CONF-970803--1 SAND96-2901C

The Requirements Discovery Process

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

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

ABSTRACT

Cost and schedule overruns are often caused by poor requirements that are produced by people who do not understand the requirements process. This paper provides a high-level overview of the requirements discovery process.

INTRODUCTION

No two systems are exactly alike in their requirements. However, there is a uniform and identifiable process for logically discovering the system requirements regardless of system purpose, size, or complexity (Grady, 1993: Bahill, Dean and Bentz, 1996). The purpose of this paper is to illuminate this process.

This paper presents the philosophy and terminology used by the New Mexico Weapons Systems Engineering Center at Sandia National Laboratories for discovering system requirements. Other organizations may use different procedures and terminology. However, we think a consensus is developing in the Systems Engineering community. It is hoped that this paper is consistent with that consensus. Like Systems Engineering in general, the statements in this paper are not dogmatic. Each statement has been rightfully violated many times (see for example Martin, 1995). However, these statements are generalizations of good engineering practices.

This paper only explains a part of the systems requirements process. Large projects should use a computer tool to help write, decompose and maintain system requirements. Many such computer tools are commercially available. Each project design team will select a specific tool and then provide training for it. Because such training is tool specific, this paper will not discuss such tools. Another part of the requirements process is modeling the proposed system. Dozens of tools are available; two recently popular ones are object-oriented design and functional decomposition (Bharathan, Poe, & Bahill, 1995). This paper does not discuss tools for modeling systems, because of the sheer magnitude of the task.

Requirements are the necessary attributes defined for a system before and during design. The customer's need is the ultimate system requirement from which all other requirements flow (Grady, 1993). In addition, requirements are statements that identify the essential needs of a system in order for it to have value and utility. Requirements may be derived or based upon interpretation of other stated requirements to assist in providing a common understanding of the desired characteristics of a system. Finally, requirements should state what the system is to do, but they should not specify how the system is to do it.

This work was supported by the United States Department of Energy under Contract DE-AC04-94AL 85000.
Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy.

DISCLAIMER

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, 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 or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

THE REQUIREMENTS DISCOVERY PROCESS

Requirements discovery is one subprocess of the Systems Design Process shown in Figure 1. Systems Design is a fractal process. It is applied at levels of greater and greater detail: It is applied to the system, then to the subsystems, then to the components, etc. It is applied to the system being designed and also to the enterprise in which the system will operate. This concept is shown in a poster that is available at: http://www.sie.arizona.edu/sysengr

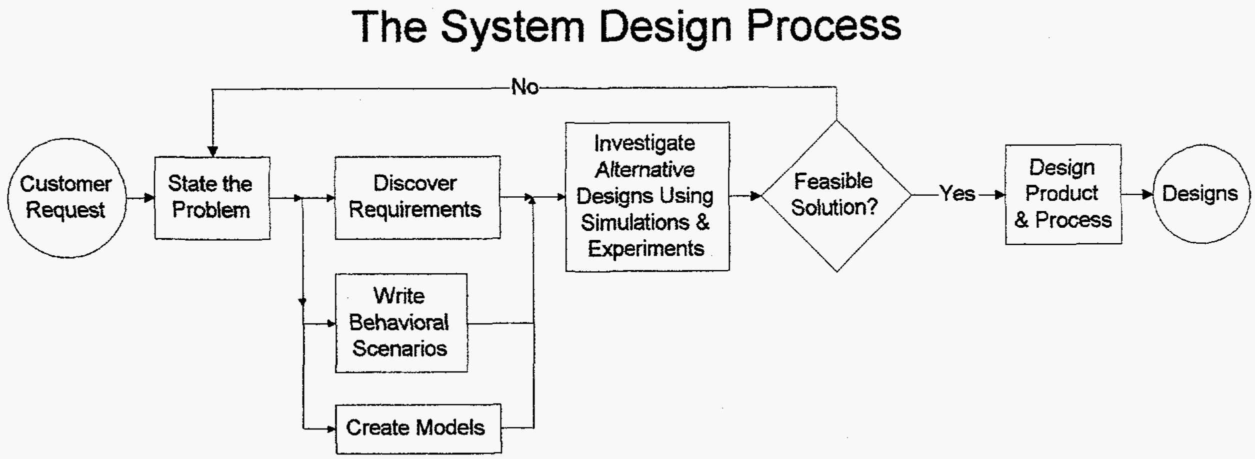

Figure 1. The system design process.

Identify Customers and Stakeholders

The first step in developing requirements is to identify the customer. The term *customer* includes anyone who has a right to impose requirements on the system. This includes end users, operators, bill payers, owners, regulatory agencies, victims, sponsors, etc. 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 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. It is imperative that the Systems Engineer understands this conflict and exposes it. The sponsor and user can then help trade off the development costs against the cost to use and maintain. Total life cycle costs are significantly larger than initial costs. For example, in one of their advertisements, Compaq proclaimed, "80% 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 $8,000.

