Previous Page

different types of systems, and there are some variations in terms of similarities and dissimilarities. To provide some insight into the variety of systems in existence, a partial listing of categories follows:[8]

Natural and man-made systems. Natural systems include those that came into being through natural processes. Examples include a river system and an energy system. Man-made systems are those that have been developed by human beings, the results of which include a wide variety of capabilities. As all man-made systems are embedded in the natural world, there are numerous interfaces that must be addressed. For instance, the development and construction of a hydroelectric power system located on a river system creates impacts on both sides of the spectrum, and it is essential that the systems approach involving both the natural and man-made segments of this overall capability be implemented.

Physical and conceptual systems. Physical systems are those made up of real components occupying space. On the other hand, conceptual systems can be an organization of ideas, a set of specifications and plans, a series of abstract concepts, and so on. Conceptual systems often lead directly into the development of physical systems, and there is a certain degree of commonality in terms of the type of processes employed. Again, the interfaces may be many, and there is a need to address these elements in the context of a higher-level system in the overall hierarchy.

Static and dynamic systems. Static systems include those having structure, but without activity (as viewed in a relatively short period of time). A highway bridge and a warehouse are examples. A dynamic system is one that combines structural components with activity. An example is a production capability combining a manufacturing facility, capital equipment, utilities, conveyors, workers, transportation vehicles, data, software, managers, and so on. Although there may be specific points in time when all system components are static in nature, the successful accomplishment of system objectives does require activity and the dynamic aspects of system operation do prevail throughout a given scenario.

Closed and open-loop systems. A closed system is one that is relatively self-contained and does not significantly interact with its environment. The environment provides the medium in which the system operates; however, the impact is minimal. A chemical equilibrium process and an electrical circuit (with a built-in feedback and control loop) are examples. Conversely, open-loop systems interact with their environments. Boundaries are crossed (through the flow of information, energy, and/or matter), and there are numerous interactions both among the various system components and up and down the overall system hierarchical structure. A system/product logistic support capability is an example.

[8]This categorization follows the general form presented in B. Blanchard and W. Fabrycky, *Systems Engineering and Analysis*, 3d ed. (Upper Saddle River, NJ: Prentice-Hall, 1998). These categories represent only a few of those that could be described.

These categories are presented to stimulate further thought relative to the definition of a system. It is not easy to classify a system as being either closed or open, and the precise relationships between natural and man-made systems may not be well defined. However, the objective here is to gain a greater appreciation for the many different considerations required in dealing with system engineering and its process. This text tends to deal mainly with *man-made* systems that are *physical* by nature, *dynamic* in operation, and of the *open-loop* variety.

The systems addressed herein may include a wide variety of *functional* entities. There are transportation systems, communication systems, manufacturing systems, information processing systems, and so on, as indicated in Figure 1.6. In each instance, there are *inputs,* there are *outputs,* there are external *constraints* imposed on

