (SSCM)*	The Aerospace Corporation	System- and subsystem-level DDT&E and FH costs of newer Class C and D Earth-orbiting small satellites
Marshall Space Flight Center Historical Cost Models*	MSFC	Subsystem-level DDT&E and FH costs for manned and unmanned spacecraft, and launch vehicles

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

called system-level costs, and are often calculated as percentages of the direct costs.

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

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

A third situation requiring additional calculation occurs when a model provides a cost estimate of the total

Learning Curve Theory

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

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

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

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

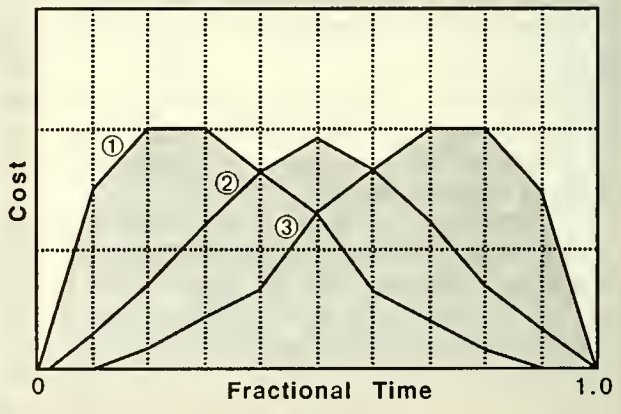
Learning curve theory may not always be applicable because, for example, the *time* rate of production has no effect on the basic equation. For more detail on learning curves, including empirical studies and tables for various learning rates, see Harold Asher, *Cost-Quantity Relationships in the Airframe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

An Example of a Cost Spreader Function: The Beta Curve

One technique for spreading estimated acquisition costs over time is to apply the *beta curve*. This fifth-degree polynomial, which was developed at JSC in the late 1960s, expresses the cumulative cost fraction as a function of the cumulative time fraction, T:

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

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



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

acquisition effort, but doesn't take into account the multi-year nature of that effort. The system engineer can use a set of "annual cost spreaders" based on the typical ramping-up and subsequent ramping-down of acquisition costs for that type of project. (See sidebar on beta curves.)

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

5.3 Effectiveness Definition and Modeling

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

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

5.3.1 Strategies for Measuring System Effectiveness

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

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

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

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

The system engineer must tailor the effectiveness measure and its measurement method to the resolution of

Practical Pitfalls in Using Effectiveness Measures in Trade Studies

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

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

This discussion so far has been predicated on one accepted measure of system effectiveness. The job of computing system effectiveness is considerably easier when the system engineer has a single measure and measurement method (model). But, as with costs, a single measure may not be possible. When it does not exist, the system engineer must fall back to computing the critical high-level, but nevertheless still underlying, system performance or technical attributes. In effect, these high-level performance or technical attributes are elevated to the status of measures of (system) effectiveness (MoEs) for trade study purposes, even though they do not represent a truly comprehensive measure of system effectiveness.

These high-level performance or technical attributes might represent one of the facets described above, or they may be only components of one. They are likely to require knowledge or estimates of lower-order performance or technical attributes. Figure 26 shows how system effectiveness might look in an hierarchical tree structure. This figure corresponds, in some sense, to Figure 25 on life-cycle cost, though rolling up by simple addition obviously does not apply to system effectiveness.

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

5.3.2 NASA System Effectiveness Measures

The facets of system effectiveness in Figure 26 are generic. Not all must apply to a particular system. The system engineer must determine which performance or technical attributes make up system effectiveness, and how they should be combined, on a system-by-system basis. Table 6 provides examples of how each facet of system effectiveness could be interpreted for specific classes of NASA flight systems. No attempt has been made to enumerate all possible performance or technical attributes, or

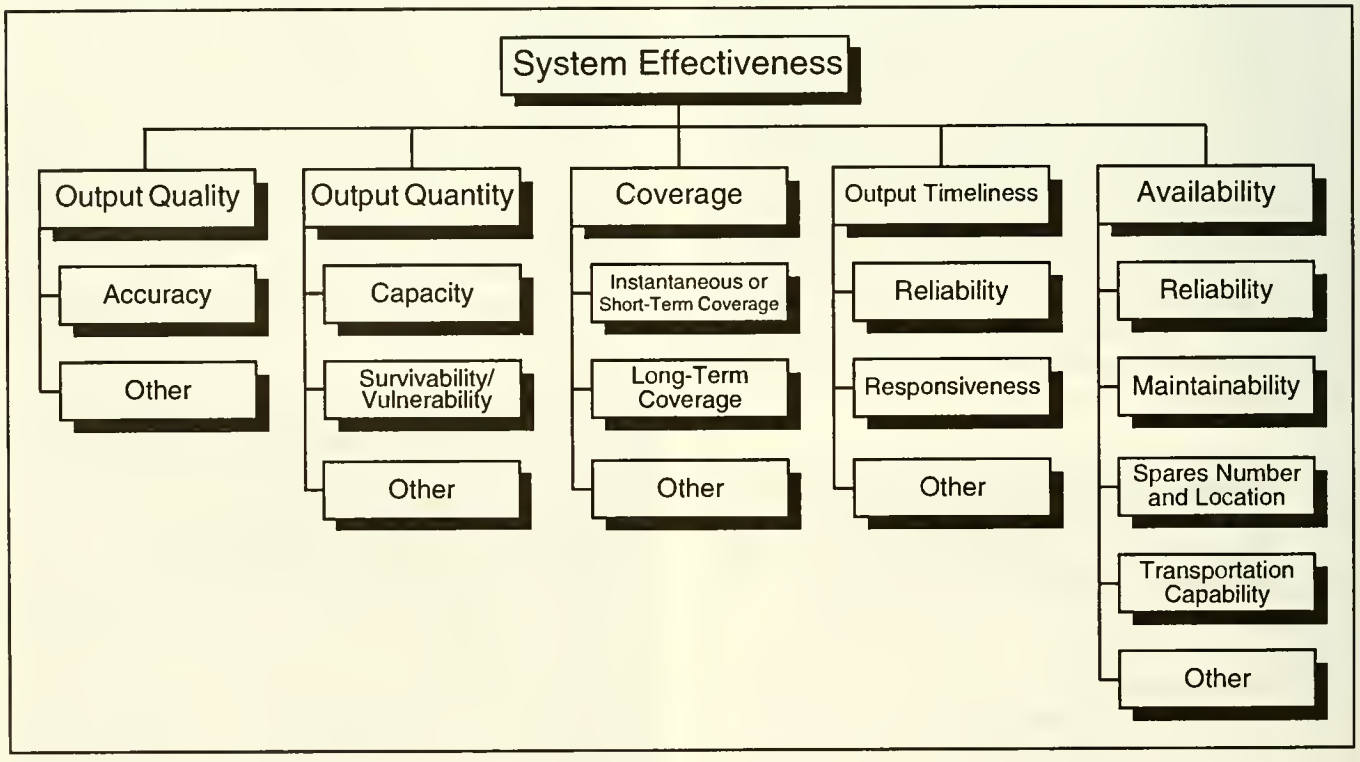


Figure 26 — System Effectiveness Components (Generic).

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

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

to fill in each possible entry in the table; its purpose is illustrative only.

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

5.3.3 Availability and Logistics Supportability Modeling

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

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

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

Measures of Availability

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

- Inherent = MTTF / (MTTF + MTTR)
- Achieved = MTTMA / (MTTMA + MMT)
- Operational = MTTMA / (MTTMA + MMT + MLDT)
  = MTTMA / (MTTMA + MDT)

where:
MTTF = Mean time to failure
MTTR = Mean time to repair (corrective)
MTTMA = Mean time to a maintenance action (corrective and preventive)
MMT = Mean (active) maintenance time (corrective and preventative)
MLDT = Mean logistics delay time (includes downtime due to administrative delays, and waiting for spares, maintenance personnel, or supplies)
MDT = Mean downtime (includes downtime due to (active) maintenance and logistics delays)

Availability measures can be also calculated at a point in time, or as an average over a period of time. A further, but manageable, complication in calculating availability takes into account degraded modes of operation for redundant systems. For systems that cannot be repaired, availability and reliability are equal. (See sidebar on page 92.)

source requirements on the system availability effects. (See sidebar on logistics supportability models.)

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

Another role for these models is to provide quantitative requirements for incorporation into the system's formal Integrated Logistics Support (ILS) Plan. Figure 27 shows the role of availability and logistics supportability models in the trade study process.

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

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

Logistics Supportability Models: Two Examples

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

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

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

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

cies. (See Section 6.5 for more on Integrated Logistics Support.)

5.4 Probabilistic Treatment of Cost and Effectiveness

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

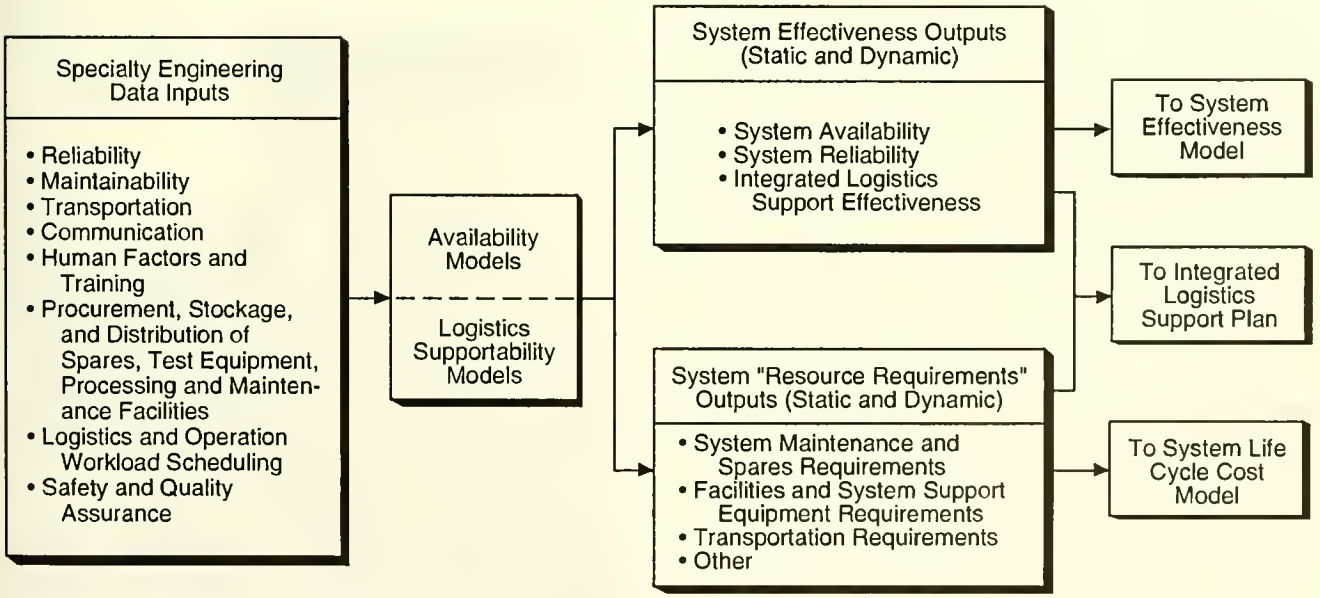


Figure 27 — Roles of Availability and Logistics Supportability Models.

5.4.1 Sources of Uncertainty in Models

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

- Uncertainty about the correctness of the model's structural equations, in particular whether the functional form chosen by the modeler is the best representation of the relationship between an equation's inputs and output
- Uncertainty in model parameters, which are, in a very real sense, also chosen by the modeler; this uncertainty is evident for model coefficients derived from statistical regression, but even known physical constants are subject to some uncertainty due to experimental or measurement error
- Uncertainty in the true value of model inputs (e.g., estimated weight or thermal properties) that describe a new system.

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

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

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

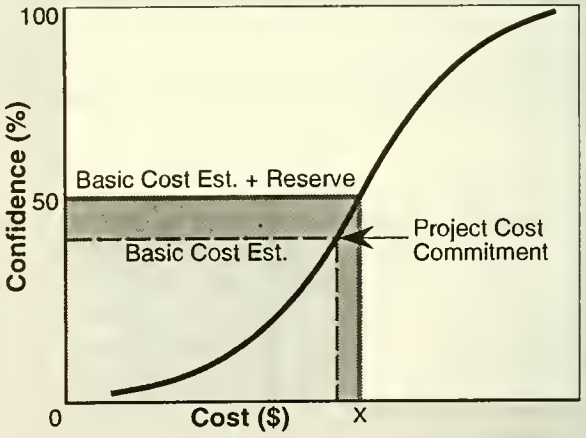
The above uncertainties are present even if the cost model inputs that describe a new system are precisely known in Phase A. This is rarely true; more often, model inputs are subject to considerable guesswork early in the project life cycle. The uncertainty in a model input can be expressed by attributing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

5.4.2 Modeling Techniques for Handling Uncertainty

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

The Cost S-Curve

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



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

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

- Analytic solution
- Decision analysis
- Monte Carlo simulation.

Analytic Solution. When the structure of a model and its uncertainties permit, a closed-form analytic solution for the required probability density (or cumulative distribution) function is sometimes feasible. Examples can be found in simple reliability models (see Figure 29).

Decision Analysis. This technique, which was discussed in Section 4.6, also can produce a cumulative distribution function, though it is necessary to descretize any continuous input probability distributions. The more probability intervals that are used, the greater the accuracy of the results, but the larger the decision tree. Furthermore, each uncertain model input adds more than linear computational complexity to that tree, making this technique less efficent in many situations than Monte Carlo simulation, described next.

Monte Carlo Simulation. This technique is often used to calculate an approximate solution to a stochastic model that is too complicated to be solved by analytic methods alone. A Monte Carlo simulation is a way of sampling input points from their respective domains in order to estimate the probability distribution of the output variable. In a simple Monte Carlo analysis, a value for each uncertain input is drawn at random from its probability distribution, which can be either discrete or continuous. This set of random values, one for each input, is used to compute the corresponding output value, as shown in Figure 28. The entire process is then repeated k times. These k output values constitute a random sample from the probability distribution over the output variable induced by the input probability distributions.

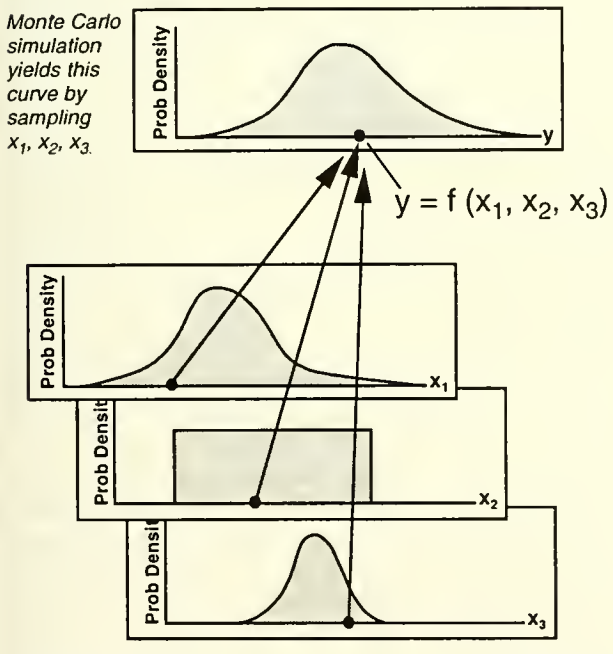


Figure 28 — A Monte Carlo Simulation with Three Uncertain Inputs.

For an example of the usefulness of this technique, recall Figures 2 (in Chapter 2) and 24 (this chapter), which show the projected cost and effectiveness of three alternative design concepts as probability "clouds." These clouds may be reasonably interpreted as the result of three system-level Monte Carlo simulations. The information displayed by the clouds is far greater than that embodied in point estimates for each of the alternatives.

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

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

Monte Carlo simulation is a fairly easy technique to apply. Also, what a particular combination of uncertainties mean can often be communicated more clearly to managers. A powerful example of this technique applied to NASA flight readiness certification is found in Moore, Ebbeler, and Creager, who combine Monte Carlo simulation with traditional reliability and risk analysis techniques.

6 Integrating Engineering Specialties Into the Systems Engineering Process

This chapter discusses the basic concepts, techniques, and products of some of the specialty engineering disciplines, and how they fit into the systems engineering process.

6.1 Role of the Engineering Specialties

Specialty engineers support the systems engineering process by applying specific knowledge and analytic methods from a variety of engineering specialty disciplines *to ensure that the resulting system is actually able to perform its mission in its operational environment.* These specialty engineering disciplines typically include reliability, maintainability, integrated logistics, test, fabrication/production, human factors, quality assurance, and safety engineering. One view of the role of the engineering specialties, then, is *mission assurance.* Part of the system engineer's job is to see that these mission assurance functions are coherently integrated into the project at the right times and that they address the relevant issues.

Another idea used to explain the role of the engineering specialties is the "Design-for-X" concept. The X stands for any of the engineering "ilities" (e.g., reliability, testability, producibility, supportability) that the project-level system engineer needs to consider to meet the project's goals/objectives. While the relevant engineering specialties may vary on NASA projects by virtue of their diverse nature, some are always needed. It is the system engineer's job to identify the particular engineering specialities needed for his/her tailored Product Development Team (PDT). The selected organizational approach to integrating the engineering specialities into the systems engineering process and the technical effort to be made should be summarized in the SEMP (Part III). Depending on the nature and scope of the project, the technical effort may also need more detailed documentation in the form of individual specialty engineering program plans.

As part of the technical effort, specialty engineers often perform tasks that are common across disciplines. Foremost, they apply specialized analytical techniques to create information needed by the project manager and system engineer. They also help define and write system requirements in their areas of expertise, and they review data packages, engineering change requests (ECRs), test results, and documentation for major project reviews. The project manager and/or system engineer need to ensure that the information and products so generated add value to the project commensurate with their cost.

The specialty engineering technical effort should also be well integrated both in time and content, not separate organizations and disciplines operating in near isolation (i.e., more like a basketball team, rather than a golf foursome). This means, as an example, that the reliability engineer's FMECA (or equivalent analysis) results are passed at the right time to the maintainability engineer, whose maintenance analysis is subsequently incorporated into the logistics support analysis (LSA). LSA results, in turn, are passed to the project-level system engineer in time to be combined with other cost and effectiveness data for a major trade study. Concurrently, the reliability engineer's FMECA results are also passed to the risk manager to incorporate critical items into the Critical Items List (CIL) when deemed necessary, and to alert the PDT to develop appropriate design or operations mitigation strategies. The quality assurance engineer's effort should be integrated with the reliability engineer's so that, for example, component failure rate assumptions in the latter's reliability model are achieved or bettered by the actual (flight) hardware. This kind of process harmony and timeliness is not easily realized in a project; it nevertheless remains a goal of systems engineering.

6.2 Reliability

Reliability can be defined as the probability that a device, product, or system will not fail for a given period of time under specified operating conditions. Reliability is an inherent system design characteristic. As a principal contributing factor in operations and support costs and in system effectiveness (see Figure 26), reliability plays a key role in determining the system's cost-effectiveness.

6.2.1 Role of the Reliability Engineer

Reliability engineering is a major specialty discipline that contributes to the goal of a cost-effective system. This is primarily accomplished in the systems engineering process through an active role in implementing specific design features to ensure that the system can perform in the predicted physical environments throughout the mission, and by making independent predictions of system reliability for design trades and for (test program, operations, and integrated logistics support) planning.

The reliability engineer performs several tasks, which are explained in more detail in NHB 5300.4(1A-1),

Reliability Relationships

The system engineer should be familiar with the following reliability parameters and mathematical relationships for continuously operated systems.

Name	Symbol	Mathematical Relationships	
Hazard Rate	$\lambda(t)$	$= -(1/R)\, dR/dt$	$= f(t) / (1 - F(t))$
Reliability	$R(t)$	$= \int_t^\infty f(\tau)\, d\tau$	$= 1 - F(t)$
Cumulative Failure Probability	$F(t)$	$= \int_0^t f(\tau)\, d\tau$	$= 1 - R(t)$
Failure Probability Density	$f(t)$	$= -dR(t)/dt$	$= \lambda(t)R(t)$

Many reliability analyses assume that failures are random so that $\lambda(t) = \lambda$ and the failure probability density follows an exponential distribution. In that case, $R(t) = \exp(-\lambda t)$, and the Mean Time To Failure (MTTF) = $1/\lambda$. Another popular assumption that has been shown to apply to many systems is a failure probability density that follows a Weibull distribution; in that case, the hazard rate $\lambda(t)$ satisfies a simple power law as a function of t. With the proper choice of Weibull parameters, the constant hazard rate can be recovered as a special case. While these (or similar) assumptions may be analytically convenient, a system's actual hazard rate may be less predictable. (Also see bathtub curve sidebar!)

Reliability Program Requirements for Aeronautical and Space System Contractors. In brief, these tasks include:

- Developing and executing a reliability program plan
- Developing and refining reliability prediction models, including associated environmental (e.g., vibration, acoustic, thermal, and EMI/EMC) models, and predictions of system reliability. These models and predictions should reflect applicable experience from previous projects.
- Establishing and allocating reliability goals and environmental design requirements
- Supporting design trade studies covering such issues as the degree of redundancy and reliability vs. maintainability
- Supporting risk management by identifying design attributes likely to result in reliability problems and recommending appropriate risk mitigations
- Developing reliability data for timely use in the project's maintainability and ILS programs
- Developing environmental test requirements and specifications for hardware qualification. The reliability engineer may provide technical analysis and justification for eliminating or relaxing qualification test requirements. These activities are usually closely coordinated with the project's verification program.
- Performing analyses on qualification test data to verify reliability predictions and validate the system reliability prediction models, and to understand and resolve anomalies
- Collecting reliability data under actual operations conditions as a part of overall system validation.

The reliability engineer works with other specialty engineers (e.g., the quality assurance, maintainability, verification, and producibility engineers) on system reliability issues. On small projects, the reliability engineer may perform some or all of these other jobs as well.

6.2.2 Reliability Program Planning

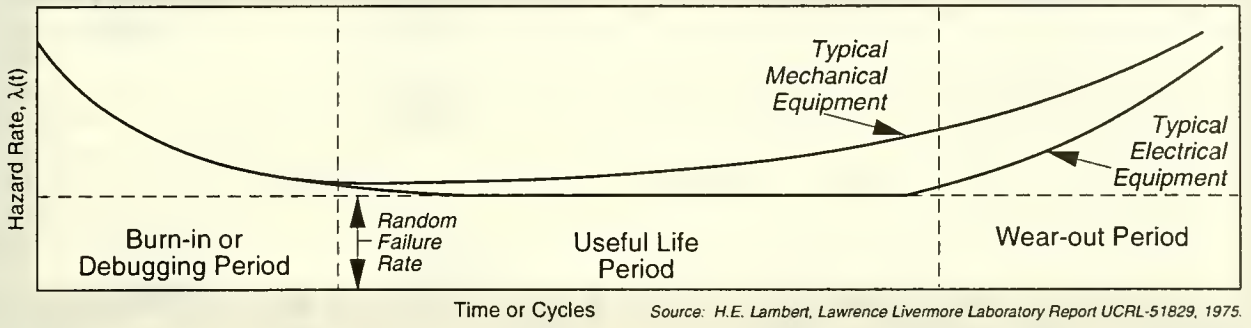
The reliability program for a project describes what activities will be undertaken in support of reliability engineering. The reliability engineer develops a reliability program considering its cost, schedule, and risk implications. This planning should begin during Phase A. The project manager/system engineer must work with the reliability engineer to develop an approprate reliability program as

Lunar Excursion Module (LEM) Reliability

Part of the reliability engineer's job is to develop an understanding of the underlying physical and human-induced causes of failures, rather than assuming that all failures are random. According to Joseph Gavin, Director of the LEM Program at Grumman, "after about 10 years of testing of individual [LEM] components and subsystems, [NASA] found something like 14,000 anomolies, only *22* of which escaped definite understanding."

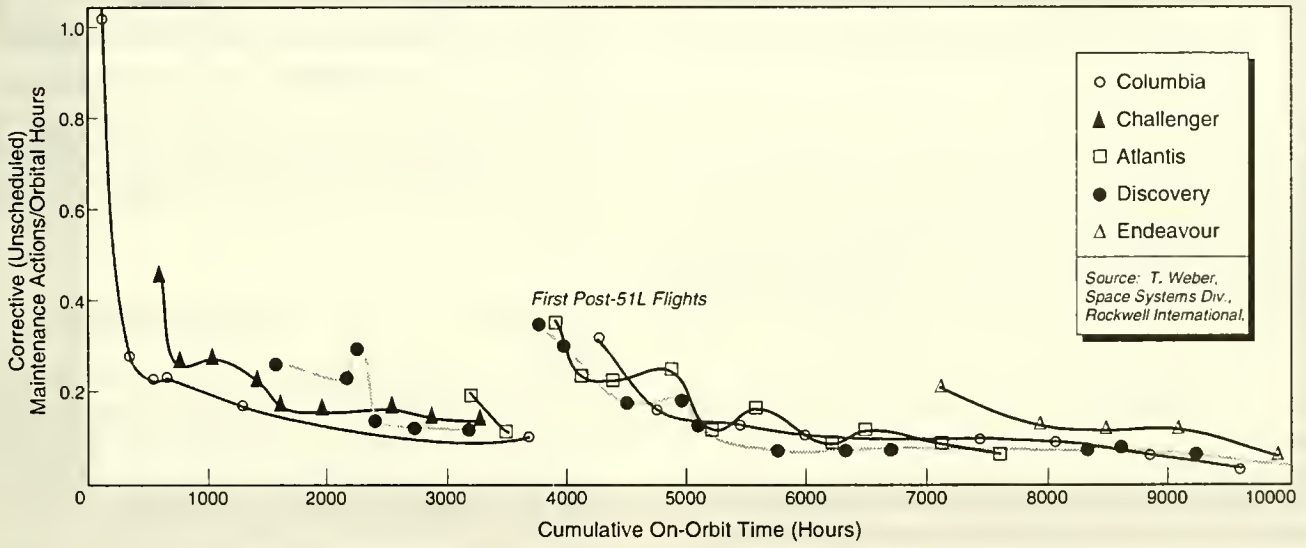
The Bathtub Curve

For many systems, the hazard rate function looks like the classic "bathtub curve" as in the graph below. Because of burn-in failures and/or inadequate quality assurance practices, $\lambda(t)$ is initially high, but gradually decreases during the infant failure rate period. During the useful life period, $\lambda(t)$ remains constant, reflecting randomly occurring failures. Later, $\lambda(t)$ begins to increase because of wearout failures. The exponential reliability formula applies only during the useful life period.