Before writing a document you should consider who the audience is going to be. For a requirements document, the audience is the client and the designers. System requirements communicate the customer's needs to the technical community that will design and build the system, and therefore they must be understandable by both. One of the most difficult tasks in creating a system is communicating with all subgroups within both groups (IEEE P1233). The client and the designers have different backgrounds and needs. Wymore (1993) suggests two different documents for these two different groups: The Operational Need Document for the client and the System Requirements Document for the design engineers.

Understand the Customer's Needs

The system design 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 the customer's environment, discover the details, and explain them. Flexible designs and rapid prototyping 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.

It is the Systems Engineer's responsibility to ensure 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 the customer 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 needs; 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.

Define and State the Problem

What is the problem we are trying 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.

Early in the process, the customer frequently fails to recognize the scope or magnitude of the problem that is to be solved. The problem should not be described in terms of a perceived solution. It is imperative that the Systems Engineer help the customer develop a problem statement that is completely independent of solutions and specific technologies. Solutions and technologies are, of course, important; however, there is a proper place for them later in the Systems Engineering process. It is the Systems Engineer's responsibility to work with the customer, asking the questions necessary to develop a complete "picture" of the problem and its scope. The Air Force customer did not know that they wanted a stealth airplane until after the engineers showed that they could do it.

Figure 2 shows the requirements discovery process. This whole diagram is the "Discover Requirements" box of the System Design Process shown in Figure 1.

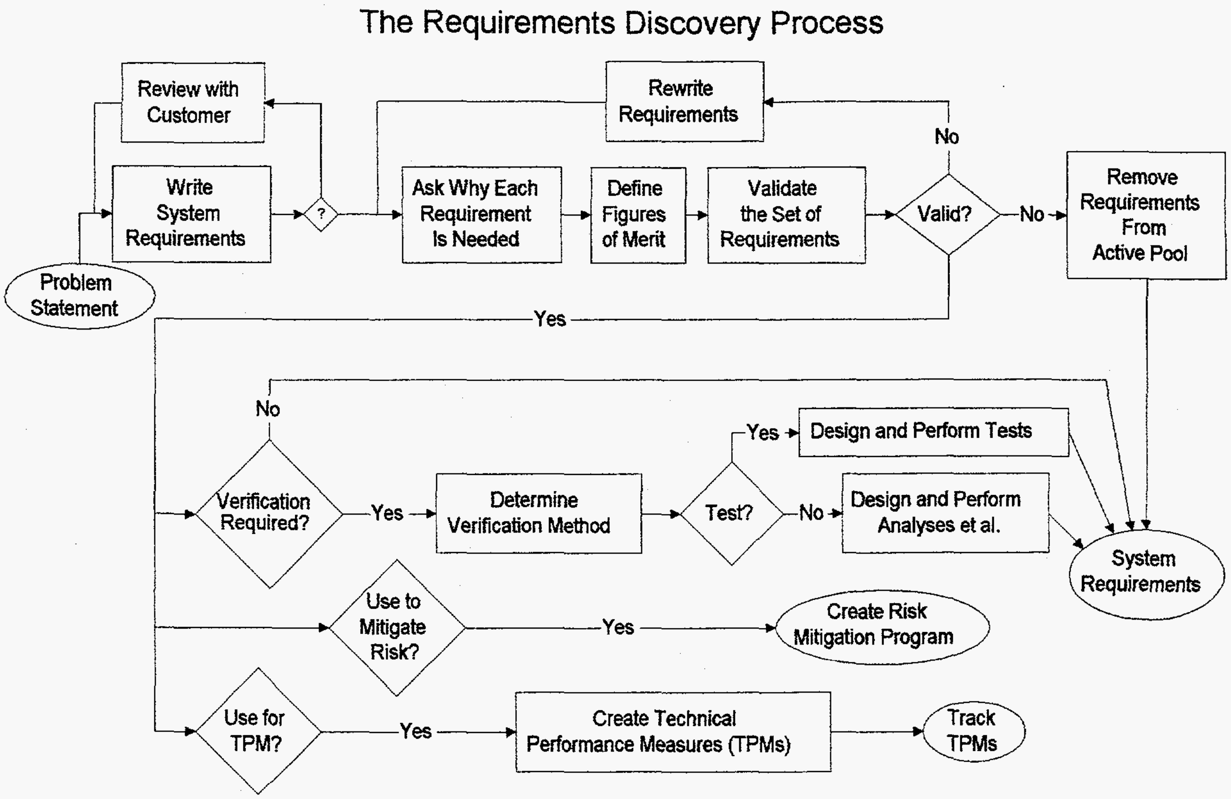
The Requirements Discovery Process
Review with Customer
Rewrite Requirements
Write System Requirements
?
Ask Why Each Requirement Is Needed
Define Figures of Merit
Validate the Set of Requirements
Valid?
No
No
Remove Requirements From Active Pool
Problem Statement
Yes
No
Yes
Design and Perform Tests
Verification Required?
Yes
Determine Verification Method
Test?
No
Design and Perform Analyses et al.
System Requirements
Use to Mitigate Risk?
Yes
Create Risk Mitigation Program
Use for TPM?
Yes
Create Technical Performance Measures (TPMs)
Track TPMs

Figure 2. The requirements discovery process.