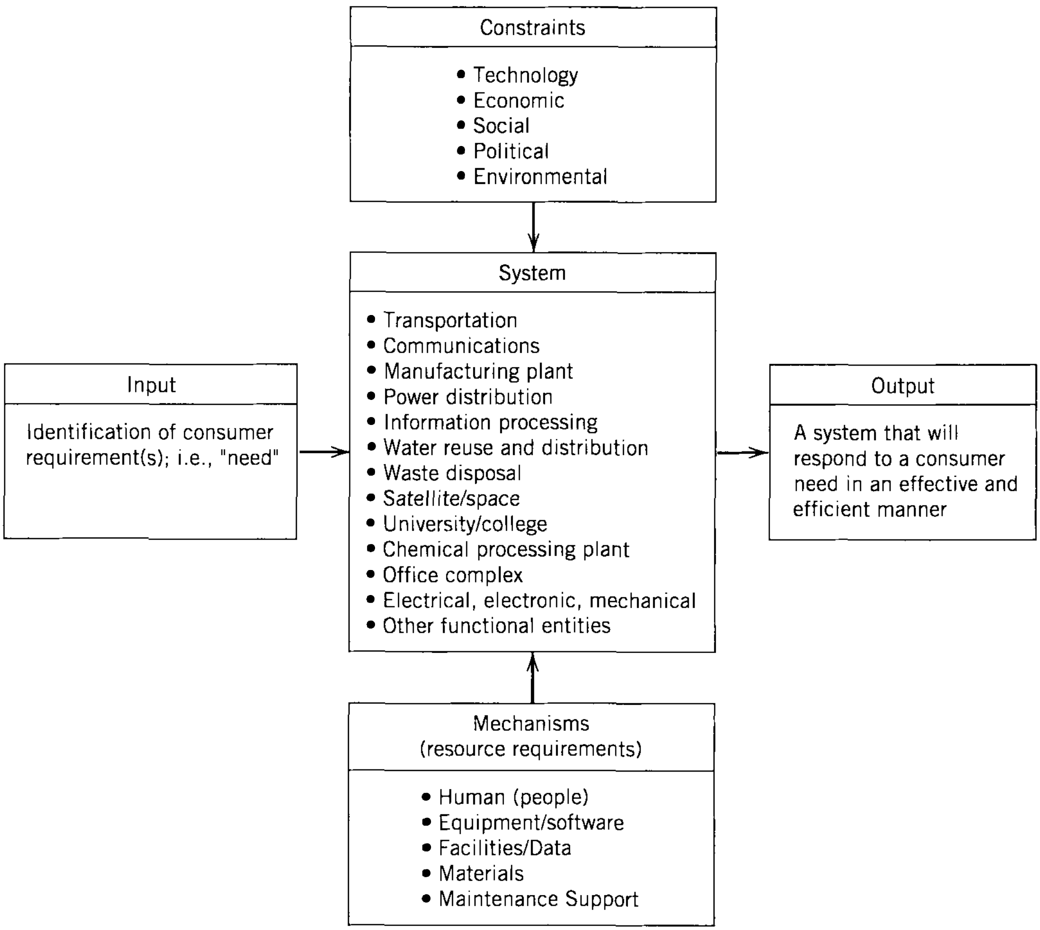


Figure 1.6 The system.

a system, and there are the required *mechanisms* necessary to realize the desired results. Within the framework of the "system," there are products and processes.

A system is composed of many different elements, including those that are directly utilized in the actual accomplishment of a mission (e.g., prime equipment, operating software, operating personnel) and the elements of maintenance support (e.g., maintenance personnel, test equipment, facilities, spares and repair parts). Although the support infrastructure is not often considered an element of a system per se, the system may not be able to complete its designated function in its absence. Thus, the support infrastructure is addressed as a major system element, presented in the context of the system life cycle. Figure 1.7 identifies the major elements of a system.

A system may be contained within some form of *hierarchy,* as shown in Figure 1.8. The question is, Are we addressing a *transportation* system, including many different types of vehicles (e.g., automotive, rail), a *vehicular* system, including many automobiles, or an *automobile,* with driver and associated support? It is not uncommon for a group of individuals to get together to discuss a particular issue, each having a different perception as to the "system" being addressed.

In regard to the systems shown in Figure 1.8, there are "upward" and "downward" impacts that must be considered. Decisions pertaining to the *vehicular* system may have an upward impact on the transportation system, and certainly will have a downward impact on the automobile. For example, the maintenance support infrastructure for the vehicular system may have to be compatible with the maintenance concept specified for the transportation system. In addition, this concept may also be imposed as a constraint in the design of the automobile. In any event, these interaction effects may be significant and must be addressed.

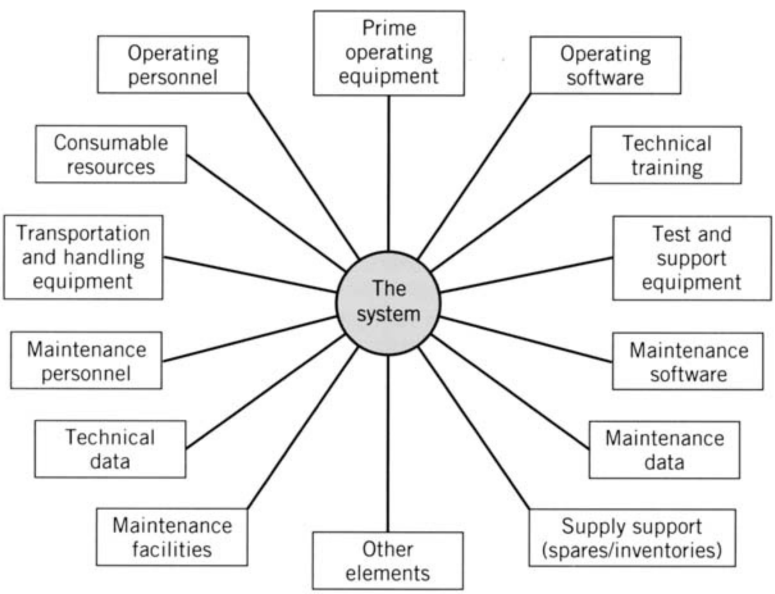


Figure 1.7 The major elements of a system.

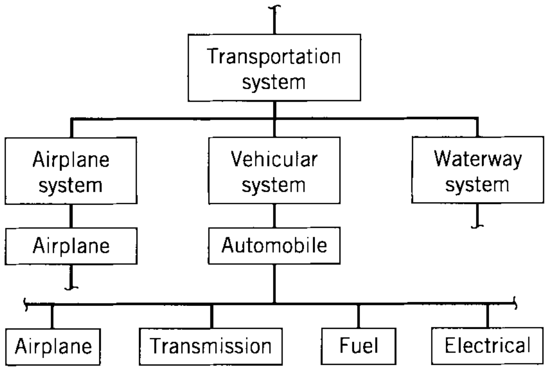


Figure 1.8 The hierarchy of systems.