Source: H.E. Lambert, Lawrence Livermore Laboratory Report UCRL-51829, 1975.

Using up-to-date Shuttle data, the following plot was obtained. Is the bathtub curve real?



many factors need to be considered in developing this program. These factors include:

- NASA payload classification. The reliability program's analytic content and its documentation of problems and failures are generally more extensive for a Class A payload than for a Class D one. (See Appendix B.3 for classification guidelines.)
- Mission environmental risks. Several mission environmental models may need to be developed. For flight projects, these include ground (transportation and handling), launch, on-orbit (Earth or other), and planetary environments. In addition, the reliability engineer must address design and verification requirements for each such environment.
- Degree of design inheritance and hardware/software reuse.

The reliability engineer should document the reliability program in a *reliability program plan*, which should be summarized in the SEMP (Part III) and updated as needed through the project life cycle; the summary may be sufficient for small projects.

6.2.3 Designing Reliable Space-Based Systems

Designing reliable space-based systems has always been a goal for NASA, and many painful lessons have been learned along the way. The system engineer should be aware of some basic design approaches for achieving reliability. These basic approaches include *fault avoidance*, *fault tolerance*, and *functional redundancy*.

Fault Avoidance. Fault avoidance, a joint objective of the reliability engineer and quality assurance engineer (see Section 6.3), includes efforts to:

- Provide design margins, or use appropriate derating guidelines, if available
- Use high-quality parts where needed. (Failure rates for Class S parts are typically one-fourth of those procured to general military specifications.)
- Consider materials and electronics packaging carefully
- Conduct formal inspections of manufacturing facilities, processes, and documentation
- Perform acceptance testing or inspections on all parts when possible.

Fault Tolerance. Fault tolerance is a system design characteristic associated with the ability of a system to continue operating after a component failure has occurred. It is implemented by having design redundancy and a fault detection and response capability. Design redundancy can take several forms, some of which are represented in Figure 29 along with their reliability relationships.

Functional Redundancy. Functional redundancy is a system design and operations characteristic that allows the system to respond to component failures in a way sufficient to meet mission requirements. This usually involves operational work-arounds and the use of components in ways that were not originally intended. As an example, a repair of the damaged *Galileo* high-gain antenna was impossible, but a work-around was accomplished by software fixes that further compressed the science data and images; these were then returned through the low-gain antenna, although at a severely reduced data rate.

These three approaches have different costs associated with their implementation: Class S parts are typically more expensive, while redundancy adds mass, volume, costs, and complexity to the system. Different approaches to reliability may therefore be appropriate for different projects. In order to choose the best balance among approaches, the system engineer must understand the system-level effects and life-cycle cost of each approach. To achieve this, trade study methods of Section 5.1 should be used in combination with reliability analysis tools and techniques.

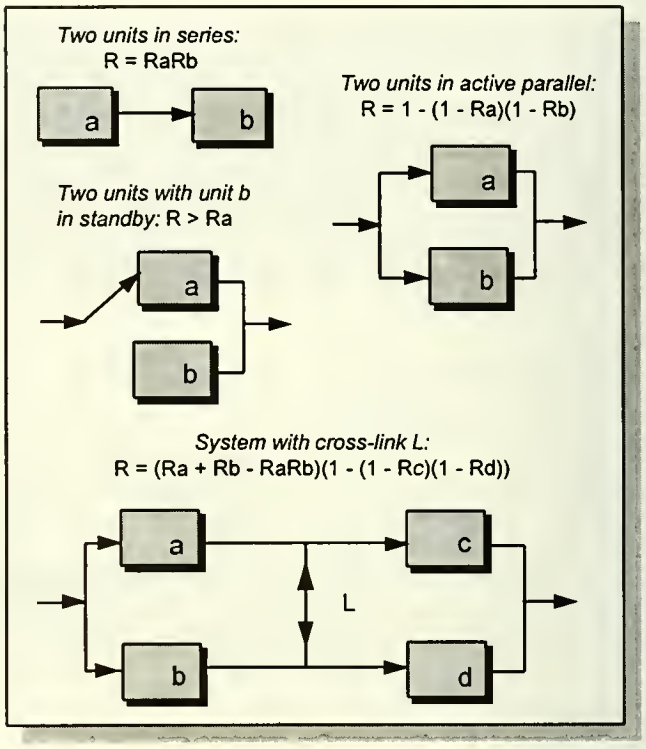


Figure 29 — Basic Reliability Block Diagrams.

6.2.4 Reliability Analysis Tools and Techniques

Reliability Block Diagrams. Reliability block diagrams are used to portray the manner in which the components of a complex system function together. These diagrams compactly describe how components are connected. Basic reliability block diagrams are shown in Figure 29.

Fault Trees and Fault Tree Analysis. A fault tree is a graphical representation of the combination of faults that will result in the occurrence of some (undesired) top event. It is usually constructed during a fault tree analysis, which is a qualitative technique to uncover credible ways the top event can occur. In the construction of a fault tree, successive subordinate failure events are identified and logically linked to the top event. The linked events form a tree structure connected by symbols called *gates*, some basic examples of which appear in the fault tree shown in Figure 30. Fault trees and fault tree analysis are often precursors to a full probabilistic risk assessment (PRA). For more on this technique, see the U.S. Nuclear Regulatory Commission *Fault Tree Handbook*.

Reliability Models. Reliability models are used to predict the reliability of alternative architectures/designs from the estimated reliability of each component. For simple systems, reliability can often be calculated by applying the rules of probability to the various components and "strings" identified in the reliability block diagram. (See Figure 29.) For more complex systems, the method of *minimal cut sets*, which relies on the rules of Boolean algebra, is often used to evaluate a system's fault tree. When individual component reliability functions are themselves uncertain, Monte Carlo simulation methods may be appropriate. These methods are described in reliability engineering textbooks, and software for calculating reliability is widely available. For a compilation of models/software, see D. Kececioglu, *Reliability, Avaliability, and Maintainability Software Handbook.*

FMECAs and FMEAs. Failure Modes, Effects, and Criticality Analysis (FMECA) and Failure Modes and Effects Analysis (FMEA) are specialized techniques for hardware failure and safety risk identification and characterization. (Also see Section 4.6.2.)

Problem/Failure Reports (P/FRs). The reliability engineer uses the Problem/Failure Reporting System (or an approved equivalent) to report reliability problems and nonconformances encountered during qualification and acceptance testing (Phase D) and operations (Phase E).

6.3 Quality Assurance

Even with the best of available designs, hardware fabrication (and software coding) and testing are subject to the vagaries of Nature and human beings. The system engineer needs to have some confidence that the system actually produced and delivered is in accordance with its functional, performance, and design requirements. Quality Assurance (QA) provides an independent assessment to the project manager/system engineer of the items produced and processes used during the project life cycle. The quality assurance engineer typically acts as the system engineer's eyes and ears in this context. The project manager/system engineer must work with the quality assurance engineer to develop a quality assurance program (the extent, responsibility, and timing of QA activities) tailored to the project it supports. As with the reliability program, this largely depends on the NASA payload classification (see Appendix B.3).

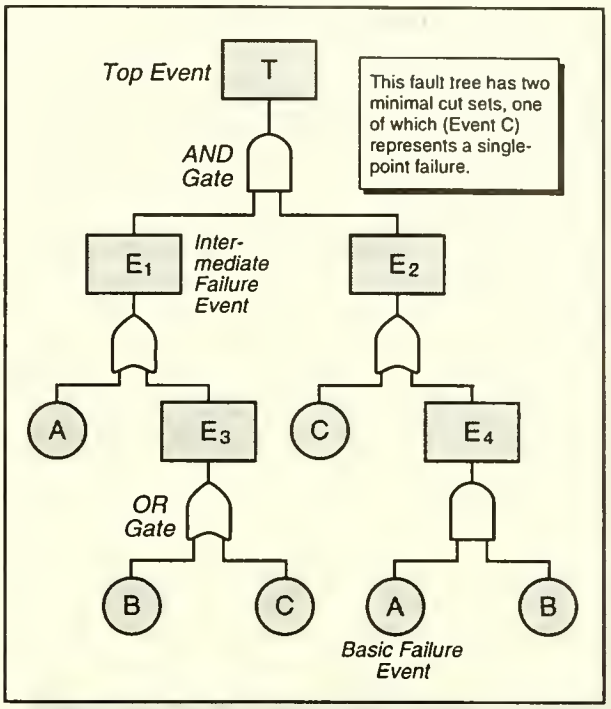


Figure 30 — A Simple Fault Tree.

6.3.1 Role of the Quality Assurance Engineer

The quality assurance engineer performs several tasks, which are explained in more detail in NHB 5300.4(1B), *Quality Program Provisions for Aeronautical and Space System Contractors.* In brief, these tasks include:

- Developing and executing a *quality assurance program plan*
- Ensuring the completeness of configuration management procedures and documentation, and monitoring the fate of ECRs/ECPs (see Section 4.7)
- Participating in the evaluation and selection of procurement sources
- Inspecting items and facilities during manufacturing/fabrication, and items delivered to NASA field centers
- Ensuring the adequacy of personnel training and technical documentation to be used during manufacturing/fabrication
- Ensuring verification requirements are properly specified, especially with respect to test environments, test configurations, and pass/fail criteria
- Monitoring qualification and acceptance tests to ensure compliance with verification requirements and

test procedures, and to ensure that test data are correct and complete
- Monitoring the resolution and close-out of nonconformances and Problem/Failure Reports (P/FRs)
- Verifying that the physical configuration of the system conforms to the "build-to" (or "code-to") documentation approved at CDR
- Collecting and maintaining QA data for subsequent failure analyses.

The quality assurance engineer also participates in major reviews (primarily SRR, PDR, CDR, and FRR) on issues of design, materials, workmanship, fabrication and verification processes, and other characteristics that could degrade product system quality.

6.3.2 Quality Assurance Tools and Techniques

PCA/FCA. The Physical Configuration Audit (PCA) verifies that the physical configuration of the system corresponds to the approved "build-to" (or "code-to") documentation. The Functional Configuration Audit (FCA) verifies that the acceptance verification (usually, test) results are consistent with the approved verification requirements. (See Section 4.8.4.)

In-Process Inspections. The extent, timing, and location of in-process inspections are documented in the quality assurance program plan. These should be conducted in consonance with the manufacturing/fabrication and verification program plans. (See Sections 6.6 and 6.7.)

QA Survey. A QA survey examines the operations, procedures, and documentation used in the project, and evaluates them against established standards and benchmarks. Recommendations for corrective actions are reported to the project manager.

Material Review Board. The Material Review Board (MRB), normally established by the project manager and chaired by the project-level quality assurance engineer, performs formal dispositions on nonconformances.

6.4 Maintainability

Maintainability is a system design characteristic associated with the ease and rapidity with which the system can be retained in operational status, or safely and economically restored to operational status following a failure. Often used (though imperfect) measures of maintainability include mean maintainance downtime, maintenance effort (workhours) per operating hour, and annual maintenance cost. However measured, maintainability arises from many factors: the system hardware and software design, the required skill levels of maintenance personnel, adequacy of diagnostic and maintenance procedures, test equipment effectiveness, and the physical environment under which maintenance is performed.

6.4.1 Role of the Maintainability Engineer

Maintainability engineering is another major specialty discipline that contributes to the goal of a cost-effective system. This is primarily accomplished in the systems engineering process through an active role in implementing specific design features to facilitate safe maintenance actions in the predicted physical environments, and through a central role in developing the integrated logistics support (ILS) system. (See Section 6.5 on ILS.)

The maintainability engineer performs several tasks, which are explained in more detail in NHB 5300.4(1E), *Maintainability Program Requirements for Space Systems*. In brief, these tasks include:

- Developing and executing a *maintainability program plan*. This is usually done in conjunction with the ILS program plan.
- Developing and refining the system maintenance concept as a part of the ILS concept
- Establishing and allocating maintainability requirements. These requirements should be consistent with the maintenance concept and traceable to system-level availability objectives.
- Performing an engineering design analysis to identify maintainability design deficiencies
- Performing analyses to quantify the system's maintenance resource requirements, and documenting them in the Maintenance Plan
- Verifying that the system's maintainability requirements and maintenance-related aspects of the ILS requirements are met
- Collecting maintenance data under actual operations conditions as part of ILS system validation.

Many of the analysis tasks above are accomplished as part of the Logistics Support Analysis (LSA), described in Section 6.5.3. The maintainability engineer also participates in and contributes to major project reviews on the above items as appropriate to the phase of the project.

6.4.2 The System Maintenance Concept and Maintenance Plan

As the system operations concept and user requirements evolve, so does the ILS concept. Central to the latter is the system maintenance concept. It serves as the basis for establishing the system's maintainability design requirements and its logistics support resource requirements (through the LSA process). In developing the system maintenance concept, it is useful to consider the mission profile, how the system will be used, its operational availability goals, anticipated useful life, and physical environments.

Traditionally, a description of the system maintenance concept is hardware-oriented, though this need not always be so. The system maintenance concept is typically described in terms of the anticipated levels of maintenance (see sidebar on maintenance levels), general repair policies regarding corrective and preventive maintenance, assumptions about supply system responsiveness, the availability of new or existing facilities, and the maintenance environment. Initially, the system maintenance concept may be based on experience with similar systems, but it should not be exempt from trade studies early in the project life cycle. These trade studies should focus on the cost-effectiveness of alternative maintenance concepts in the context of overall system optimization.

Maintenance Levels for Space Station *Alpha*

As with many complex systems, the maintenance concept for *Alpha* calls for three maintenance levels: *organizational, intermediate,* and *depot* (or *vendor*). The system engineer should be familiar with these terms and the basic characteristics associated with each level. As an example, consider *Alpha*:

Level	Work Performed	Spares
Organizational	On-orbit crew performs ORU remove-and-replace, visual inspections, minor servicing and calibration.	Few.
Intermediate	KSC maintenance facility repairs ORUs, performs detailed inspections, servicing, calibrations, and some modifications.	Extensive.
Depot/ Vendor	Factory performs major overhauls, modifications, and complex calibrations.	More extensive, or fabricated as needed.

The Maintenance Plan, which appears as a major technical section in the Integrated Logistics Support Plan (ILSP), documents the system maintenance concept, its maintenance resource requirements, and supporting maintainability analyses. The Maintenance Plan provides other inputs to the ILSP in the areas of spares, maintenance facilities, test and support equipment, and, *for each level of maintenance*, it provides maintenance training programs, facilities, technical data, and aids. The supporting analyses should establish the feasibility and credibility of the Maintenance Plan with aggregate estimates of corrective and preventive maintenance workloads, initial and recurring spares provisioning requirements, and system availability. Aggregate estimates should be the result of using best-practice maintainability analysis tools and detailed maintainabilty data suitable for the LSA. (See Section 6.5.3.)

6.4.3 Designing Maintainable Space-Based Systems

Designing NASA space-based systems for maintainability will be even more important in the future. For that reason, the system engineer should be aware of basic design features that facilitate IVA and EVA maintenance. Some examples of good practice include:

- Use coarse and fine installation alignment guides as necessary to assure ease of Orbital Replacement Unit (ORU) installation and removal
- Have minimum sweep clearances between interface tools and hardware structures; include adequate clearance envelopes for those maintenance activities where access to an opening is required
- Define reach envelopes, crew load/forces, and general work constraints for IVA and EVA maintenance tasks
- Consider corrective and preventive maintenance task frequencies in the location of ORUs
- Allow replacement of an ORU without removal of other ORUs
- Choose a system thermal design that precludes degradation or damage during ORU replacement or maintenance to any other ORU
- Simplify ORU handling to reduce the likelihood of mishandling equipment or parts
- Encourage commonality, standardization, and interchangeability of tooling and hardware items to ensure a minimum number of items
- Select ORU fasteners to minimize accessibility time consistent with good design practice

- Design the ORU surface structure so that no safety hazard is created during the removal, replacement, test, or checkout of any ORU during IVA or EVA maintenance; include cautions/warnings for mission or safety critical ORUs
- Design software to facilitate modifications, verifications, and expansions
- Allow replacement of software segments on-line without disrupting mission or safety critical functions
- Allow on- or off-line software modification, replacement, or verification without introducing hazardous conditions.

6.4.4 Maintainability Analysis Tools and Techniques

Maintenance Functional Flow Block Diagrams (FFBDs). Maintenance FFBDs are used in the same way as system FFBDs, described in Appendix B.7.1. At the top level, maintenance FFBDs supplement and clarify the system maintenance concept; at lower levels, they provide a basis for the LSA's maintenance task inventory.

Maintenance Time Lines. Maintenance time line analysis (see Appendix B.7.3) is performed when time-to-restore is considered a critical factor for mission effectiveness and/or safety. (Such cases might include EVA and emergency repair procedures.) A maintenance time line analysis may be a simple spreadsheet or, at the other end, involve extensive computer simulation and testing.

FMECAs and FMEAs. Failure Modes, Effects, and Criticality Analysis (FMECA) and Failure Modes and Effects Analysis (FMEA) are specialized techniques for hardware failure and safety risk identification and characterization. They are discussed in this handbook under risk management (see Section 4.6.2) and reliability engineering (see Section 6.2.4). For the maintainability engineer, the FMECA/FMEA needs to be augmented at the LRU/ORU level with failure prediction data (i.e., MTTF or MTBF), failure detection means, and identification of corrective maintenance actions (for the LSA task inventory).

Maintainability Models. Maintainability models are used in assessing how well alternative designs meet maintainability requirements, and in quantifying the maintenance resource requirements. Modeling approaches may range from spreadsheets that aggregate component data, to complex *Markov* models and stochastic simulations. They often use reliability and time-to-restore data at the LRU/ORU level obtained from experience with similar components in existing systems. Some typical uses to which these models are put include:

- Annual maintenance hours and/or maintenance downtime estimates
- System MTTR and availability estimates (see sidebar on availability measures on page 86)
- Trades between reliability and maintainability
- Optimum LRU/ORU repair level analysis (ORLA)
- Optimum (reliability-centered) preventive maintenance analysis
- Spares requirements analysis
- Mass/volume estimates for (space-based) spares
- Repair vs. discard analysis.

LSA and LSAR. The Logistics Support Analysis (LSA) is the formal technical mechanism for integrating supportability considerations into the systems engineering process. Many of the above tools and techniques provide maintainability inputs to the LSA, or are used to develop LSA outputs. Results of the LSA are captured in Logistics Support

Maintainability Lessons Learned from HST Repair (STS-61)

When asked (for this handbook) what maintainability lessons were learned from their mission, the STS-61 crew responded with the following:

- The maintainability considerations designed into the Hubble Space Telescope (HST) worked.
- For spacecraft in LEO, don't *preclude* a servicing option; this means, for example, including a grapple fixture even though it has a cost and mass impact.
- When servicing is part of the maintenance concept, make sure that it's applied throughout the spacecraft. (The HST Solar Array Electronics Box, for example, was not designed to be replaced, but had to be nevertheless!)
- Pay attention to details like correctly sizing the hand holds, and using connectors and fasteners designed for easy removal and reattachment.

Other related advice:

- Make sure ground-based mock-ups and drawings exactly represent the "as-deployed" configuration.
- Verify tool-to-system interfaces, especially when new tools are involved.
- Make provision in the maintainability program for high-fidelity maintenance training.

Analysis Record (LSAR) data tables, which formally document the baselined ILS system. (See Section 6.5.3.)

Problem/Failure Reports (P/FRs). The maintainability engineer uses the Problem/Failure Reporting System (or an approved equivalent) to report maintainability problems and nonconformances encountered during qualification and acceptance testing (Phase D) and operations (Phase E).

6.5 Integrated Logistics Support

The objective of Integrated Logistics Support (ILS) activities within the systems engineering process is to ensure that the product system is supported during development (Phase D) and operations (Phase E) in a cost-effective manner. This is primarily accomplished by early, concurrent consideration of supportability characteristics, performing trade studies on alternative system and ILS concepts, quantifying resource requirements for each ILS element using best-practice techniques, and acquiring the support items associated with each ILS element. During operations, ILS activities support the system while seeking improvements in its cost-effectiveness by conducting analyses in response to actual operational conditions. These analyses continually reshape the ILS system and its resources requirements. Neglecting ILS or poor ILS decisions invariably have adverse effects on the life-cycle cost of the resultant system.

6.5.1 ILS Elements

According to NHB 7120.5, the scope of ILS includes the following nine elements:

- *Maintenance:* the process of planning and executing life-cycle repair/services concepts and requirements necessary to ensure sustained operation of the system
- *Design Interface:* the interaction and relationship of logistics with the systems engineering process to ensure that supportability influences the definition and design of the system so as to reduce life-cycle cost
- *Technical Data:* the recorded scientific, engineering, technical, and cost information used to define, produce, test, evaluate, modify, deliver, support, and operate the system
- *Training:* the processes, procedures, devices, and equipment required to train personnel to operate and support the system
- *Supply Support:* actions required to provide all the necessary material to ensure the system's supportability and usability objectives are met
- *Test and Support Equipment*: the equipment required to facilitate development, production, and operation of the system
- *Transportation and Handling:* the actions, resources, and methods necessary to ensure the proper and safe movement, handling, packaging, and storage of system items and materials
- *Human Resources and Personnel Planning:* actions required to determine the best skills-mix, considering current and future operator, maintenance, engineering, and administrative personnel costs
- *System Facilities:* real property assets required to develop and operate a system.

6.5.2 Planning for ILS

ILS planning should begin early in the project life cycle, and should be documented in an *ILS program plan*. This plan describes what ILS activities are planned, and how they will be conducted and integrated into the systems engineering process. For major projects, the ILS program plan may be a separate document because the ILS system (ILSS) may itself be a major system. For smaller projects, the SEMP (Part III) is the logical place to document such information. An important part of planning the ILS program concerns the strategy to be used in performing the Logistics Support Analysis (LSA) since it can involve a major commitment of logistics engineering specialists. (See Section 6.5.3.)

Documenting *results* of ILS activities through the project life cycle is generally done in the Integrated Logistics Support Plan (ILSP). The ILSP is the senior ILS document used by the project. A preliminary ILSP should be prepared by the completion of Phase B and subsequently maintained. This plan documents the project's logistics support concept, responsibility for each ILS element by project phase, and LSA results, especially trade study results. For major systems, the ILSP should be a distinct and separate part of the system documentation. For smaller systems, the ILSP may be integrated with other system documentation. The ILSP generally contains the following technical sections:

- *Maintenance Plan* — Developed from the system maintenance concept and refined during the system design and LSA processes. (NMI 5350.1A, *Maintainability and Maintenance Planning Policy*, and

NHB 5300.4(1E), *Maintainability Program Requirements for Space Systems*, do not use the term *ILS*, but they nevertheless mandate almost all of the steps found in an LSA. See Section 6.4.2 for more details on the maintenance plan.)

- *Personnel and Training Plan* — Identifies both operator and maintenance training, including descriptions of training programs, facilities, equipment, technical data, and special training aids. According to NMI 5350.1A/NHB 5300.4(1E), the maintenance training element is part of the maintenance plan.
- *Supply Support Plan* — Covers required quantities of spares (reparable and expendable) and consumables (identified through the LSA), and procedures for their procurement, packaging, handling, storage, and transportation. This plan should also cover such issues as inventory management, breakout screening, and demand data collection and analysis. According to NMI 5350.1A/NHB 5300.4(1E), the spares provisioning element is part of the maintenance plan.
- *Test and Support Equipment Plan* — Covers required types, geographical location, and quantities of test and support equipment (identified through the LSA). According to NMI 5350.1A/NHB 5300.4(1E), it is part of the maintenance plan.
- *Technical Data Plan* — Identifies procedures to acquire and maintain all required technical data. According to NMI 5350.1A/NHB 5300.4(1E), technical data for training is part of the maintenance plan.
- *Transportation and Handling Plan* — Covers all equipment, containers, and supplies (identified through the LSA), and procedures to support packaging, handling, storage, and transportation of system components
- *Facilities Plan* — Identifies all real property assets required to develop, test, maintain, and operate the system, and identifies those requirements that can be met by modifying existing facilities. It should also provide cost and schedule projections for each new facility or modification.
- *Disposal Plan* — Covers equipment, supplies, and procedures for the safe and economic disposal of all items (e.g., condemned spares), including ultimately the system itself.

The cost of ILS (and hence the life-cycle cost of the system) is driven by the inherent reliability and maintainability characteristics of the system design. The project-level system engineer must ensure that these considerations influence the design process through a well-conceived ILS program. In brief, a good-practice approach to achieving cost-effective ILS includes efforts to:

- Develop an ILS program plan, and coordinate it with the SEMP (Part III)
- Perform the technical portion of the plan, i.e., the Logistics Support Analysis, to select the best combined system and ILS alternative, and to quantify the resulting logistics resource requirements
- Document the selected ILS system and summarize the logistics resource requirements in the ILSP
- Provide supportability inputs to the system requirements and/or specifications
- Verify and validate the selected ILS system.

6.5.3 ILS Tools and Techniques: The Logistics Support Analysis

The Logistics Support Analysis (LSA) is the formal technical mechanism for integrating supportability considerations into the systems engineering process. The LSA is performed iteratively over the project life cycle so that successive refinements of the system design move toward the supportability objectives. To make this happen, the ILS engineer identifies supportability and supportability-related design factors that need to be considered in trade studies during the systems engineering process. *The project-level system engineer imports these considerations largely through their impact on projected system effectiveness and life-cycle cost.* The ILS engineer also acts as a *system* engineer (for the ILSS) by identifying ILSS functional requirements, performing trade studies on the ILSS, documenting the logistics support resources that will be required, and overseeing the verification and validation of the ILSS.

