1.6 REQUIREMENTS

Requirements for a system address the needs and objectives of the stakeholders. Just as there is a hierarchy associated with the physical components of the system, there is a hierarchy of requirements. At the top of the hierarchy are mission requirements, which relate to needs associated with missions or activities that are important to one or more groups of stakeholders. These *mission requirements* typically involve the interaction of several systems, one or more of which include individuals or groups of people and are therefore stated in the context of the operation of the system in question with these other systems, called the meta-system or supersystem or system of systems. Mission requirements represent stakeholder preferences for the increased ability to perform their activities with the introduction of the system in question at a lower cost and in a faster time than the existing capability.

Stakeholders' requirements are statements by the stakeholders about the system's capabilities that define the constraints and performance parameters within which the system is to be designed. Systems engineers take these high-level, stakeholders' requirements and derive a consistent set of more detailed engineering statements of requirements as the design progresses. For the purposes of this introduction, requirements are divided into constraints and performance indices. Some constraints are simple, for example, the system must be painted with a specific shade of green. Other constraints are the minimally acceptable level associated with a performance requirement. A performance requirement defines a desired direction of performance associated with an objective of the stakeholders for the system. For an elevator system (which is used throughout this book), a performance requirement might be to minimize passengers' waiting time. For any performance requirement, there must also be a minimum acceptable performance constraint or threshold and a design goal associated with the index; this threshold dictates that no matter how wonderful a design's performance is on other objectives, performance below this threshold on this requirement makes the design unacceptable. This is a very strong statement of needs, and so minimal acceptable thresholds must be established very carefully.

Every major organization, governmental or commercial, has established its own guidelines for system or product development. The names and organizations of the several requirements documents vary somewhat but cover similar material. Table 1.6

TABLE 1.6 Typical Requirements Documents

Document Titles	Document Contents
Problem Situation or Mission Element Need Statement, and Systems Engineering Management Plan (SEMP)	• Definition of stakeholders and their relationships • Stakeholders' description of the problem and its context • Description of the current system • Definition of mission requirements • Definition of the systems engineering management structure and support tools for developing the system
Stakeholders' Need or Stakeholders' Requirements Document (StkhldrsRD)	• Definition of the problem needing solution by the system (including the context and external systems with which the system must interact) • Definition of the operational concept on which the system will be based • Creation of the structure for defining requirements • Description of the requirements in the stakeholders' language in great breadth but little depth • Trace of every requirement to a recorded statement or opinion of the stakeholders • Description of trade-offs between performance requirements, including cost and operational effectiveness
System Requirements Document (SysRD)	• Restatement of the operational concept on which the system will be based • Definition of the external systems in engineering terms • Restatement of the operational requirements in engineering language • Trace of every requirement to the previous document • Justification of engineering version of the requirements in terms of analyses, expert opinions, stakeholder meetings • Description of test plan for each requirement
System Requirements Validation Document	• Documents analyses to show that the requirements in the SysRD are consistent, complete, and correct, to the degree possible • Demonstrates that there is at least one feasible solution to the design problem as defined in the SysRD

summarizes the common major requirements documents that are produced during the beginning of the design phase. The Problem Statement (or Mission Element Need Statement in the military) gets the process rolling and identifies a problem for which a solution in the form of a system (new or improved) is needed. This document supports and documents a decision-making process to start a system development effort. The Systems Engineering Management Plan (SEMP) then defines the systems engineering development system.

Stakeholders' requirements are found in the Stakeholders' Requirements Document (StkhldrsRD). This document is produced with or by the stakeholders and is written in their language(s). Systems engineers need to be involved in a substantial way in this activity, although not all systems engineers share this view. Experience has shown that if this document is left to the stakeholders, the document will be very incomplete. The systems engineers can play a major facilitation role among the various groups of stakeholders as well as bring an assortment of tools to bear on a difficult problem, the creation of this document. These tools (a major focus of this book) ensure a greater completeness and consistency. The methods and tools presented here are equally applicable *in* the rest of the systems engineering process.

The systems engineer then begins restating and "deriving" requirements in engineering terms, called *system* requirements, so that the systems engineering design problem can be solved. This derivation of the StkhldrsRD becomes the Systems Requirements Document (SysRD).

It is critical that the requirements in all of these documents address "what" and "how well" the system must perform certain tasks. Requirements do not provide solutions but rather define the problem to be solved.

The Systems Requirements Validation Document defines requirements associated with the verification, validation, and acceptance of the system during integration. These requirements are high-level requirements that state the needs of the stakeholders for qualifying the design of the system. These requirements form the basis of the problem definition for creating the qualification system that will be used during integration. In addition to defining the high-level qualification requirements, this document should demonstrate that if the systems engineering process continues, an acceptable solution is possible. Unfortunately, this "existence proof" of a feasible solution is seldom produced in practice, leading to a major downfall of many systems engineering efforts. Namely, the realization that many months (or years) later not all of the requirements can be satisfied, and the stakeholders must relax the requirements that the engineers promised could be met.

