SANDIA REPORT

SAND97–0412 • UC–706
Unlimited Release
Printed February 1997

A Road Map for Implementing Systems Engineering

Frank F. Dean, Bo Bentz, A. Terry Bahill

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550
for the United States Department of Energy
under Contract DE-AC04-94AL85000

Approved for public release; distribution is unlimited.

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.
NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Prices available from (615) 576-8401, FTS 626-8401

Available to the public from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Rd
Springfield, VA 22161

NTIS price codes
Printed copy: A03
Microfiche copy: A01

SAND 97-0412
Unlimited Release
Printed February 1997

Distribution
Category UC–706

A Road Map for Implementing Systems Engineering

Frank F. Dean
New Mexico Weapons Systems Engineering Center
Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185-0435

Bo Bentz
A. Terry Bahill
Systems and Industrial Engineering
University of Arizona
Tucson, AZ 85721-0020

Abstract

Studies by academia, industry, and government indicate that applying a sound systems engineering process to development programs is an important tool for preventing cost and schedule overruns and performance deficiencies. There is an enormous body of systems engineering knowledge. Where does one start? How can we apply the principles of systems engineering in the Sandia environment? This road map is intended to be an aid to answering these questions.

Acknowledgment

Many members of 2100 contributed to this document. The authors wish to thank especially Ed Barkocy, Glenn Bell, Mike Skaggs, Ernie Aryers, Jeanne Lewis, Joe Lopez, and Keith Ortiz. The authors also wish to thank the many people who contributed to this document: Deadra R. Apodaca, Patty Guyer, Ron Andree, Merrillee Dolan, and Sharon O'Connor of Tech Reps, Inc. and Veronica Urioste of Jobs Plus.

CONTENTS

Acknowledgment ii

Introduction 1
The System Life Cycle 1
A Systems Engineering Process 3
Common Sense and Systems Engineering 4
The System Design and Development Process 4
Confusing Systems Engineering Terminology 4

What's in It for Me? 4
Systems Engineering Is Cost-Effective 4
DOE Documents 8
QC-1 and *QC-2* 8
EP401099 and *EP401100* 8
DOE Order 430.1 8
DOE/AL Nuclear Weapons Development and Production Manual 8
Commercial Standards: ISO 9000 9
Systems Engineering, Total Quality Management (TQM), Integrated Product Development (IPD), and Concurrent Engineering 9

What's It Going to Cost Me? 9

A System or a System of Systems? 10

Document Objective 10

Stage 1. Developing Requirements 11
Step 1. Identify Customers and Stakeholders 11
Step 2. Understand the Customer's Needs 13
Step 3. Define and State the Problem 13
Step 4. Write System Requirements 14
Step 5. Consult with the Customer 15
Step 6. Define Figures of Merit 15
Definitions for Steps 7 and 8 19
Validating a System 19
Validating Requirements 19
Verifying a System 19
Verifying Requirements 19
Verification and Validation 19
Step 7. Prescribe System Verification Process 20

Step 8. Validate System Requirements 20
Step 9. Define Technical Performance Measures.....20
Step 10. Risk Mitigation.....20
Step 11. System Requirements Review (SRR) 21
Mission Concept Review 21
System Requirements Review (SRR) 21
System Definition Review 22
Preliminary Design Review (PDR).....22
Critical Design Review (CDR) 22
Production Readiness Review (PRR) 22
Production Review.....22
System Test Review.....22
Post-Operational Review 23

Stage 2. Creating and Evaluating Concepts 23
Step 12. Create and Evaluate Alternative System Concepts 23
Step 13. Sensitivity Analysis.....24
Step 14. Functional Decomposition 24
Step 15. Begin Subsystem Requirements and Design Projects.....25
Step 16. System Design (Define System Architecture).....25
Step 17. Analysis Models, Prototypes, and Simulations 26
Step 18. Conceptual Design Review 27

Stage 3. Engineering Design and Development.....27
Construct a Product-Oriented WBS 28
Stable and Controlled Requirements.....28
Detailed Engineering Design.....28
Manufacturing Engineering.....28
Designing and Managing Interfaces 28
Configuration Management.....29
Engineering Analysis and Modeling 29
Analysis and Testing Are Synergistic.....29
The Roles of Analysis and Testing Change During the Development Cycle.....30
System Integration.....31
Prototype Development and System Validation.....31
Development Test and Evaluation (DT&E).....31

Stage 4. System Verification.....31
Verification Plan 32
Analysis and Modeling in Support of Verification 32
Operational Testing and Evaluation 32
Traceability Analyses 32

Stage 5. System Production....33
Manufacturing, Production, and Assembly....33
Production Acceptance Test and Evaluation....33

Stage 6. Operation, Maintenance, and Modification....33

Stage 7. Retirement, Disposal, Recycle, and Replacement....33

Elements Common to All of the Systems Engineering Stages....34
Total Quality Management....34
Project Management....34
Documentation....34

Systems Engineering Standards....34

References and Bibliography....35

LIST OF FIGURES

Figure 1. The System Life Cycle 2
Figure 2. The Systems Engineering Process 3
Figure 3. The Systems Engineering Process (continued) 5
Figure 4. The Management Process 5
Figure 5. The Technical Process 6
Figure 6. The System Definition and Design Process 6
Figure 7. Most Cost Associated with Designing and Manufacturing a Product Are Locked In Early in the Design Activity 7
Figure 8. The Requirements Discovery Process 12
Figure 9. Scoring Function for the Amount of Computer Memory 16
Figure 10. Performance Figure of Merit 17
Figure 11. Cost Figure of Merit 17
Figure 12. Performance Figure of Merit 18
Figure 13. The Trade-off Between Range and Throw Weight 18
Figure 14. Timing of the Major Reviews 23
Figure 15. Systems Engineering Is a Fractal Process 26

"Our vision is for Lockheed Martin to be recognized as the world's premier systems engineering and technology enterprise."

Dan Tellep (CEO) and Norm Augustine (President), May 5, 1995

Introduction

The purpose of systems engineering is to increase a system's probability of success and reduce the risk of failure. Most projects require either formal or informal systems engineering. Systems engineering is not the sole responsibility of systems engineers: all engineers must practice systems engineering. This document is intended to be used as a road map for integrating modern systems engineering disciplines and modern product realization processes into projects. It is intended that the process outlined in this document be used as suggestions and guidelines rather than as a rigid procedure. The term "product realization" as used by Sandia is essentially identical to the term "systems engineering" as used by Lockheed Martin and the DoD. Since the principal end user and customer for our weapon system projects is the DoD, we chose to use their language. The systems engineering team collected and documented many of the lessons learned and best practices from our past weapons development activities as well as many of those suggested by Lockheed Martin and the DoD. We believe that projects that follow this road map will be on a course that meets the DOE requirements specified in QC-1 and EP401099. We hope that readers will have additional contributions. Since this document is in a state of continuous improvement, your inputs and contributions are important and greatly appreciated.

The steps described in this document are the systems engineering steps for a typical systems development project. These steps are equally valid for development of new products, modifications of existing products, and replacement of components or subsystems within existing products. Additional systems development activities such as project and program management are not included. Systems engineering and project management overlap considerably. It is the responsibility of the systems engineer and the project manager to coordinate these activities and eliminate duplications.

The System Life Cycle

It is very important that engineers pay attention to the whole system life cycle. The system life cycle has seven general phases: (1) discovering system requirements, (2) creating and evaluating concepts, (3) engineering design and development, (4) system verification, (5) system production, (6) operation, maintenance, and modification, and (7) retirement, disposal, recycle, and replacement. The relationships of these phases is illustrated in Figure 1. Figure 1 is a modification of what is commonly found in the system engineering literature. This figure incorporates the observations of W.C. Nickell, 12300 that each stage of the

systems engineering life cycle must be supported by relevant, viable research activities. In fact, for one-of-a-kind systems like a dam or a power plant, the system verification stage occurs after the system production stage. However, the system life cycle can be different for different industries, products, and customers (Chapman, Bahill, and Wymore 1992; Wymore 1993; Kerzner 1995; and Shishko 1995).

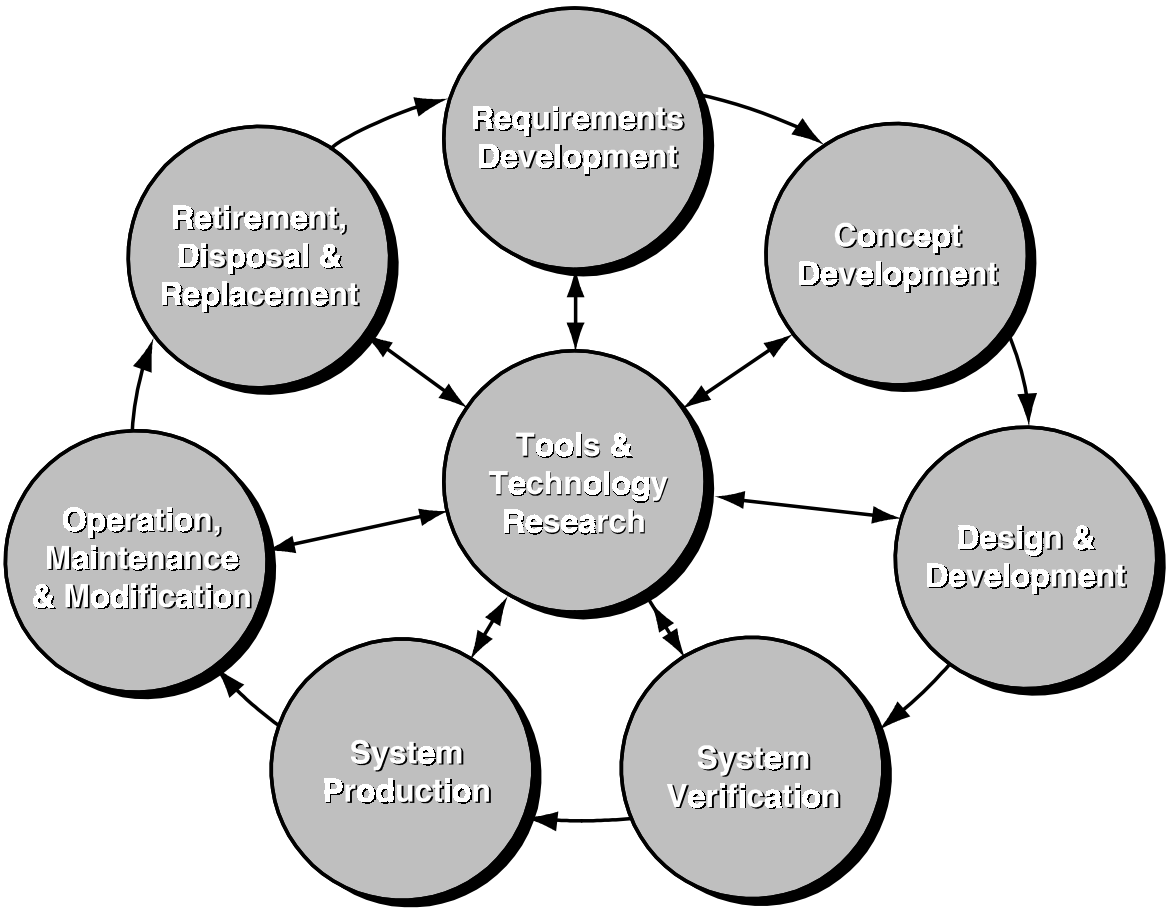


Figure 1. The System Life Cycle

The stages used in this document were chosen to be consistent with the system life cycle model previously used for Sandia weapons development projects. We have deliberately used the term "stage" to break away from the traditional Department of Energy (DOE) "phase" terminology. If we are going to do more with less in the future, we will need to break from tradition and find better processes and practices. The traditional DOE systems engineering process is a serial process. If we want to shorten the product development cycle, we will need to conduct many of the systems engineering steps in parallel. One of the objectives of this document is to identify the essential elements of systems engineering as it is applied to Sandia's mission. Once this is done, the systems engineer can identify which project activities can be done in parallel. This can be called concurrent engineering.

A Systems Engineering Process

Systems engineering should be thought of as a problem-solving discipline with its own set of tools and processes. It is a relatively young discipline and is still evolving. The tools, processes, and even the terminology are still immature in comparison with the traditional engineering disciplines.

There are two common ways to view system engineering activities. One way is to start with the customer's point of view and look at successive levels of detail. The Lockheed Martin systems engineering process described below takes this approach. The second view is to look at the various activities that must be accomplished as the systems engineering project moves along in time. This is the theme of this report. A high-level summary of this view is presented below.

A good systems engineering process contains both technical and management functions. Lockheed Martin has found that it is important to the success of its projects to clarify the roles and responsibilities associated with these two fundamental categories of systems engineering activities. Figures 2, 3, 4, and 5 are illustrations of Lockheed Martin's systems engineering process (SEP).