The LSA process found in MIL-STD-1388-1A can serve as a guideline, but its application in NASA should be tailored to the project. Figures 31a and 31b show the LSA process in more detail as it proceeds through the NASA project life cycle. Each iteration uses more detailed inputs and provides more refinement in the output so that by the time operations begin (Phase E), the full complement of logistics support resources has been identified and the ILSS verified. The first step at each iteration is to *understand the mission, the system architecture/design, and the ILSS parameters.* Specifically, the first step encompasses the following activities:

- Receiving (from the project-level system engineer) factors related to the intended use of the system such as the operations concept, mission duration,

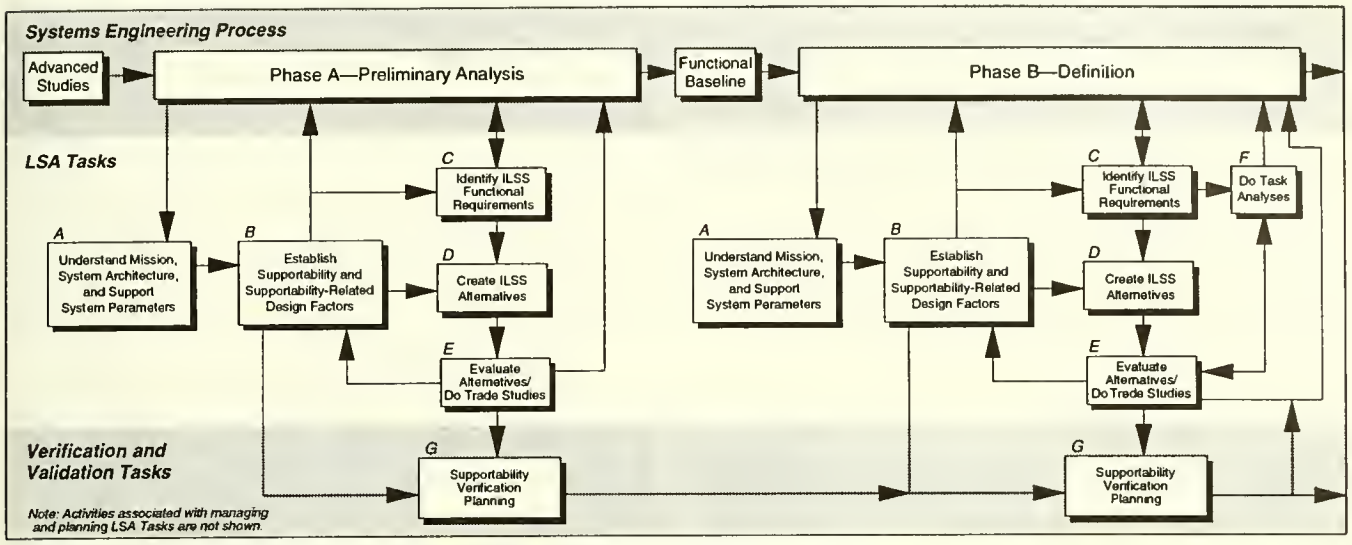


Figure 31a —Logistics Support Analysis Process Flow (Phases A and B).

number of units, orbit parameters, space transportation options, allocated supportability characteristics, etc.

- Documenting existing logistics resource capabilities and/or assets that may be cost-effective to apply to or combine with the ILSS for the system being developed
- Identifying technological opportunities that can be exploited. (This includes both new technologies in the system architecture/design that reduce logistics support resource requirements as well as new technologies within the ILSS that make it less expensive to meet *any* level of logistics support resource requirements.)
- Documenting the ILS concept and initial "strawman" ILSS, or updating (in later phases) the baseline ILSS.

The ILS engineer uses the results of these activities to *establish supportability and supportability-related design factors*, which are passed back to the project-level system engineer. This means:

- Identifying and estimating the magnitude of supportability factors associated with the various system and operations concepts being considered. Such factors might include operations team size, system RAM (reliability, availability, and maintain-

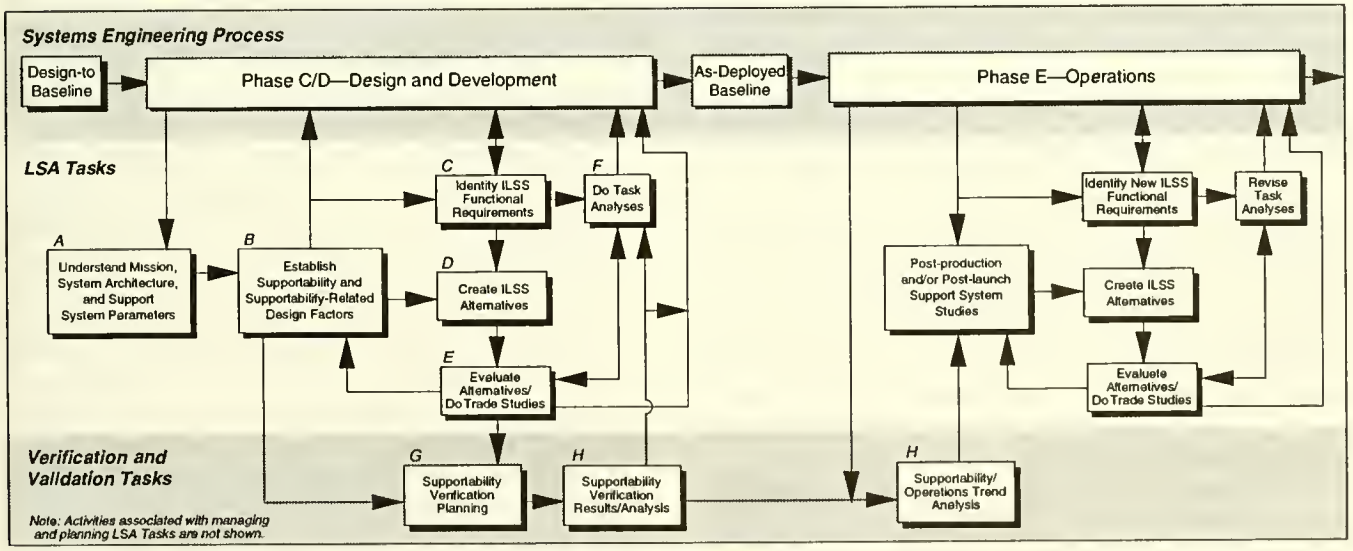


Figure 31b — Logistics Support Analysis Process Flow (Phases C/D and E).

ability) parameters, estimated annual IVA/EVA maintenance hours and upmass requirements, etc.
- Using the above to assist the project-level system engineer in projecting system effectiveness and life-cycle cost, and establishing system availability and/or system supportability goals. (See NHB 7120.5, and this handbook, Section 5.3.3.)
- Identifying and characterizing the system supportability risks. (See NHB 7120.5, and this handbook, Section 4.6.)
- Documenting supportability-related design constraints.

The heart of the LSA lies in the next group of activities, during which systems engineering and analysis are applied to the ILSS itself. The ILS engineer must first *identify the functional requirements for the ILSS.* The functional analysis process establishes the basis for a task inventory associated with the product system and, with the task inventory, aids in the identification of system design deficiencies requiring redesign. The task inventory generally includes corrective and preventive maintenance tasks, and other operations and support tasks arising from the ILSS functional requirements. A principal input to the inventory of corrective and preventive maintenance tasks, which is typically constructed by the maintainability engineer, is the FMECA/FMEA (or equivalent analysis). The FMECA/FMEA itself is typically performed by the reliability engineer. The entire task inventory is documented in Logistics Support Analysis Record (LSAR) data tables.

The ILS engineer then *creates plausible ILSS alternatives*, and *conducts trade studies* in the manner described earlier in Section 5.1. The trade studies focus on different issues depending on the phase of the project. In Phases A and B, trade studies focus on high-level issues such as whether a spacecraft in LEO should be serviceable or not, what mix of logistics modules seems best to support an inhabited space station, or what's the optimum number of maintenance levels and locations. In Phases C and D, the focus changes, for example to an individual end-item's optimum repair level. In Phase E, when the system design and its logistics support requirements are essentially understood, trade studies often revisit issues in the light of operational data. These trade studies almost always rely on techniques and models especially created for the purpose of doing a LSA. For a catalog of LSA techniques and models, the system engineer can consult the *Logistics Support Analysis Techniques Guide* (1985), Army Materiel Command Pamphlet No. 700-4.

By the end of Phase B, the results of the ILSS functional analyses and trade studies should be sufficiently refined and detailed to provide quantitative data on the logistics support resource requirements. This is accomplished by *doing a task analysis* for each task in the task inventory. These requirements are formally documented by amending the LSAR data tables. Together, ILSS trade studies, LSA models, and LSAR data tables provide the project-level system engineer with important life-cycle cost data and measures of (system) effectiveness (MoEs), which are successively refined through Phases C and D as the product system becomes better defined and better data become available. The relationship between inputs (from the specialty engineering disciplines) to the LSA process and its outputs can be seen in Figure 27 (see Section 5.3).

In performing the LSA, the ILS engineer also determines and documents (in the LSAR data tables) the logistics resource requirements for Phase D system integration and verification, and deployment (e.g., launch). For most spacecraft, this support includes pre-launch transportation and handling, storage, and testing. For new access-to-space systems, support may be needed during an extended period of developmental launches, and for inhabited space stations, during an extended period of on-orbit assembly operations. The ILS engineer also contributes to risk management activities by considering the adequacy of spares provisioning, and of logistics plans and processes. For example, spares provisioning must take into account the pos-

MIL-STD 1388-1A/2B

NHB 7120.5 suggests MIL-STD 1388 as a guideline for doing a LSA. MIL-STD 1388-1A is divided into five sections:

- LSA Planning and Control *(not shown in Figures 31a and 31b)*
- Mission and Support System Definition *(shown as boxes A and B)*
- Preparation and Evaluation of Alternatives *(shown as boxes C, D, and E)*
- Determination of Logistics Support Resource Requirements *(shown as box F)*
- Supportability Assessment *(shown as boxes G and H).*

MIL-STD 1388-1A also provides useful tips and encourages principles already established in this handbook: functional analysis, successive refinement of designs through trade studies, focus on system effectiveness and life-cycle cost, and appropriate models and selection rules.

MIL-STD 1388-2B contains the LSAR relational data table formats and data dictionaries for documenting ILS information and LSA results in machine-readable form.

sibility that production lines will close during the anticipated useful life of the system.

As part of verification and validation activity, the ILS engineer performs *supportability verification planning* and gathers supportability verification/test data during Phase D. These data are used to identify and correct deficiencies in the system design and ILSS, and to update the LSAR data tables. During Phase E, supportability testing and analyses are conducted under actual operational conditions. These data provide a useful legacy to product improvement efforts and future projects.

6.5.4 Continuous Acquisition and Life-Cycle Support

LSA documentation and supporting LSAR data tables contain large quantities of data. Making use of these data in a timely manner is currently difficult because changes occur often and rapidly during definition, design, and development (Phases B through D). Continuous Acquisition and Life-Cycle Support (CALS) — changed in 1993 from Computer-Aided Acquisition and Logistics Support — technology can reduce this dilemma by improving the digital exchange of data across NASA field centers and between NASA and its contractors. Initial CALS efforts within the logistics engineering community focused on developing CALS digital data exchange standards; current emphasis has shifted to database integration and product definition standards, such as STEP (Standard for the Exchange of Product) Model Data.

CALS represents a shift from a paper- (and labor-) intensive environment to a highly automated and integrated one. Concommitant with that are expected benefits in reduced design and development time and costs, and in the improved quality of ILS products and decisions. CALS cost savings accrue primarily in three areas: concurrent engineering, configuration control, and ILS functions. In a concurrent engineering environment, NASA's multi-disciplinary PDTs (which may mirror and work with those of a system contractor) can use CALS technology to speed the exchange of and access to data among PDTs. Availability of data through CALS on parts and suppliers also permits improved parts selection and acquisition. (See Section 3.7.2 for more on concurrent engineering.)

Configuration control also benefits from CALS technology. Using CALS to submit, process, and track ECRs/ECPs can reduce delays in approving or rejecting them, along with the indirect costs that delays cause. Although concurrent engineering is expected to reduce the number of ECRs/ECPs during design and development (Phases C and D), their timely disposition can produce significant cost savings. (See Section 4.7.2 for more on configuration control.)

Lastly, CALS technology potentially enables ILS functions such as supply support to be performed simultaneously and with less manual effort than at present. For example, procurement of design-stable components and spares can begin earlier (to allow earlier testing); at the same time, provisioning for other components can be further deferred (until design stability is achieved), thus reducing the risk of costly mistakes. Faster vendor response time also means reduced spares inventories during operations.

Can NASA Benefit from CALS?

The DoD CALS program was initiated in 1985; since 1988, it has been required on new DoD systems. According to Clark, potential DOD-wide savings from CALS exceeds $160M (FY92$). However, GAO studies have been critical of DoD's CALS implementation. These criticisms focused on CALS' limited ability to share information among users.

For NASA field centers to realize savings from CALS, new enabling investments in hardware, software, and training may be required. While many of NASA's larger contractors have already installed CALS technology, the system engineer wishing to employ CALS must recognize that both CALS and non-CALS approaches may be needed to interact with small business suppliers, and that proprietary contractor data, even when digitized, needs to be protected.

6.6 Verification

Verification is the process of confirming that deliverable ground and flight hardware and software are in compliance with functional, performance, and design requirements. The verification process, which includes planning, requirements definition, and compliance activities, begins early and continues throughout the project life cycle. These activities are an integral part of the systems engineering process. *At each stage of the process, the system engineer's job is to understand and assess verification results, and to lead in the resolution of any anomolies.* This section describes a generic NASA verification process that begins with a verification program concept and continues through operational and disposal verification. Whatever process is chosen by the program/project should be documented in the SEMP.

The objective of the verification program is to ensure that all functional, performance, and design requirements (from Level I program/project requirements through Level *n* requirements) have been met. Each project devel-

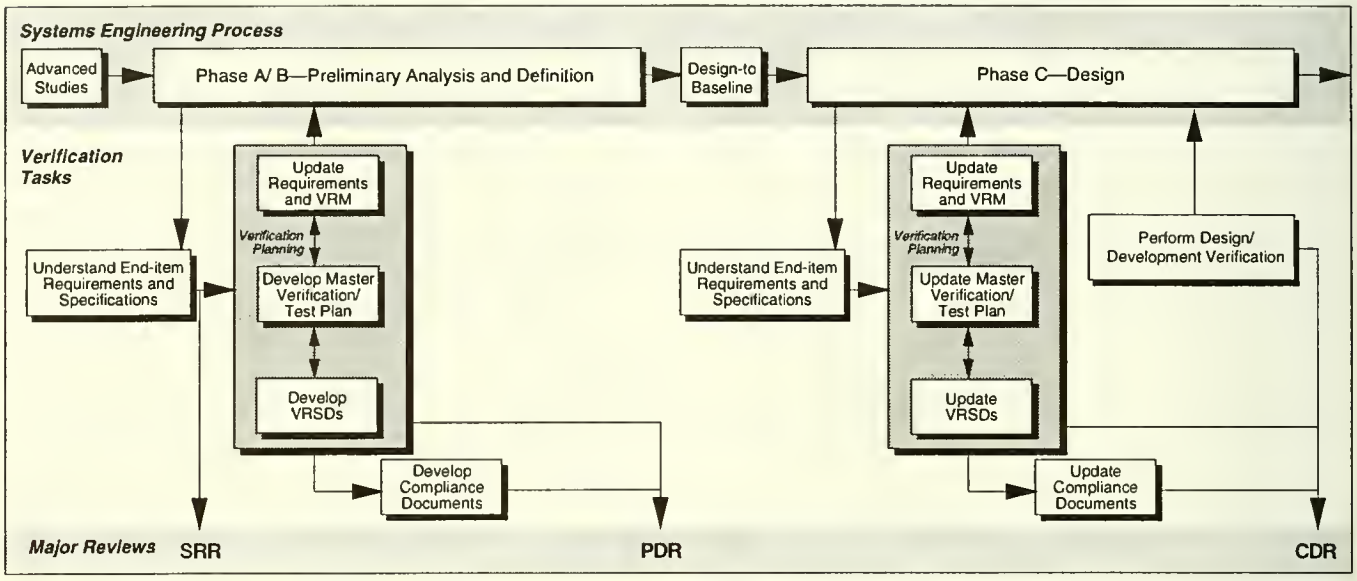


Figure 32a — Verification Process Flow (Phases A/B and C).

ops a verification program considering its cost, schedule, and risk implications. No one program can be applied to every project, and each verification activity and product must be assessed as to its applicability to a specific project. The verification program requires considerable coordination by the verification engineer, as both system design and test organizations are typically involved to some degree throughout.

6.6.1 Verification Process Overview

Verification activities begin in Phase A of a project. During this phase, inputs to the project's integrated master schedule and cost estimates are made as the verification program concept takes shape. These planning activities increase in Phase B with the refinement of requirements, costs, and schedules. In addition, the system's requirements are assessed to determine preliminary methods of verification and to ensure that the requirements *can* be

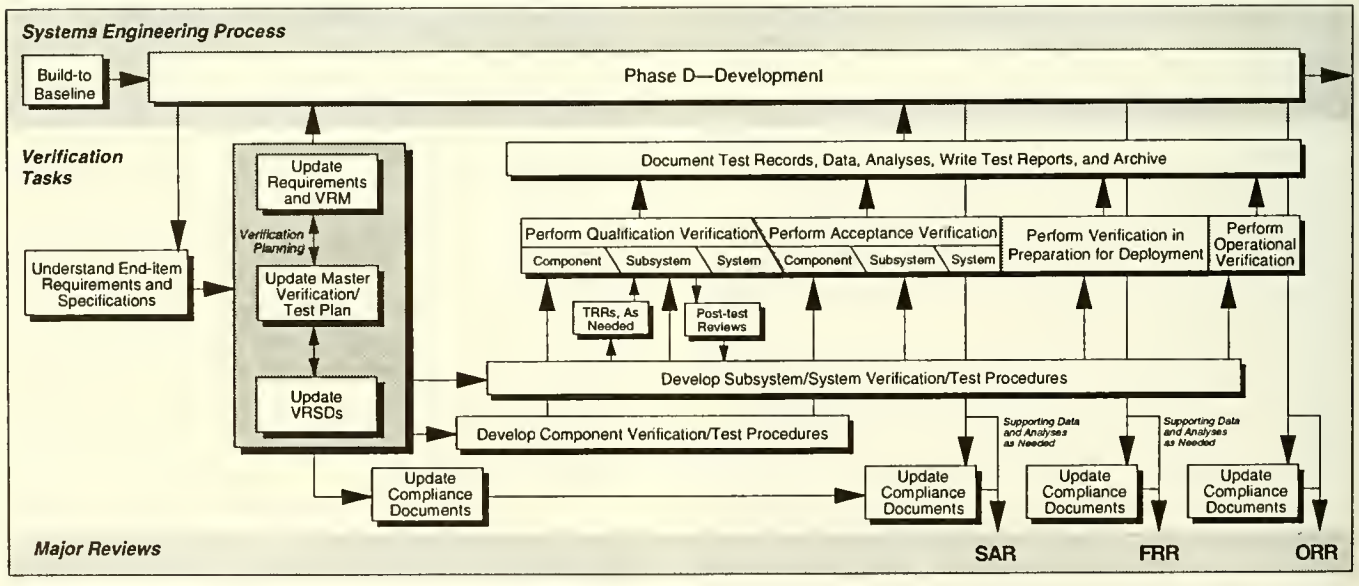


Figure 32b — Verification Process Flow (Phase D).

verified. The outputs of Phase B are expanded in Phase C as more detailed plans and procedures are prepared. In Phase D, verification activities increase substantially; these activities normally include qualification and acceptance verification, followed by verification in preparation for deployment and operational verification. Figures 32a and 32b show this process through the NASA project life cycle. (Safety reviews as applied to verification activities are not shown as separate activities in the figures.)

The Verification Program Concept. A verification program should be tailored to the project it supports. The project manager/system engineer must work with the verification engineer to develop a verification program concept. Many factors need to be considered in developing this concept and the subsequent verification program. These factors include:

- Project type, especially for flight projects. Verification methods and timing depend on the type of flight article involved (e.g., an experiment, payload, or launch vehicle).
- NASA payload classification. The verification activities and documentation required for a specific flight article generally depend upon its NASA payload classification. As expected, the verification program for a Class A payload is considerably more comprehensive than that for a Class D payload. (See Appendix B.3 for classification guidelines.)
- Project cost and schedule implications. Verification activities can be significant drivers of a project's cost and schedule; these implications should be considered early in the development of the verification program. Trade studies should be performed to support decisions about verification methods and requirements, and the selection of facility types and locations. As an example, a trade study might be made to decide between performing a test at a centralized facility or at several decentralized locations.
- Risk implications. Risk management must be considered in the development of the verification program. Qualitative risk assessments and quantitative risk analyses (e.g., a FMECA) often identify new concerns that can be mitigated by additional testing, thus increasing the extent of verification activities. Other risk assessments contribute to trade studies that determine the preferred methods of verification to be used and when those methods should be performed. As an example, a trade might be made between performing a modal test versus determining modal characteristics by a less costly, but less revealing, analysis. The project manager/system engineer must determine what risks are acceptable in terms of the project's cost and schedule.
- Availability of verification facilities/sites and transportation assets to move an article from one location to another (when needed). This requires coordination with the ILS engineer.
- Acquisition strategy (i.e., in-house development or system contract). Often a NASA field center can shape a contractor's verification process through the project's Statement of Work (SoW).
- Degree of design inheritance and hardware/software reuse.

Verification Methods and Techniques. The system engineer needs to understand what methods and techniques the verification engineer uses to verify compliance with requirements. In brief, these methods and techniques are:

- Test
- Analysis
- Demonstration
- Similarity
- Inspection
- Simulation
- Validation of records.

Verification by *test* is the actual operation of equipment during ambient conditions or when subjected to specified environments to evaluate performance. Two subcategories can be defined: functional testing and environmental testing. *Functional testing* is an individual test or series of electrical or mechanical performance tests conducted on flight or flight-configured hardware and/or software at conditions equal to or less than design specifications. Its purpose is to establish that the system perfoms satisfactorily in accordance with design and performance specifications. Functional testing generally is performed at ambient conditions. Functional testing is performed before and after each environmental test or major move in order to verify system performance prior to the next test/operation. *Environmental testing* is an individual test or series of tests conducted on flight or flight-configured hardware and/or software to assure it will perform satisfactorily in its flight environment. Environmental tests include vibration, acoustic, and thermal vacuum. Environmental testing may be combined with functional testing if test objectives warrant.

Verification by *analysis* is a process used in lieu of (or in addition to) testing to verify compliance to specifications/requirements. The selected techniques may include

systems engineering analysis, statistics and qualitative analysis, computer and hardware simulations, and computer modeling. Analysis may be used when it can be determined that: (1) rigorous and accurate analysis is possible; (2) testing is not feasible or cost-effective; (3) similarity is not applicable; and/or (4) verification by inspection is not adequate.

Verification by *demonstration* is the use of actual demonstration techniques in conjunction with requirements such as maintainability and human engineering features. Verification by *similarity* is the process of assessing by review of prior acceptance data or hardware configuration and applications that the article is similar or identical in design and manufacturing process to another article that has previously been qualified to equivalent or more stringent specifications. Verification by *inspection* is the physical evaluation of equipment and/or documentation to verify design features. Inspection is used to verify construction features, workmanship, and physical dimensions and condition (such as cleanliness, surface finish, and locking hardware). Verification by *simulation* is the process of verifying design features and performance using hardware or software other than flight items. Verification by *validation of records* is the process of using manufacturing records at end-item acceptance to verify construction features and processes for flight hardware.

Verification Stages. Verification stages are defined periods of verification activity when different verification goals are met. In this handbook, the following verification stages are used for flight systems:

- Development
- Qualification
- Acceptance
- Preparation for deployment (also known as prelaunch)
- Operational (also known as on-orbit or in-flight)
- Disposal (as needed).

Analyses and Models

Analyses based on models are used extensively throughout a program/project to verify and determine compliance to performance and design requirements. Most verification requirements that cannot be verified by a test activity are verified through analyses and modeling. The analysis and modeling process begins early in the project life cycle and continues through most of Phase D; these analyses and models are updated periodically as actual data that are used as inputs become available. Often, analyses and models are validated or corroborated by the results of a test activity. Any verification-related results should be documented as part of the project's archives.

The *development stage* is the period during which a new project or system is formulated and implemented up to the manufacturing of qualification or flight hardware. Verification activities during this stage (e.g., breadboard testing) provide confidence that the system can accomplish mission goals/objectives. When tests are conducted during this stage, they are usually performed by the design organization, or by the design and test organizations together. Also, some program/project requirements may be verified or partially verified through the activities of the PDR and CDR, both of which occur during this stage. Any development activity used to formally satisfy program/project requirements should have quality assurance oversight.

The *qualification stage* is the period during which the flight (protoflight approach) or flight-type hardware is verified to meet functional, performance, and design requirements. Verifications during this stage are conducted on flight-configured hardware at conditions more severe than acceptance conditions to establish that the hardware will perform satisfactorily in the flight environments with sufficient margin. The *acceptance stage* is the period during which the deliverable flight end-item is shown to meet functional, performance, and design requirements under conditions specified for the mission. The acceptance stage ends with shipment of the flight hardware to the launch site.

The *preparation for deployment stage* begins with the arrival of the flight hardware and/or software at the launch site and terminates at launch. Requirements verified during this stage are those that demand the integrated vehicle and/or launch site facilities. The *operational verification stage* begins at liftoff; during this stage, flight systems are verified to operate in space environment conditions, and requirements demanding space environments are verified. The *disposal stage* is the period during which disposal requirements are verified.

6.6.2 Verification Program Planning

Verification program planning is an interactive and lengthy process occurring during all phases of a project, but more heavily during Phase C. The verification engineer develops a preliminary definition of verification requirements and activities based on the program/project and mission requirements. An effort should be made throughout a project's mission and system definition to phrase requirements in absolute terms in order to simplify their verification. As the system and interface requirements are es-

tablished and refined, the verification engineer assesses them to determine the appropriate method of verification or combination thereof. These requirements and the method(s) of verification are then documented in the appropriate requirements document.

Using the methods of verification to be performed *for each verification stage*, along with the levels (e.g., part, subsystem, system) at which the verifications are to be performed, and any environmental controls (e.g., contamination) that must be maintained, the verification engineer outlines a preliminary schedule of verification activities associated with development, qualification, and acceptance of the system. This preliminary schedule should be in accordance with project milestones, and should be updated as verification activities are refined.