1.2.3 The System Life Cycle

As shown in Figure 1.9, the *life cycle* includes the entire spectrum of activity for a given system, commencing with the identification of need and extending through system design and development, production and/or construction, operational use and sustaining maintenance and support, and system retirement and material disposal. As the activities in each phase interact with the activities in other phases, it is essential to consider the overall life cycle in addressing system-level issues, particularly if one is to properly assess the risks associated with the decision-making process throughout.

Although the life-cycle phases conveyed in Figure 1.9 reflect a more generic sequential approach, the specific activities (and the duration of each) may vary somewhat, depending on the nature, complexity, and purpose of the system. Needs may change, obsolescence may occur, and the levels of activity may be different, depending on the type of system and where it fits in the overall hierarchical structure of activities and events. In addition, the various phases of activity may overlap somewhat, as illustrated in the two examples presented in Figure 1.10.

Figure 1.10 shows how an airplane, a ground transportation vehicle, or an electronic device may progress through conceptual design, preliminary design, detail de-

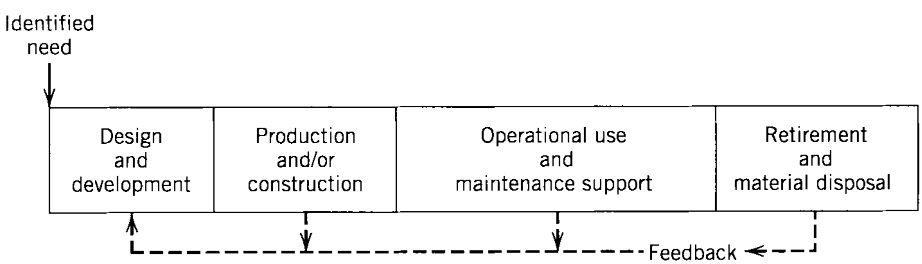


Figure 1.9 The system life cycle.

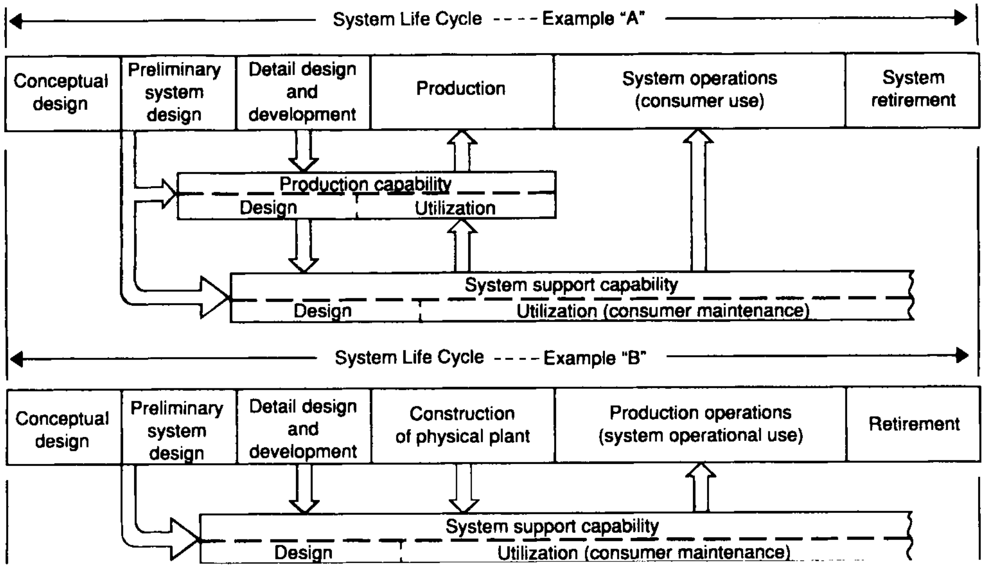


Figure 1.10 Examples of system life cycles.

sign, production, and so on, as reflected through the series of activities for Example "A." When this example is evaluated further, the top row of activities is applicable to those elements of the system that relate directly to the accomplishment of the mission (e.g., an automobile). At the same time, there are two closely related life cycles of activity that must also be considered. The design, construction, and operation of the production capability, which can have a significant impact on the operations of the prime elements of the system, should be addressed concurrently along with the system maintenance and support activity. Further, these activities must be addressed early during the conceptual and preliminary design of those prime elements represented by the top row. Although all of these activities may be presented through an illustrated single flow, as conveyed in Figure 1.9, the breakout in Figure 1.10 is intended to emphasize the importance of addressing *all* aspects of the total system process and the various interactions that may occur.

Example "B" in Figure 1.10 is presented to cover the major phases associated with a manufacturing plant, a chemical processing plant, or a satellite ground tracking facility, where the construction of a "one-of-a-kind" system configuration is required. Again, the maintenance and support capability is identified separately in order to indicate degree of importance and to suggest that there are many interaction effects that must be considered.