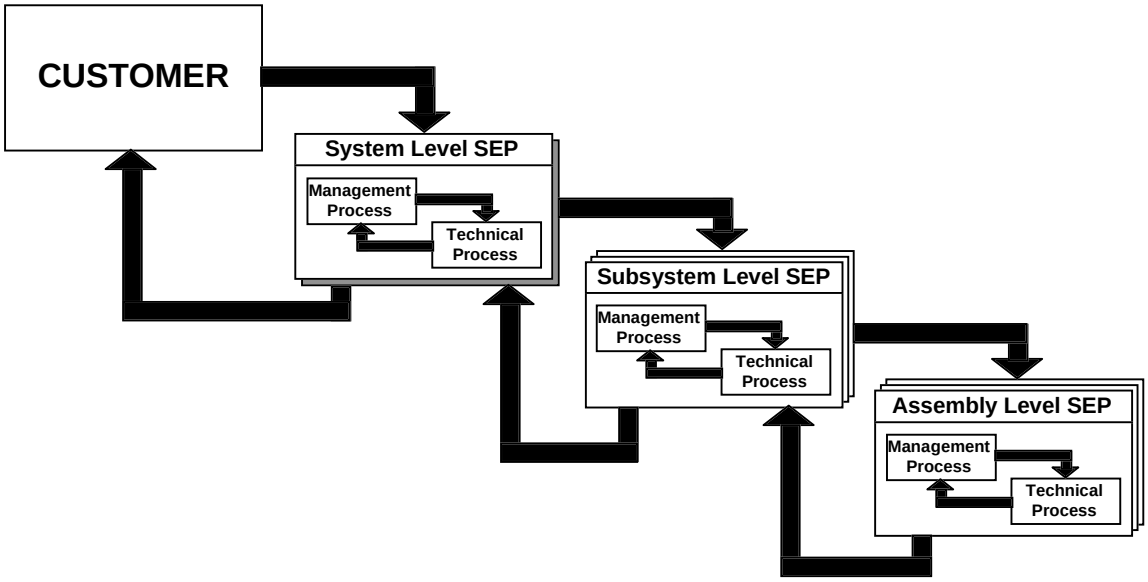


Figure 2. The Systems Engineering Process

Figure 2 illustrates the relationship of the "ultimate" customer with the different levels of the systems engineering process. The systems organization is the customer for each of the subsystem organizations. In addition, each subsystem organization may be the customer for the assembly or component organizations. For complex systems there may be many more levels than illustrated in Figure 2. The most important message from Figure 2 is that the process used for the system is the same as the process used for the subsystems and the same as the process used for the assemblies.

Figure 3 illustrates the inputs and outputs to a typical systems engineer process level. The process is recursive or fractal in nature. The input and output types are the same at the system level, the subsystem level, or the component level. (The details are quite different, of course.)

Figure 4 illustrates the activities in the management process and Figure 5 illustrates the activities in the technical process. This "road map" is written from the systems level perspective. In the text there are notations where one needs to repeat this process for each of the subsystems. This road map focuses primarily on the technical process activities.

Common Sense and Systems Engineering

Systems Engineering, like Project Management, is mostly common sense. There are those who seek to make it complicated and esoteric, but it's mostly common sense and good engineering practices.

The System Design and Development Process

The system engineering literature frequently refers to the first stages of the system life cycle in Figure 1 as the system design and development process. A flow diagram for this process is presented as Figure 6. This diagram ties together the material discussed in Stages 1, 2, and 3.

Confusing Systems Engineering Terminology

Like many disciplines, systems engineering has evolved its own terminology. Systems engineering is a relatively young discipline, and common systems engineering terms have a wide range of meanings and interpretations. Frequently, meanings vary from industry to industry, from company to company within a specific industry, and even from office to office of a single company. The industry leaders such as Lockheed Martin and Boeing are investing considerable effort to standardize processes and terminology within their businesses. We have collected an extensive glossary of the common systems engineering terms. A copy is available upon request.

What's in It for Me?

Systems Engineering Is Cost-Effective

Numerous studies, documented in the NCOSE publications, show that using sound, proven systems engineering principles is an important contributor to minimizing cost and schedule overruns and performance deficiencies. A Boeing Systems Engineering Experiment (Frantz 1995) demonstrated that projects implementing a formal systems engineering process, similar to the one described in this road map, an result in a significant reduction (factors of 2 to 3) in the total time for system design process and the total cost of the system design process. In addition, the Boeing projects that used the formal systems engineering process produced a

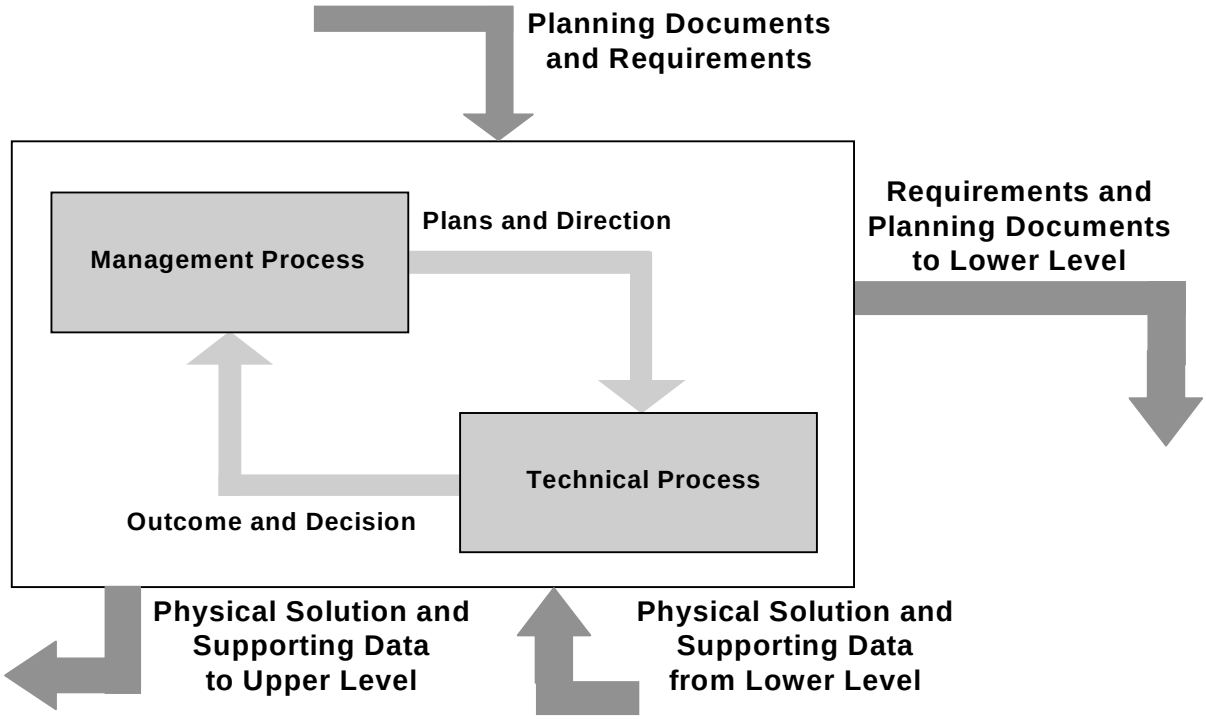


Figure 3. The Systems Engineering Process (continued)