During planning, the verification engineer also identifies the documentation necessary to support the verification program. This documentation normally includes: (1) a *Verification Requirements Matrix* (VRM), (2) a *Master Verification Plan* (MVP), (3) a *Verification Requirements and Specifications Document* (VRSD), and (4) a *Verification Requirements Compliance Document* (VRCD). Documentation for test procedures and reports may also be defined. Because the system engineer should be familar with these basic elements of a verification process, each of these is covered below.

Verification Requirements Matrix. The Verification Requirements Matrix (VRM) is that portion of a requirements document (generally a System Requirements Document or CI specification) that defines how each functional, performance, and design requirement is to be verified, the stage in which verification is to occur, and (sometimes) the applicable verification levels. The verification engineer develops the VRM in coordination with the design, systems engineering, and test organizations. VRM contents are tailored to each project's requirements, and the level of detail in VRMs may vary. The VRM is baselined as a result of the PDR, and essentially establishes the basis for the verification program. A sample VRM for a CI specification is shown in Appendix B.9.

Master Verification Plan. The Master Verification Plan (MVP) is the document that describes the overall verification program. The MVP provides the content and depth of detail necessary to provide full visibility of all verification activities. Each major activity is defined and described in detail. The plan encompasses qualification, acceptance, pre-launch, operational, and disposal verification activities for flight hardware and software. (Development stage verification activities are not normally documented in the plan, but may be documented elsewhere.) The plan provides a general schedule and sequence of events for major verification activities. It also describes test software, Ground Support Equipment (GSE), and facilities necessary to support the verification activities. The verification engineer develops the plan through a thorough understanding of the verification program concept, the requirements in the Program (i.e., Level I) Requirements Document (PRD), System/Segment (i.e., Level II) Requirements Document (SRD), and/or the CI specification, and the methods identified in the VRM of those documents. Again, the development of the plan requires that the verification engineer work closely with the design, systems engineering, and test organizations. A sample outline for this plan is illustrated in Appendix B.10.

Verification Reports

A verification report should be provided for each analysis and, at a minimum, for each major test activity, such as functional testing, environmental testing, and end-to-end compatibility testing. If testing occurs over long periods of time or is separated by other activities, verification reports may be needed for each individual test activity, such as functional testing, acoustic testing, vibration testing, and thermal vacuum/thermal balance testing. Verification reports should be completed within a few weeks following a test, and should provide evidence of compliance with the verification requirements for which it was conducted. The verification report should include as appropriate:

- Verification objectives and degree to which they were met
- Description of verification activity
- Test configuration and differences from flight configuration
- Specific result of each test and each procedure including annotated tests
- Specific result of each analysis
- Test performance data tables, graphs, illustrations, and pictures
- Description of deviations from nominal results, problems/failures, approved anomaly corrective actions, and re-test activity
- Summary of non-conformance/discrepancy reports including dispositions
- Conclusion and recommendations relative to success of verification activity
- Status of support equipment as affected by test
- Copy of as-run procedure
- Authentication of test results and authorization of acceptability.

Verification Requirements and Specifications Document. The Verification Requirements and Specifications Document (VRSD) defines the detailed requirements and specifications for the verification of a flight article, including the ground system/segment. The VRSD specifies requirements and specifications for activities covering qualification through operational verification. Requirements are also defined for flight software verification after the software has been installed in the flight article. The VRSD should cover verifications by *all* methods; some programs/projects, however, use a document that defines only requirements to be satisfied by test.

The VRSD should include all requirements defined in Level I, II, and III requirements documents plus derived requirements. The VRSD defines the acceptance criteria and any constraints for each requirement. The VRSD typically identifies the locations where requirements will be verified. On large programs/projects, a VRSD is normally developed for each verification activity/location (e.g., thermal-vacuum testing), and is tailored to include requirements for that verification activity only. The verification engineer develops the VRSD from an understanding of the requirements, the verification program concept, and the flight article. The VRSD is baselined prior to the start of the verification activity. The heart of the VRSD is a data table that includes the following fields:

- A numerical designator assigned to each requirement
- A statement of the specific requirement to be verified
- The "pass/fail" criteria and tolerances for each requirement
- Any constraints that must be observed
- Any remarks to aid in the understanding of the requirement
- Location where the requirement will be verified.

The VRSD, along with flight article drawings and schematics, is the basis for the development of verification procedures, and is also used as one of the bases for development of the Verification Requirements Compliance Document (VRCD).

Verification Requirements Compliance Document. The Verification Requirements Compliance Document (VRCD) provides the evidence of compliance to each Level 1 through Level *n* design, performance, safety, and interface requirement, and to each VRSD requirement. The flowdown to VRSD requirements completes the full requirements traceability. Compliance with all the requirements ensures that Level I requirements have been met.

The VRCD defines, for each requirement, the method(s) of verification and corresponding compliance information for each method employed. The compliance information provides either the actual data, or a reference to the location of the actual data that shows compliance with the requirement. (The document also shows any non-compliances by referencing the related Non-Compliance Report (NCR) or Problem/Failure Report (P/FR); following resolution of the anomaly, the document specifies appropriate reverification information.) The compliance information may reference a verification report, an automated test program, a verification procedure, an analysis report, or a test. The inputting of compliance information into the compliance document occurs over a lengthy period of time, and on large systems and payloads, the effort may be continuous. The information in the compliance document must be up-to-date for the *System Acceptance Review(s)* (SAR) and *Flight Readiness Review* (FRR). The compliance document is not baselined because compliance information is input to the document throughout the entire project life cycle. It is, however, an extremely important part of the project's archives.

The heart of the Verification Requirements Compliance Document is also a data table with links to the corresponding requirements. The VRCD includes the following fields:

- A numerical designator assigned to each requirement
- A numerical designator that defines the document where the requirement is defined
- A statement of the specific requirement for which compliance is to be defined
- Verification method used to verify the requirement
- Location of the data that show compliance with the requirement statement. This information could be a test, report, procedure, analysis report, or other information that fully defines where the compliance data could be found. Retest information is also shown.
- Any non-conformances that occurred during the verification activities
- Any statements of compliance information as to any non-compliance or acceptance by means other than the method identified, such as a waiver.

Verification Procedures. The verification procedures are documents that provide step-by-step instructions for performing a given verification activity. The procedure is tailored to the verification activity that is to be performed to satisfy a requirement, and could be a test, demonstration, or any other verification-related activity. The procedure is

written to satisfy requirements defined by the VRSD, and is submitted prior to the *Test Readiness Review* (TRR) or the start of the verification activity in which the procedure is used. (See sidebar on TRRs.)

Procedures are also used to verify the acceptance of facilities, electrical and mechanical ground support equipment, and special test equipment. The information generally contained in a procedure is as follows, but it may vary according to the activity and test article:

- Nomenclature and identification of the test article or material
- Identification of test configuration and any differences from flight configuration
- Identification of objectives and criteria established for the test by the applicable verification specification
- Characteristics and design criteria to be inspected or tested, including values, with tolerances, for acceptance or rejection
- Description, in sequence, of steps and operations to be taken
- Identification of computer software required
- Identification of measuring, test, and recording equipment to be used, specifying range, accuracy, and type
- Certification that required computer test programs/support equipment and software have been verified prior to use with flight hardware
- Any special instructions for operating data recording equipment or other automated test equipment as applicable
- Layouts, schematics, or diagrams showing identification, location, and interconnection of test equipment, test articles, and measuring points
- Identification of hazardous situations or operations
- Precautions and safety instructions to ensure safety of personnel and prevent degradation of test articles and measuring equipment
- Environmental and/or other conditions to be maintained with tolerances
- Constraints on inspection or testing
- Special instructions for non-conformances and anomalous occurrences or results
- Specifications for facility, equipment maintenance, housekeeping, certification inspection, and safety and handling requirements before, during, and after the total verification activity.

The procedure may provide blank spaces for recording of results and narrative comments in order that the completed procedure can serve as part of the verification report. The as-run and certified copy of the procedure is maintained as part of the project's archives.

Test Readiness Reviews

A Test Readiness Review (TRR) is held prior to each major test to ensure the readiness of all ground, flight, and operational systems to support the performance of the test. A review of the detailed status of the facilities, Ground Support Equipment (GSE), test design, software, procedures, and verification requirements is made. The test activities and schedule are outlined and personnel responsibilities are identified. Verification emphasis is directed toward ensuring that all verification requirements that have been identified for the test have been included in the test design and procedures.

6.6.3 Qualification Verification

Qualification stage verification activities begin after completion of development of the flight hardware designs, and include analyses and testing to ensure that the flight or flight-type hardware (and software) will meet functional and performance requirements in anticipated environmental conditions. Qualification tests generally are designed to subject the hardware to worst case loads and environmental stresses. Some of the verifications performed to ensure hardware compliance to worst case loads and environments are vibration/acoustic, pressure limits, leak rates, thermal vacuum, thermal cycling, electromagnetic interference and electromagnetic compatibility (EMI/EMC), high and low voltage limits, and life time/cycling. During this stage, many performance requirements are verified, while analyses and models are updated as test data are acquired. Safety requirements, defined by hazard analysis reports, may also be satisfied by qualification testing.

Qualification usually occurs at the component or subsystem level, but could occur at the system level as well. When a project decides against building dedicated qualification hardware, and uses the flight hardware itself for qualification purposes, the process is termed *protoflight*. Additional information on protoflight testing is contained in MSFC-HDBK-670, *General Environmental Test Guidelines (GETG) for Protoflight Instruments and Experiments*.

6.6.4 Acceptance Verification

The acceptance stage verification activities provide the assurance that the flight hardware and software are in

compliance with all functional, performance, and design requirements, and are ready for shipment to the launch site. The acceptance stage begins with the acceptance of each individual component or piece part for assembly into the flight article and continues through the SAR.

Some verifications cannot be performed after a flight article, especially a large one, has been assembled and integrated (e.g., due to inaccessability). When this occurs, these verifications are performed *during* fabrication and integration, and are known as *in-process tests*. Acceptance testing, then, begins with in-process testing and continues through functional testing, environmental testing, and end-to-end compatibility testing. Functional testing normally begins at the component level and continues at the systems level, ending with all systems operating simultaneously. All tests are performed in accordance with requirements defined in the VRSD. When flight hardware is unavailable, or its use is inappropriate for a specific test, simulators may be used to verify interfaces. Anomalies occurring during a test are documented on the appropriate reporting system (NCR or P/FR), and a proposed resolution should be defined before testing continues. Major anomalies, or those that are not easily dispositioned, may require resolution by a collaborative effort of the system engineer, and the design, test, and other organizations. Where appropriate, analyses and models are validated and updated as test data are acquired.

6.6.5 Preparation for Deployment Verification

The pre-launch verification stage begins with the arrival of the flight article at the launch site and concludes at liftoff. During this stage, the flight article is processed and integrated with the launch vehicle. The launch vehicle could be the Shuttle, some other launch vehicle, or the flight article could be part of the launch vehicle. Verifications requirements for this stage are defined in the VRSD. When the launch site is the *Kennedy Space Center*, the Operations and Maintenance Requirements and Specifications Document (OMRSD) is used in lieu of the VRSD.

Verifications performed during this stage ensure that no visible damage to the system has occurred during shipment and that the system continues to function properly. If system elements are shipped separately and integrated at the launch site, testing of the system and system interfaces is generally required. If the system is integrated into a carrier, the interface to the carrier must also be verified. Other verifications include those that occur following integration into the launch vehicle and those that occur at the launch pad; these are intended to ensure that the system is functioning and in its proper launch configuration. Contingency verifications and procedures are developed for any contingencies that can be foreseen to occur during prelaunch and countdown. These contingency verifications and procedures are critical in that some contingencies may require a return of the launch vehicle or flight article from the launch pad to a processing facility.

Software IV&V

Some project managers/system engineers may wish to add IV&V (Independent Verification and Validation) to the software verification program. IV&V is a process whereby the products of the software development life cycle are independently reviewed, verified, and validated by an organization that is neither the developer nor the acquirer of the software. The IV&V agent should have no stake in the success or failure of the software; the agent's only interest should be to make sure that the software is thoroughly tested against its requirements.

IV&V activities duplicate the project's V&V activities step-by-step during the life cycle, with the exception that the IV&V agent does no informal testing. If IV&V is employed, formal acceptance testing may be done only once, by the IV&V agent. In this case, the developer formally demonstrates that the software is ready for acceptance testing.

6.6.6 Operational and Disposal Verification

Operational verification provides the assurance that the system functions properly in a (near-) zero gravity and vacuum environment. These verifications are performed through system activation and operation, rather than through a verification activity. Systems that are assembled on-orbit must have each interface verified, and must function properly during end-to-end testing. Mechanical interfaces that provide fluid and gas flow must be verified to ensure no leakage occurs, and that pressures and flow rates are within specification. Environmental systems must be verified. The requirements for all operational verification activities are defined in the VRSD.

Disposal verification provides the assurance that the safe deactivation and disposal of all system products and processes has occurred. The disposal stage begins in Phase E at the appropriate time (i.e., either as scheduled, or earlier in the event of premature failure or accident), and concludes when all mission data have been acquired and verifications necessary to establish compliance with disposal requirements are finished. Both operational and disposal verification activities may also include validation assess-

ments — that is, assessments of the degree to which the system accomplished the desired mission goals/objectives.

6.7 Producibility

Producibility is a system characteristic associated with the ease and economy with which a completed design can be transformed (i.e., fabricated, manufactured, or coded) into a hardware and/or software realization. While major NASA systems tend to be produced in small quantities, a particular producibility feature can be critical to a system's cost-effectiveness, as experience with the Shuttle's thermal tiles has shown.

6.7.1 Role of the Production Engineer

The production engineer supports the systems engineering process (as a part of the multi-disciplinary PDT) through an active role in implementing specific design features to enhance producibility, and by performing the production engineering analyses needed by the project. These tasks and analyses include:

- Performing the manufacturing/fabrication portion of the system risk management program (see Section 4.6). This is accomplished by conducting a rigorous production risk assessment and by planning effective risk mitigation actions.
- Identifying system design features that enhance producibility. Efforts usually focus on design simplification, fabrication tolerances, and avoidance of hazardous materials.
- Conducting producibility trade studies to determine the most cost-effective fabrication/manufacturing process
- Assessing production feasibility within project constraints. This may include assessing contractor and principal subcontractor production experience and capability, new fabrication technology, special tooling, and production personnel training requirements.
- Identifying long-lead items and critical materials
- Estimating production costs as a part of life-cycle cost management
- Developing production schedules
- Developing approaches and plans to validate fabrication/manufacturing processes.

The results of these tasks and production engineering analyses are documented in the Manufacturing Plan with a level of detail appropriate to the phase of the project. The production engineer also participates in and contributes to major project reviews (primarily PDR and CDR) on the above items, and to special interim reviews such as the Production Readiness Review (ProRR).

6.7.2 Producibility Tools and Techniques

Manufacturing Functional Flow Block Diagrams (FFBDs). Manufacturing FFBDs are used in the same way system FFBDs, described in Appendix B.7.1, are used. At the top level, manufacturing FFBDs supplement and clarify the system's manufacturing sequence.

Risk Management Templates. The risk management templates of DoD 4245.7M, *Transition from Development to Production ...Solving the Risk Equation*, are a widely recognized series of risks, risk responses, and lessons learned from DoD experience. These templates, which were designed to reduce risks in production, can be tailored to individual NASA projects.

Producibility Assessment Worksheets. These worksheets, which were also developed for DoD, use a judgment-based scoring approach to help choose among alternative production methods. See *Producibility Measurement for DoD Contracts.*

Producibility Models. Producibility models are used in addressing a variety of issues such as assessing the feasibility of alternative manufacturing plans, and estimating production costs as a part of life-cycle cost management. Specific producibility models may include:

- Scheduling models for estimating production output, and for integrating system enhancements and/or spares production into the manufacturing sequence
- Manufacturing or assembly flow simulations, e.g., discrete event simulations of factory activities
- Production cost models that include learning and production rate sensitivities. (See sidebar page 82.)

Statistical Process Control/Design of Experiments. These techniques, long applied in manufacturing to identify the causes of unwanted variations in product quality and reduce their effects, have had a rebirth under TQM. A collection of currently popular techniques of this new *quality engineering* is known as *Taguchi methods.* For first-hand information on Taguchi methods, see his book, *Quality Engineering in Production Systems*, 1989. A handbook approach to to some of these techniques can be found in the

Navy's *Producibility Measurement Guidelines: Methodologies for Product Integrity.*

6.8 Social Acceptability

NASA systems must be acceptable to the society that funds them. The system engineer takes this into account by integrating mandated social concerns into the systems engineering process. For some systems, these concerns can result in significant design and cost penalties. Even when social concerns can be met, the planning and analysis associated with doing so can be time-consuming (even to the extent of affecting the project's critical path), and use significant specialized engineering resources. The system engineer must include these costs in high-level trade studies of alternative architectures/designs.

6.8.1 Environmental Impact

NASA policy and federal law require all NASA actions that may impact the quality of the environment be executed in accordance with the policies and procedures of the National Environmental Policy Act (NEPA). For any NASA project or other major NASA effort, this requires that studies and analyses be produced explaining how and why the project is planned, and the nature and scope of its potential environmental impact. These studies must be performed whether the project is conducted at NASA Headquarters, a field center, or a contractor facility, and must properly begin at the earliest period of project planning (i.e., not later than Phase A). Findings, in the form of an Environmental Assessment (EA) and, if warranted, through the more thorough analyses of an Environmental Impact Statement (EIS), must be presented to the public for review and comment. (See sidebar on NEPA.)

At the outset, some NASA projects will be of such a magnitude and nature that an EIS is clearly going to be required by NEPA, and some will clearly not need an EIS. Most major NASA projects, however, fall in between, where the need for an EIS is *a priori* unclear; in such cases an EA is prepared to determine whether an EIS is indeed required. *NASA's experience since 1970 has been that projects in which there is the release — or potential release — of large or hazardous quantities of pollutants (rocket exhaust gases, exotic materials, or radioactive substances), require an EIS.* For projects in this category, an EA is not performed, and the project's analyses should focus on and support the preparation of an EIS.

The NEPA process is meant to ensure that the project is planned and executed in a way that meets the national environmental policy and goals. First, the process helps the system engineer shape the project by putting potential environmental concerns in the forefront during Phase A. Secondly, the process provides the means for reporting to the public the project's rationale and implementation method. Finally, it allows public review of and comment on the planned effort, and requires NASA to consider and respond to those comments. The system engineer should be aware of the following NEPA process elements.

What is NEPA?

The National Environmental Policy Act (NEPA) of 1969 declares a national environmental policy and goals, and provides a method for accomplishing those goals. NEPA requires an Environmental Impact Statement (EIS) for *"major federal actions significantly affecting the quality of the human environment."*

Some environmental impact reference documents include:

- National Environmental Policy Act (NEPA) of 1969, as amended (40 CFR 1500-1508)
- *Procedures for Implementing the National Environmental Policy Act* (14 CFR 1216.3)
- *Implementing the Requirements of the National Evnironmental Policy Act*, NHB 8800.11
- Executive Order 11514, *Protection and Enhancement of Environmental Quality*, March 5, 1970, as amended by Executive Order 11991, May 24, 1977
- Executive Order 12114, *Environmental Effects Abroad of Major Federal Actions*, January 4, 1979.

Environmental Assessment (EA). An EA is a concise public document that serves to provide sufficient evidence and analyses for determining whether to prepare either an EIS or a Finding of No Significant Impact (FONSI). The analyses performed should identify the environmental effects of all reasonable alternative methods of achieving the project's goals/objectives so that they may be compared. The alternative of taking no action (i.e., not doing the project) should also be studied. Although there is no requirement that NASA select the alternative having the least environmental impact, there must be sufficient information available to make clear what those impacts would be, and to describe the reasoning behind NASA's preferred selection. The environmental analyses are an integral part of the project's systems engineering process.

The EA is the responsibility of the NASA Headquarters Program Associate Administrator (PAA) responsi-

ble for the proposed project or action. The EA can be carried out at Headquarters or at a NASA field center. Approval of the EA is made by the responsible PAA. Most often, approval of the EA takes the form of a memorandum to the Associate Administrator (AA) for Management Systems and Facilities (Code J) stating either that the project requires an EIS, or that it does not. If an EIS is found to be necessary, a Notice of Intent (NOI) to prepare an EIS is written; if an EIS is found to be unnecessary, a Finding of No Significant Impact (FONSI) is written instead.

Finding of No Significant Impact (FONSI). A FONSI should briefly present the reasons why the proposed project or action, as presented in the EA, has been judged to have no significant effect on the human environment, and does not therefore require the preparation of an EIS. The FONSI for projects and actions that are national in scope is published in the Federal Register, and is available for public review for a 30-day period. During that time, any supporting information is made readily available on request.

Notice of Intent (NOI). A Notice of Intent to file an EIS should include a brief description of the proposed project or action, possible alternatives, the primary environmental issues uncovered by the EA, and NASA's proposed scoping procedure, including the time and place of any scoping meetings. The NOI is prepared by the responsible Headquarters PAA and published in the Federal Register. It is also sent to interested parties.

Scoping. The responsible Headquarters PAA must conduct an early and open process for determining the scope of issues to be addressed in the EIS, and for identifying the significant environmental issues. Scoping is also the responsibility of the Headquarters PAA responsible for the proposed project or action; however, the responsible Headquarters PAA often works closely with the Code J AA. Initially, scoping must consider the full range of environmental parameters en route to identifying those that are significant enough to be addressed in the EIS. Examples of the environmental categories and questions that should be asked in the scoping process are contained in NHB 8800.11, *Implementing the Provisions of the National Environmental Policy Act*, Section 307.d.

Environmental Impact Statement (EIS). The EA and scoping elements of the NEPA process provide the responsible Headquarters PAA with an evaluation of significant environmental effects and issues that must be covered in the EIS. Preparation of the EIS itself may be carried out by NASA alone, or with the assistance or cooperation of other government agencies and/or a contractor. If a contractor is used, the contractor should execute a disclosure statement prepared by NASA Headquarters indicating that the contractor has no interest in the outcome of the project.

The section on environmental consequences is the analytic heart of the EIS, and provides the basis for the comparative evaluation of the alternatives. The analytic results for each alternative should be displayed in a way that highlights the choices offered the decision maker(s). An especially suitable form is a matrix showing the alternatives against the categories of environmental impact (e.g., air pollution, water pollution, endangered species). The matrix is filled in with (an estimate of) the magnitude of the environmental impact for each alternative and category. The subsequent discussion of alternatives is an extremely important part of the EIS, and should be given commensurate attention.

NASA review of the draft EIS is managed by the Code J AA. When submitted for NASA review, the draft EIS should be accompanied by a proposed list of federal, state and local officials, and other interested parties.

External review of the draft EIS is also managed by the Code J AA. A notice announcing the release and availability of the draft EIS is published in the Federal Register, and copies are distributed with a request for comments. Upon receipt of the draft, the Environmental Protection Agency (EPA) also places a notice in the Federal Register, and the date of that publication is the date that all time limits related to the draft's release begin. A minimum of 45 days must be allowed for comments. Comments from external reviewers received by the Code J AA will be sent to the office responsible for preparing the EIS. Each comment should be incorporated in the final EIS.

The draft form of the final EIS, modified as required by the review process just described, should be forwarded to the Code J AA for a final review before printing and distribution. The final version should include satisfactory responses to all responsible comments. While NASA need not yield to each and every opposing comment, NASA's position should be rational, logical, and based on data and arguments stronger than those cited by the commentors opposing the NASA views.

According to NHB 8800.11, *Implementing the Provisions of the National Environmental Policy Act* (Section 309.b), "an important element in the EIS process is involvement of the public. Early involvement can go a long way toward meeting complaints and objections regarding a proposed action, and experience has taught that a fully informed and involved public is considerably more supportive of a proposed action. When a proposed action is believed likely to generate significant public concern, the public should be brought in for consultation in the early planning stages. If an EIS is warranted, the public should

be involved both in scoping and in the EIS review. Early involvement can help lead to selection of the best alternative and to the least public objection."

Record of Decision (ROD). When the EIS process has been completed and public review periods have elapsed, NASA is free to make and implement the decision(s) regarding the proposed project or action. At that time, a Record of Decision (ROD) is prepared by the Headquarters PAA responsible for the project or action. The ROD becomes the official public record of the consideration of environmental factors in reaching the decision. The ROD is not published in the Federal Register, but must be kept in the official files of the program/project in question and made available on request.

6.8.2 Nuclear Safety Launch Approval

Presidential Directive/National Security Council Memorandum-25 (PD/NSC-25) requires that flight projects calling for the use of radioactive sources follow a lengthy analysis and review process in order to seek approval for launch. The nuclear safety launch approval process is separate and distinct from the NEPA compliance process. While there may be overlaps in the data-gathering for both, the documentation required for NEPA and nuclear safety launch approval fulfill separate federal and NASA requirements. While NEPA is to be done at the earliest stages of the project, launch approval officially begins with Phase C/D.

Phase A/B activities are driven by the requirements of the EA/EIS. At the earliest possible time (not later than Phase A), the responsible Headquarters PAA must undertake to develop the project EA/EIS and a Safety Analysis/Launch Approval Plan in coordination with the nuclear power system integration engineer and/or the launch vehicle integration engineer. A primary purpose of the EA/EIS is to ensure a comprehensive assessment of the rationale for choosing a radioactive source. In addition, the EA/EIS illuminates the environmental effects of alternative mission designs, flight systems, and launch vehicles, as well as the relative nuclear safety concerns of each alternative.

The launch approval engineer ensures that the following specific requirements are met during Phase A:

- Conduct a radioactive source design trade study that includes the definition, spacecraft design impact evaluation, and cost trades of all reasonable alternatives
- Identify the flight system requirements that are specific to the radioactive source
- For nuclear power alternatives, identify flight system power requirements and alternatives, and define the operating and accident environments to allow DOE (U.S. Department of Energy) to assess the applicability of existing nuclear power system design(s).

During Phase B, activities depend on the specifics of the project's EA/EIS plan. The responsible Headquarters PAA determines whether the preparation and writing of the EA/EIS will be done at a NASA field center, at NASA Headquarters, or by a contractor, and what assistance will be required from other field centers, the launch facility, DOE, or other agencies and organizations. The launch approval engineer ensures that the following specific requirements are met during Phase B:

- Update and refine the project, flight system, launch vehicle, and radioactive source descriptions
- Update and refine the radioactive source design trade study developed during Phase A
- Assist DOE where appropriate in conducting a preliminary assessment of the mission's nuclear risk and environmental hazards.