Although there may be variations in approaches, the nomenclature used, the duration of different phases, and so on, it is still appropriate that systems be viewed in terms of their respective life cycles. The past is replete with examples in which major decisions have been made in the early stages of system acquisition based on the "short term" only. In other words, in the design and development of a new system, the consideration for *production/construction* and/or *maintenance and support* of that system was inadequate. These activities were considered later, and, in many instances, the consequences of this "after-the-fact" approach were costly, as discussed in Section 1.1.[9]

1.2.4 Definition of *System Engineering*

System engineering may be defined in a number of ways, depending on one's background and personal experience. The inaugural issue of *Systems Engineering*, published by the International Council on Systems Engineering (INCOSE), describes a variety of approaches.[10] However, there is a basic theme throughout that deals with a top-down process, which is life-cycle oriented, involving the integration of functions, activities, and organizations.

More recently, the Fellows of INCOSE developed a consensus definition as follows:

> System engineering is an engineering discipline whose responsibility is to create and execute an interdisciplinary process to ensure that the customer and stakeholder's needs are satisfied in a high-quality, trustworthy, and cost and schedule efficient manner throughout a system's entire life cycle. This process is usually comprised of the following seven tasks: *S*tate the problem; *I*nvestigate alternatives; *M*odel the system; *I*ntegrate; *L*aunch the system; *A*ssess performance; and *R*e-evaluate (*SIMILAR*). The systems engineering process is not sequential. The functions are performed in a parallel and iterative manner.[11]

The Department of Defense (DOD) defines system engineering as:

> An approach to translate approved operational needs and requirements into operationally suitable blocks of systems. The approach shall consist of a top-down, iterative process of requirements analysis, functional analysis and allocation, design synthesis and verification, and system analysis and control. Systems engineering shall permeate design, manufacturing, test and evaluation, and support of the product. Systems engineering principles shall influence the balance between performance, risk, cost, and schedule.

[9]Referring to Figure 1.10, the emphasis as presented addresses the three life cycles, including (1) the life cycle pertaining to the mission-related elements of the system, (2) the production capability, and (3) the maintenance and support capability. There is a fourth life cycle that is equally important but not highlighted in the figure, and this pertains to the design and implementation of the *retirement and material recycling/disposal* capability. One needs to design for producibility, design for supportability/serviceability, and design for recyclability and disposability.

[10]Inaugural Issue, *Systems Engineering*, Journal of the International Council on Systems Engineering, Vol. 1, no. 1, (July/September 1994).

[11]"A Guide to the Systems Engineering Body of Knowledge (SEBok)—Introduction." *INSIGHT* Vol. 5, no. 1, published by INCOSE, April 2002.

More specifically:

> The systems engineering process shall:[12]
>
> 1. Transform approved operational needs and requirements into an integrated system design solution through concurrent consideration of all life-cycle needs (i.e., development, manufacturing, test and evaluation, deployment, operations, support, training, and disposal; and
> 2. Ensure the interoperability and integration of all operational, functional, and physical interfaces. Ensure that system definition and design reflect the requirements for all system elements: hardware, software, facilities, people, and data; and
> 3. Characterize and manage technical risks.
>
> The key systems engineering activities that shall be performed are requirements analysis, functional analysis/allocation, design synthesis and verification, and system analysis and control.

A slightly different definition (preferred by the author) states that system engineering is:

> The application of scientific and engineering efforts to: (1) transform an operational need into a description of system performance parameters and a system configuration through the use of an iterative process of definition, synthesis, analysis, design, test and evaluation, and validation; (2) integrate related technical parameters and ensure the compatibility of all physical, functional, and program interfaces in a manner the optimizes the total definition and design; and (3) integrate reliability, maintainability, usability (human factors), safety, producibility, supportability (serviceability), disposability, and other such factors into a total engineering effort to meet cost, schedule, and technical performance objectives.[13]

Basically, system engineering is *good* engineering with certain designated areas of emphasis, a few of which are noted as follows:

1. A top-down approach is required, viewing the system as a *whole.* Although engineering activities in the past have very adequately covered the design of various system components, the necessary *overview* and an understanding of how these components effectively fit together has not always been present.

2. A *life-cycle* orientation is required, addressing all phases to include system design and development, production and/or construction, distribution, operation, sustaining maintenance and support, and retirement and material phaseout. Emphasis in

[12]Department of Defense Regulation 5000.2R, "Mandatory Procedures for Major Defense Acquisition Programs (MDAPS) and Major Automated Information System (MAIS) Acquisition Programs," Chapter 5, Paragraph C5.2, April 5, 2002.

[13]This is a slightly modified version of the definition of systems engineering that was included in the original version of MIL-STD-499, "Systems Engineering" (Washington, DC: Department of Defense, July 1969).

the past has been placed primarily on system design activities, with little (if any) consideration given to their impact on production, operations, support, and disposal.

3. A better and more complete effort is required relative to the initial *identification of system requirements,* relating these requirements to specific design goals, the development of appropriate design criteria, and the follow-on analysis effort to ensure the effectiveness of early decision making in the design process. In the past, the early "front-end" analysis effort, as applied to many new systems, has been minimal. This, in turn, has required greater individual design efforts downstream in the life cycle, many of which are not well integrated with other design activities and require modification later on.