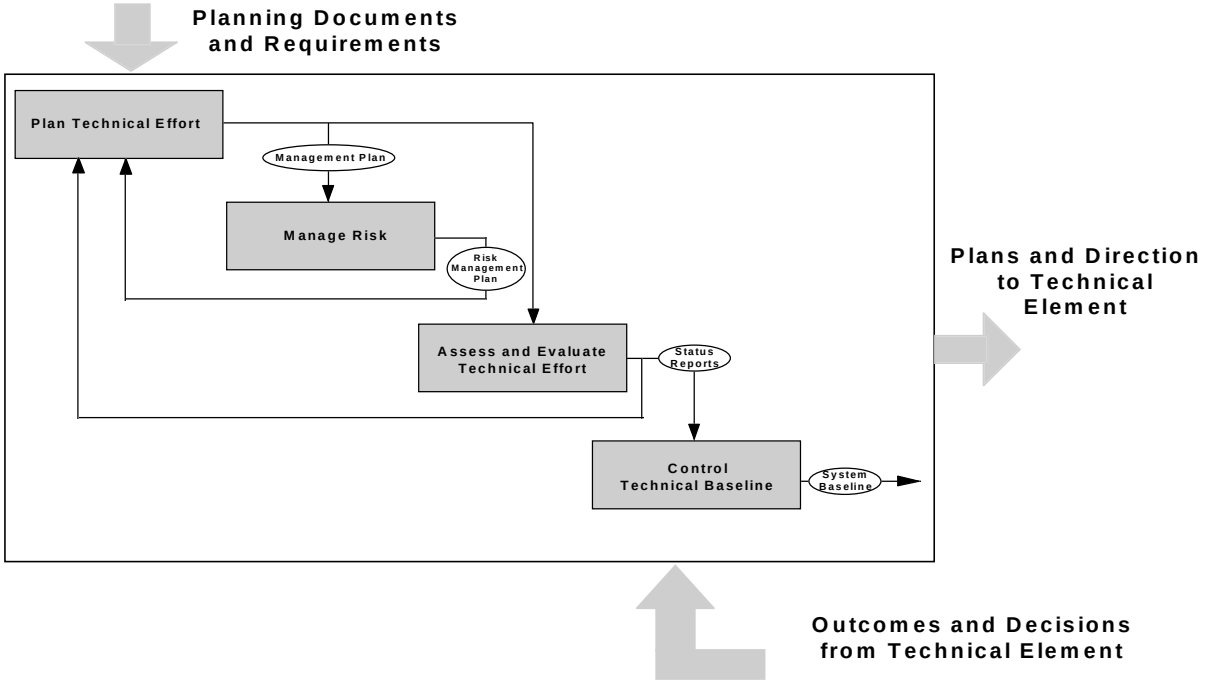


Figure 4. The Management Process

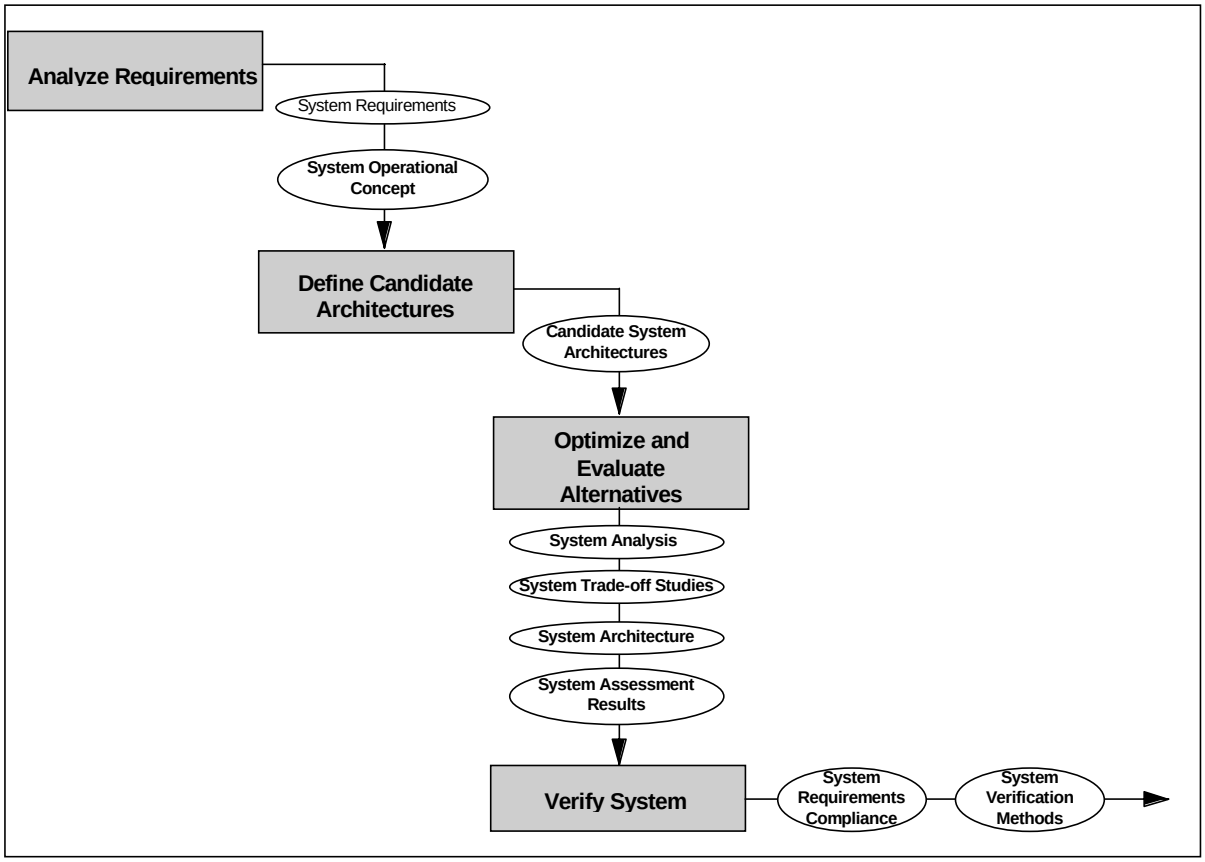


Figure 5. The Technical Process

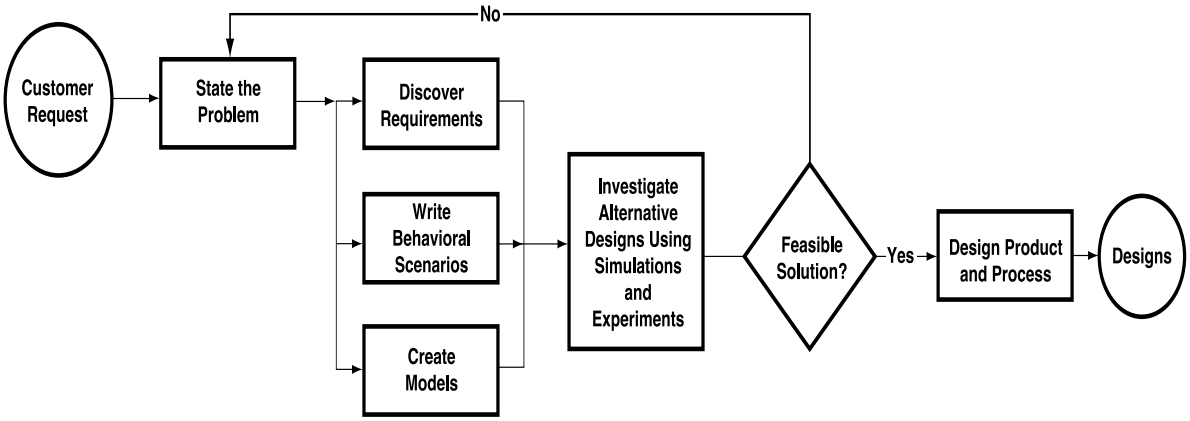


Figure 6. The System Definition and Design Process

product superior to that produced by the other projects that participated in the Boeing Systems Engineering Experiment.

One reason for the success of formal systems engineering processes is that they are front-end-loaded processes. The systems engineering processes tend to uncover system incompatibilities, design flaws, and other errors early in the process, before most costs are locked in. This is the time when it's inexpensive to fix the problems. Figure 7 is a good illustration of when costs are locked in versus when they actually accrue. Most costs are locked in long before they actually accrue.

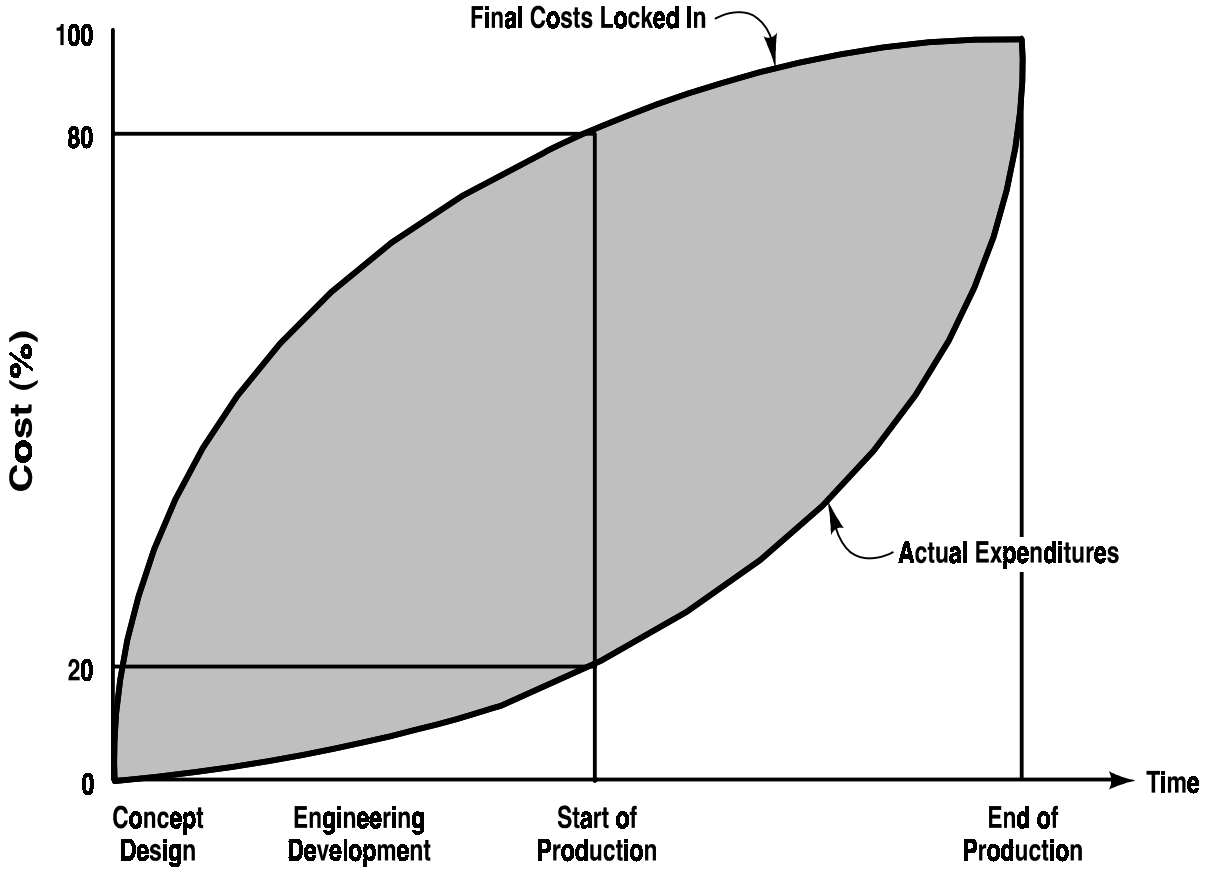


Figure 7. Most Costs Associated with Designing and Manufacturing a Product Are Locked In Early in the Design Activity

DOE Documents

Elements of Systems Engineering and Project Management are described in several DOE documents. The information in the DOE documents is valuable, and we hope that we have captured the essence of each of these DOE documents in this road map document. Below we have listed some of the documents we have reviewed.

QC-1 and *QC-2*

Revision 8 of QC-1 merges QC-1 and QC-2 into a single document. QC-1 is a DOE/AL document that describes the minimum DOE quality requirements over the entire life cycle of a product. This road map is consistent with the requirements of QC-1. Projects that follow the steps and the process of this road map document will meet QC-1 requirements. Furthermore, QC-1 Rev. 8 is consistent with current DoD and industrial standards for implementing systems engineering processes for product development.

EP401099 and *EP401100*

The 1996 version of the DOE interagency Engineering Procedure EP401099/B - Product Realization Process replaces both the old EP401099 and EP401100. This road map document defines a system engineering process that meets the requirements of EP401099/B. EP401099/B provides a high-level description of a systems definition and design process similar to Figure 6. The material in the Stage 1, 2, and 3 sections of this road map provide the details for implementing the EP401099/B process.

EP401099/B describes a systems engineering process without calling it "systems engineering." One should interpret the EP401099 terms "Product Realization Team" (PRT) and "PRT leader" as "systems engineering team" and "lead systems engineer" or "systems engineering manager."

DOE Order 430.1

DOE Order 430.1, Life-Cycle Asset Management replaces at least 13 DOE Orders. It is expected that Sandia will come under this order in the near future. Eleven guidebooks describe contractor implementation of this order. The process described in this document is almost identical to that described in the systems engineering guidebook, "Project Execution and Engineering Management Planning." This is not accidental. Both are derived from the same source standard, Draft MIL-STD-499B.

DOE/AL Nuclear Weapons Development and Production Manual

The US Department of Energy/Albuquerque Field Office produced and maintains the Nuclear Weapons Development and Production Manual. This road map presents an implementation strategy for Section 3 of the Nuclear Weapons Development and Production Manual.

Commercial Standards: ISO 9000

Many industry leaders are using documented, formal systems engineering processes and Systems Engineering Management Plans (SEMP) as a method for implementing ISO 9000. ISO 9000 certification concerns the process for designing and manufacturing the product, not the product itself.

Systems Engineering, Total Quality Management (TQM), Integrated Product Development (IPD), and Concurrent Engineering

Systems engineering, TQM, IPD, and concurrent engineering are interrelated and complementary processes and activities. A good systems engineering process is an essential part of TQM. Furthermore the philosophies and strategies of TQM, IPD, and concurrent engineering are key ingredients in designing and implementing an effective systems engineering process. According to the NASA System Engineering Handbook (Shisko 1995), "TQM is the realization that an operating organization is a particular kind of system and should be engineered as one."

What's It Going to Cost Me?

There is a large body of information addressing the costs associated with implementing rigorous project management and system engineering processes. These sources include publications by the Project Management Institute, the International Council on Systems Engineering, and university textbooks.

Both the project management community and the systems engineering community claim ownership of many of the system development process activities. The Lockheed Martin systems engineering process provides the first step in sorting these costs by breaking the systems engineering process into a management element and a technical element.
The bottom line is that it's difficult to summarize the cost estimates in the literature. But we'll try anyway.

The majority of sources estimate that the cost of the project management activities is approximately 5% of the total project cost. For example, if the project is designing a facility, then the project management cost of the design activity is approximately 5% of the total cost of producing the design package. If the project is building the facility, the project management costs are approximately 5% of the total cost required to bring the facility into operation.

There is a wider range of estimates for the cost of the system engineering activities than for the project management activities. System engineering cost estimates range from 5% to 15% of the total project cost. Many of the systems engineering references consider project management to be a subset of systems engineering. It's reasonable to assume that the higher estimates include many of the project management activities.

For planning purposes, 10% to 15% of the total project costs should be budgeted for combined system engineering and project management activities. It's the project manager's responsibility to eliminate duplication of efforts by systems engineering and project management.

A System or a System of Systems?

Most systems development projects consist of four distinct but closely coupled systems: (1) the engineering design and development system that produces and verifies the designs for the product system, (2) the production system, which actually manufactures or produces the product system (3) the logistics system that the end user needs to supply, operate, and maintain the product system, and (4) the product system itself. The product system is the system delivered to the end user that is intended to solve the user's problem or meet the user's needs.

In some applications there is a fifth system. This is the system of inspections, tests, and analyses needed to verify continued operational readiness and performance of the product system. Sandia weapons system examples include surveillance tests and JTA tests of stockpile systems.

Document Objective

The objective of this document is to pull together many of the fundamental elements and lessons learned in systems engineering. Our intent for this document is to provide some guidance for developing a systems engineering process tailored to the needs of any specific project. Many organizations document the project's systems engineering process in the "Systems Engineering Management Plan." Details for a SEMP are described in both the IEEE 1220 and DoD (EIA 632) Systems Engineering Standards.

This document describes the activities recommended for a typical project. One might conclude from this document, though incorrectly, that the steps occur in a linear sequence. In general, it is true that step n will usually begin before step n+1 begins. (In the terminology of project management, this is a start-to-start constraint.) With real-world projects, many of the steps will proceed in parallel. Some groups of steps will actually form iterative loops. Occasionally other constraints may change the actual sequence of the steps. In general, however, the steps in the order described here are a good guide.

The definitions of the stages in this document are consistent with the definitions of the phases or stages used in industry, academia, DoD, and DOE literature (with the exception of EP401099).

The activities recommended here are applicable to any project or study. They are applicable to weapon system projects, for example, an AF&F (Arming, Fuzing, and Firing) project. Each of the AF&F subsystem projects (such as a programmer, radar, or fireset) and component

projects (such as a battery or an impact fuze) can follow the activities prescribed in this document. These activities are also applicable to other hardware projects, such as an upgrade or modification to a weapon, a shipping container, or a computer system.

Stage 1. Developing Requirements

Systems development projects usually begin with a requirements development process. Customarily the requirements development (Stage 1) activities are a stand-alone project with a separate project plan, budget, schedule, and statement of work. The systems engineering team assembled to complete this activity typically has a wide range of expertise, including engineering, analysis, manufacturing, testing, reliability, logistics, cost analysis, and human factors. Management structures vary widely. In some projects the team may be lead by a first-level manager, and in other cases it may be lead by a senior technical staff member. The essential principle is that a wide range of disciplines be competently represented at the earliest stages of the project. The work to be accomplished, the schedule, the milestones, and the deliverable will be described in the Project Plan or the Systems Engineering Management Plan.