Systems engineers have always desired to demonstrate the importance of requirements and getting the requirements right, for example, complete, consistent, and correct. In the mid-1970s, three organizations (GTE [Daly, 1977], IBM [Fagan, 1974], and TRW [Boehm, 1976]) conducted independent studies of software projects. These studies addressed the relative cost to fix a problem based upon where in the system cycle the problem was found. Boehm [1981] and Davis [1990, p. 25] compared the results of the three studies (see the first row of Table 1.7). The costs have been normalized so that the relative cost to repair an average problem found in the coding phase is 10 units. These results stood for 20 years. The next eight rows of

TABLE 1.7 Comparison of the Relative Cost to Fix Software in Various Life Cycle Phases

Source	Phase Requirements Issue Found			
	Requirements	Design	*Code*	Test
Boehm (1981)	1	5	*10*	50
Hoffman (2001)	1	3	*5*	37
Cigital (2003)	1	3	*7*	51
Rothman (2000)		5	*33*	75
Rothman Case B (2000)			*10*	40
Rothman Case C			*10*	40
Rothman (2002)	1	20	*45*	250
Pavlina (2003)	1	10	*100*	1000
McGibbon (2003)		5		50
Mean	1	7.3	*25.6*	177
Median	1	5	*10*	50.5

Table 1.7 show results from recent studies, summarized in Haskins et al. [2004]. As can be seen, the results have held up well. Getting the requirements right is a very difficult task, and therefore a task that is fraught with errors. An error that is caught during requirements development can be fixed for about 10% of the cost associated with an error caught during coding. Errors caught during maintenance in the operation of the system cost about 20 times that of an error caught during coding and 200 times the cost of an error caught during requirements development. Unfortunately, many of these errors are not caught until late in the life cycle, causing the expenditure of significant money.

6.2 REQUIREMENTS

Many authors have defined the term *requirement.* The list below provides several definitions that highlight key concepts (the italics are the author's).

Sailor [1990]: Identifiable capabilities expressed as *performance measurables* of functions that the system must possess to meet the mission objectives.

MIL-STD 499B [Military Standard, 1993]: Identifies the *accomplishment levels* needed to achieve specific objectives.

Chambers and Manos [1992]: The attributes of the final design that must be a part of any *acceptable solution to the design problem.*

Grady [1993]: An *essential attribute* for a system or an element of a system, *coupled by a relation statement with value and units information for the attribute.*

Davis [2005]: An *externally observable characteristic of a desired* system.

The requirements for a system set up standards and measurement tools for judging the success of the system design. These requirements should be viewed hierarchically. At the top are mission-level requirements that establish how the stakeholders will benefit by introducing the system in question into the super-system of the system. These *mission requirements* relate to objectives of the stakeholders that are defined in the context of the super system, not the system itself. For example, Boeing identified two primary mission requirements when starting on the Boeing 777 commercial aircraft: trip cost per seat and total trip cost. Each airline company that purchases a Boeing 777 is the meta-system that most influences an aircraft company during the development phase.

Stakeholders' requirements are developed next in the context of these mission requirements and should focus on the boundary of the system. If the stakeholders' requirements are defined internally to the system, the risk of having design statements embedded in the requirements goes up substantially. A major emphasis of this chapter is that the stakeholders' requirements should be as design independent as possible. Boeing's stakeholders' requirements for the Boeing 777 included such topics as liftable weight of the aircraft at specified conditions, the empty weight of the aircraft,

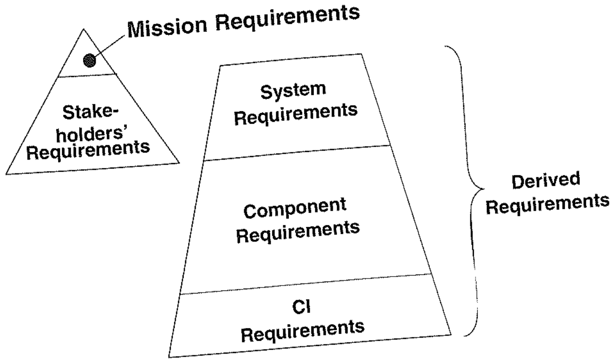

FIGURE 6.1 Requirements hierarchies.

the drag force on the aircraft for certain specified flight conditions, and the fuel consumption of the aircraft at certain specified flight conditions.

As discussed in Chapter 1, *system requirements* are a translation (or derivation) of the stakeholders' requirements into engineering terminology. Once this translation occurs, the derivation process of requirements continues. Recall from Chapter 1 that the goal of the design process is to create a system specification that can be developed into specifications for the system's components, which are then segmented into specifications for the system configuration items (CIs). As a result, the design process creates two hierarchies of requirements as shown in Figure 6.1.