The launch approval engineer is also responsible for coordinating the activities, interfaces, and record-keeping related to mission nuclear safety issues. The following tasks are managed by the launch approval engineer:

- Develop the project EA/EIS and Safety Analysis/Launch Approval Plan
- Maintain a database of documents related to EA/EIS and nuclear safety launch approval tasks. This database will help form and maintain the audit trail record of how and why technical decisions and choices are made in the mission development and planning process. Attention to this activity early on saves time and expense later in the launch approval process when the project may be called upon to explain why a particular method or alternative was given greater weight in the planning process.
- Provide documentation and review support as appropriate in the generation of mission data and trade studies required to support the EA/EIS and safety analyses
- Establish a project point-of-contact to the launch vehicle integration engineer, DOE, and NASA Headquarters regarding support to the EA/EIS and nuclear safety launch approval processes. This includes responding to public and Congressional que-

Table 7 — Planetary Protection Categories

Category	Application	Summarized Requirements
1	Moon, Sun, and Mercury	Certification of category only.
2	All missions other than to the above or Mars	Avoidance of accidental impact with solar system object by spacecraft and launch vehicle. Documentation of final disposition of launched hardware. An inventory of organic materials for landers and probes.
3	Flybys and orbiters to Mars	Stringent limitations on the probability of impact. For orbiters, requirements on orbital lifetime or requirements for microbial cleanliness of spacecraft.
4	Mars landers	Stringent limitation on the probability of hard impact by spacecraft elements not intended to impact Mars. Microbial cleanliness of lander hardware surfaces directly established by bioassay.
5	Any sample return mission other than from the Moon or Mercury	Outbound requirements per category of a lander mission to target planet. Requirements for inbound leg of such missions have not yet been finalized, but are likely to include sterilization of any hardware that contacted the target planet before its return to Earth, and the containment of any returned samples.

ries regarding radioactive source safety issues, and supporting proceedings resulting from any litigation that may occur.

- Provide technical analysis support as required for the generation of accident and/or command destruct environment for the radioactive source safety analysis. The usual technique for the technical analysis is a probabilistic risk assessment (PRA). See Section 4.6.3.

6.8.3 Planetary Protection

The U.S. is a signatory to the United Nation's *Treaty of Principles Governing the Activities of States in the Exploration and Use of Outer Space, Including the Moon and Other Celestial Bodies.* Known as the "Outer Space" treaty, it states in part (Article IX) that exploration of the Moon and other celestial bodies shall be conducted "so as to avoid their harmful contamination and also adverse changes in the environment of the Earth resulting from the introduction of extraterrestrial matter." NASA policy (NMI 8020.7D) specifies that the purpose of preserving solar system conditions is for future biological and organic constituent exploration. It also establishes the basic NASA policy for the protection of the Earth and its biosphere from planetary and other extraterrestrial sources of contamination.

The general regulations to which NASA flight projects must adhere are set forth in NHB 8020.12B, *Planetary Protection Provisions for Robotic Extraterrestrial Missions.* Different requirements apply to different missions, depending on which solar system object is targeted and the spacecraft or mission type (flyby, orbiter, lander, sample-return, etc.). For some bodies (such as the Sun, Moon, Mercury), there are no outbound contamination requirements. Present requirements for the outbound phase of missions to Mars, however, are particularly rigorous. Planning for planetary protection begins in Phase A, during which feasibility of the mission is established. Prior to the end of Phase A, the project manager must send a letter to the Planetary Protection Officer (PPO) within the Office of the AA for Space Science stating the mission type and planetary targets, and requesting that the mission be assigned a planetary protection category. Table 7 shows the current planetary protection categories and a summary of their associated requirements.

Prior to the Preliminary Design Review (PDR) at the end of Phase B, the project manager must submit to the NASA PPO a Planetary Protection Plan detailing the actions that will be taken to meet the requirements. The project's progress and completion of the requirements are reported in a Planetary Protection Pre-Launch Report submitted to the NASA PPO for approval. The approval of this report at the Flight Readiness Review (FRR) constitutes the final approval for the project and must be obtained for permission to launch. An update to this report, the Planetary Protection Post-Launch Report, is prepared to report any deviations from the planned mission due to actual launch or early mission events. For sample return missions only, additional reports and reviews are required: prior to launch toward the Earth, prior to commitment to Earth reentry, and prior to the release of any extraterrestrial sample to the scientific community for investigation. Finally, at the formally declared end-of-mission, a Planetary Protection End-of-Mission Report is prepared. This document reviews the entire history of the mission in comparison to the original Planetary Protection Plan, and documents the degree of compliance with NASA's planetary protection requirements. This document is typically reported on by the NASA PPO at a meeting of the Committee on Space Research (COSPAR) to inform other spacefaring nations of NASA's degree of compliance with international planetary protection requirements.

Appendix A — Acronyms

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

AA	Associate Administrator (NASA)
APA	Allowance for Program Adjustment
ACWP	Actual Cost of Work Performed
AGE	Aerospace Ground Equipment
AHP	Analytic Hierarchy Process
BCWP	Budgeted Cost of Work Performed
BCWS	Budgeted Cost of Work Scheduled
C/SCSC	Cost/Schedule Control System Criteria
CALS	Continuous Acquisition and Life-Cycle Support
CCB	Configuration (or Change) Control Board
CDR	Critical Design Review
CER	Cost Estimating Relationship
CI	Configuration Item
CIL	Critical Items List
CoF	Construction of Facilities
COSPAR	Committee on Space Research
COTR	Contracting Office Technical Representative
CPM	Critical Path Method
CR	Change Request
CSCI	Computer Software Configuration Item
CSM	Center for Systems Management
CWBS	Contract Work Breakdown Structure
DCR	Design Certification Review
DDT&E	Design, Development, Test and Evaluation
DoD	(U.S.) Department of Defense
DOE	(U.S.) Department of Energy
DR	Decommissioning Review
DSMC	Defense Systems Management College
EA	Environmental Assessment
EAC	Estimate at Completion
ECP	Engineering Change Proposal
ECR	Engineering Change Request
EIS	Environmental Impact Statement
EMC	Electromagnetic compatibility
EMI	Electromagnetic interference
EOM	End of Mission
EPA	(U.S.) Environmental Protection Agency
EVA	Extravehicular Activities
EVM	Earned Value Measurement
FCA	Functional Configuration Audit
FFBD	Functional Flow Block Diagram
FH	Flight Hardware
FMEA	Failure Modes and Effects Analysis
FMECA	Failure Modes, Effects, and Criticality Analysis
FONSI	Finding of No Significant Impact
FRR	Flight Readiness Review
GAO	General Accounting Office
GOES	Geosynchonous Orbiting Environmental Satellite
GSE	Ground Support Equipment
HQ	NASA Headquarters
HST	Hubble Space Telescope
I&V	Integration and Verification
ILS	Integrated Logistics Support
ILSP	Integrated Logistics Support Plan
ILSS	Integrated Logistics Support System
IOP	Institutional Operating Plan
IRAS	Infrared Astronomical Satellite
IV&V	Independent Verification and Validation
IVA	Intravehicular Activities
LEM	Lunar Excursion Module (*Apollo*)
LEO	Low Earth Orbit
LMEPO	Lunar/Mars Exploration Program Office
LMI	Logistics Management Institute
LOOS	Launch and Orbital Operations Support
LRU	Line Replaceable Unit
LSA	Logistics Support Analysis
LSAR	Logistics Support Analysis Record
MDT	Mean Downtime
MCR	Mission Concept Review
MDR	Mission Definition Review
MESSOC	Model for Estimating Space Station Operations Cost
MICM	Multi-variable Instrument Cost Model
MLDT	Mean Logistics Delay Time
MMT	Mean Maintenance Time
MNS	Mission Needs Statement
MoE	Measure of (system) Effectiveness
MRB	Material Review Board
MRR	Mission Requirements Review
MTBF	Mean Time Between Failures
MTTF	Mean Time To Failure
MTTMA	Mean Time To a Maintenance Action
MTTR	Mean Time To Repair/Restore
NAR	Non-Advocate Review
NCR	Non-Compliance (or Non-Conformance) Report
NEPA	National Environmental Policy Act
NHB	NASA Handbook
NMI	NASA Management Instruction
NOAA	(U.S.) National Oceanic and Atmospheric Administration
NOI	Notice of Intent
OMB	Office of Management and Budget

OMRSD	Operations and Maintenance Requirements and Specifications Document (KSC)
ORLA	Optimum Repair Level Analysis
ORR	Operational Readiness Review
ORU	Orbital Replacement Unit
P/FR	Problem/Failure Report
PAA	Program Associate Administrator (NASA)
PAR	Program/Project Approval Review
PBS	Product Breakdown Structure
PCA	Physical Configuration Audit
PDR	Preliminary Design Review
PDT	Product Development Team
PDV	Present Discounted Value
PERT	Program Evaluation and Review Technique
POP	Program Operating Plan
PPAR	Preliminary Program/Project Approval Review
PPO	Planetary Protection Officer
PRA	Probabilistic Risk Assessment
PRD	Program Requirements Document
ProRR	Production Readiness Review
QA	Quality Assurance
QFD	Quality Function Deployment
RAM	Reliability, Availability, and Maintainability
RAS	Requirements Allocation Sheet
RID	Review Item Discrepancy
RMP	Risk Management Plan
ROD	Record of Decision
RTG	Radioisotope Thermoelectric Generator
SAR	System Acceptance Review
SDR	System Definition Review
SEB	Source Evaluation Board
SEMP	Systems Engineering Management Plan
SEPIT	Systems Engineering Process Improvement Task
SEWG	Systems Engineering Working Group (NASA)
SI	*Le Système International d' Unités* (the international [metric] system of units)
SIRTF	Space Infrared Telescope Facility
SOFIA	Stratospheric Observatory for Infrared Astronomy
SoSR	Software Specification Review
SoW	Statement of Work
SSR	System Safety Review
SRD	System/Segment Requirements Document
SRM&QA	Safety, Reliability, Maintainability, and Quality Assurance
SRR	System Requirements Review
STEP	Standard for the Exchange of Product (model data)
STS	Space Transportation System
SSA	Space Station *Alpha*
SSF	Space Station *Freedom*
TBD	To Be Determined; To Be Done
TDRS	Tracking and Data Relay Satellite
TLA	Time Line Analysis
TLS	Time Line Sheet
TPM	Technical Performance Measure(ment)
TQM	Total Quality Management
TRR	Test Readiness Review
V&V	Verification and Validation
VMP	Verification Master Plan
VRCD	Verification Requirements Compliance Document
VRM	Verification Requirements Matrix
VRSD	Verification Requirements and Specifications Document
WBS	Work Breakdown Structure
WFD	Work Flow Diagram

NASA Organizations

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

Appendix B — Systems Engineering Templates and Examples

Appendix B.1 — A Sample SEMP Outline

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

Systems Engineering Management Plan

Title Page
Introduction

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

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

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

Appendix B.2 — A "Tailored" WBS for an Airborne Telescope

Figure B-1 shows a partial Product Breakdown Structure (PBS) for the proposed Stratospheric Observatory for Infrared Astronomy (SOFIA), a 747SP aircraft outfitted with a 2.5 to 3.0 m telescope. The PBS has been elaborated for the airborne facility's telescope element. The PBS level names have been made consistent with the sidebar on page 3 of this handbook.

Figures B-2 through B-5 show a corresponding Work Breakdown Structures (WBSs) based on the principles in Section 4.3 of this handbook. At each level, the prime product deliverables from the PBS are WBS elements. The WBS is completed at each level by adding needed service (i.e., functional) elements such as management, systems engineering, integration and test, etc. The integration and test WBS element at each level refers to the activities of unifying prime product deliverables at that level.

Although the SOFIA project is used as an illustration in this appendix, the SOFIA WBS should be tailored to fit actual conditions at the start of Phase C/D as determined by the project manager. One example of a condition that could substantially change the WBS is international participation in the project.

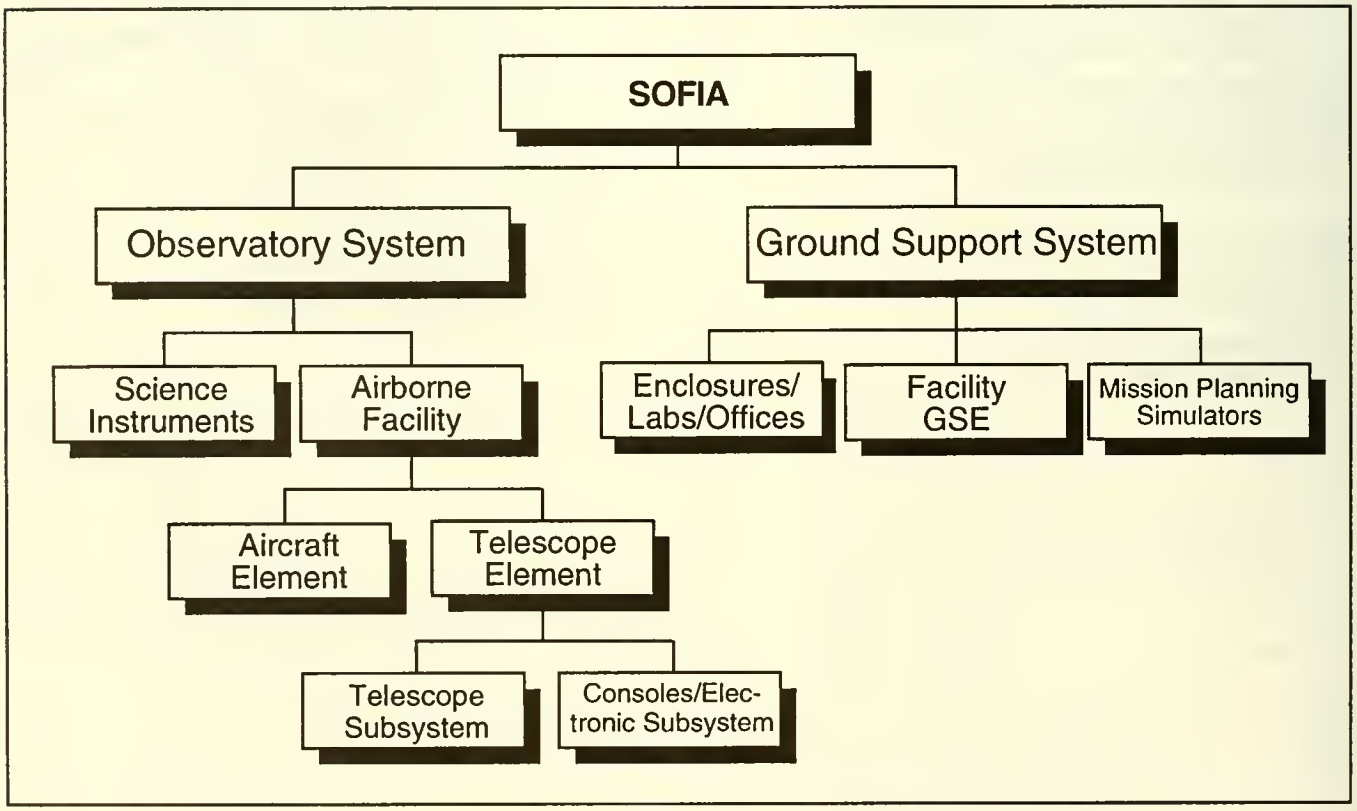


Figure B-1 — Stratospheric Observatory for Infrared Astronomy (SOFIA) Product Breakdown Structure.

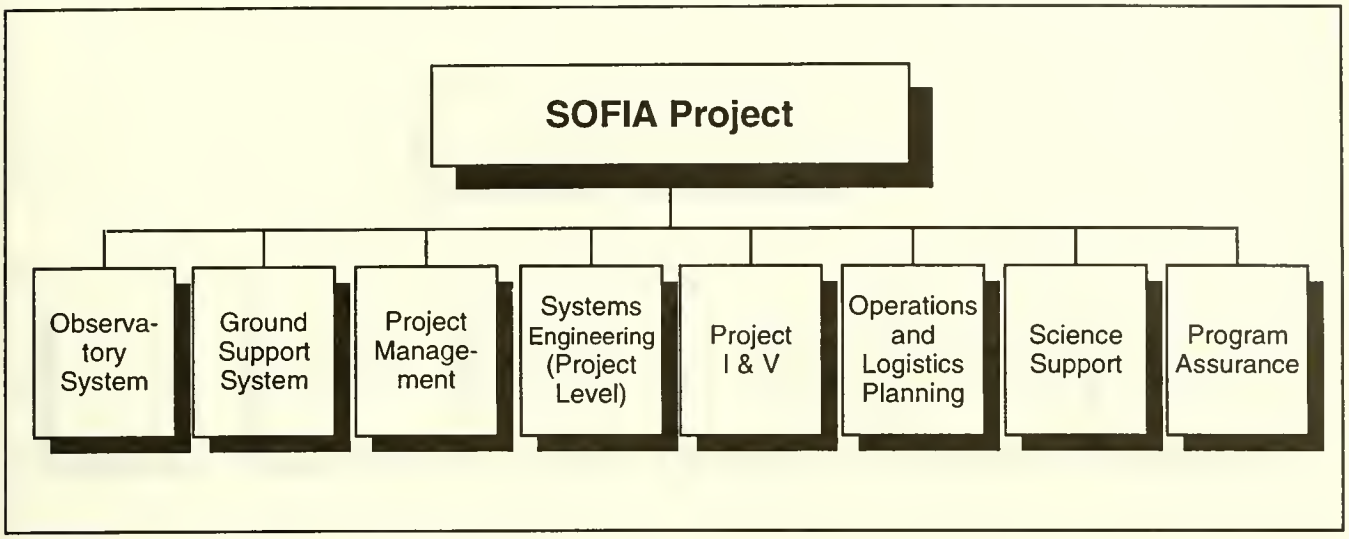


Figure B-2 — SOFIA Project WBS (*Level 3*).

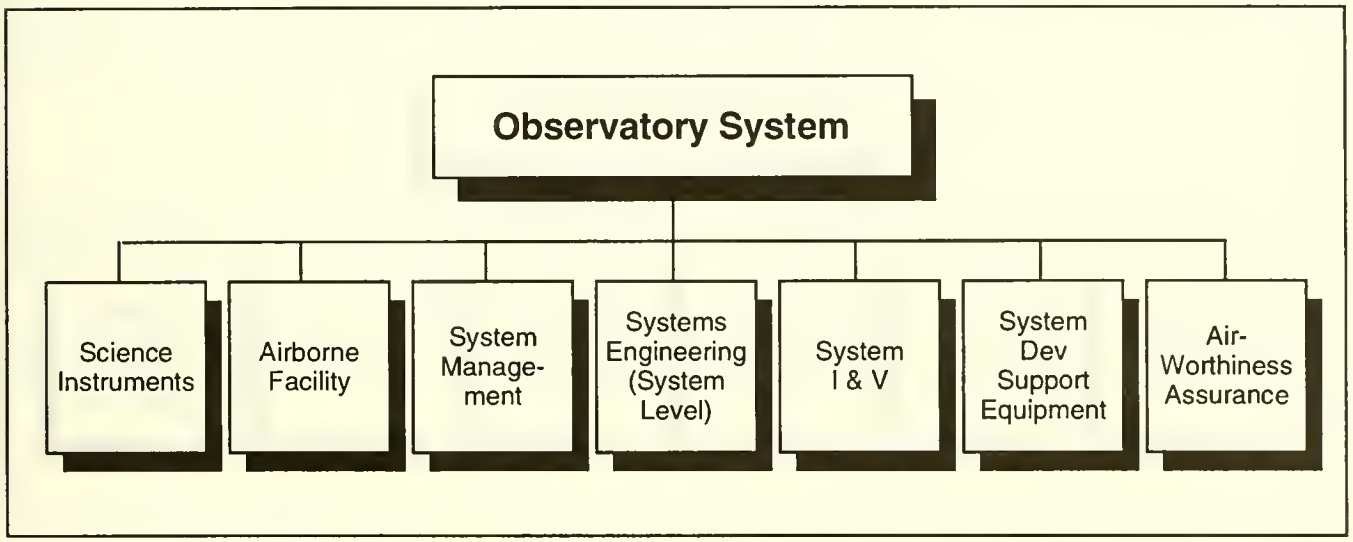


Figure B-3 — SOFIA Observatory System WBS (*Level 4*).

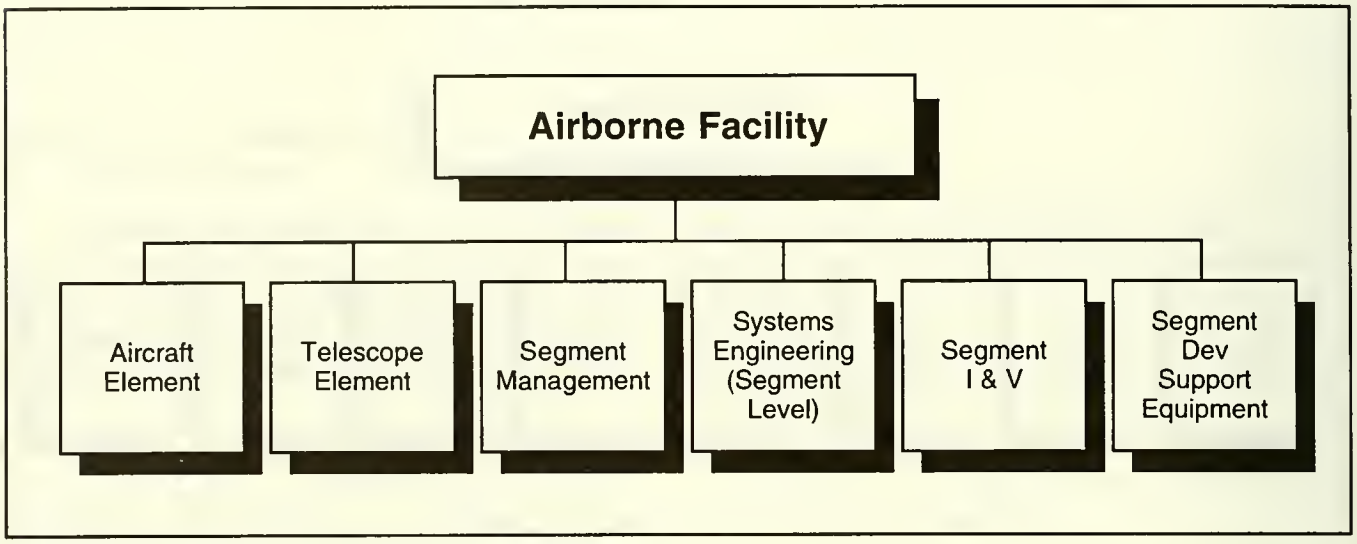


Figure B-4 — SOFIA Airborne Facility WBS (*Level 5*).

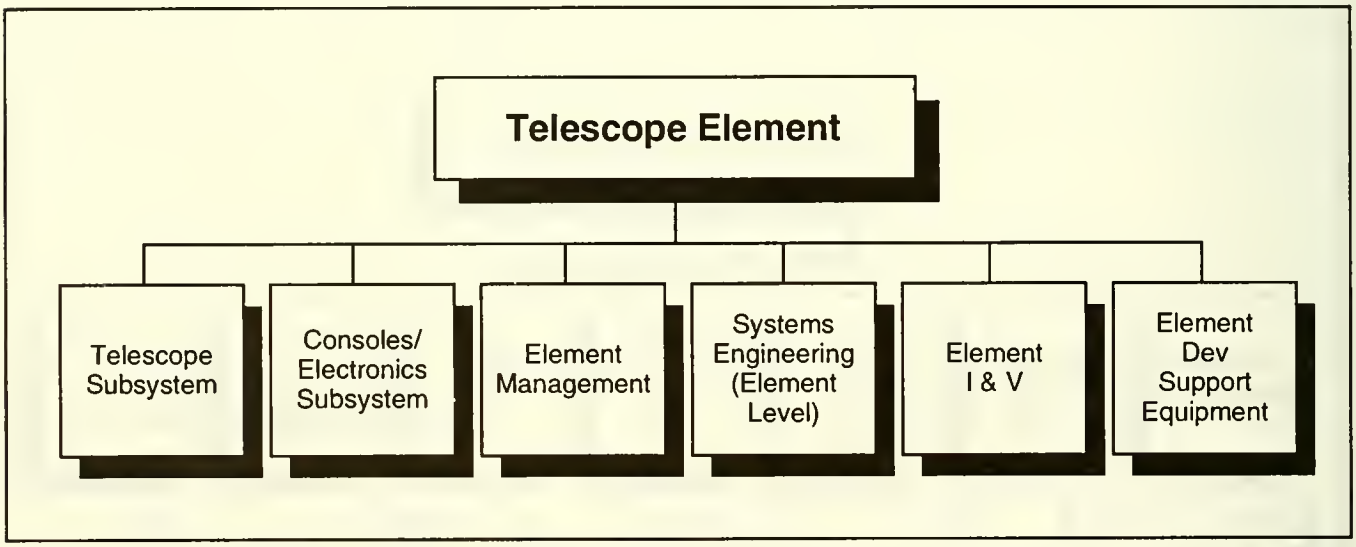


Figure B-5 — SOFIA Telescope Element WBS (*Level 6*).