Traditional systems engineering approaches recommended that all Stage 1 activities be completed prior to beginning Stage 2 activities. The traditional approach does not always apply to today's projects because many of the Stage 2 activities are frequently started, at least at a preliminary level, during Stage 1. This document will not address that aspect, however, because our goal is to make the systems engineering process understandable to new staff and engineers unfamiliar with the details of systems engineering. Our philosophy is to list the fundamental elements or steps that the systems engineer ought to consider when planning a systems engineering project. The extent and details of implementation, the ordering of the steps, and the interactions among the steps are left as exercises for the systems engineering team on the project. Past DOE weapons development programs spread the Stage 1 activities over the Pre-Phase 1, Phase 1, Phase 2, Phase 2A, and beginning of Phase 3 programs. DoD and Commercial experience indicates that developing a good set of requirements before beginning the creating and evaluating concepts stage is a key ingredient for reducing development costs and time.

Elements of Stage 1 and their interrelationships are illustrated in Figure 8.

Step 1. Identify Customers and Stakeholders

The first step in developing requirements is to identify the customers. The terms customer and stakeholder include anyone who has the right to impose requirements on the system. This includes end users, operators, bill payers, owners, regulatory agencies, victims, sponsors, manufacturers, transporters, maintainers, etc. Because systems engineering delivers both a product and the process for manufacturing it, we must also consider the manufacturers and producers. All facets of the customer must be kept in mind during system design. For example, in evaluating the cost of a system, the total life cycle cost and the cost to society

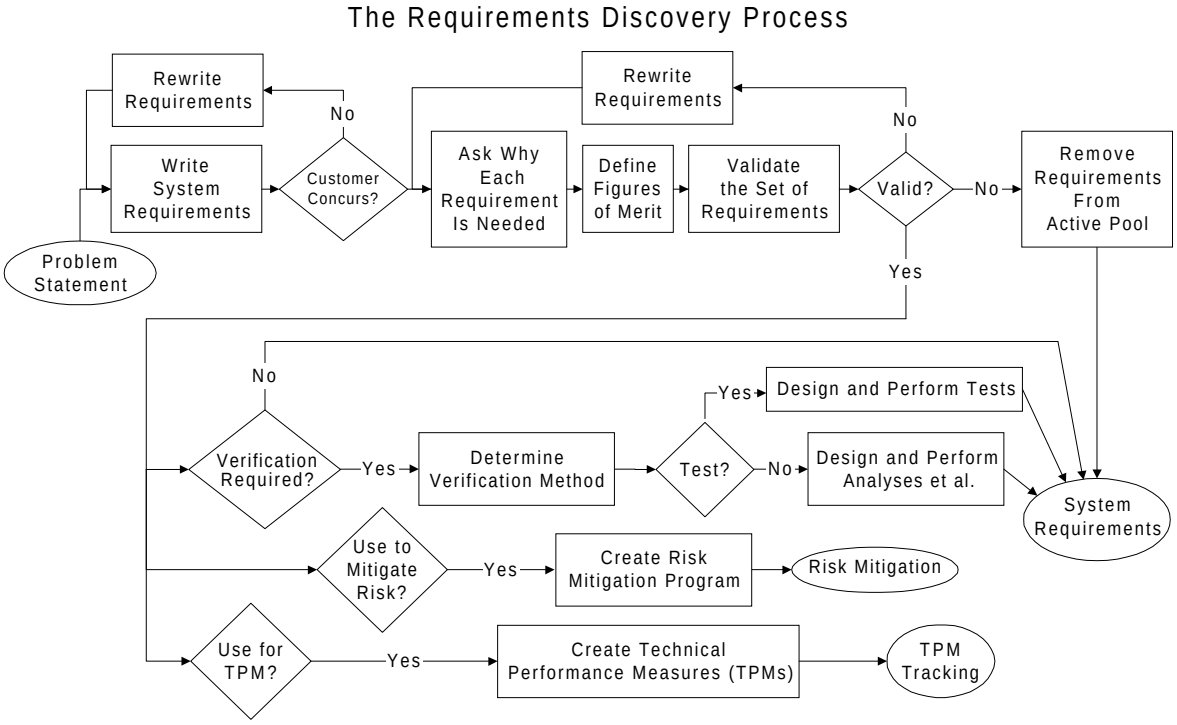


Figure 8. The Requirements Discovery Process

should be considered. Frequently, the end user does not fund the cost of development. This often leads to products that are expensive to own, operate, and maintain over the entire life of the product because the organization funding development saves a few dollars in the development process. The systems engineer must understand this conflict and expose it. The sponsor and user can then help trade off the development costs against the cost to own and maintain. Total life cycle costs are significantly larger than initial costs. For example, in one of its advertisements, Compaq proclaimed, "Eighty percent of the lifetime cost of your company's desktops comes after you purchase them." In terms of the personal computer, if total life cycle costs were $10,000, purchase cost would have been $2,000 and maintenance and operation costs $8,000.

Let us now illustrate some customer and stakeholder roles for a commercial airliner, such as the Boeing 777. The users of the product are passengers that fly on the airplane. The operators are the crew that fly the plane and the mechanics that maintain it. The bill payers are the airline companies, such as United, TWA, etc. The owners are the stockholders of these companies. The Federal Aviation Administration (FAA) writes the regulations and certifies the airplane. Among others, people who live near the airport are victims of noise and air pollution. If the plane is tremendously successful, McDonnell Douglas (the manufacturer of a competing airplane) would also be a victim.

The users and operators of the process would be the employees in the manufacturing plant. The bill payer would be Boeing. The owner would be the stockholders of Boeing. Occupational Safety and Health Administration (OSHA) would be among the regulators. Victims would include physically injured workers and, according to Deming, workers stuck doing mindless, repetitive tasks who have little control over the output but are reviewed for performance (Latzko and Saunders 1995).

Step 2. Understand the Customer's Needs

The system design process must begin with a complete understanding of the customer's needs. The information necessary to begin a design usually comes from preliminary studies and specific customer requests. Frequently the customer is not aware of the details of what is needed. Systems engineers must enter each customer's environment, discover the details, and explain them. Flexible design and rapid prototypes facilitate identification of details that might have been overlooked. Talking to the customer's customer and the supplier's supplier can also be useful. This activity is frequently referred to as "mission analysis."

The systems engineer is responsible for ensuring that all information concerning the customer's needs is collected. The systems engineer must also ensure that the definitions and terms used have the same meaning for everyone involved. Several direct interviews with many customers and stakeholders are necessary to ensure that all of the customer's needs are stated and that they are clear and understandable. The customer might not understand the needs; he may be responding to someone else's requirements. Often a customer will misstate his need. For example, a person might walk into a hardware store and say he needs a half-inch drill bit. But what he actually needs is a half-inch *hole* in a metal plate, and a chassis-punch might be more suitable.

Customers did not know that they wanted a stealth airplane until after the engineers told them it was possible.

Step 3. Define and State the Problem

What is the problem the system is required to solve? Answering this question is one of the systems engineer's most important and often overlooked tasks. An elegant solution to the wrong problem is less than worthless (Wymore 1993).

Early in the process the customer frequently fails to recognize the scope or magnitude of the problem that is to be solved. The problem statement should not be described in terms of a perceived solution. The systems engineer must help the customer develop a problem statement that is completely independent of solutions and specific technologies. The systems engineer's responsibility is to work with the customer, asking the necessary questions to develop a complete "picture" of the problem and its scope. Through interaction with the customer (and much deliberation) a clear, concise, and complete problem statement (independent of a solution) can be derived. Experience indicates that this problem statement is frequently quite different from the original customer request.

Step 4. Write System Requirements

The process of developing and specifying requirements is often referred to as Requirements Analysis. The systems engineer must interact with the customer to develop the requirements. The systems engineer must involve the customer in the process of defining, clarifying, and prioritizing the requirements. The prudent systems engineer involves users, bill payers, regulators, manufacturers, maintainers, and other key players in the process.

The customer will usually differentiate between mandatory (hard) requirements and preference (soft) requirements. Mandatory requirements are those requirements that must be met in order for the system to be acceptable (e.g., laws). Mandatory requirements are either met or are not. Preference requirements will often have desired target values associated with them that are usually called "goals." In the context of this paper, the terms "target value" and "goal" are used to describe an upper bound, a lower bound, or an optimal value for a preference requirement (e.g., reliability should be at least x, or cost should not exceed y). Preference requirements are those requirements for which either more or less will result in increased happiness (utility value) of the customer. An important function of the systems engineer is separating the "wants from the musts and the shoulds from the shalls."

Next, systems engineering must discover the functions the system must perform in order to satisfy its purpose. The systems functions form the basis for dividing the system into subsystems. QFD is useful for identifying system functions (Bahill and Chapman 1993; Bicknell and Bicknell 1994).

Although this implies that requirements are transformed into functions in a serial manner, that is not the case. It is actually a parallel and iterative process. First we look at system requirements, then at system functions. Then we reassess the requirements and the functions again, etc.

In the past, requirements analysis involved a progression from performance requirements to functional requirements to design requirements. For example, a teenage boy might express the operational need this way: "Hey, Dad, we need speakers in the car that will make your insides rumble during drum solos." The father would translate this into the performance requirement: "For bass frequencies, we need 110dB of sound output." Then the systems engineer would convert this into the functional requirement: "Amplify the radio's output to produce 115 watts in the frequency range 20 to 500 Hz." Finally, after a trip to the audio shop, the design engineer would transform this into the design requirement, "Use Zapco Z100S1VX power amplifiers with JL Audio 12 W1-8 speakers." But this implies a sequential process; in today's environments, the requirements process is concurrent and iterative.

The systems engineer's responsibility is to work intensely with the customer and/or those who represent the customer to develop the customer requirements. Many types of requirements exist, including performance, cost, test, acceptance, technology, trade-off and schedule. SAND96-1620 is an exclusive discussion of requirements and their sources.

To reiterate, requirements development is an iterative process. As the system development process matures, more requirements--including top-level requirements--will come to light. The customer usually is not completely sure of what he requires of the system, and the systems engineer probably is not yet sure of exactly what the system can do.

Step 5. Consult with the Customer

The systems engineer's responsibility is to confirm with the customer that the requirements are correct and that the set of requirements is complete. The systems engineer must meet with the customer regularly to review, correct, and approve the requirements. The customer must agree that if these requirements are met, then the system should meet his needs and expectations. This process terminates in the Systems Requirements Review (SRR) described in Step 11. Step 5 should be conducted repeatedly during requirements development and should prevent surprises and embarrassments at the SRR.

At these meetings it is important to ask why each requirement is needed. This can help eliminate unneeded requirements. It can also help reveal the requirements behind the stated requirements. It may be easier to satisfy the requirements behind the requirements than the stated requirements themselves.

Step 6. Define Figures of Merit

Figures of merit are the criteria by which the different designs will be "judged." Each figure of merit must have a fully described unit of measurement. Units of power could be horsepower, for example, and units of cost could be dollars (or inverse dollars, if it is desirable to consistently have "more is better" situations). If, for example, a figure of merit were acceleration, then the unit of measurement could be seconds taken to accelerate from 0 to 60 mph. The units of measurement can be anything, as long as they measure the appropriate criteria, are fully described, and are used consistently for all designs. A figure of merit describes how effectively a preference requirement has been met. For example, the car went from 0 to 60 in 6.5 seconds. These values are the ones put into scoring functions, as shown in Figure 9, to give the requirements scores, which are in turn used to perform trade-off studies. Such measurements are made throughout the development of the system.

A scoring function is used to give a system (alternative design) a normalized score that reflects how well each criterion has been met. The scoring function translates the value measured for a figure of merit into a normalized score. The use of scoring functions allows different criteria to be compared and traded off against each other. In other words, scoring functions allow apples and oranges to be compared and nanoseconds to be compared to billions of dollars.

Figures 10, 11, and 12 show scoring functions for a system requirement, a subsystem requirement, and a component requirement respectively.