4. An *interdisciplinary* effort (or team approach) is required throughout the system design and development process to ensure that all design objectives are met in an effective manner. This necessitates a complete understanding of the many different design disciplines and their interrelationships, particularly for large projects.

Inherent within the system engineering process is a "top-down/bottom-up" development approach, as illustrated in Figure 1.11. The emphasis throughout this text is the shaded area; that is, the front-end requirements analysis activity. Traditionally, the

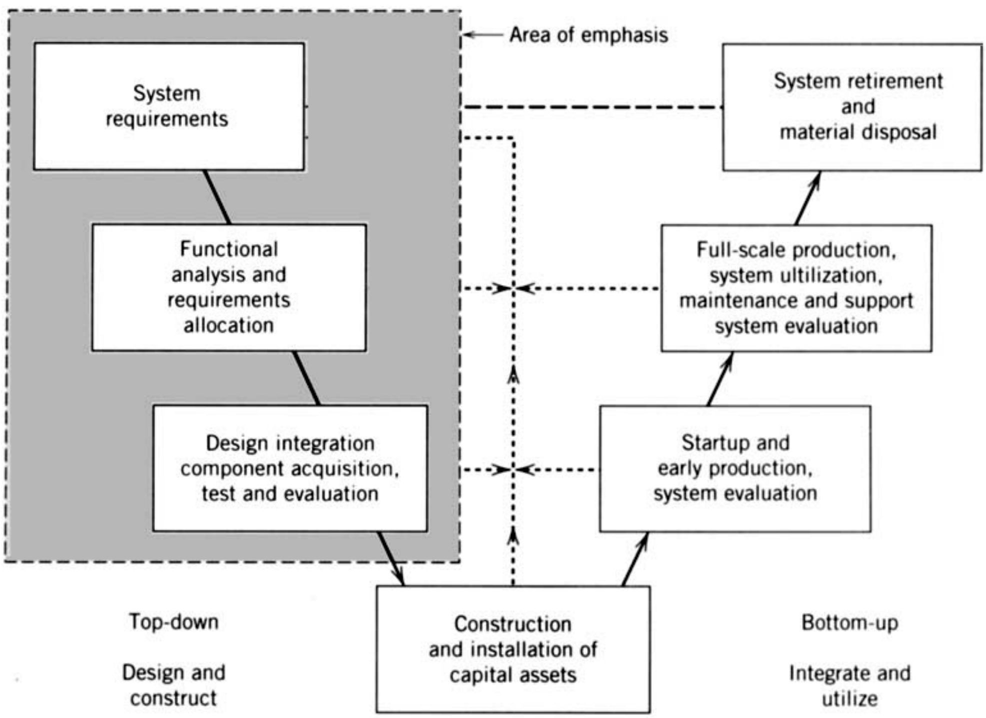


Figure 1.11 Top-down/bottom-up system development process.

requirements have not been well defined from the beginning, resulting in some rather extensive and costly efforts during the final integration and test activity.

Figure 1.12 presents an extension of the basic life-cycle phases shown in Figure 1.9, describing typical activities that occur in each phase, identifying various configuration *baselines* that should be established as one progresses from the initial identification of need to the development of a fully operational system, and including the iterative steps inherent within the system engineering process. Although the presentation of information in the figure may lead the reader to believe that the system acquisition process is very complex, the objective is to show this as a *process* in itself. Every time there is a newly identified *need,* there are certain steps through which one should evolve—that is, conceptual design, preliminary design, and so on. Even if the effort (in terms of the resources expended) is minimal, there is still the requirement for design activities at the *system level* and on down. The objective is to view these phase-related activities as a process within itself and to identify the baselines where the design evolves from one level of definition to the next. "Tailoring" the activities in Figure 1.12 to the system in question is essential for the successful implementation of the system engineering process.

The system engineering process per se includes the basic steps of requirements analysis, functional analysis, requirements allocation, design optimization and trade-offs, synthesis, evaluation, and so on (refer to blocks 0.1, 0.2, 0.3, etc., in Figure 1.12). These steps are *iterative* by nature, evolving from the system-level definition to the subsystem level, detailed level, and on down to the component. Further, these steps are not necessarily accomplished in a serial sequence, but are interactive with the appropriate *feedback* provisions at each step in the process. Although the requirements may vary somewhat from program to program, the purpose of this figure is to provide a baseline for future reference as different topics are presented throughout this text.

In block 0.2 (Figure 1.12), the accomplishment of the *functional analysis* will lead to the identification of resources in terms of the need for hardware, software, people, facilities, data, and the like. The functional analysis identifies the "WHATs" from a requirements perspective, and this leads to the accomplishment of trade-offs and the description of the "HOWs" pertaining to the completion of functions. Figure 1.13 illustrates the identification of hardware, software, and human requirements (from the functional analysis), and the subsequent life cycles associated with the development of each of these resources. One of the goals of system engineering is to "justify" these resource requirements through a top-down approach and to ensure the proper development of each through a fully integrated system as one progresses through the design of its various elements.