Appendix B.3 — Characterization, Mission Success, and SRM&QA Cost Guidelines for Class A–D Payloads

Appendix B.3 is Attachment A of NM1 8010.1A, *Classification of NASA Payloads.*

	Class A	Class B	Class C	Class D
Characterization	High priority, minimum risk	High priority, medium risk	Medium priority, medium/high risk	High risk, minimum cost
Typical factors used to determine payload classifications	High national prestige; long hardware life required; high complexity; highest cost; long program duration; critical launch constraints; retrieval/ reflight or in-flight maintenance to recover from problems is not feasible.	High national prestige; medium hardware life required; high to medium complexity; high cost; medium program duration; some launch constraints; retrieval/reflight or in-flight maintenance to recover from problems is difficult or not feasible.	Moderate national prestige; short hardware life required; medium to low complexity; medium cost; short program duration; few launch constraints; retrieval/reflight or in-flight maintenance to recover from problems may be feasible.	Little national presitge; short hardware life required; low complexity; low cost; short program duration; non-critical launch time/orbit constraints; re-flyable or economically replaceable; in-flight maintenance may be feasible.
Achievement of mission success criteria	All affordable programmatic and other measures are taken to achieve minimum risk. The highest practical product assurace standards are utilized.	Compromises are used to permit somewhat reduced costs while maintaining a low risk to the overall mission success and a medium risk of achieving only partial success.	Moderate risks of not achieving mission success are accepted to permit significant cost savings. Reduced product assurance requirements are allowed.	Significant risk of not achieving mission success is accepted to permit minimum costs. Minimal product assurance requirements are allowed.
Estimated relative [1] SRM&QA cost factors	1.0	0.7 × Class A	0.4 × Class A	0.1 × Class A

Note [1]: There are wide variations in the methods for specifying and accounting for "SRM&QA costs". For Class A programs, these costs are typically in the range of 10.15% of the total program cost. The relative SRM&QA cost factors specified here are intended to require substantive differences in the SRM&QA programs (and the associated costs) for the various program classifications in order to establish a meaningful ladder of cost/risk levels.

Appendix B.4 — A Sample Risk Management Plan Outline

1.0 Introduction
- 1.1 Purpose and Scope of the RMP
- 1.2 Applicable Documents and Definitions
- 1.3 Program/Project (or System) Description

2.0 Risk Management Approach
- 2.1 Risk Management Philosophy/Overview
- 2.2 Management Organization and Responsibilities
- 2.3 Schedule, Milestones, and Reviews
- 2.4 Related Program Plans
- 2.5 Subcontractor Risk Management
- 2.6 Program/Project Risk Metrics

3.0 Risk Management Methodologies, Processes, and Tools
- 3.1 Risk Identification and Characterization
- 3.2 Risk Analysis
- 3.3 Risk Mitigation and Tracking

4.0 Significant Identified Risks*
- 4.1 Technical Risks
- 4.2 Programmatic Risks
- 4.3 Supportability Risks
- 4.4 Cost Risks
- 4.5 Schedule Risks

** Each subsection contains risk descriptions, characterizations, analysis results, mitigation actions, and reporting metrics .*

Appendix B.5 — An Example of a Critical Items List

```
                        SHUTTLE CRITICAL ITEMS LIST - ORBITER

 SUBSYSTEM       :LANDING DECELERATION    FMEA NO 02-1  -001  -1          REV:02/09/82
.ASSEMBLY        :MAIN LANDING GEAR       ABORT:                  CRIT.   FUNC:     1
.P/N RI          :MC621-0011                                      CRIT.    HDW:     1
.P/N VENDOR      :1170100 MENASCO                 VEHICLE   102   099      103     104
.QUANTITY        :2                       EFFECTIVITY:       x     x        x       x
.                :LEFT HAND               PHASE(S)    PL     LO    00     DO  X  LS
.                :RIGHT HAND
.                                  REDUNDANCY SCREEN:        A-N/A    B-N/A   C-N/A
.
.PREPARED BY:                      APPROVED BY:                       APPROVED BY (NASA):
.DES             L L RHODES        DES _____________________          SSM _____________________
.REL             A L DOBNER        REL _____________________          REL _____________________
.
.
.ITEM: MLG STRUT
.       MLG SHOCK STRUT INNER AND OUTER CYLINDER AND LOAD CARRYING MEMBERS.
.FUNCTION:
.       MLG LOAD CARRYING MEMBERS CYLINDER - DAMPER, WHERE A PASSAGE OF
        HYDRAULIC FLUID THROUGH AN ORIFICE ABSORBS THE ENERGY OF IMPACT AND
        WHERE DRY NITROGEN IS USED AS THE ELASTIC MEDIUM TO RESTORE THE
        UNSPRUNG PARTS TO THEIR EXTENDED POSITION.
.FAILURE MODE:  STRUCTURAL FAILURE
.
.CAUSE(S):
.       STRESS CORROSION.  PIECE-PART STRUCTURAL FAILURE.  OVERLOAD.
.EFFECT(S) ON (A) SUBSYSTEM (B) INTERFACES (C) MISSION (D) CREW/VEHICLE:
.       (A) LOSS OF SUBSYSTEM FUNCTION.  (B) NONE.  (C) NONE.  (D) PROBABLE
        LOSS OF VEHICLE IF MAIN STRUT FAILS ON LANDING.
.DISPOSITION & RATIONALE (A) DESIGN (B) TEST (C) INSPECTION (D) FAILURE HISTORY:
.       (A) UNDER WORST CASE LOADING (FLAT STRUT) THE STRUT IS CAPABLE OF
        WITHSTANDING ONE LANDING AT THE NORMAL LANDING DESIGN GROSS WEIGHT OF
        207,000 LBS. AND SINK SPEED OF 9.6 FEET PER SECOND WITH CORRESPONDING
        LANDING ROLLOUT AND BRAKING CONDITIONS, WITH NO YIELDING OF THE
        STRUCTURAL MEMBERS.  (B) ACCEPTANCE INCLUDES VERIFICATION THAT
        CERTIFIED MATERIALS AND PROCESSES WERE USED.  CERTIFICATION INCLUDES A
        FATIGUE LOAD TEST SPECTRUM (REF MC62-0011 TABLES 10-11) REPRESENTING THE
        EQUIVALENT LOADING FOR THE LIFE OF EACH LANDING GEAR WITH A SCATTER
        FACTOR OF 4.0  THE STATIC LOAD TESTS INCLUDED A TAXI BUMP (65K
        PAYLOAD), VEHICLE WEIGHT 227 KIPS/AND A RIGHT TURN/WHICH IS THE WORST
        CASE CONDITIONS WITHOUT FAILURE.  (C) DURING TURNAROUND-VISUALLY INSPECT
        FOR DAMAGE.  USE MORE TO SUPPORT SUSPECT AREAS.  AT MANUFACTURER-RAW
        MATERIAL VERIFIED-VISUAL INSPE./ID PERFORMED-PARTS PROTECTION, COATING
        AND PLATING PROCESSES VERIF. BY INSPECTION.-MANUF., INSTL. AND ASSY.
        OPERATIONS VERIF. BY SHOP TRAVELER MIPS-CORROSION PROTECTION PROVISIONS
        VERIF. NDE OF SURFACE AND SUB-SURFACE DEFECTS VERIF. BY INSPECTION.
        PROPERLY MONITORED HANDLING AND STORAGE ENVIRONMENT VERIFIED. MATL. AND
        EQUIPMENT CONFORMANCE TO CONTRACT REQMTS. VERIFIED BY INSP.-FINDINGS
        VERIFIED BY AUDIT 9-25-78.  (D) DURING DROP TEST PROGRAM, THE OUTER
        GLAND NUT FAILED.  MENASCO REDESIGNED AND CHANGED FROM ALUMINUM TO STEEL
        MATL.  THE SNUBBER RING P/N 1170134-1 WAS REDESIGNED.  UPPER BEARING
        1170107-1 WAS REPLACED BY A SOLID ALUMINUM-BRONZE BEARING.
```

Appendix B.6 — A Sample Configuration Management Plan Outline

1.0 Introduction
- 1.1 Description of the CIs
- 1.2 Program Phasing and Milestones
- 1.3 Special Features

2.0 Organization
- 2.1 Structure and Tools
- 2.2 Authority and Responsibility
- 2.3 Directives and Reference Documents

3.0 Configuration Identification
- 3.1 Baselines
- 3.2 Specifications

4.0 Configuration Control
- 4.1 Baseline Release
- 4.2 Procedures
- 4.3 CI Audits

5.0 Interface Management
- 5.1 Documentation
- 5.2 Interface Control

6.0 Configuration Traceability
- 6.1 Nomenclature and Numbering
- 6.2 Hardware Identification
- 6.3 Software and Firmware Identification

7.0 Configuration Status Accounting and Communications
- 7.1 Data Bank Description
- 7.2 Data Bank Content
- 7.3 Reporting

8.0 Configuration Management Audits

9.0 Subcontractor/Vendor Control

Appendix B.7 — Techniques of Functional Analysis

Appendix B.7 is reproduced from the *Defense Systems Management Guide*, published January 1990 by the Defense Systems Management College, Ft. Belvoir, VA.

• • •

System requirements are analyzed to identify those functions which must be performed to satisfy the objectives of each functional area. Each function is identified and described in terms of inputs, outputs, and interface requirements from top down so that subfunctions are recognized as part of larger functional areas. Functions are arranged in a logical sequence so that any specified operational usage of the system can be traced in an end-to-end path. Although there are many tools available, functional identification is accomplished primarily through the use of 1) functional flow block diagrams (FFBDs) to depict task sequences and relationships, 2) N^2 diagrams to develop data interfaces, and 3) time line analyses to depict the time sequence of time-critical functions.

B.7.1 Functional Flow Block Diagrams

The purpose of the FFBD is to indicate the sequential relationship of all functions that must be accomplished by a system. FFBDs depict the time sequence of functional events. That is, each function (represented by a block) occurs following the preceding function. Some functions may be performed in parallel, or alternate paths may be taken. The duration of the function and the time between functions is not shown, and may vary from a fraction of a second to many weeks. The FFBDs are function oriented, not equipment oriented. In other words, they identify "what" must happen and do not assume a particular answer to "how" a function will be performed.

FFBDs are developed in a series of levels. FFBDs show the same tasks identified through functional decomposition and display them in their logical, sequential relationship. For example, the entire flight mission of a spacecraft can be defined in a top level FFBD, as shown in Figure B-6. Each block in the first level diagram can then be expanded to a series of functions, as shown in the second level diagram for "perform mission operations." Note that the diagram shows both input (transfer to operational orbit) and output (transfer to space transportation system orbit), thus initiating the interface identification and control process. Each block in the second level diagram can be progressively developed into a series of functions, as shown in the third level diagram on Figure B-6. These diagrams are used both to develop requirements and to identify profitable trade studies. For example, does the spacecraft antenna acquire the tracking and data relay satellite (TDRS) only when the payload data are to be transmitted, or does it track TDRS continually to allow for the reception of emergency commands or transmission of emergency data? The FFBD also incorporates alternate and contingency operations, which improve the probability of mission success. The flow diagram provides an understanding of total operation of the system, serves as a basis for development of operational and contingency procedures, and pinpoints areas where changes in operational procedures could simplify the overall system operation. In certain cases, alternate FFBDs may be used to represent various means of satisfying a particular function until data are acquired, which permits selection among the alternatives.

B.7.2 N^2 Diagrams

The N^2 diagram has been used extensively to develop data interfaces, primarily in the software areas. However, it can also be used to develop hardware interfaces. The basic N^2 chart is shown in Figure B-7. The system functions are placed on the diagonal; the remainder of the squares in the N x N matrix represent the interface inputs and outputs. Where a blank appears, there is no interface between the respective functions. Data flows in a clockwise direction between functions (e.g., the symbol F_1 F_2 indicates data flowing from function F_1 to function F_2). The data being transmitted can be defined in the appropriate squares. Alternatively, the use of circles and numbers permits a separate listing of the data interfaces as shown in Figure B-8. The clockwise flow of data between functions that have a feedback loop can be illustrated by a larger circle called a control loop. The identification of a critical function is also shown in Figure B-8, where function F_4 has a number of inputs and outputs to all other functions in the upper module. A simple flow of interface data exists between the upper and lower modules at functions F_7 and F_8. The lower module has complex interaction among its functions. The N^2 chart can be taken down into successively lower levels to the hardware and software component functional levels. In addition to defining the data that must be supplied across the interface, the N^2 chart can pinpoint areas where conflicts could arise.

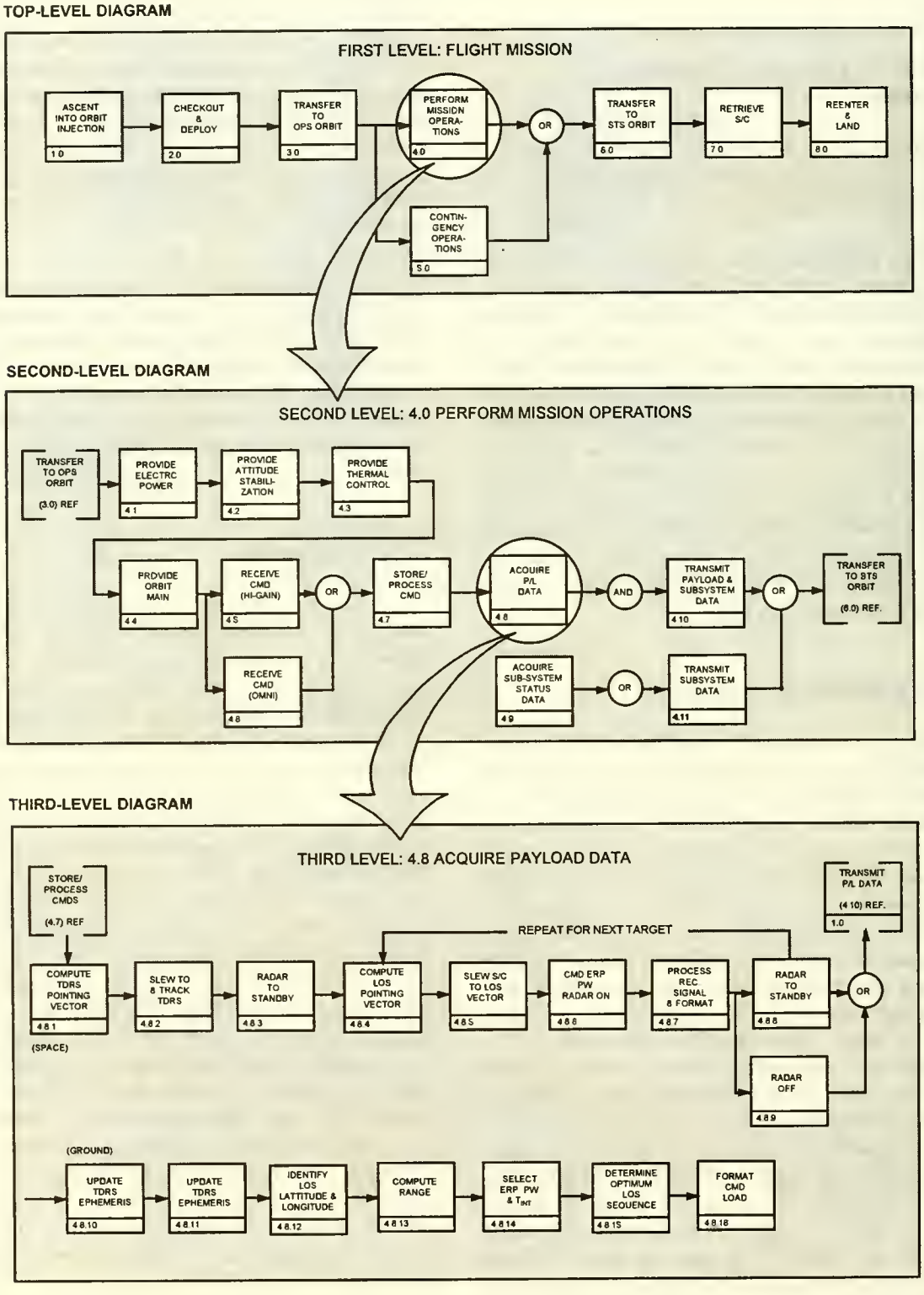


Figure B-6 — Development of Functional Flow Block Diagrams.

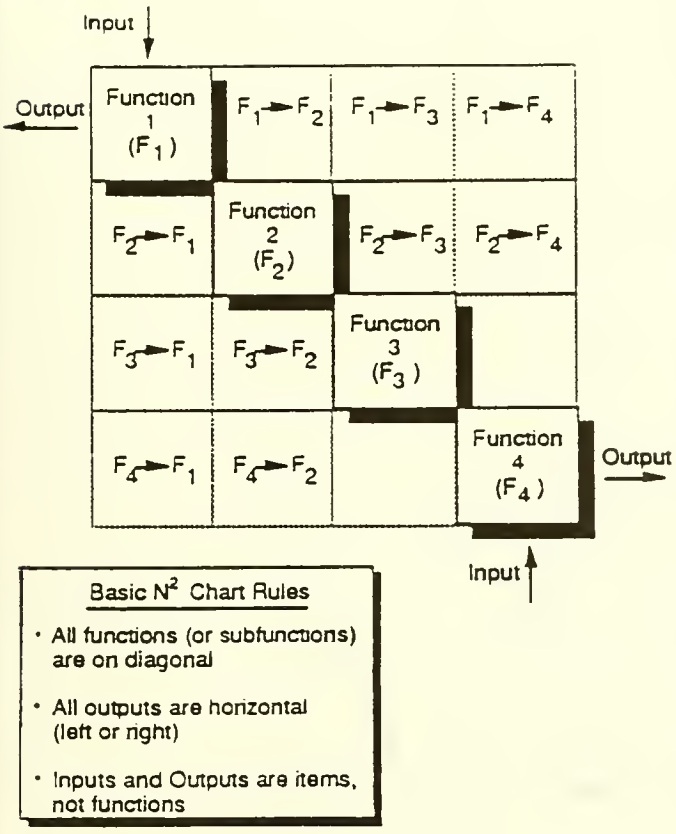


Figure B-7 — N^2 Chart Definition.

B.7.3 Time Line Analysis

Time line analysis adds consideration of functional durations and is used to support the development of design requirements for operation, test and maintenance functions. The time line sheet (TLS) is used to perform and record the analysis of time critical functions and functional sequences. Additional tools such as mathematical models and computer simulations may be necessary. Time line analysis is performed on those areas where time is critical to the mission success, safety, utilization of resources, minimization of down time, and/or increasing availability. Not all functional sequences require time line analysis, only those in which time is a critical factor. The following areas are often categorized as time critical: 1) functions affecting system reaction time, 2) mission turnaround time, 3) time countdown activities, and 4) functions requiring time line analysis to determine optimum equipment and/or personnel utilization. An example of a high level TLS for a space program is shown in Figure B-9.

For time critical function sequences, the time requirements are specified with associated tolerances. Time line analyses play an important role in the trade-off process between man and machine. The decisions between automatic and manual methods will be made and will determine what times are allocated to what subfunctions. In addition to defining subsystem/component time requirements, time line analysis can be used to develop trade studies in areas other than time consideration (e.g., should the spacecraft location be determined by the ground network or by onboard computation using navigation satellite inputs? Figure B-10 is an example of a maintenance TLS which illustrates that availability of an item (a distiller) is dependent upon the completion of numerous maintenance tasks accomplished concurrently. Furthermore, it illustrates the traceability to higher level requirements by referencing the appropriate FFBD and requirement allocation sheet (RAS).

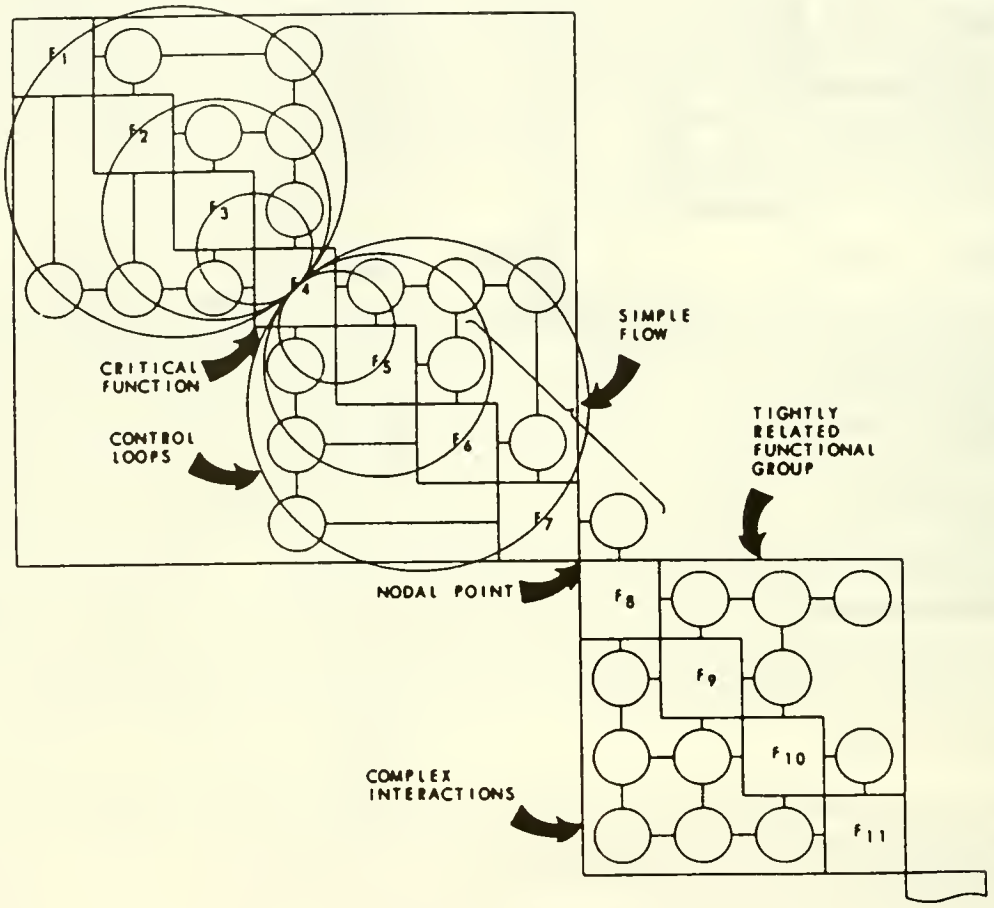


Figure B-8 — N^2 Chart Key Features (from "The N^2 Chart", R. Lano, © 1977 TRW Inc.)

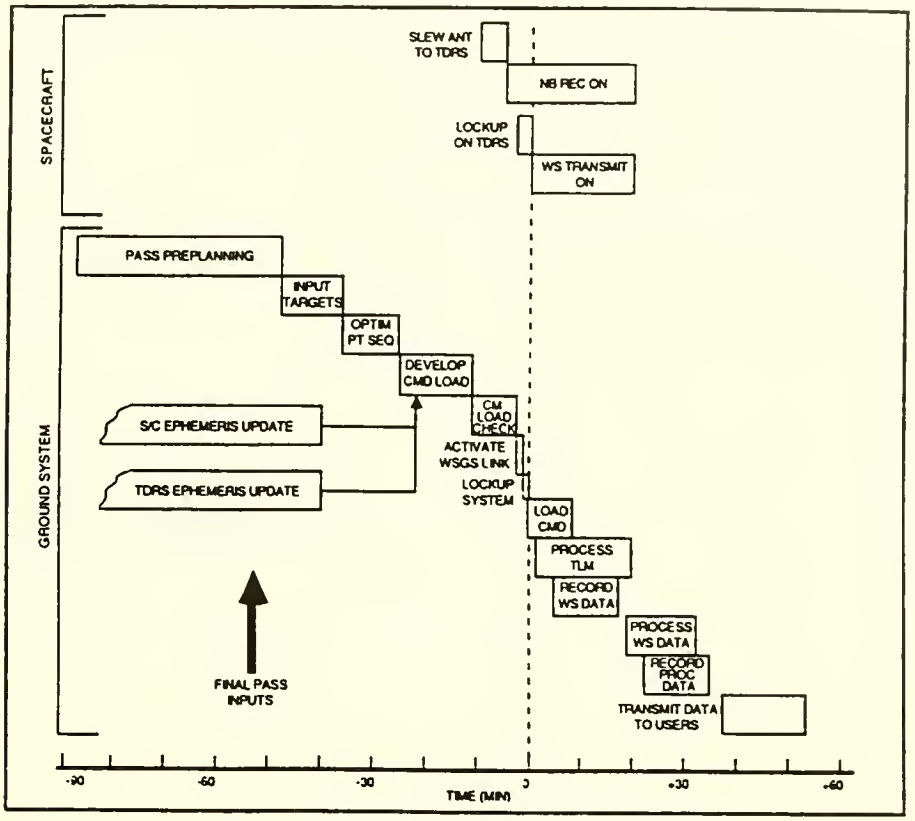


Figure B-9 — Flight Mission Time Lines.

TIME LINE SHEET	(A) FUNCTION - PERFORM PERIODIC MAINT ON VC DISTILLER	(B) LOCATION - ENGINE ROOM 3	(C) TYPE OF MAINT - SCHEDULED 200 HR PM
(D) SOURCE - FFBD 37.5X3	(E) FUNCTION & TASKS - RAS 37.5X37	(F) TIME - HOURS .5 1.0	

TASK SEQ. #	TASK	CREW MEMBER	
.01	INSPECT COMPRESSOR BELT	A2	.3H
.02	LUBRICATE BLOWDOWN PUMP	B1	.2H
.03	CHECK MOUNTING BOLTS	B1	.1H
.04	CLEAN BREATHER CAP	B1	.1H
.05	CLEAN FOOD STRAINER	C1	.5H
.06	REPLACE OIL	B1	.2H
.07	REPLACE FILTER	C1	.4H
.08	REPLACE V-DRIVE BELT	D1	.9H
.09	CLEAN & INSPECT CONTROL PANEL	C1	.1H
.10	INSTALL NEW DIAPHRAGMS	A2	.7H
.11	CLEAN CONTROLS	B1	.1H

TOTAL MANHOURS — 3.6 MH
ELAPSED TIME — 1.0 H

Figure B-10 — Sample Maintenance Time Line Sheet.

Appendix B.8 — The Effect of Changes in ORU MTBF on Space Station *Freedom* Operations

The reliability of Space Station *Freedom*'s (SSF) Orbital Replacement Units (ORUs) has a profound effect on its operations costs. This reliability is measured by the Mean Time Between Failures (MTBF). One study of the effects, by Dr. William F. Fisher and Charles Price, was *SSF External Maintenance Task Team Final Report* (JSC, July 1990). Another, by Anne Accola, et al., shows these effects parametrically. Appendix B.8 excerpts this paper, *Sensitivity Study of SSF Operations Costs and Selected User Resources* (presented at the International Academy of Astronautics Symposium on Space Systems Costs Methodologies and Applications, May 1990).

• • •

There are many potential tradeoffs that can be performed during the design stage of SSF. Many of them have major implications for crew safety, operations cost, and achievement of mission goals. Operations costs and important non-cost operations parameters are examined. One example of a specific area of concern in design is the reliability of the ORUs that comprise SSF. The implications of ORU reliability on logistics upmass and downmass to and from SSF are great, thus affecting the resources available for utilization and for other operations activities. In addition, the implications of reliability on crew time available for mission accomplishment (i.e., experiments) vs. station maintenance are important.