Seldom does one optimize all requirements. Moving from one alternative to another will improve at least one criterion; and worsen at least one criterion; that is, there will be trade-offs. A trade-off (function) is a give-and-take relationship between preference requirements. Recall that preference requirements have a range of values for which we either prefer more or prefer less. There can be trade-offs, for example, if we get more of one requirement and less of another (assuming in both cases more is better). To be less abstract, let us discuss the

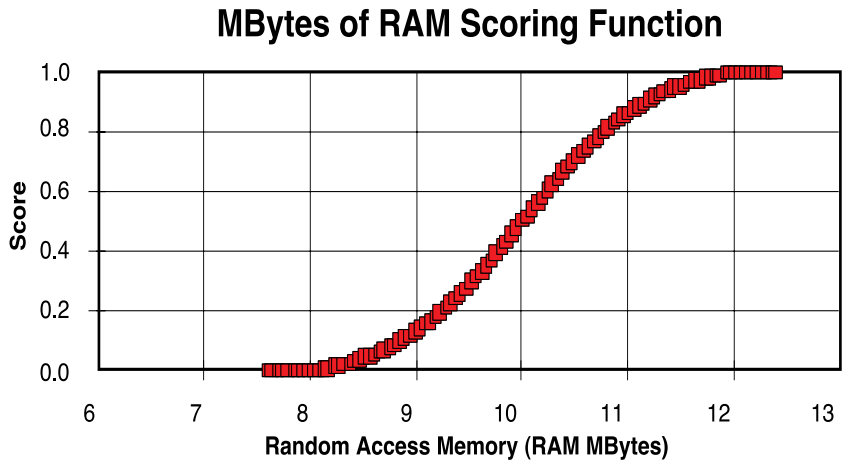


Figure 9. Scoring Function for the Amount of Computer Memory

trade-off between cost and power of a new vehicle. The trade-off is obvious: you might be willing to spend more to get more power, or you might be willing to give up a bit of power in order to reduce cost. Which car you choose, i.e., a car with more power and higher cost, or a car with less power and lower cost, will depend on which criterion is more important to you, cost or power.

Other examples of preference requirements for which trade-offs may occur are (1) yield and circular error probable (CEP), (2) yield and weight, (3) CEP and range, (4) radiation hardness and range, and (5) throw weight and range. The trade-off curve of throw weight and range is illustrated in Figure 13.

The use of scoring functions in industry has not yet flourished, although scoring functions are endemic to academia. Scoring functions, however, are a very useful tool and should be seriously considered. Deriving the parameters for the scoring functions is a difficult task, but the analysis will have to be done sometime in the systems engineering process. Eventually, alternative designs will have to be compared.

Trade-off studies are defined, conducted, and documented at the various levels of functional or physical detail to support requirements, functional decomposition/allocation, and design

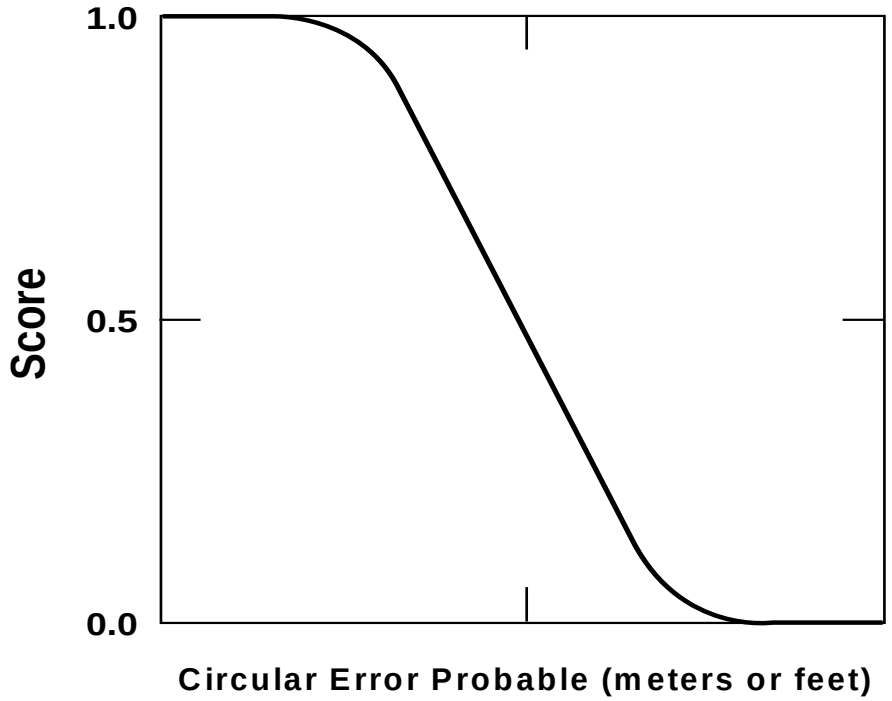


Figure 10. Performance Figure of Merit

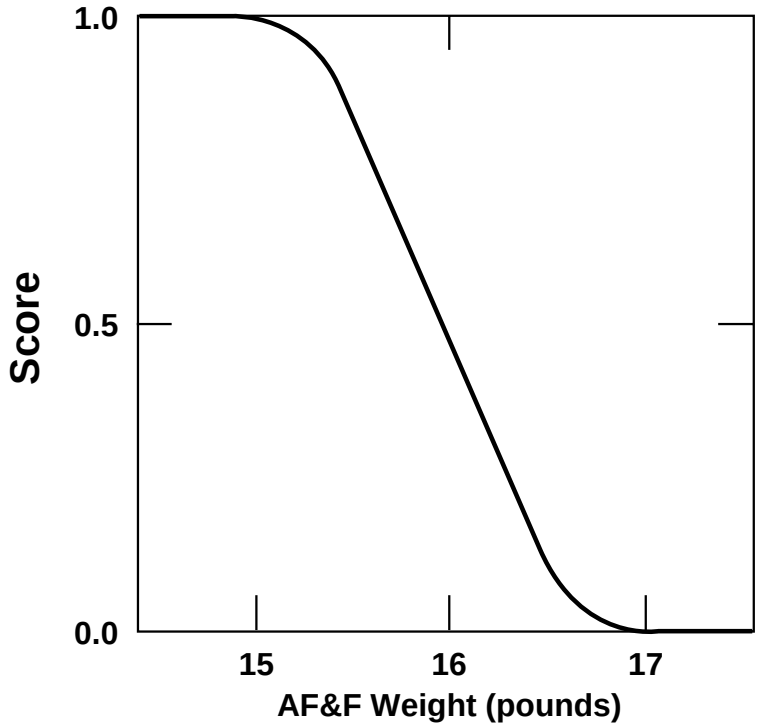


Figure 11. Cost Figure of Merit

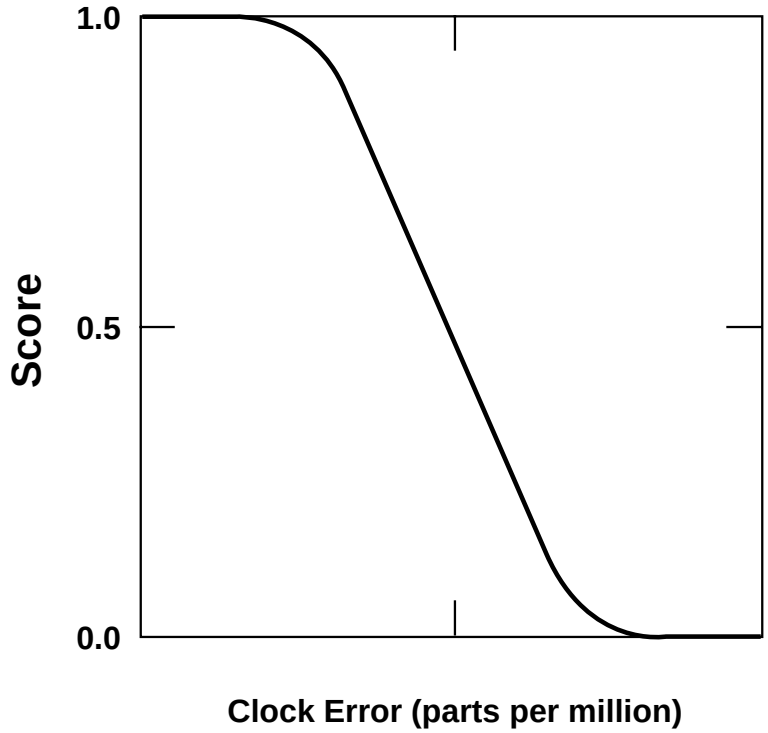


Figure 12. Performance Figure of Merit

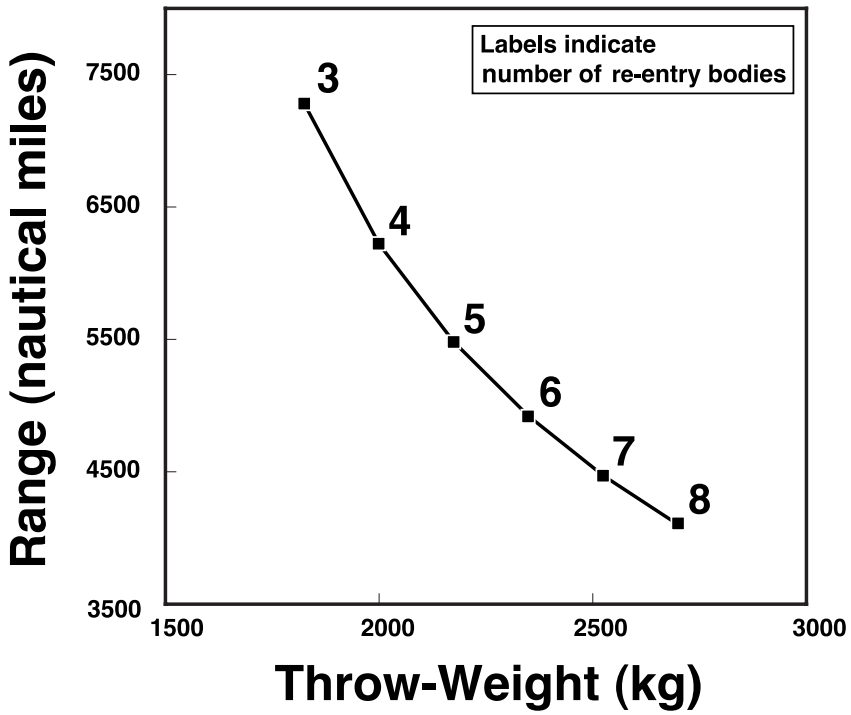


Figure 13. The Trade-off Between Range and Throw Weight

alternative decisions. Or, as specifically designated, trade-off studies may support the decision needs of the systems engineering process. The level of detail of a study should be commensurate with cost, schedule, performance, and risk impacts.

Trade-off studies are conducted at a variety of levels: for example, among the alternative concepts, among the subsystem of each concept, among the alternatives for each subsystem, and so on.

At this time it may be useful to examine the following definitions.

Definitions for Steps 7 and 8

Validating a System—This means building the right system; making sure that the system does what it is supposed to do. It determines the correctness of an end product, compliance of the system with the customer's needs, and completeness of the system**.**

Validating Requirements—This means ensuring that the set of requirements is consistent, ensuring that a real-world solution can be built that satisfies the requirements, and proving that such a system satisfies its requirements. If systems engineering discovers that the customer has requested a perpetual-motion machine, the project should be stopped.

Verifying a System—This means building the system right; i.e., ensuring that the system complies with its requirements. Verifying a system determines the conformance of the system to its design requirements. It also guarantees the consistency of the product with itself and with the previous prototypes at the end of each phase. In other words, it guarantees the honest and smooth transition from model to prototype to preproduction unit to production unit.

Verifying Requirements—This means an examination, analysis, test, or demonstration that proves whether a requirement has been satisfied. This process is iterative. The requirements should be verified with respect to the model, the prototype, the preproduction unit, and the production unit.

Verification and Validation—MIL-STD-1521B (and most systems engineers) and MIL-STD-2167A (and most software engineers) use the words “verification” and “validation” in almost the exact opposite fashion (Grady 1994). For systems engineers, to validate requirements is to prove that it is possible to satisfy them. System verification, on the other hand, is a process of proving that a system meets its requirements. To add further confusion, ISO 9000 tells you to verify that a design meets the requirements and validate that the product meets the requirements. NASA has a different spin. It says that verification consists of proving that a system (or a subsystem) complies with its requirements, whereas validation consists of proving that the total system accomplishes its purpose (Shishko 1995). Thus, it is necessary for all (sponsor, users, bill payers, etc.) to agree on the definitions of verification and validation as these terms pertain to your system.

Step 7. Prescribe System Verification Process

A critical element of the requirements development process is describing the tests that will be used to prove compliance of the final system with its requirements. How the system will be tested must be specified; i.e., the testing requirements must be complete. Specifying the testing procedure will aid in the detection of untestable requirements. The specification of the testing process informs the producers of the systems how the system will be tested. In other words, the producers of the system know how they will be graded. This process frequently uncovers overlooked requirements. It also lessens the probability of finding yourself in the position of not being able to demonstrate to the customer that your product meets his needs. Requirements can be verified by test, analysis, demonstration, inspection, logical argument, modeling, or simulation.