Figure 1.14 presents the system engineering approach from a different perspective. As one progresses through the life cycle, there is a need to ensure the full "traceability" of requirements from the system level and on down to the component. As *technical performance measures* (TPMs), or the applicable *metrics,* are established for the system, these measures must be allocated or apportioned to the next level, appropriate *design criteria* are identified, and these criteria must be reflected and supportive from the top down. Further, the appropriate methods/tools must be applied in

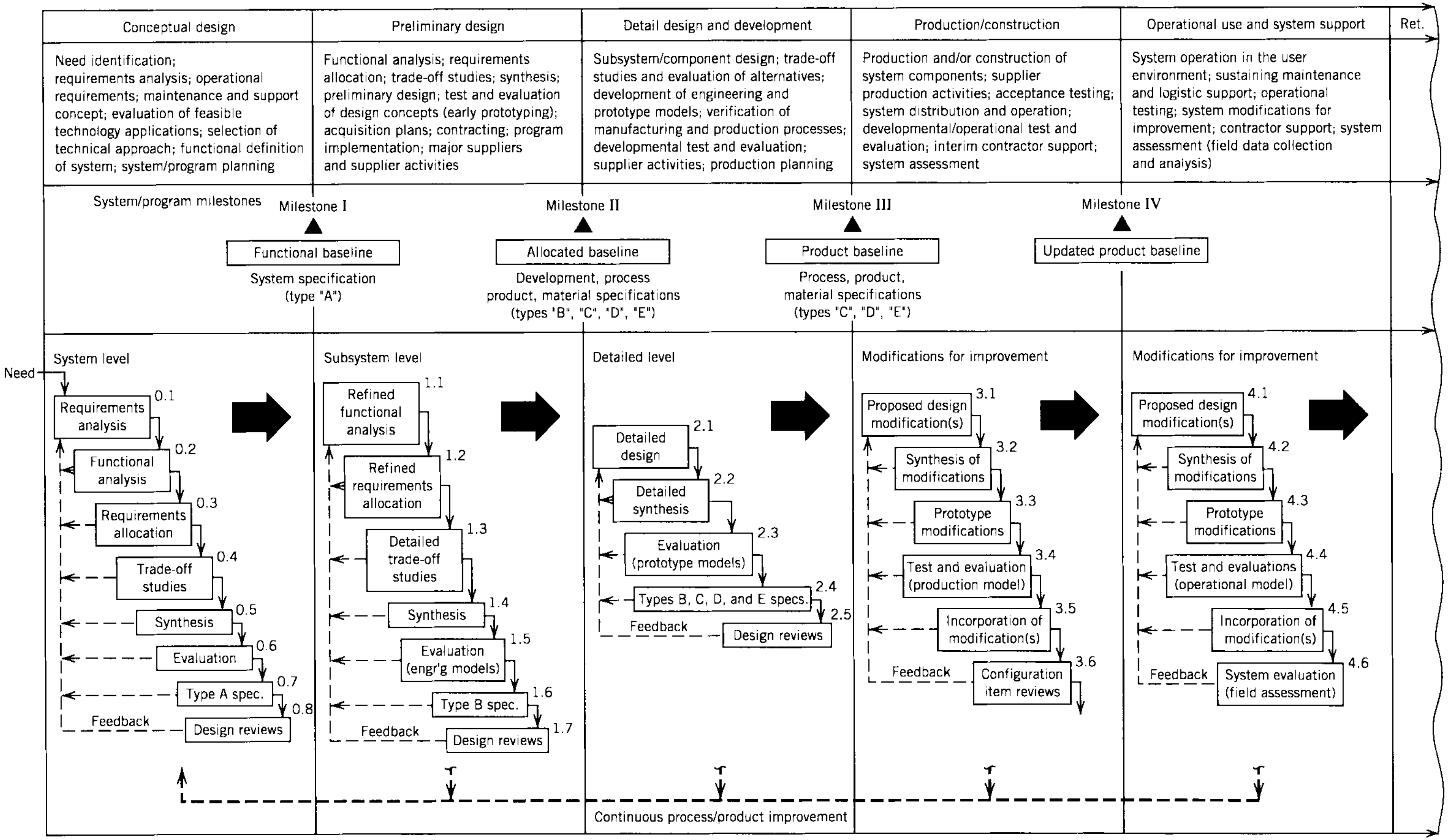


Figure 1.12 System engineering within the acquisition process. *Source:* B. S. Blanchard, D. Verma, and E. Peterson, *Maintainability: A Key to Effective Serviceability and Maintenance Management* (New York: John Wiley & Sons, Inc., 1995). This material is used by permission of John Wiley & Sons, Inc.