The MTBF effect on operations cost is shown in Figure B-11. Repair and spares costs are influenced greatly by varying MTBF. Repair costs are inversely proportional to MTBF, as are replacement spares. The initial spares costs are also influenced by variables other than MTBF. The combined spares cost, consisting of initial and replacement spares are not as greatly affected as are repair costs. The five-year operations cost is increased by only ten percent if all ORU MTBF are halved. The total operations cost is reduced by three percent if all ORU MTBF are doubled. It would almost appear that MTBF is not as important as one would think. However, MTBF also affects available crew time and available upmass much more than operations cost as shown in Figures B-12 and B-13.

Available crew time is a valuable commodity because it is a limited resource. Doubling the number of ORU replacements (by decreasing the MTBF) increases the maintenance crew time by 50 percent, thus reducing the amount of time available to perform useful experiments

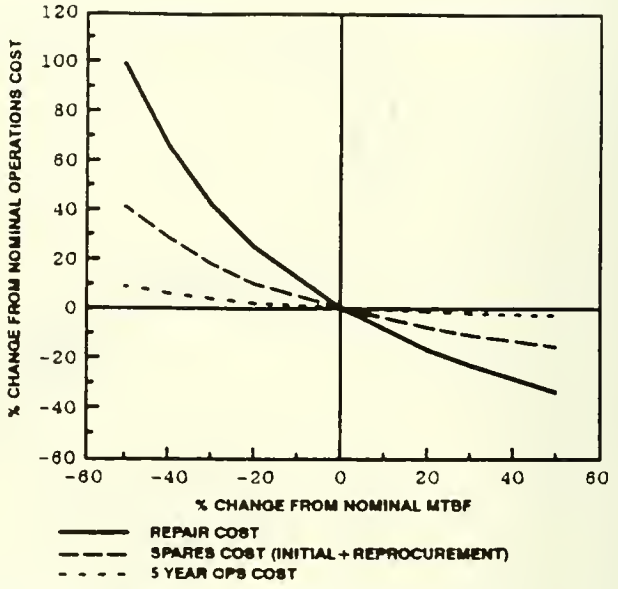


Figure B-11 — Effect of MTBF on Operations Cost.

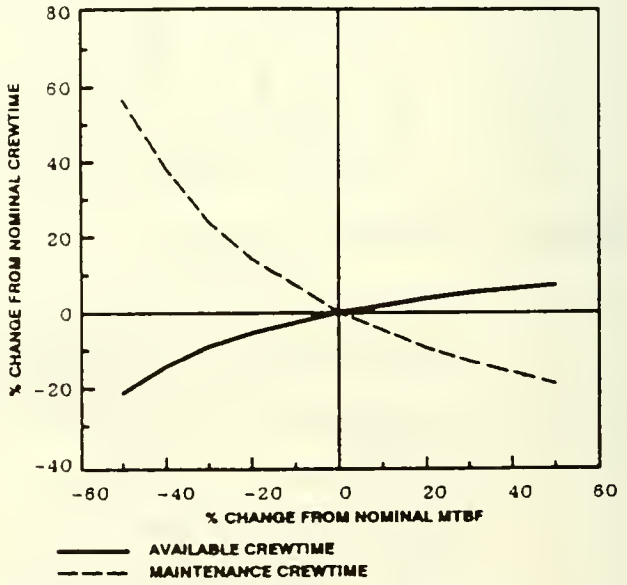


Figure B-12 — Effect of MTBF on Crew Time.

or scientific work by 22 percent. By halving the ORU replacements, the maintenance crew time decreases by 20 percent and the available crew time increases by eight percent.

Available upmass is another valuable resource because a fixed number of Space Shuttle flights can transport only a fixed amount of payload to the SSF. Extra ORUs taken to orbit reduces available upmass that could be used to take up experimental payloads. Essentially, by doubling the number of ORU replacements, the available upmass is

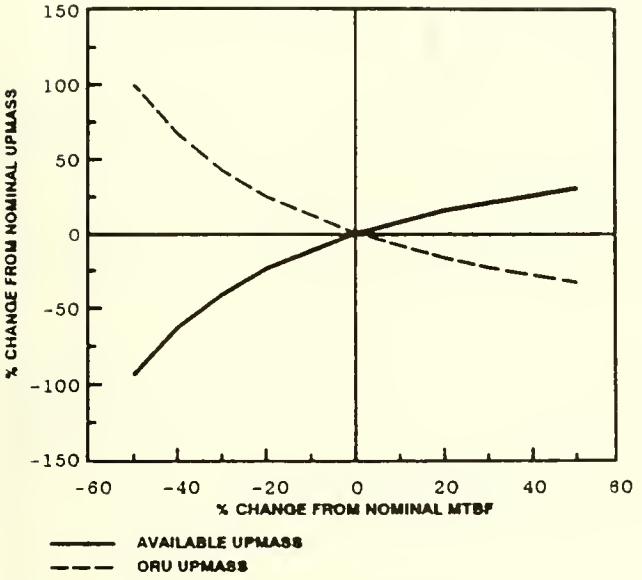


Figure B-13 — Effect of MTBF on Upmass.

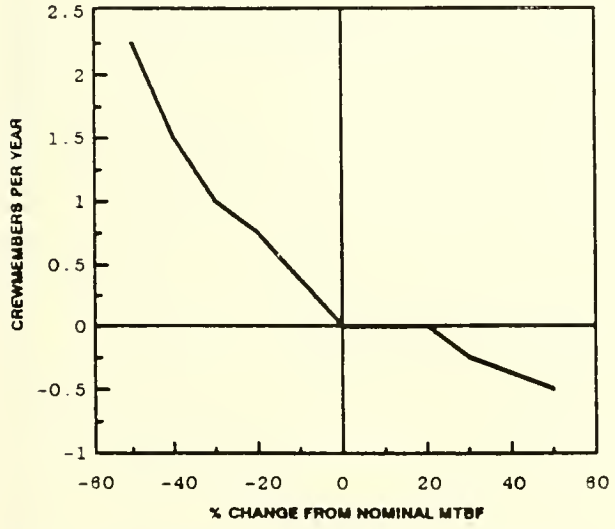


Figure B-14 — Effect of MTBF on Number of Crew (Available Crew Time Maintained).

driven to zero. Conversely, halving the number of ORU replacements increases the available upmass by 30 percent.

Although the effects of MTBF on resources is interesting, it is a good idea to quantify the effectiveness of the scenarios based on total cost to maintain the nominal resources. Figure B-14 shows the number of crew members needed each year to maintain the available crew time. The figure shows that to maintain the nominal available crew time after doubling the number of ORU replacements, the Station would need two extra crew members. It should be noted that no attempt was made to assess the design capability or design cost impacts to accommodate these extra crew members. The savings of crew due to halving the number of ORU replacements is small, effectively one less crew member for half the year.

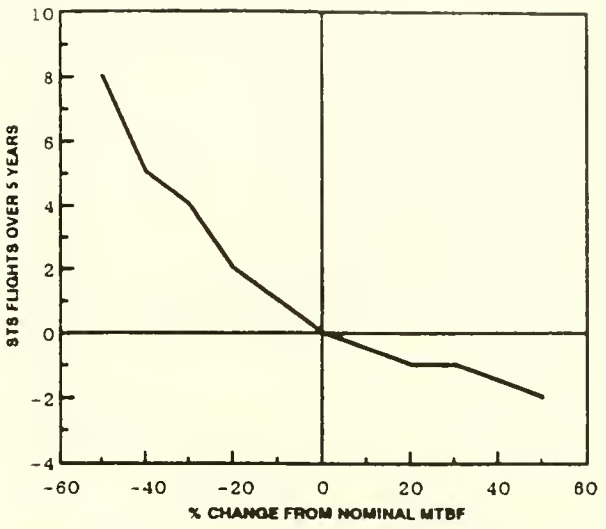


Figure B-15 — Effect of MTBF on Number of STS Flights (Available Upmass Maintained).

Figure B-15 shows the number of Space Shuttle flights over five years needed to maintain the nominal available upmass. The Space Shuttle flights were rounded upward to obtain whole flights. Doubling the number of ORU replacements would mean eight extra Space Shuttle flights would be needed over five years. Halving the ORU replacements would require two fewer Space Shuttle flights over five years. No attempt was made to assess the Space Shuttle capability to provide the extra flights or the design cost impacts to create the ORUs with the different reliabilities.

Figure B-16 shows the effect of assessing the cost impact of the previous two figures and combining them with the five-year operations cost. The influence of MTBF is effectively doubled when the resources of available upmass and crew time are maintained at their nominal values.

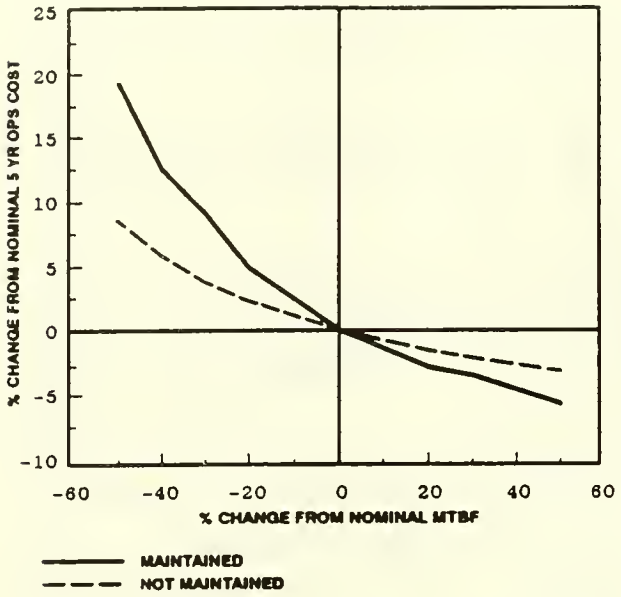


Figure B-16 — Effect of MTBF on Five-year Operations Cost (Maintaining vs. Not Maintaining Available Upmass and Crew Time).

Appendix B.9 — An Example of a Verification Requirements Matrix

Appendix B.9 is a small portion of the Verification Requirements Matrix (VRM) from the National Launch System Level III System Requirements Document, originally published at the Marshall Space Flight Center. Its purpose here is to illustrate the content and one possible format for a VRM. The VRM coding key for this example is shown on the next page.

Paragraph Number	Requirement Statement	A	B	C	D	E	F
3.2	3.2 PROVISIONS FOR MAN-RATING	0					
6.2	Basic vehicle design shall include design safety factors, reliability, and health monitoring necessary for manned flight.	2.4, 3.5					
3.2	The design of all flight-critical systems shall utilize high reliability parts and components, as defined in Section 3.20.5.14.	2.1, 3.5	4.1	4.1			
3.2	All critical systems whose failure could result in loss of vehicle shall utilize, as a minimum, fail-safe design.	3.2, 3.5					
3.2	The design shall provide an emergency detection system (EDS) as defined in Section 3.4.10.3	3.5					
3.3	3.3 OPERATIONS REQUIREMENTS	0					
3.3.1	3.3.1 Ground Operations Launch Vehicle Requirements	0					
3.3.1.1	3.3.1.1 Operation Readiness Requirements	0					
3.3.1.1.1	The 1.5 Stage LV and HLV shall provide an operational time fraction (see Section 6.1.1) of at least TBD for total 1.5 Stage LV and HLLV flight rates up to 44 flights per year.	2.4					
3.3.1.1.2	At each launch site, Launch Vehicles shall beet a 90 percent probability of being able to conduct launches within ten days of their scheduled dates (as defined no later than when vehicle integration begins).	2.4					
3.3.1.1.2	The LVs shall also meet a 95 percent probability of being able to conduct launches within twenty days of their scheduled dates.	2.4					
3.3.1.1.2	The LVs shall have the capability for launch in daylight or darkness	2.4					
3.3.1.1.3	The LVs shall incorporate means of discharging electrical potential differences between the Payload Carrier, payload, LV elements (e.g., Payload Carrier Adapter), and ground in accordance with Section 3.20.5.7.	2.4, 3.5			7.14, 7.17		
3.3.1.1.4	3.3.1.1.4 Sustainable Launch Rates	0					
3.3.1.1.4.1	The LVs shall be designed to support a maximum scheduled launch rate of three flights per year from KSC.	3.5					
3.3.1.1.4.2	The 1.5 Stage LV shall be designed to support a maximum scheduled launch rate of 14 flights per year (includes 4 flights to meet resiliency requirements) from CCAFS.	3.5					

Verification Requirements Matrix Coding Key

Verification Method/Level		Verification Stages	
Code	Method	Code	Stage
0	Title of Information Only	A	Development
1.0	Similarity	B	Qualification
1.1	Component Similarity	C	Acceptance
1.2	Subsystem Similarity	D	Prelaunch
2.0	Analysis	E	Flight/Operational
2.1	Component Analysis	F	Post Flight/Disposal
2.2	Subsystem Analysis		
2.3	Integrated Element Analysis		
2.4	Integrated Vehicle Analysis		
3.0	Inspection		
3.1	Component Inspection		
3.2	Subsystem Inspection		
3.3	Integrated Element Inspection		
3.4	Integrated Vehicle Inspection		
3.5	Review of Design Documentation		
4.0	Validation ot Records		
4.1	Component Validation of Records		
4.2	Subsystem Validation of Records		
4.3	Integrated Element Validation of Records		
4.4	Integrated Vehicle Validation of Records		
5.0	Demonstration		
5.1	Component Demonstration		
5.2	Subsystem Demonstration		
5.3	Integrated Element Demonstration		
5.4	Integrated Vehicle Demonstration		
6.0	Simulation		
6.1	Component Simulation		
6.2	Subsystem Simulation		
6.3	Integrated Element Simulation		
6.4	Integrated Vehicle Simulation		
7.0	Test		
7.1	Component Functional Test		
7.2	Component Environmental Test		
7.3	Component EMI/EMC Test		
7.4	Component Proof Test		
7.5	Other Component Test		
7.6	Subsystem Functional Test		
7.7	Subsystem Environmental Test		
7.8	Subsystem Proof Test		
7.9	Other Subsystem Test		
7.10	Integrated Element Functional Test		
7.11	Integrated Element Environmental Test		
7.12	Integrated Element EMI/EMC Test		
7.13	Integrated Element Interface Test		
7.14	Other Integrated Element Test		
7.15	Integrated Vehicle Functional Test		
7.16	Integrated Vehicle Environmental Test		
7.17	Integrated Vehicle Interface Test		
7.18	Hot Firing Test		

Appendix B.10 — A Sample Master Verification Plan Outline

1.0 Introduction
- 1.1 Scope
- 1.2 Applicable Documents
- 1.3 Document Maintenance and Control

2.0 Program/Project Description
- 2.1 Program/Project Overview and Verification Master Schedule
- 2.2 Systems Descriptions
- 2.3 Subsystems Descriptions

3.0 Integration and Verification (I&V) Organization and Staffing
- 3.1 Program/Project Management Offices
- 3.2 NASA Field Center I&V Organizations
- 3.3 International Partner I&V Organizations
- 3.4 Prime Contractor I&V Organization
- 3.5 Subcontractor I&V Organizations

4.0 Verification Team Operational Relationships
- 4.1 Verification Team Scheduling and Review Meetings
- 4.2 Verification and Design Reviews
- 4.3 Data Discrepancy Reporting and Resolution Procedures

5.0 Systems Qualification Verification
- 5.1 Tests*
- 5.2 Analyses
- 5.3 Inspections
- 5.4 Demonstrations

6.0 Systems Acceptance Verification
- 6.1 Tests*
- 6.2 Analyses
- 6.3 Inspections
- 6.4 Demonstrations

7.0 Launch Site Verification

8.0 On-Orbit Verification

9.0 Post-Mission/Disposal Verification

10.0 Verification Documentation

11.0 Verification Methodology

12.0 Support Equipment
- 12.1 Ground Support Equipment
- 12.2 Flight Support Equipment
- 12.3 Transportation, Handling, and Other Logistics Support
- 12.4 TDRSS/NASCOM Support

13.0 Facilities

** This section contains subsections for each type of test, e.g., EMI/EMC, mechanisms, thermal/vacuum. This further division by type applies also to analyses, inspections, and demonstrations.*

Appendix C — Use of the Metric System

C.1 NASA Policy

It is NASA policy (see NMI 8010.2A and NHB 7120.5) to:

- Adopt the International System of Units, known by the international abbreviation *SI* and defined by ANSI/IEEE Std 268-1992, as the preferred system of weights and measurements for all major system development programs.
- Use the metric system in procurements, grants and business-related activities to the extent economically feasible.
- Permit continued use of the inch-pound system of measurement for existing systems.
- Permit hybrid metric and inch-pound systems when full use of metric units is impractical or will compromise safety or performance.

C.2 Definitions of Units

Parts of Appendix C are reprinted from IEEE Std 268-1992, *American National Standard for Metric Practice*, Copyright © 1992 by the Institute of Electrical and Electronics Engineers, Inc. The IEEE disclaims any responsibility or liability resulting from the placement and use in this publication. Information is reprinted with the permission of the IEEE.

• • •

Outside the United States, the comma is widely used as a decimal marker. In some applications, therefore, the common practice in the United States of using the comma to separate digits into groups of three (as in 23,478) may cause ambiguity. To avoid this potential source of confusion, recommended international practice calls for separating the digits into groups of three, counting from the decimal point toward the left and the right, and using a thin space to separate the groups. In numbers of four digits on either side of the decimal point the space is usually not necessary, except for uniformity in tables.

C.2.1 SI Prefixes

The names of multiples and submultiples of SI units may be formed by application of the prefixes and symbols shown in the sidebar. (The unit of mass, the *kilogram*, is the only exception; for historical reasons, the *gram* is used as the base for construction of names.)

Prefixes for SI Units

Factor	Prefix	Sym.	Pronunciation
10^{24}	yotta	Y	YOTT-a (*a* as in *a*bout)
10^{21}	zetta	Z	ZETT-a (*a* as in *a*bout)
10^{18}	exa	E	EX-a (*a* as in *a*bout)
10^{15}	peta	P	PET-a (as in *petal*)
10^{12}	tera	T	TERR-a (as in *terrace*)
10^{9}	giga	G	GIGa (*g* as in *g*iggle, *a* as in *a*bout)
10^{6}	mega	M	MEG-a)as in *mega*phone)
10^{3}	kilo	k	KILL-oh**
10^{2}	hecto*	h	HECK-toe
10	deka*	da	DECK-a (as in *deca*hedron)
1			
10^{-1}	deci*	d	DESS-ih (as in *deci*mal)
10^{-2}	centi*	c	SENT-ih (as in *centi*pede)
10^{-3}	milli	m	MILL-ih (as in *mili*tary)
10^{-6}	micro	μ	MIKE-roe (as in *micro*phone)
10^{-9}	nano	n	NAN-oh (*a* as in ant)
10^{-12}	pico	p	PEEK-oh
10^{-15}	femto	f	FEM-toe
10^{-18}	atto	a	AT-toe (*a* as in h*a*t)
10^{-21}	zepto	z	ZEP-toe (*e* as in st*e*p)
10^{-24}	yocto	y	YOCK-toe

* The prefixes that do not represent 1000 raised to a power (that is, *hecto, deka, deci*, and *centi*) should be avoided where practical.
** The first syllable of every prefix is accented to assure that the prefix will retain its identity. *Kilometer* is *not* an exception.

C.2.2 Base SI Units

ampere (A) The *ampere* is that constant current which, if maintained in two straight parallel conductors of infinite length, of negligible circular cross section, and placed one meter apart in vacuum, would produce between these conductors a force equal to 2×10^{-7} newton per meter of length.

candela (cd) The *candela* is the luminous intensity, in a given direction, of a source that emits monochromatic radiation of frequency 540×10^{12} Hz and that has a radiant intensity in that direction of 1/683 watt per steradian.

kelvin (K) The *kelvin*, unit of thermodynamic temperature, is the fraction 1/273.16 of the thermodynamic temperature of the triple point of water.

kilogram (kg) The *kilogram* is the unit of mass; it is equal to the mass of the international prototype of the kilogram. (The international prototype of the kilogram is a particular cylinder of platinum-iridium alloy which is preserved in a vault at Sèvres, France, by the International Bureau of Weights and Measures.)

meter (m) The *meter* is the length of the path traveled by light in a vacuum during a time interval of 1 299 792 458 of a second.

mole (mol) The *mole* is the amount of substance of a system which contains as many elementary entities as there are atoms in 0.012 kilogram of carbon-12. **Note:** When the mole is used, the elementary entities must be specified and may be atoms, molecules, ions, electrons, other particles, or specified groups of such particles.

second (s) The *second* is the duration of 9 192 631 770 periods of the radiation corresponding to the transition between the two hyperfine levels of the ground state of the cesium-133 atom.

C.2.3 Supplementary SI Units

radian (rad) The *radian* is the plane angle between two radii of a circle that cut off on the circumference an arc equal in length to the radius.

steradian (sr) The *steradian* is the solid angle that, having its vertex in the center of a sphere, cuts off an area of the surface of the sphere equal to that of a square with sides of length equal to the radius of the sphere.

C.2.4 Derived SI Units with Special Names

In addition to the units defined in this subsection, many quantities are measured in terms of derived units which do not have special names — such as *velocity* in **m/s**, *electric field strength* in **V/m**, *entropy* in **J/K**.

becquerel (Bq = 1/s) The *becquerel* is the activity of a radionuclide decaying at the rate of one spontaneous nuclear transition per second.

degree Celsius (°C = K) The *degree Celsius* is equal to the kelvin and is used in place of the kelvin for expressing Celsius temperature defined by the equation $t = T - T_0$, where t is the Celsius temperature, T is the thermodynamic temperature, and T_0 = 273.15 K (by definition).

coulomb (C = A·s) *Electric charge* is the time integral of electric current; its unit, the *coulomb*, is equal to one ampere second.

farad (F = C/V) The *farad* is the capacitance of a capacitor between the plates of which there appears a difference of potential of one volt when it is charged by a quantity of electricity equal to one coulomb.

gray (Gy = J/kg) The *gray* is the absorbed dose when the energy per unit mass imparted to matter by ionizing radiation is one joule per kilogram. (The *gray* is also used for the ionizing radiation quantities: specific energy imparted, kerma, and absorbed dose index.)

henry (H = Wb/A) The *henry* is the inductance of a closed circuit in which an electromotive force of one volt is produced when the electric current in the circuit varies uniformly at a rate of one ampere per second.

hertz (Hz = 1/s) The *hertz* is the frequency of a periodic phenomenon of which the period is one second.

joule (J = N·m) The *joule* is the work done when the point of application of a force of one newton is displaced a distance of one meter in the direction of the force.

lumen (lm = cd·sr) The *lumen* is the luminous flux emitted in a solid angle of one steradian by a point source having a uniform intensity of one candela.

lux (lx = lm/m^2) The *lux* is the illuminance produced by a luminous flux of one lumen uniformly distributed over a surface of one square meter.

newton (N = kg·m/s^2) The *newton* is that force which, when applied to a body having a mass of one kilogram, gives it an acceleration of one meter per second squared.

ohm (Ω = V/A) The *ohm* is the electric resistance between two points of a conductor when a constant difference of potential of one volt, applied between these two points, produces in this conductor a current of one ampere, this conductor not being the source of any electromotive force.

pascal (Pa = N/m^2) The *pascal* [which, in the preferred pronunciation, rhymes with *rascal*] is the pressure or stress of one newton per square meter.

siemens (S = A/V) The *siemens* is the electric conductance of a conductor in which a current of one ampere is produced by an electric potential difference of one volt.

sievert (Sv = J/kg) The *sievert* is the dose equivalent when the absorbed dose of ionizing radiation multiplied by the dimensionless factors Q (quality factor) and N (product of any other multiplying factors) stipulated by the International Commission on Radiological Protection is one joule per kilogram.

tesla ($T = Wb/m^2$) The *tesla* is the magnetic flux density of one weber per square meter. In an alternative approach to defining the magnetic field quantities the *tesla* may also be defined as the magnetic flux density that produces on a one-meter length of wire carrying a current of one ampere, oriented normal to the flux density, a force of one newton, magnetic flux density being defined as an axial vector quantity such that the force exerted on an element of current is equal to the vector product of this element and the magnetic flux density.

volt (V = W/A) The *volt* (unit of electric potential difference and electromotive force) is the difference of electric potential between two points of a conductor carrying a constant current of one ampere, when the power dissipated between these points is equal to one watt.

watt (W = J/s) The *watt* is the power that represents a rate of energy transfer of one joule per second.

weber (Wb = V·s) The *weber* is the magnetic flux that, linking a circuit of one turn, produces in it an electromotive force of one volt as it is reduced to zero at a uniform rate in one second.

C.2.5 Units in Use with SI

Time The SI unit of time is the *second.* This unit is preferred and should be used if practical, particularly when technical calculations are involved. In cases where time relates to life customs or calendar cycles, the minute, hour, day and other calendar units may be necessary. For example, vehicle speed will normally be expressed in kilometers per hour.

minute (min) 1 min = 60 s
hour (h) 1 h = 60 min = 3600 sec
day (d) 1 d = 24 h = 86 400 sec
week, month, etc.

Plane angle The SI unit for plane angle is the *radian.* Use of the degree and its decimal submultiples is permissible when the radian is not a convenient unit. Use of the minute and second is discouraged except for special fields such as astronomy and cartography.

degree (°) $1° = (\pi/180)$ rad
minute (′) $1' = (1/60)° = (\pi/10\,800)$ rad
second (″) $1'' = (1/60)' = (\pi/648\,000)$ rad

Area The SI unit of area is the *square meter* (m^2). The hectare (ha) is a special name for the square hectometer (hm^2). Large land or water areas are generally expressed in hectares or in square kilometers (km^2).