Write System Requirements

The Systems Engineer must interact with the customer to write the system requirements. The Systems Engineer must involve the customer in the process of defining, clarifying, and prioritizing the requirements. It is prudent to involve users, bill payers, regulators, manufacturers, maintainers, and other key players in the process.

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

Although this makes it sound as if 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 re-examine the requirements and then re-examine the functions. Then we re-assess the requirements and again the functions, etc.

Review Requirements with Customer

The Systems Engineer must continually consult with the customer to ensure that the requirements are correct and complete. The customer should be satisfied that if these requirements are met, then the system will do what it really needs to do. All parties must agree to a way of measuring system performance to ensure that the system does what the customer wants it to do. The Systems Engineer and the customer should identify which requirements can be used as trade-off requirements. Sometimes the customer is not available for consultation. In such unfortunate situations, a surrogate customer will have to be used.

At these reviews 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.

Define Figures of Merit

Figures of merit are the criteria on 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). Suppose 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. The value of 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 the scoring functions 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.

Validate System Requirements

Validating requirements means ensuring that the set of requirements is consistent, that a real-world solution can be built that satisfies the requirements, and that it can be proven 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. Each requirement should be technically feasible, and fit within 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 test, demonstration, inspection, logical argument, modeling, or simulation.

Describe Verification Process

A critical element of the requirements development process is describing the tests, analysis or data that will be used to prove compliance of the final system with its requirements. Each test must explicitly link to a specific requirement; this will help expose untestable requirements. Describing the system tests informs the producers how the system will be tested, so that they know how they will be "graded." This process frequently uncovers overlooked requirements. At this time it may be useful to examine the following definitions.

Validating a System: 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: Ensuring that the set of requirements is consistent, that a real-world solution can be built that satisfies the requirements, and that it can be proven 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: Building the system right; 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 at the end of each phase, with itself and with the previous prototypes. In other words, it guarantees the honest and smooth transition from model to prototype to preproduction unit to production unit.

Verifying Requirements: 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 DoD-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 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 to agree on the definitions of verification and validation as these terms pertain to your system.

Define Technical Performance Measures

Technical performance measures (TPMs), or metrics, are used to track the progress of the design and manufacturing process. TPMs 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. It is hoped that as the design and manufacturing process progresses, the TPM values of the prototypes and preproduction units will come closer and closer to the goals.

Mitigate Risk

Identifying and mitigating project risk is the responsibility of management at all levels in the company. Each item that poses a threat to the cost, schedule or performance of the project must be identified and tracked. The following information should be recorded for each identified risk: name, description, type, origin, probability, severity, impact, identification number, identification date, work breakdown structure element number, risk mitigation plan, responsible team, needed resolution date, closure criteria, principal engineer, current status, date, signature of team leader. Forms useful in identifying and mitigating risk are given in Kerzner (1995) and Grady (1995).

Models (or computer simulations) are often used to reduce risk. Low risk portions of the system should be modeled at a high level of abstraction, whereas high risk portions should be modeled with fine resolution.

Review System Requirements

The system requirements must be reviewed with the customer many times. At a minimum requirements should be reviewed at the end of the modeling phase, after testing the prototypes, before commencement of production, and after testing production units.

The main objectives of these reviews are to find missing requirements, eliminate unneeded requirements, ensure that the requirements have been met, and verify that the system satisfies customer needs. At these reviews, 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. These reviews should be formal. The results and conclusions of the reviews should be documented. The Systems Engineer is responsible for initiating and conducting these reviews.

REFERENCES

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

Bahill, A.T., Dean, F.F. & Bentz, A. (1996). Discovering System Requirements, SAND XXX Sandia National Laboratories.

Bharathan, K., Poe, G.L. & 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. XXX pages

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

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.

Hooks, I. (1994). *Writing Good Requirements*, Proceedings NCOSE, pp. 197-203.

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

Kar, P. & 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.

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

LaPlue, L., Garcia, R.A., & 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.

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.

Shishko, R. (1995). *NASA Systems Engineering Handbook*, SP-6105.

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

BIOGRAPHY

A. Terry Bahill is a Professor of Systems Engineering at the University of Arizona in Tucson. He received his Ph.D. in electrical engineering and computer science from the University of California, Berkeley, in 1975. He holds a U.S. patent for the Bat Chooser™, a system that computes the Ideal Bat Weight™ for individual batters. He is Editor of the CRC Press Book Series on Systems Engineering, and is a Fellow of The Institute of Electrical and Electronics Engineers (IEEE).

Frank F. Dean is the Manager of the New Mexico Weapon Development Operations Department at Sandia National Laboratories in Albuquerque. Prior to that, as a Distinguished Member of the technical staff, he directed Systems Engineering studies and projects. He has a B.S. and an M.S. in mathematics from Illinois Institute of Technology, Chicago. He is a member of NCOSE and PMI.