Step 8. Validate System Requirements

Validating requirements means ensuring that the requirements are consistent (no requirement contradicts another) and that a real-world solution can be built (or already exists) and tested to prove that it satisfies the requirements. If you discover that the customer has requested a perpetual-motion machine, the project should be stopped. Each requirement should be technically feasible and fit within the budget, schedule, and other constraints. Requirements are often validated by reference to an existing system that meets most of the requirements. The requirements that are not satisfied by the existing system are validated by argument, modeling, or simulation.

The systems engineer is responsible for validating the requirements during the SRR.

Step 9. Define Technical Performance Measures

Technical performance measures (TPMs), or metrics, are used to track the progress of the design and manufacturing process. They are measurements that are made during the design and manufacturing process to evaluate the likelihood of satisfying the system requirements. Not all requirements have TPMs, just the most important ones. In the beginning of the design and manufacturing process, the prototypes will not meet the TPM goals. Therefore, the TPM values are only required to be within a tolerance band. Ideally, as the design and manufacturing processes progress the TPM values of the prototypes and preproduction units will come closer and closer to the goals.

Step 10. Risk Mitigation

Identifying and mitigating program risk is the responsibility of management at all levels of the company. Each item that poses a threat to the cost, schedule, or performance of the project must be evaluated. The Risk Analysis process estimates both the probability and consequence of each risk item. The output of the Risk Analysis process includes a prioritized list of risks and a recommendation of those risks that should be managed and tracked. The following information should be recorded for each identified risk: name, description, type, origin,

probability, severity, identification number, identification date, identification on the work breakdown structure, risk mitigation plan responsible team, needed resolution date, principal engineer, current status, date, and signature of team leader. Forms useful in identifying and mitigating risk are given in chapter 17 of Kerzner (1995), section 4.10 of Grady (1995), and the DSMC Risk Management Guide.

Step 11. System Requirements Review (SRR)

The Stage 1 activities are usually concluded with the Systems Requirements Review (SRR). The SRR must be conducted with and for the customer. The customer's approval of the requirements is necessary before proceeding with the project. The systems engineer is responsible for initiating and conducting the review. The main objectives of this review are to find missing requirements, eliminate unneeded requirements, ensure that the requirements can be met, and verify that the system satisfies customer needs. At this review trade-offs will usually have to be made between performance, schedule, and cost. Additional objectives include assessing the maturity of the development effort, recommending whether to proceed to the next phase of the project, and committing additional resources. This review should be formal with documented results and conclusions. The Defense Systems Management College's (DSMC) Systems Engineering Management Guide (1990) and the Electronics Industries Association's IS/632 (1994) are excellent sources for identifying what should be addressed in the various systems reviews.

The SRR is one of many requirements reviews with the customer. The requirements must be reviewed with the customer several times during Stage 1 and also during Stage 2. Some of these reviews will be more formal than others. At a minimum, requirements should be reviewed at the end of the modeling phase and after testing production units.

This is a good place to introduce the typical reviews one finds in a good systems engineering process. The following definitions based on Sage (1992) and Shishko (1995) might be useful. They are arranged in chronological order. Although these definitions are written with a singular noun, they are often implemented with a collection of reviews. Each system, subsystem, subsubsystem, etc. will be reviewed, and the totality of these constitutes the indicated review.

Mission Concept Review

The Mission Concept Review (MCR) and the Mission Definition Review (MDR) are the first formal reviews. They examine the mission objectives and the functional and performance requirements. If the organization does not have a Vision or Mission statement, then you should write one. A similar review is called the Mission Requirements Review in the DOE Interagency Engineering Procedures Manual EP401063/E-Design Reviews (1996).

System Requirements Review (SRR)

This review demonstrates that the product development team understands the mission and the system requirements. It confirms the system requirements are sufficient to meet mission

objectives. It ensures the performance and cost figures of merit are realistic, and the verification plan is adequate. At the end of the SRR, the requirements are placed into a formal configuration management system with approvals required for changes. Changing requirements after this review will impact schedule and cost. A similar review is called the Conceptual Design Review in EP401063/E-Design Reviews manual (1996).

System Definition Review

This examines the proposed system architecture, the proposed system design, and the flow down of functions to the major subsystems. It ensures that the verification plan is complete.

Preliminary Design Review (PDR)

This demonstrates that the preliminary design meets all the system requirements with acceptable risk. System development and verification tools are identified, and the Work Breakdown Structure is examined. Full-scale engineering design begins after this review. A similar review is called the Baseline Design Review in EP401063/E-Design Reviews manual (1996).

Critical Design Review (CDR)

This verifies that the design meets the requirements. The CDR examines the system design in full detail, ensures that technical problems and design anomalies have been resolved, checks the technical performance measures, and ensures that the design maturity justifies the decision to commence manufacturing. Few requirements should be changed after this review. The Final Design Review referred to in EP401063/E-Design Reviews manual (1996) is a similar review.

Production Readiness Review (PRR)

For some systems there is a long phase when prototypes are built and tested. At the end of this phase, and before production begins, there is a production readiness review.

Production Review

After production has been established, a review is conducted to determine whether any cost, performance, or scheduling problems that may have developed require additional design activity. This Production Review will identify any difficulties or changes that have occurred since the PRR.

System Test Review

The system is tested to verify that it satisfies its requirements. Technical performance measures are compared to their goals. The results of these tests are presented at the System Acceptance and Operational Readiness Reviews.

Post-Operational Review

After the system is operational but while it is still in production, an evaluation of the system is performed. Field experience may be used to identify operational, maintenance, or other problems that could be rectified by changing production processes or other redesign activities.

Figure 14 shows the timing of these major reviews.

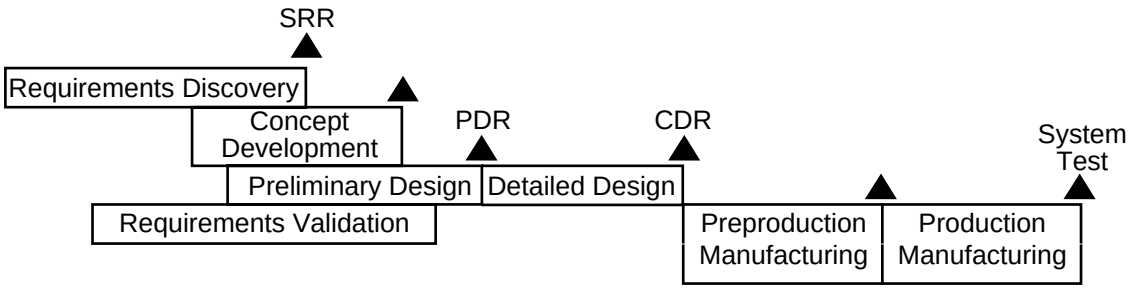


Figure 14. Timing of the Major Reviews

Stage 2. Creating and Evaluating Concepts

The majority of projects in an engineering development facility such as Sandia are concluded at the end of Stage 1; either the project is terminated or the Stage 1 results are passed on to another company or organization. The Stage 1 results are themselves a product. In some cases, the systems engineering team will begin the process at Stage 2 because another company or organization has completed and documented Stage 1. It follows, then, that the Stage 2 activities are customarily a stand-alone project with a separate project plan, budget, schedule, and statement of work. Often the first activities of this project will be to review and update the Stage 1 project results irrespective of who performed the Stage 1 activities. The DOE Phase 1, Phase 2, and Phase 2A projects focus on accomplishing the activities defined in Stage 2.

Step 12. Create and Evaluate Alternative System Concepts

Several alternative designs (concepts) should be proposed. The systems engineering designs should be evaluated based on the figures of merit defined during Stage 1. The systems engineer should be able to get an overall score for each design (if scoring functions are being used). Since the scores are normalized, comparisons can be done straight across designs; i.e., the design with the best score is the best design. This analysis should be redone whenever more data are available. For example, figures of merit will be computed initially based on estimates by the design engineers. After system models are constructed, analyzed, and simulation data are evaluated, the figures of merit should be reevaluated. Next, prototypes should be constructed, measured, and tested. Finally, tests should be run on the final system. Doing this analysis throughout the entire system development process helps to identify quickly those designs good enough to pursue. For the design of complex systems, alternative

designs reduce project risk. Areas of potential high risk should be identified and feasibility studies should be initiated. These activities are discussed in the risk management plan.

If the figures of merit scores for the alternative designs seem wrong, perhaps the weights of the figures of merit were not distributed correctly or the scoring functions are not correctly defined. Or more likely, perhaps important figures of merit have not been included. The weights, scoring functions, and figures of merit can be modified during the system development process.

Step 13. Sensitivity Analysis

Sensitivity analysis shows how sensitive the system performance is to changes in parameters, e.g., changes in values of figures of merit, weights, or scoring functions. A sensitivity analysis can determine which figures of merit have the largest impact on the overall score of a system concept and, hence, can determine which data must be accurate. When experiments are very expensive, it is valuable to know which data have the most impact on a system, i.e., which data have to be the most accurate. Some figures of merit will be found to have very little impact on the system, so it is unnecessary to be extremely accurate when calculating their values (or collecting data). In these ways, sensitivity analysis is used to find out what is most important and to help allocate resources. The latter step is an important part of advanced manufacturing.

Step 14. Functional Decomposition

At this point in the systems engineering process the systems engineer is still working in terms of the functions the system must perform in order to accomplish its mission. The systems engineer may or may not choose to map the functional elements of the system onto physical hardware and software elements at this time. For new systems based on existing systems, it may be appropriate to associate hardware with the functional elements. On the other hand, for unprecedented or revolutionary system solutions, the systems engineer may want to delay identifying or defining hardware until after completion of the functional decomposition of the subsystems.

When analyzing or re-engineering an existing system systems engineers do functional analysis to see what the system does in order to improve its performance (often called “value engineering”). And they also do functional decomposition to see what the system is supposed to do. In this manner, they can describe the present state of the system, the desired (or goal) state of the system, and suggest how the system design can be changed.

Systems engineers do functional decomposition on new systems (1) to map functions to physical components, thereby ensuring that each function has an acknowledged owner, (2) to map functions to system requirements, and (3) to ensure that all necessary functions are listed and that no unnecessary functions are requested. The list becomes the basis for the work breakdown structure and is the part of the systems development process where similar functions and requirements are grouped together. This forms the basis for partitioning the

system into subsystems and components, and it is also the basis for assigning requirements to subsystems and implementing requirements traceability.

Partitioning the system into subsystems is one of the most important activities of the systems engineering process. Improper partitioning can cause more interfaces than may be necessary, overly complicated interfaces, and generate unnecessary requirements. Good partitioning should minimize and simplify the number of internal and external interfaces.

Typically system-level requirements are passed down to the subsystems as Control Documents or Control Drawings (CDs). Similarly, CDs are used to pass requirements down from subsystems to components. Interface Control Documents or Interface Control Drawings (ICDs) are use to define interface requirements between subsystems and between components.

The typical functions of a nuclear weapon are to prevent unauthorized use; prevent nuclear detonation in accident scenarios; allow authorized use; detonate at designated time or location; and select fuzing or detonation mode.

The important outputs of the functional decomposition step are a functional description of each subsystem and a set of functional requirements for each subsystem.

A note of caution is in order: if the systems engineer is not careful, the system architecture is likely to be influenced more by a company's organizational structure than the functions that the system must accomplish (needs of the customer).

Step 15. Begin Subsystem Requirements and Design Projects

The subsystems engineers begin the systems engineering process for the subsystems. The steps are the same as they are for the system, which illustrates the recursive or fractal nature of the systems engineering process (Figure 15). The subsystems systems engineer should begin at Step 1 to determine the requirements for his subsystem in his language.

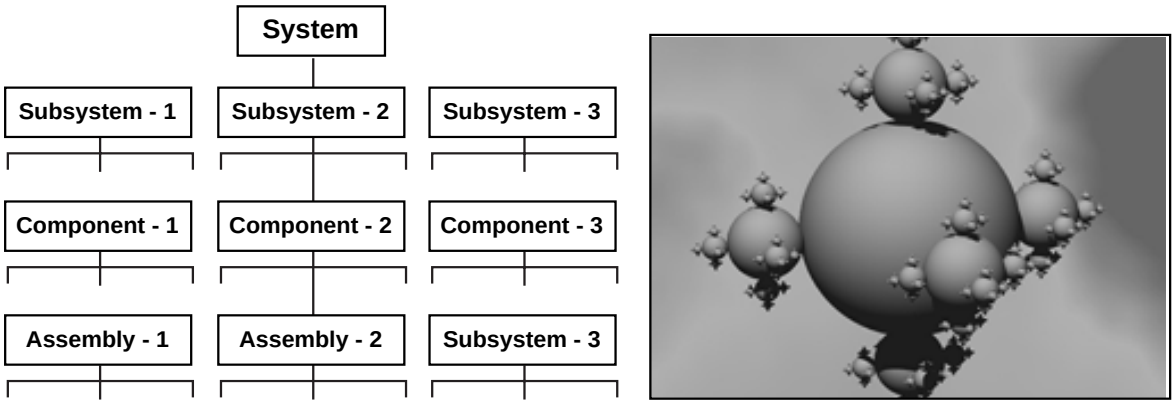
At this point the interfaces among the subsystems and with the system must be negotiated, formally documented, and formally approved. The interfaces must be subjected to formal and rigorous configuration management and change control.