Volume The SI unit of volume is the *cubic meter.* This unit, or one of the regularly formed multiples such as the cubic centimeter, is preferred. The special name *liter* has been approved for the cubic decimeter, but use of this unit is restricted to volumetric capacity, dry measure, and measure of fluids (both liquids and gases). No prefix other than *milli-* or *micro-* should be used with *liter.*

Mass The SI unit of mass is the *kilogram.* This unit, or one of the multiples formed by attaching an SI prefix to *gram* (g), is preferred for all applications. The megagram (Mg) is the appropriate unit for measuring large masses such as have been expressed in tons. However, the name *ton* has been given to several large mass units that are widely used in commerce and technology: the long ton of 2240 lb, the short ton of 2000 lb, and the metric ton of 1000 kg (also called *tonne* outside the USA) which is almost 2205 lb. None of these terms are SI. The term *metric ton* should be restricted to commercial usage, and no prefixes should be used with it. Use of the term *tonne* is deprecated.

Others The ANSI/IEEE standard lists the *kilowatthour* (1 kWh = 3.6 MJ) in the category of "Units in Use with SI Temporarily". The SI unit of energy, the *joule,* together with its multiples, is preferred for all applications. The kilowatthour is widely used, however, as a measure of electric energy. This unit should not be introduced into any new areas, and eventually, it should be replaced by the megajoule. In that same "temporary" category, the standard also defines the *barn* ($1\ b = 10^{-28}\ m^2$) for cross section, the *bar* ($1\ bar = 10^5$ Pa) for pressure, the *curie* ($1\ Ci = 3.7 \times 10^{10}$ Bq) for radionuclide activity, the *roentgen* ($1\ R = 2.58 \times 10^{-4}$ C/kg) for X- and gamma-ray exposure, the *rem* for dose equivalent (1 rem = 0.01 Sv), and the *rad* (1 rd = 0.01 Gy) for absorbed dose.

C.3 Conversion Factors

One of the many places a complete set of conversion factors can be found is in ANSI/IEEE Std 268-1992. The abridged set given here is taken from that reference. Symbols of SI units are given in bold face type and in parentheses. Factors with an asterisk (*) between the number and its power of ten are exact by definition. To conform with the international practice, this section uses spaces — rather than commas — in number groups.

To convert from	to	Multiply by
acre foot	meter^3 (**m^3**)	1.233 5 E+03
acre	meter^2 (**m^2**)	4.046 9 E+03
astronomical unit	meter (**m**)	1.495 979 E+11
atmosphere (standard)	pascal (**Pa**)	1.013 25*E+05
barrel (for petroleum, 42 gal)	meter^3 (**m^3**)	1.589 873 E–01
board foot	meter^3 (**m^3**)	2.359 737 E–03
British thermal unit (International Table)	joule (**J**)	1.055 056*E+03
calorie (thermochemical)	joule (**J**)	4.184*E+00
calorie (International Table)	joule (**J**)	4.186 8*E+00
centimeter of mercury (0 °C)	pascal (**Pa**)	1.333 22 E+03
centimeter of water (4 °C)	pascal (**Pa**)	9.806 38 E+01
cup	milliliter (**mL**)	2.366 E+02
curie	becquerel (**Bq**)	3.7*E+10
day	second (**s**)	8.64*E+04
day (sidereal)	second (**s**)	8.616 409 E+04
degree (angle)	radian (**rad**)	1.745 329 E–02
degree Celsius	kelvin (**K**)	$T_K \equiv t°C + 273.15$
degree Fahrenheit	degree Celsius	$t°C \equiv (t°F - 32)/1.8$
degree Fahrenheit	kelvin (**K**)	$T_K \equiv (t°F + 459.67)/1.8$
degree Rankine	kelvin (**K**)	$T_K \equiv T°R/1.8$
dyne	newton (**N**)	1*E–05
electronvolt	joule (**J**)	1.602 19 E–19
erg	joule (**J**)	1*E–07
fathom	meter (**m**)	1.828 8 E+00
foot	meter (**m**)	3.048*E–01
foot of water (39.2 °F)	pascal (**Pa**)	2.989 07 E+03
footcandle	lux (**lx**)	1.076 391 E+01
footlambert	candela per meter^2 (**cd/m^2**)	3.426 259 E+00
ft·lbf	joule (**J**)	1.355 818 E+00
ft·lbf/s	watt (**W**)	1.355 818 E+00
ft·poundal	joule (**J**)	4.214 011 E–02
g (standard acceleration of free fall)	meter per second^2 (**m/s^2**)	9.806 65*E+00

To convert from	to	Multiply by
gallon (US liquid)	meter^3 (m^3)	3.785 412 E–03
gauss	tesla (**T**)	1*E–04
grain	kilogram (**kg**)	6.479 891*E–05
horsepower (550 ft·lbf/s)	watt (**W**)	7.456 999 E+02
hour	second (**s**)	3.6*E+03
hour (sidereal)	second (**s**)	3.590 170 E+03
inch	meter (**m**)	2.54*E–02
inch of mercury (32 °F)	pascal (**Pa**)	3.386 39 E+03
inch of water (60 °F)	pascal (**Pa**)	2.490 89 E+02
kilogram-force (kgf)	newton (**N**)	9.806 65 *E+00
kilowatt hour (kW·hr or kWh)	joule (**J**)	3.6*E+06
kip (1000 lbf)	newton (**N**)	4.448 222 E+03
knot (international)	meter per second (**m/s**)	5.144 444 E–01
lambert	candela per meter^2 (**cd/m^2**)	1/π*E+04
light year	meter (**m**)	9.461 E+15
liter	meter^3 (m^3)	1*E–03
maxwell	weber (**Wb**)	1*E–08
mho	siemens (**S**)	1*E+00
micron	meter (**m**)	1*E–06
mil	meter (**m**)	2.54*E–05
mile (international)	meter (**m**)	1.609 344*E+03
mile (US statute)	meter (**m**)	1.609 347 E+03
mile (nautical)	meter (**m**)	1.852*E+03
ounce (avoirdupois)	kilogram (**kg**)	2.834 952 E–02
ounce (troy or apothecary)	kilogram (**kg**)	3.110 348 E–02
ounce (US fluid)	meter^3 (m^3)	2.957 353 E–05
parsec	meter (**m**)	3.085 678 E+16
pica (printer's)	meter (**m**)	4.217 518 E–03
pound (mass)(avoirdupois)(lb or lbm)	kilogram (**kg**)	4.535 923 7*E–01
poundal	newton (**N**)	1.382 550 E–01
pound force (lbf)	newton (**N**)	4.448 222 E+00
quad	joule (**J**)	1.055 E+18
quart (US dry)	meter^3 (m^3)	1.101 22 E–03
quart (US liquid)	meter^3 (m^3)	9.463 53 E–04
rad (absorbed dose)	gray (**Gy**)	1*E–02
rem (dose equivalent)	sievert (**Sv**)	1*E–02
roentgen	coulomb per kilogram (**C/kg**)	2.58 E–04
slug	kilogram (**kg**)	1.459 390 E+01
tablespoon	milliliter (**mL**)	1.479 E+01
teaspoon	milliliter (**mL**)	4.929 E+00

To convert from	to	Multiply by
therm (US)	joule (**J**)	1.054 804*E+08
ton (explosive energy of TNT)	joule (**J**)	4.184 E+09
ton of refrigeration (12 000 Btu/h)	watt (**W**)	3.517 E+03
ton (short, 2000 lb)	kilogram (**kg**)	9.071 847 E+02
yard	meter (**m**)	9.144*E–01
year (365 days)	second (**s**)	3.153 6*E+07
year (sidereal)	second (**s**)	3.155 815 E+07

Bibliography

Air Force, Department of the, *Producibility Measurement for DoD Contracts*, Office of the Assistant for Reliability, Maintainability, Manufacturing, and Quality, SAF/AQXE, Washington, DC, 1992. **Referred to on page(s) 111.**

———, *Unmanned Space Vehicle Cost Model, Seventh Edition*, SD TR-88-97, Air Force Systems Command/Space Division (P. Nguyen, et al.), August 1994. **Referred to on page(s) 81.**

Agrawal, Brij N., *Design of Geosynchronous Spacecraft*, Prentice-Hall, Inc., Englewood Cliffs, NJ 07632, 1986. **Referred to on page(s) 1.**

Armstrong, J.E., and Andrew P. Sage, *An Introduction to Systems Engineering*, John Wiley & Sons, New York, 1995. **Referred to on page(s) 1.**

Army, Department of the, *Logistics Support Analysis Techniques Guide*, Pamphlet No. 700-4, Army Materiel Command, 1985. **Referred to on page(s) 102.**

Asher, Harold, *Cost-Quantity Relationships in the Airframe Industry*, R-291, The Rand Corporation, 1956. **Referred to on page(s) 82.**

Barclay, Scott, et al., *Handbook for Decision Analysis*, Decisions and Designs, Inc., McLean, VA, September 1977. **Referred to on page(s) 43.**

Blanchard, B.S., and W.J. Fabrycky, *Systems Engineering and Analysis* (2nd Edition), Prentice-Hall, Inc. Englewood Cliffs, NJ, 1990. **Referred to on page(s) 1.**

———, *System Engineering Management*, John Wiley & Sons, Inc., New York, 1991. **Referred to on page(s) 1.**

Boden, Daryl, and Wiley J. Larson (eds), *Cost-Effective Space Mission Operations*, publication scheduled for August 1995. **Referred to on page(s) 1.**

Boehm, Barry W., "A Spiral Model of Software Development and Enhancement", *Computer*, pp 61–72, May 1988. **Referred to on page(s) 13.**

Carter, Donald E., and Barbara Stilwell Baker, *Concurrent Engineering: The Product Development Environment for the 1990s*, Addison-Wesley Publishing Co., Inc., Reading, MA, 1992. **Referred to on page(s) 25.**

Chamberlain, Robert G., George Fox and William H. Duquette, *A Design Optimization Process for Space Station Freedom*, JPL Publication 90-23, June 15, 1990. **Referred to on page(s) 18.**

Chestnut, Harold, *Systems Engineering Tools*, John Wiley & Sons, Inc., New York, 1965. **Referred to on page(s) 1.**

———, *Systems Engineering Methods*, John Wiley & Sons, Inc., New York, 1965. **Referred to on page(s) 1.**

Churchman, C. West, Russell L. Ackoff and E. Leonard Arnoff, *Introduction to Operations Research*, John Wiley & Sons, Inc., New York, 1957.

Clark, Philip, et al., *How CALS Can Improve the DoD Weapon System Acquisition Process*, PL207R1, Logistics Management Institute, November 1992. **Referred to on page(s) 103.**

Committee on Space Research, *COSPAR Information Bulletin No. 38* (pp. 3-10), June 1967. **Referred to on page(s) 115.**

Defense, Department of, *Transition from Development to Production*, DoD 4245.7-M, 1985. **Referred to on page(s) 40, 111.**

———, *Metric System, Application in New Design*, DOD-STD-1476A, 19 November 1986.

———, *Logistics Support Analysis* and *Logistics Support Analysis Record*, MIL-STD-1388-1A/2B, revised 21 January 1993, Department of Defense. **Referred to on page(s) 86, 100, 102.**

———, *Failure Modes, Effects, and Criticality Analysis*, MIL-STD-1629A, Department of Defense. **Referred to on page(s) 41.**

Defense Systems Management College, *Scheduling Guide for Program Managers*, GPO #008-020-01196-1, January 1990.

———, *Test and Evaluation Management Guide*, GPO #008-020-01303-0, 1993.

———, *Integrated Logistics Support (ILS) Guide*, DTIC/NTIS #ADA 171-087, 1993.

———, *Systems Engineering Management Guide*, DTIC/NTIS #ADA223-168, 1994. **Referred to on page(s) 1, 40.**

———, *Risk Management: Concepts and Guidance*, GPO #008-020-01164-9, 1994.

DeJulio, E., *SIMSYLS User's Guide*, Boeing Aerospace Operations, February 1990. **Referred to on page(s) 87.**

de Neufville, R., and J.H. Stafford, *Systems Analysis for Engineers and Managers*, McGraw-Hill, New York, 1971. **Referred to on page(s) 1, 43.**

Electronic Industries Association (EIA), *Systems Engineering*, EIA/IS-632, 2001 Pennsylvania Ave. NW, Washington, DC, 20006, December 1994. **Referred to on page(s) 1, 4.**

Executive Order 11514, *Protection and Enhancement of Environmental Quality*, March 5, 1970, as amended by Executive Order 11991, May 24, 1977. **Referred to on page(s) 112.**

Executive Order 12114, *Environmental Effects Abroad of Major Federal Actions*, January 4, 1979. **Referred to on page(s) 112.**

Fisher, Gene H., *Cost Considerations in Systems Analysis*, American Elsevier Publishing Co. Inc., New York, 1971. (Also published as R-490-ASD, The Rand Corporation, December 1970.) **Referred to on page(s) 1, 67.**

Forsberg, Kevin, and Harold Mooz, "The Relationship of System Engineering to the Project Cycle", Center for Systems Management, 5333 Betsy Ross Dr., Santa Clara, CA 95054; also available in *A Commitment to Success*, Proceedings of the first annual meeting of the National Council for Systems Engineering and the 12th annual meeting of the American Society for Engineering Management, Chattanooga, TN, 20–23 October 1991. **Referred to on page(s) 20.**

Fortescue, Peter W., and John P.W. Stark (eds), *Spacecraft Systems Engineering*, John Wiley and Sons Ltd., Chichester, England, 1991. **Referred to on page(s) 1.**

Gavin, Joseph G., Jr. (interview with), *Technology Review*, Vol. 97, No. 5, July 1994. **Referred to on page(s) 92.**

General Accounting Office, *DOD's CALS Initiative*, GAO/AIMD-94-197R, September 30, 1994. **Referred to on page(s) 103.**

Goddard Space Flight Center, *Goddard Multi-variable Instrument Cost Model (MICM)*, Research Note #90-1, Resource Analysis Office (Bernard Dixon and Paul Villone, authors), NASA/GSFC, May 1990. **Referred to on page(s) 81.**

Green, A.E., and A.J. Bourne, *Reliability Technology*, Wiley Interscience, 1972.

Griffin, Michael D., and James R. French, *Space Vehicle Design*, AIAA 90-8, American Institute of Aeronautics and Astronautics, c/o TASCO, P.O. Box 753, Waldorf, MD 20604-9961, 1990. **Referred to on page(s) 1.**

Hickman, J.W., et al., *PRA Procedures Guide*, The American Nuclear Society and The Institute of Electrical and Electronics Engineers, NUREG/CR-2300, Washington, DC, January 1983. **Referred to on page(s) 42.**

Hillier, F.S. and G.J. Lieberman, *Introduction to Operations Research*, 2nd Edition, Holden-Day, Inc., 1978.

Hodge, John, "The Importance of Cost Considerations in the Systems Engineering Process", in the *NAL Monograph Series: Systems Engineering Papers*, NASA Alumni League, 922 Pennsylvania Ave. S.E., Washington, DC 20003, 1990. **Referred to on page(s) 14.**

Hood, Maj. William C., *A Handbook of Supply Inventory Models*, Air Force Institute of Technology, School of Systems and Logistics, September 1987.

Hughes, Wayne P., Jr. (ed.), *Military Modeling*, Military Operations Research Society, Inc., 1984.

IEEE, *American National Standard for Metric Practice*, ANSI/IEEE Std 268-1992 (supersedes IEEE Std

268-1979 and ANSI Z210.1-1976), American National Standards Institute (ANSI), 1430 Broadway, New York, NY 10018. **Referred to on page(s) 139.**

———, *IEEE Trial-Use Standard for Application and Management of the Systems Engineering Process*, IEEE Std 1220-1994., 345 E 47th St., New York, NY 10017, February 28, 1995. **Referred to on page(s) 1.**

Jet Propulsion Laboratory, *The JPL System Development Management Guide*, Version 1, JPL D-5000, NASA/JPL, November 15, 1989.

Johnson Space Center, *NASA Systems Engineering Process for Programs and Projects*, Version 1.0, JSC-49040, NASA/JSC, October 1994. **Referred to on page(s) xi, 22.**

Kececioglu, Dimitri, and Pantelis Vassiliou, *1992-1994 Reliability, Avaliability, and Maintainability Software Handbook*, Reliability Engineering Program, University of Arizona, 1992. **Referred to on page(s) 95.**

Keeney, R.L., and H. Raiffa, *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*, John Wiley and Sons, Inc., New York, 1976. **Referred to on page(s) 76.**

Kline, Robert, et al., *The M-SPARE Model*, Logistics Management Institute, NS901R1, March 1990. **Referred to on page(s) 87.**

Langley Research Center, *Systems Engineering Handbook for In-House Space Flight Projects*, LHB 7122.1, Systems Engineering Division, NASA/LaRC, August 1994.

Lano, R., *The N^2 Chart*, TRW Inc., 1977. **Referred to on page(s) 68, 127.**

Larson, Wiley J., and James R. Wertz (eds), *Space Mission Analysis and Design* (2nd edition), published jointly by Microcosm, Inc., 2601 Airport Dr., Suite 230, Torrance, CA 90505 and Kluwer Academic Publishers, P.O. Box 17, 3300 AA Dordrecht, The Netherlands, 1992. **Referred to on page(s) 1.**

———, *Reducing Space Mission Cost*, publication scheduled for November 1995. **Referred to on page(s) 1.**

Leising, Charles J., "System Architecture", in *System Engineering at JPL*, course notes (contact: Judy Cobb, Jet Propulsion Laboratory), 1991. **Referred to on page(s) 11.**

Lexicon — Glossary of Technical Terms and Abbreviations Including Acronyms and Symbols, Draft/Version 1.0, produced for the NASA Program/Project Management Initiative, Office of Human Development (Code ND), NASA Headquarters, by DEF Enterprises, P.O. Box 590481, Houston, TX 77259, March 1990. **Referred to on page(s) 117.**

Marshall Space Flight Center, *Program Risk Analysis Handbook*, NASA TM-100311, Program Planning Office (R.G. Batson, author), NASA/MSFC, August 1987.

———, *Systems Engineering Handbook, Volume 1 — Overview and Processes; Volume 2 — Tools, Techniques, and Lessons Learned*, MSFC-HDBK-1912, Systems Analysis and Integration Laboratory, Systems Analysis Division, NASA/MSFC, February 1991. **Referred to on page(s) 75, 89.**

———, *Verification Handbook, Volume 1 — Verification Process; Volume 2 — Verification Documentation Examples*, MSFC-HDBK-2221, Systems Analysis and Integration Laboratory, Systems Integration Division, Systems Verification Branch, February 1994.

———, *General Environmental Test Guidelines (GETG) for Protoflight Instruments and Experiments*, MSFC-HDBK-670, Systems Analysis and Integration Laboratory, Systems Integration Division, Systems Verification Branch, February 1994. **Referred to on page(s) 109.**

McCormick, Norman, *Reliability and Risk Analysis*, Academic Press, Orlando, FL, 1981.

Miles, Ralph F., Jr. (ed.), *Systems Concepts — Lectures on Contemporary Approaches to Systems*, John Wiley & Sons, New York, 1973. **Referred to on page(s) 1.**

Moore, N., D. Ebbeler and M. Creager, "A Methodology for Probabilistic Prediction of Structural Failures of Launch Vehicle Propulsion Systems", American Institute of Aeronautics and Astronautics, 1990. **Referred to on page(s) 89.**

Morgan, M. Granger, and Max Henrion, *Uncertainty: A Guide to Dealing with Uncertainty in Quantitative Risk and Policy Analysis*, Cambridge University Press, Cambridge, England, 1990.

Morris, Owen, "Systems Engineering and Integration and Management for NASA Manned Space Flight Programs", in *NAL Monograph Series: Systems Engineering Papers*, NASA Alumni League, 922 Pennsylvania Ave. S.E., Washington, DC 20003, 1990. **Referred to on page(s) 9, 11.**

NASA Headquarters, *Initial OSSA Metrication Transition Plan*, Office of Space Science and Applications (Code S), NASA/HQ, September 1990.

———, NHB 1700.1B, *NASA Safety Policy and Requirements Document*, Office of Safety and Mission Assurance (Code Q), NASA/HQ, June 1993. **Referred to on page(s) 55.**

———, NHB 5103.6B, *Source Evaluation Board Handbook*, Office of Procurement (Code H), NASA/HQ, October 1988. **Referred to on page(s) 75.**

———, NHB 5300.4 (1A-1), *Reliability Program Requirements for Aeronautical and Space System Contractors*, Office of Safety and Mission Assurance (Code Q), NASA/HQ, January 1987. **Referred to on page(s) 91, 92.**

———, NHB 5300.4 (1B), *Quality Program Provisions for Aeronautical and Space System Contractors*, Office of Safety and Mission Assurance (Code Q), NASA/HQ, April 1969. **Referred to on page(s) 95.**

———, NHB 5300.4 (1E), *Maintainability Program Requirements for Space Systems*, Office of Safety and Mission Assurance (Code Q), NASA/HQ, March 1987. **Referred to on page(s) 96, 100.**

———, NHB 7120.5 (with NMI 7120.4), *Management of Major System Programs and Projects*, Office of the Administrator (Code A), NASA/HQ, November 1993 (revision forthcoming). **Referred to on page(s) ix, xi, 13, 14, 17, 99, 102, 139.**

———, NHB 8800.11, *Implementing the Requirements of the National Environmental Policy Act*, Office of Management Systems and Facilities (Code J), NASA/HQ, June 1988. **Referred to on page(s) 113.**

———, NHB 8020.12B, *Planetary Protection Provisions for Robotic Extraterrestrial Missions*, Office of Space Science (Code S), NASA/HQ, forthcoming. **Referred to on page(s) 115.**

———, NHB 9501.2B, *Procedures for Contractor Reporting of Correlated Cost and Performance Data*, Financial Management Division (Code BF), NASA/HQ, February 1985. **Referred to on page(s) 59.**

———, NMI 5350.1A, *Maintainability and Maintenance Planning Policy*, Office of Safety and Mission Assurance (Code Q), NASA/HQ, September 26, 1991. **Referred to on page(s) 99, 100.**

———, NMI 7120.4 (with NHB 7120.5), *Management of Major System Programs and Projects*, Office of the Administrator (Code A), NASA/HQ, November 1993 (revision forthcoming). **Referred to on page(s) ix, xi, 3, 13.**

———, NMI 8010.1A, *Classification of NASA Payloads*, Safety Division (Code QS), NASA/HQ, 1990. **Referred to on page(s) 38, 123.**

———, NMI 8010.2A, *Use of the Metric System of Measurement in NASA Programs*, Office of Safety and Mission Quality (Code Q), NASA/HQ, June 11, 1991. **Referred to on page(s) 139.**

———, NMI 8020.7D, *Biological Contamination Control for Outbound and Inbound Planetary Spacecraft*, Office of Space Science (Code S), NASA/HQ, December 3, 1993. **Referred to on page(s) 115.**

———, NMI 8070.4A, *Risk Management Policy*, Safety Division (Code QS), NASA/HQ, undated. **Referred to on page(s) 37, 44.**

National Environmental Policy Act (NEPA) of 1969, as amended (40 CFR 1500-1508). **Referred to on page(s) 112.**

Navy, Department of the, *Producibility Measurement Guidelines: Methodologies for Product Integrity*, NAVSO P-3679, Office of the Assistant Secretary for Research, Development, and Acquisition, Washington DC, August 1993. **Referred to on page(s) 112.**

Nuclear Regulatory Commission, U.S., (by N.H. Roberts, et al.), *Fault Tree Handbook*, NUREG-0492, Office of Nuclear Regulatory Research, January 1980. **Referred to on page(s) 94.**

Office of Management and Budget, *Guidelines and Discount Rates for Benefit-Cost Analysis of Federal Programs, Circular A-94, OMB, October 1992.* **Referred to on page(s) 79.**

Pace, Scott, *U.S. Access to Space: Launch Vehicle Choices for 1990-2010*, The Rand Corporation, R-3820-AF, March 1990.

Presidential Directive/National Security Council Memorandum-25 (PD/NSC-25), *Scientific of Technological Experiments with Possible Large-Scale Adverse Environmental Effects and Launch of Nuclear Systems into Space*, December 14, 1977. **Referred to on page(s) 114.**

Procedures for Implementing the National Environmental Policy Act (14 CFR 1216.3), **Referred to on page(s) 112.**

Reilly, Norman B., *Successful Systems Engineering for Engineers and Managers*, Van Nostrand Reinhold, New York, 1993. **Referred to on page(s) 1.**

Ruskin, Arnold M., "Project Management and System Engineering: A Marriage of Convenience", Claremont Consulting Group, La Cañada, California; presented at a meeting of the Southern California Chapter of the Project Management Institute, January 9, 1991. **Referred to on page(s) 11.**

Saaty, Thomas L., *The Analytic Hierarchy Process*, McGraw-Hill, New York, 1980. **Referred to on page(s) 75.**

Sage, Andrew P., *Systems Engineering*, John Wiley & Sons, New York, 1992. **Referred to on page(s) 1.**

Shishko, R., *Catalog of JPL Systems Engineering Tools and Models (1990)*, JPL D-8060, Jet Propulsion Laboratory, 1990.

———, *MESSOC Version 2.2 User Manual*, JPL D-5749/Rev. B, Jet Propulsion Laboratory, October 1990. **Referred to on page(s) 81.**

Sivarama Prased, A.V., and N. Somasehara, "The AHP for Choice of Technologies: An Application", *Technology Forecasting and Social Change*, Vol. 38, pp. 151–158, September 1990.

Smith, Jeffrey H., Richard R. Levin and Elisabeth J. Carpenter, *An Application of Multiattribute Decision Analysis to the Space Station Freedom Program — Case Study: Automation and Robotics Technology Evaluation,* JPL Publication 90-12, Jet Propulsion Laboratory, May 1, 1990. **Referred to on page(s) 76.**

SP-7012 (NASA Special Publication), *The International System of Units; Physical Constants and Conversion Factors*, E.A. Mechtly, published by the NASA Scientific and Technical Information Office, 1973; U.S. Government Printing Office, (stock number 3300-00482).

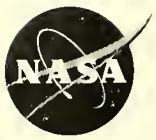
Steuer, R., *Multiple Criteria Optimization*, John Wiley and Sons, Inc., New York, 1986.

Stewart, Rodney D., *Cost Estimating*, 2nd ed., John Wiley and Sons, Inc., New York, 1991.

Taguchi, Genichi, *Quality Engineering in Production Systems*, New York: McGraw-Hill, 1989. **Referred to on page(s) 111.**

Wagner, G. M., *Principles of Operations Research — With Applications to Managerial Decisions*, Prentice Hall, 1969.

Index



National Aeronautics and
Space Administration

To request additional copies of SP-6105, call (U.S.) 202-554-8677.