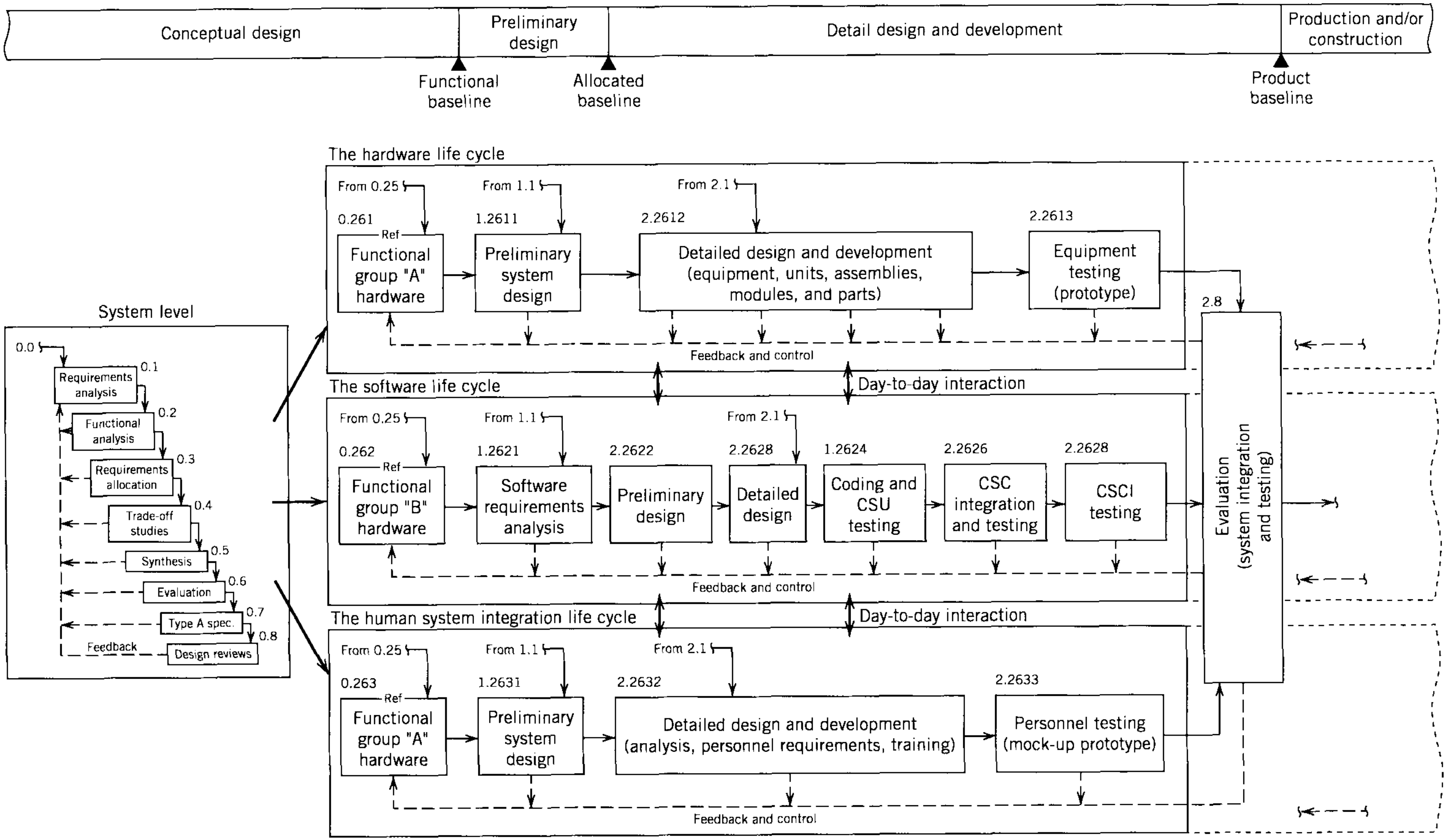


Figure 1.13 The integration of the hardware, software, and human life cycles.