The subsystems engineers will usually come back to the systems engineer for requirements redefinition or subsystem functionality clarification; therefore, the systems engineer acts as the subsystems engineer's customer.

Step 16. System Design (Define System Architecture)

The overall system should be divided into subsystems of similar complexity. Subsystem requirements continue to be defined and refined. Interfaces among subsystems are negotiated and defined. Reusability should be considered in creating subsystems. For new designs, subsystems should be created so that they can be reused in future products. For redesign, subsystems should be created to maximize the use of existing (particularly commercially

available) parts. Systems engineers must also decide whether to make or buy the subsystems. They must determine if there are commercially available subsystems that satisfy the requirements. If nothing satisfies all the requirements, then modification of an existing subsystem should be considered. If this proves unsatisfactory, then some subsystems will have to be designed. Engineers designing one subsystem must understand the other



The systems engineering process is applied at levels of greater and greater detail. It is applied to the system, then to the subsystems, then to the components, etc. Similarly, for the fractal pattern above the same algorithm was applied at the large structural level, then at the medium-scale level, then at the fine detail level, etc.

Figure 15. Systems Engineering Is a Fractal Process

subsystems that their system will interact with. This is where the results of the trade studies support negotiating and defining interfaces among the subsystems. Flexibility may be more important than optimality.

Hardware, software, and bioware must be considered. Bioware applies to living things that use, operate, or are a part of the system. For example, in designing a racetrack, the horses or dogs are a part of the bioware. Considering bioware involves providing for education, safety, human factors, facilities, handling equipment, and assembly tools.

At this early stage in the development process it is important to consider design for manufacturability, design for reliability, design for modeling and analysis, and design for test.

Step 17. Analysis, Models, Prototypes, and Simulations

System analysis, modeling, and prototyping should be directed towards providing early technical data to evaluate the feasibility of the conceptual design to meet the design requirements. Model and prototype data are also useful for trade-off studies. The models or prototypes are necessary to identify and reduce the risk of using new technology, and they

provide insight into whether or not the real system will meet the requirements and constraints. Models or prototypes may be part of the complete system or may be a partial description of the system or one of its subsystems. Models or prototypes may be analytical, may use physical hardware, or may be a combination of both. The system architecture only needs to be defined to supply gross answers that are needed during this phase.

The modeling, prototyping, and simulation done at this stage will usually be general because the concepts and all of their subsequent details have not been fully defined yet. This early modeling, prototyping, and simulation can also help identify those concepts that are not worth pursuing.

Step 18. Conceptual Design Review

The conceptual design review is held at the end of Stage 2. The objectives of the conceptual review are to (1) verify that the conceptual designs produced in Stage 2 can be expected to satisfy the customer's needs and requirements when the designs are built, (2) assess the likelihood that systems represented by the design concepts can actually be built, (3) assess the adequacy and accuracy of the resource estimation for future phases, (4) assess the maturity of the development effort, and (5) recommend whether or not to proceed to Stage 3, and (6) commit the required resources. EIA 632 has an excellent description of the purpose and issues to be discussed in the various systems engineering reviews.

Stage 3. Engineering Design and Development

Customarily the Stage 3 activities are a stand-alone project with its own unique project plan, budget, schedule, and statement of work. Frequently the first activities of the Stage 3 project will be to review and update the results of the Stage 1 and 2 projects. On occasion the first activities of the Stage 3 project will be to redo extensive portions of the Stage 1 and 2 activities.

Stage 3 translates the vision into reality. Stage 3 is where the product is created, where the detailed engineering and design take place. The activities that take place in Stage 3 are essentially the same as the activities that made up the DOE Phase 3 and DOE Phase 4 projects. The main difference is that many of the Stage 3 activities can and should be concurrent rather than sequential. In the past, many of the Phase 3 and Phase 4 activities were performed sequentially. With modern analysis, simulation, prototype manufacturing facilities, and continuously improving production processes, concurrency is a better approach.

Most of the Stage 3 activities are conducted in parallel. The sequence can be fine-tuned for specific applications. In general, one would want to start developing a detailed Work Breakdown Structure (WBS) very early in Stage 3. However, initiating configuration management too early stifles the design process and may needlessly limit design approaches. On the other hand, waiting too long to institute Configuration Management creates chaos and greatly increases the probability of cost and schedule overruns. The following paragraphs describe some of the main Stage 3 activities.

Construct a Product-Oriented WBS

The systems engineer can now begin developing a work breakdown structure. The work breakdown structure should be based on products and physical hardware. The tasks and activities needed to realize products are frequently described in the work packages for each WBS element. The top level will be the system and necessary support and logistical products. Products will include hardware, software, manufacturing processes, and documentation such as study reports, specifications, and drawings.

This work breakdown structure will be developed while the subsystem and subsequent components are being defined.

Stable and Controlled Requirements

When full-scale engineering design and development begins, changes in requirements can have very expensive consequences. At the beginning of Stage 3 a formal change control process must be in place for the requirements.

Detailed Engineering Design

Detailed specifications, drawings, and descriptions for the system and its subsystems must be defined. These designs will be refined as greater detail is introduced and as the system becomes further defined.

Manufacturing Engineering

The systems engineer is responsible for defining the process or manufacturing system that will build or manufacture the customer's system. For large-scale, complex systems, this is typically approached as a systems engineering project.

Designing and Managing Interfaces

Interfaces between subsystems and between the main system and the external world must be designed. Care should be taken to ensure that interfaces between subsystems follow natural or functional flows. When the same information travels back and forth among different subsystems, a natural way to combine the subsystems into larger, more general subsystems may have been overlooked. Subsystems should be defined to minimize the amount of information that must be interchanged or transmitted among the subsystems. Well-designed subsystems send finished products to other subsystems. If this is not the case, the systems engineer should review and revise the functional decomposition of the system. He should look for an alternative or more appropriate decomposition.

Once the interfaces are defined and agreed upon, they must be subjected to a formal change control and approval process.

Negotiation, definition, and management of external interfaces are usually a difficult and time-consuming processes. These interfaces usually involve multiple agencies and companies, each with their own priorities and agenda.

Configuration Management

Configuration management (also called modification management) ensures that any changes in requirements, design, or implementation are controlled, carefully identified, and accurately recorded. All stakeholders should have an opportunity to comment on proposed changes. Decisions to adopt a change must be captured in a database and reflected in system documentation. (This documentation is a time-frozen design for costing, building, testing, etc.) All concerned parties must be notified of changes to ensure that they are all working on the same design. The phrase "requirements tracking" is now being used for an important subset of configuration management.

Configuration management should begin early in the process and continue throughout the entire systems engineering process.

Engineering Analysis and Modeling

Engineering analysis and modeling can be divided into three broad categories. First there is the analysis and modeling that simulates the functional performance of the system, the subsystems, and their interfaces. Second, there is the environmental analysis that simulates the wide range of environments found in a typical System Stockpile to Target Sequence (STS) document and the response of the system and subsystems to these environments. And third, there is the analysis that supports system verification, which is discussed in the next stage. It's not unusual for the same analyst to use the same tools and the same modeling principles for each of these categories. However, the results and the data produced by the analysts have greatly differing customers and roles. The details, accuracy, and time urgency requirements for the analysis is frequently different for each category. Furthermore these requirements typically become more restrictive as the project progresses in the systems development life cycle. Excessive detail early in the development life cycle may not add value to the project.

A wide variety of systems and subsystems analysis is required using a wide variety of tools. Typical analysis include electrical circuit analysis, mechanical response analysis, thermal analysis, aerodynamic analysis, electromagnetic analysis, radiation transport, and others.

Analysis and Testing Are Synergistic

Analysis and testing are synergistic activities, each gaining value from the other. An effective design strategy is to apply them both in an iterative process, building on the results of preceding activities. They both become more complex and realistic as the system design matures.

Successful environmental tests are always preceded by extensive analysis. One can not realistically test physical hardware until analysis has determined which STS environments stress the hardware the most. Analysis also indicates where to find the largest responses within the system hardware. This information is critical to proper selection and placement of the instrumentation sensors. Analysis also determines which orientation of the system hardware relative to the environment provides the greatest stimulation of the responses. Without analysis we can only guess at the proper test orientation, test levels, test conditions, instrumentation locations, and calibration.

Successful analysis and modeling depends upon meaningful inputs from experiments and tests with realistic physical hardware. The equations and parameters used in many computer models come from experiments and tests. Experiments allow us to collect accurate input data such as material properties for the computer models. Test with realistic physical hardware provide a method of calibrating the computer models in known conditions. This improves confidence in the results of the model over a wider range of STS conditions.

Frequently there are details in the design of the physical hardware that are too complex for the current generation of computers, analysis software, and models. Examples of these details include interfaces, nonlinear joints, elastic plastic transitions, shock effects, spallation effects, etc. Many of these features require zoning and time steps so fine that the analysis is beyond the scope of today's computers. So we use test data to develop simple representations of these features or responses for incorporation into the computer models.

During the life cycle, system hardware can be subjected to combinations of STS environments either sequentially or simultaneously. Physically simulating combinations of STS environments in tests is usually difficult, expensive, and many times impossible. Analysis and modeling provides the necessary simulation of these scenarios.

The Roles of Analysis and Testing Change During the Development Cycle

Early in the systems development process testing and analysis provide the engineers with a picture of the environments in which the system must function. Testing and analysis provide the design engineers with feedback concerning the performance of the design in various STS scenarios. Later in the system development process, prototypes provide the designers with feedback, the analysts with model validation information, and the testers with criteria for designing and conducting system validation tests. During the final stage of the system development program, testing and analysis are used to validate and verify the system. Frequently full-scale system tests, because of their complexity and cost, can address only a limited (although important) subset of system scenarios. Complete validation and verification is accomplished by a very closely coupled series of tests and analyses that address the full range of operational scenarios and environments.

System Integration

System integration means bringing subsystems together to produce the overall system. System integration demonstrates whether the subsystems will interact to satisfy the customer's needs or not. If the subsystems do not interact correctly, or if their compilation does not build the desired system, the subsystems requirements and their interface specifications must be reviewed and refined. Systems integration may need to be done several times during the systems engineering process. Mechanical models of all of the subsystems are produced periodically and fit-tested to assure that the final system can be assembled. Similarly, electrical breadboards and prototypes are periodically connected to assure compatibility of electrical circuits. Similar checks should be done periodically with the various software systems. Optimistically, subsystem interface and functionality problems will be detected during the modeling, simulation, and prototyping activities, when correcting problems is much less expensive.

Prototype Development and System Validation

System prototypes should be developed and built to aid in early detection of problems in system functionality, internal interfaces, and external interfaces. Prototypes should perform adequately in the environment in which the real system will operate (or as close to that environment as possible). Prototypes are often used to validate the system. System validation demonstrates that a real-world solution exists that meets all of the requirements or that a system can be built. If the prototype can be shown to meet the requirements set forth for the system, then the prototype validates the system. System validation is often referred to as “building the right system” (as opposed to system verification, sometimes referred to as “building the system right”).

Development Test and Evaluation (DT&E)

Development Test and Evaluation (DT&E) is performed to assist the engineering and development process to define the overall system operational capability. These activities support the development effort by providing technical information necessary for design decisions. They also help the designers determine if they are designing the right system to meet the system requirements. To do this better, DT&E should be integrated with the modeling effort. The analysis effort supports the experimental effort; likewise, the experimental effort supports the analysis effort. This increases the technical database and increases the effectiveness of the evaluation of the design.

Stage 4. System Verification

The main elements of the System Verification stage are (1) the verification plan, (2) analysis and modeling, (3) operational testing and evaluation, and (4) traceability analyses.

Verification provides the means for determining whether the system satisfies its requirements, how well it functions in the operational environment, and whether or not it should continue

into production. Verification, whether analytical or experimental, is the part of the system engineering process that is designed to obtain, verify, evaluate, or provide data to answer the fundamental question of whether the system satisfies its requirements. Testing is performed in support of the project to provide a technical database for decisions.

Verification Plan

The verification plan is essentially a more detailed version of the verification requirements defined in Stage 1. The requirements defined in this step should be traceable to those defined in Stage 1. Verification is accomplished with a combination of analysis, modeling, and physical testing. The verification plan establishes which analyses and data will be used to verify the system and how that data will be gathered and analyzed. Testing is an important element of verification; however, verification extends beyond testing into intensive data analysis. Requirements frequently specify a wide range of operational environments and conditions. Testing over the entire envelope of conditions is usually impractical. Test results are usually combined with analysis to verify the system over the entire range.