The stakeholders' requirements are produced in conjunction with the stakeholders of the system, based upon the operational needs of these stakeholders. Some systems engineers believe the systems engineering process begins when the StkhldrsRD arrives; however, the position taken here and supported by Pragmatic Principle [DeFoe, 1993] of the International Council on Systems Engineering (INCOSE) is that the systems engineers must be involved with the stakeholders to have any hope of producing a useful StkhldrsRD, note italicized items. In fact, the process described in this chapter is focused on methods and models for developing a valid and complete StkhldrsRD.

The Systems Requirements Document (SysRD), which is derived from the StkhldrsRD, is a translation from the language of stakeholders to the language of engineers. The system's requirements are traced directly from the stakeholders' requirements.

Note the term *stakeholder* is used in the above discussion in place of the more common term *user*. This is to emphasize the fact that there are usually multiple categories of users of a system: owner and/or bill payer, developer, producer or manufacturer, tester, deployer, trainer, operator, user, victim, maintainer, sustainer, product improver, and decommissioner. Each stakeholder has a significantly different perspective of the system and the system's requirements. If one perspective is singled out as the only appropriate one, the developers of the system will miss key information, and the system will be viewed negatively or as a failure from the other perspectives.

The systems engineering process for creating a system design is decision rich. That is, the systems engineer is searching via a great deal of analysis and experience to find a very good (optimum is usually not possible to determine) solution that satisfies all of the mandatory requirements of the stakeholders and delivers as much performance as possible within the guidelines of cost and schedule.

This *search process* involves making many decisions about the system's physical character (or resources) and allocations of functions to resources that are usually only revisited if absolutely necessary. This search process occurs as the top-down onion-peeling process of systems engineering occurs. Figure 6.1 shows derived requirements at the component level (which may be several layers of the onion) and the CI (or bottom) level. Chapters 7 through 10 will describe this process of architecture development and creation of appropriate derived requirements, supported by analysis and judgment. To continue the story of the Boeing 777, Boeing created requirements for a major subsystem of the 777 – the engine. These derived requirements for the engine included the weight of the engine (derived from the weight of the empty aircraft), the thrust of the engine at specified conditions (derived from the liftable weight of the aircraft), the drag of the engine at specified conditions (derived from the drag of the aircraft), and the fuel consumption of the engine at specified conditions (derived from the fuel consumption of the aircraft).

A major impediment to this design process being successful is the overconstraint of the solution space by the stakeholders' requirements. The systems engineers job is to work with the stakeholders to define the stakeholders' requirements so as to make sure that there is significant design freedom within these requirements and that many feasible designs exist. Stakeholders and (all too often) engineers are willing to constrain the requirements space very tightly without fully understanding or appreciating the potential value of the design options that they are eliminating. The stakeholders' requirements process defined in this chapter takes explicit account of this need to have and define a large tradable region in design space for the systems engineers to search with quantitative techniques utilizing the priorities of the stakeholders.

TABLE 6.1 Roles and Responsibilities during Requirements Generation

Who has the right to have a stakeholders' requirement?	Any individual/organization with a need involved in the development (design and qualification), production, deployment, training, operation, maintenance, support, refinement, decommissioning of **and payment** for the system.
What does one call a requirer?	Customer or stakeholder.
Who must respond to the requirer(s) having a requirement and how?	System's **requirements team**, a collection of *stakeholders* and *systems engineers*. Response is acceptance, request for clarification, or rejection.
By what criteria does the Systems Requirements Team respond?	This team establishes the external systems diagram and fundamental objectives hierarchy of the system, and then determines if the requirement fits within the scope of the system's boundary and fundamental objective. Stakeholders' requirements also have to be assessed for the proper level of abstraction. A requirement should not be too strategic (mission-oriented) or means (or solution) oriented.
How does one know that the requirement is "right"?	There is no right or wrong, only acceptable or unacceptable at this time. Over time, some of the stakeholders' requirements will change.
How are these requirements conveyed to the people who get involved once a requirer has enunciated a requirement?	The system's requirements team documents the collection of stakeholders' requirements. This Stakeholders' Requirements Document (StkhldrsRD) is distributed to the stakeholders and systems engineers. Included in this document is a discussion of the operational concept of the system and the external systems and context associated with the system, that is, how each stakeholder expects to interact with the system. By reviewing the Stakeholders' Requirements Document, each stakeholder can see how the requirement she or he suggested fits into the envisioned operation of the system, and can judge whether this vision makes sense from her/his perspective.
What does the Systems Requirements Team do next?	The system's stakeholders' requirements team remains active throughout the system's life cycle. During design there will be many occasions when the system's stakeholders' requirements must be reviewed and modified. These occasions will diminish in frequency once the system is deployed, but the requirements process is still critical as requirements changes and system modifications are envisioned, agreed to, developed, and fielded.