Analysis and Modeling in Support of Verification

Analysis and modeling are usually essential elements of the verification process. Physical tests are usually limited to a few special cases and environments.

Analysis and modeling are frequently required to verify compliance over the entire operating range and assess performance in combinations of environments.

Furthermore, analysis and modeling are frequently essential elements of the process for identifying the proper test environments and conditions.

Operational Testing and Evaluation

At the end of the development phase, the major question that must be answered is whether or not the system meets its requirements and satisfies the customer's needs. Operational testing and evaluation (OT&E) is performed to estimate the system operational effectiveness and suitability. The results from this testing are the basis for production recommendations. This testing provides data that confirm that the system can be produced, that it will meet the requirements, and that it will allow production verification testing to be performed.

Traceability Analyses

This activity should document activities, rationale, results, and impacts from the verification testing and analysis to ensure that the activities are traceable to system requirements. It should document the tests or analysis performed and the refinement of the hardware or model, procedures, and results sufficiently so that the test or model can be recreated at a later time. This should be coordinated with the verification plan.

Stage 5. System Production

Manufacturing, Production, and Assembly

Stage 5 consists of all the activities necessary to deliver the final products to the customer. These activities range from the processes used to accept raw materials and supplied parts, through acceptance testing of the end product. For past systems, manufacturing typically involved producing components and subsystems at a variety of DOE facilities (Allied Signal, Rocky Flats, Oak Ridge, Mound, and Pinellas), DoD facilities, and facilities of commercial suppliers. For most weapon systems final integration and assembly occurred at the DOE's Pantex Facility.

Production Acceptance Test and Evaluation

This activity is known as Production Acceptance Test and Evaluation (PAT&E) or Qualification Testing. PAT&E is performed on production items in order to determine and demonstrate that the production system meets the system requirements. Typical acceptance testing programs include both nondestructive and destructive testing of new product. This testing does not end when the system goes into production, but continues throughout production. Testing of the system will be necessary to determine if the production process is correct and repeatable. Testing will also determine if the system continues to meet the design requirements and production quality requirements. Finally, testing may also involve, if possible, evaluation of systems that have been in use to provide more data for design improvements, reliability, aging, and failure modes.

Stage 6. Operation, Maintenance, and Modification

One of the systems engineer's responsibilities is to ensure that the system satisfies its requirements throughout its life cycle. Therefore, the responsibility does not terminate when the system is delivered to the customer. The systems engineer must plan for the maintenance of the system. Further, the systems engineer should work with the customer to modify the system as new requirements arise or existing requirements change during the life of the system. During Engineering Design the systems engineers must create measures that can be used to assess the performance of the system during the Operation Stage. These measures can be used to continually improve system performance and quality. Typical systems engineering responsibilities during this stage induce stockpile surveillance, Joint Test Assembly (JTA), Limited-Life Component Exchanges (LLCEs), Nuclear Explosive Safety Study (NESS), studies, Stockpile Improvement Programs (SIPs), etc.

Stage 7. Retirement, Disposal, Recycle, and Replacement

The systems engineer should determine the life span of the product and make arrangements for its retirement, disposal, and replacement. All aspects of the system, including the final

stages of retirement and disposal, must be considered and accounted for. When disposing of a system or replacing one, recycling should always be considered.

Elements Common to All of the Systems Engineering Stages

There are many activities or processes that are common to all stages of the system development process and product life cycle. These include total quality management, project management, risk management, reviews, and documentation.

Total Quality Management

Everyone must continually look for ways to improve the quality of the system design process. Major tools used in this process include concurrent engineering, quality function deployment (QFD), and Taguchi's quality engineering techniques.

Project Management

Project management is the planning, organizing, directing, and controlling of company resources to meet specific goals and objectives within time, within cost, and at the desired performance level. The Project Management Institute's (PMI 1996) *A Guide to the Project Management Body of Knowledge* and Kerzner's (1995) *Project Management* are excellent sources of information.

Documentation

All of these systems engineering activities must be documented in a common repository. The stored information should be location, platform, and display independent, which means any person on any computer using any tool should be able to operate on the fundamental data. Results of trade-off analyses should be included, and it should also be explained why major decisions were made. Examples include cost plans, analysis, conclusions and results, test reports, decision memorandums, etc. Wayne Wymore of the University of Arizona recommends an excellent approach to system documentation (Wymore 1993).

Systems Engineering Standards

There are several industry and government standards for implementing systems engineering, including DoD, IEEE, EIA, and ANSI standards. The original standard is DoD MIL-STD-499A. It was to be replaced by MIL-STD-499B, but the Perry directive to reduce military standards was implemented while MIL-STD-499B was in the final stages of the approval process. The Electronics Industry Association subsequently released the MIL-STD-499B as standard EIA-632. The American National Standards Institute is working with the DoD, EIA, and IEEE to revise EIA-632 and issue it as an ANSI standard. Drafts of the ANSI standard are scheduled to be released in late 1996. The IEEE 1220 standard is an adaptation of MIL-STD-499B, and is to be used as a standard for commercial (non-DoD) products.

References and Bibliography

ASD Directorate of Systems Engineering (ASD/ENS) and Defense Systems Management College Systems Engineering Department (DSMC/FD-SE), Defense Systems Management College (1992). *Military Standard Systems Engineering MIL-STD-499B (2.0)* (draft not yet approved and subject to modification).

AT&T ATS Engineering Standard Process (1994). *Systems Engineering Process*, 2nd edition. AT&T Guilford Center Documentation Center.

Bahill, A.T., and Chapman, W.L. (1993). A tutorial on quality function deployment. *Engineering Management J,* 5(3):24-35.

Bharathan, K., Poe, G.L., and Bahill, A.T. (1995). Object-Oriented Systems Engineering. *Systems Engineering in the Global Market Place*, proceedings of the Fifth Annual Symposium of the National Council on Systems Engineering, July 22-26, St. Louis, MO.

Bicknell, K.D., and Bicknell, B.A. (1994). *The Road Map to Repeatable Success: Using QFD to Implement Changes.* Boca Raton: CRC Press.

Blanchard, B.S. (1991). *System Engineering Management.* John Wiley & Sons, Inc., New York.

Boardman, J. (1990). *Systems Engineering: An Introduction.* Prentice Hall, New York.

Booch, G. (1994). *Object-Oriented Analysis and Design.* Menlo Park, CA: Benjamin/Cummings Publishing Company, Inc.

Chapman, W.L., and Bahill, A.T. (1996). Design Modeling and Production, in *The Engineering Handbook,* (Ed.) R.C. Dorf, pp. 1732-1737. Boca Raton: CRC Press.

Chapman, W.L., Bahill, A.T., and Wymore, W. (1992). *Engineering Modeling and Design.* Boca Raton: CRC Press.

Defense Systems Management College *(1990). Systems Engineering Management Guide.* Washington, DC: USGPO.

Defense Systems Management College (1989). *Risk Management Concepts and Guidance*. Washington, DC: USGPO.

Deming, W.E. (1992). *Out of the Crisis.* Massachusetts Institute of Technology, Center for Advanced Engineering Study, Cambridge, Mass.

Department of Energy (1995). DOE Order 430.1, "Life-Cycle Asset Management (LCAM)," August 24, 1995.

Electronic Industries Association Engineering Department (1994). *EIA/IS-632 Interim Standard for Systems Engineering*, Washington, D.C.

Frantz, W.F. (1995). *The Impact of Systems Engineering on Quality and Schedule: Empirical Evidence.* The Boeing Company, Operations Technology, Seattle.

Funk, P.A., and Larson, D.L. (1994). Design features influencing thermal performance of solar box cookers, presented at the International Winter Meeting, paper No. 94-6546, American Society of Agricultural Engineers.

Gause, D.C., and Weinberg, G.M. (1990). *Are Your Lights On? How to Figure Out What the Problem Really Is*. New York: Dorset House Publishing.

Grady, J.O. (1993). *System Requirements Analysis.* New York: McGraw Hill Inc.

Grady, J.O. (1994). *System Integration.* Boca Raton: CRC Press.

Grady, J.O. (1995). *System Engineering Planning and Enterprise Identity.* Boca Raton: CRC Press.

Harvey, J.R., and Michalowski, S. (1993). *Nuclear Weapons Safety and Trident: Issues and Options.* A Report of the Center for International Security and Arms Control. Stanford University.

Hooks, I. (1994). *Writing Good Requirements*, "Systems Engineering: A Competitive Edge in a Changing World," proceedings of the Fourth Annual Symposium of the National Council on Systems Engineering, August 10-12, San Jose, CA.

IEEE (1993). *IEEE P1233 Guide For Developing System Requirements Specifications.* IEEE Standards Dept., NY.

IEEE (1994). *IEEE P1220 Standard for Systems Engineering.* IEEE Standards Dept., NY.

Jacobson, I., Ericsson, M., and Jacobson, A. (1995). *The Object Advantage: Business Process Reengineering with Object Technology.* New York: Addison-Wesley.

Kar, P., and Bailey, M. (1996). Characteristics of good requirements, Systems Engineering Practices & Tools, *Proceedings of the Sixth Annual International Symposium of the International Council on Systems Engineering*, Vol. II pp. 284-291, July 7-11, Boston.

Karnavas, W.J., Sanchez, P., and Bahill, A.T. (1993). Sensitivity analyses of continuous and discrete systems in the time and frequency domains. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-23: 488-501.

Katz, R. (1994). *Contemporary Logic Design*. Benjamin Cummings.

Kerzner, H. (1995). *Project Management: a Systems Approach to Planning, Scheduling, and Controlling.* New York: Van Nostrand Reinhold.

LaPlue, L., Garcia, R.A., and Rhodes, R. (1995). A rigorous method for formal requirements definition, *Systems Engineering in the Global Market Place*, Proceeding of the Fifth Annual Symposium of the National Council on Systems Engineering, July 22-26, St. Louis, pp. 401-406.

Latzko, W.J., and Saunders, D.M. (1995). *Four Days with Dr. Deming.* Reading, Mass: Addison-Wesley.

Lawton, R. (1993). *Creating a Customer-Centered Culture*. Milwaukee: ASQC.

Lockheed Missiles and Space Company, Inc. (1994). *SSD Systems Engineering Manual, Code 66,* Sunnyvale, California.

Martin, J. (1995). Requirements methodology: Shattering myths about requirements and the management thereof, *Systems Engineering in the Global Market Place*, Proceeding of the Fifth Annual Symposium of the National Council on Systems Engineering, July 22-26, St. Louis, pp. 473-480.

Martin, J. (1996). *Systems Engineering Guideline.* Boca Raton: CRC Press.

MIL-STD-1521B, Technical Reviews and Audits for Systems.

MIL-STD-499B (1993). Draft Military Standard for Systems Engineering, AFSC/EN. (Note: This standard was not signed by the Department of Defense. They said that government should not be in the business of writing standards and that they will adopt standards written by professional societies.)

National Council on Systems Engineering (1993). *Systems Engineering Process Activities, A "How To" Guide* (draft), San Francisco Bay Area Chapter, Sunnyvale, CA

National Council on Systems Engineering (1995). *Systems Engineering in the Global Market Place*, Proceedings of the Fifth Annual Symposium of the National Council on Systems Engineering, July 22-26, St. Louis.

PMI (Project Management Institute) (1996). *A Guide to the Project Management Body of Knowledge (LPMBOK).* Upper Darby, Pa.: Project Management Institute.

PMI (Project Management Institute) (1996). *A Guide to the Project Management Body of Knowledge (PMBOK).* Upper Darby, PA: Project Management Institute.

Rechtin, E. & Maier, M. (1996). *Systems Architecting.* Boca Raton: CRC Press.

Reilly, N.B. (1990). *Successful Systems Engineering for Engineers and Managers,* Van Nostrand Reinhold, New York.

Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., and Lorenson, W. (1991). *Object-Oriented Modeling and Design*. New York: Prentice Hall.

Sage, A.P. (1992). *Systems Engineering*. New York: John Wiley.

Sandia National Laboratories (1996). Interagency Engineering Procedure EP401063/E-Design Reviews, Albuquerque, NM.

Shand, R.M. (1994). User Manuals as Project Management Tools, *IEEE Transactions on Professional Communications*, 37, 75-80 and 123-142.

Shishko, R. (1995). *NASA Systems Engineering Handbook*, SP-6105. Headquarters: National Aeronautics and Space Administration.

Sommerville, I. (1989). *Software Engineering.* Reading, Mass: Addison-Wesley.

Szidarovszky, F., Gershon, M., and Duckstein, L. (1986). *Techniques for Multiobjective Decision Making in Systems Management.* Amsterdam: Elsevier Science Publishers.

Wymore, W. (1993). *Model-Based Systems Engineering.* Boca Raton: CRC Press.