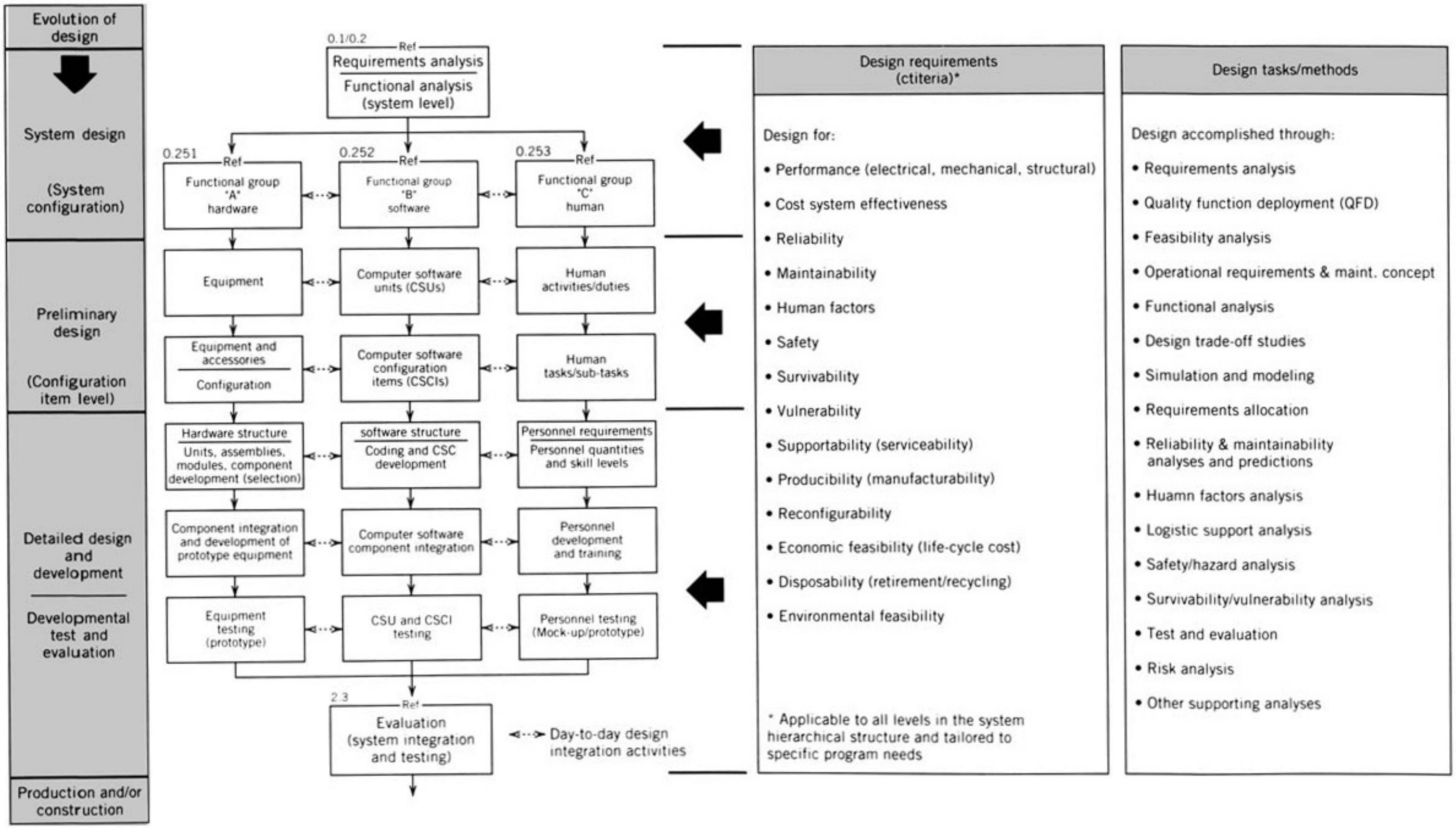


Figure 1.14 The top-down "traceability" of requirements.

the design process to ensure that the overall objectives of the system are met. Inherent within the system engineering process is the need to ensure that this traceability is maintained and to cause the integration of the appropriate techniques/methods/tools to facilitate the development process in an effective and efficient manner.

In summary, the system engineering process is continuous, iterative, and incorporates the necessary feedback provisions to ensure convergence. Figure 1.15 illustrates the *feedback* capability that must be built into the process, applied at the system level, to the subsystem level, and so on, as illustrated in Figure 1.12.

System engineering per se is not considered an engineering discipline in the same sense as civil engineering, mechanical engineering, reliability engineering, or any other design specialty area. Actually, system engineering involves efforts pertaining to the overall design and development process employed in the evolution of a system from the point when a need is first identified, through production and/or construction and the ultimate installation of that system for consumer use. The objective is to meet the requirements of the consumer in an effective and efficient manner. The system engineering process is covered further in Chapter 2. Finally, the concepts and principles associated with system engineering are not necessarily new or novel. A review of the literature in Appendix A indicates that many of the principles identified herein were being promoted back in the 1950s and early 1960s. However, in many instances, the system engineering process has not been implemented very well (if at all). Yet, at this point in time, there is a need to emphasize these concepts more than ever.

1.2.5 System Architecture

System architecture is a term often used to define a system in conceptual terms and at the highest level in its environment. An "architecture" deals with a top-level system structure (configuration), its operational interfaces, and anticipated utilization profiles (mission scenarios) and describes how the various elements of the system interact with each other. The system architecture shown in Figure 1.12 evolves as a result of a needs analysis, the completion of a feasibility analysis, and the definition of system operational requirements (i.e., block 0.1). These activities are discussed further in Chapter 2.[14]

1.2.6 System Science

Often, in addressing the subject of system engineering, one uses the terms "system science" and "system engineering" interchangeably. For the purposes of this text, *system science* deals primarily with the observation, identification, description, ex-

[14]System architecture is discussed further in a number of references included in Appendix A. Three such references are (a) E. Rechtin, and Mark Maier, *The Art of Systems Architecting* (CRC Press, 1996); (b) James N. Martin, *Systems Engineering Guide Book: A Process for Developing Systems and Products,* Boca Raton, FL (CRC Press, 1997); and (c) EIA/IS 632, *Systems Engineering,* Washington, DC: Electronic Industries Association (EIA) (latest edition). It should be noted that the development of "system architecture" constitutes a critical initial step in the system engineering process.

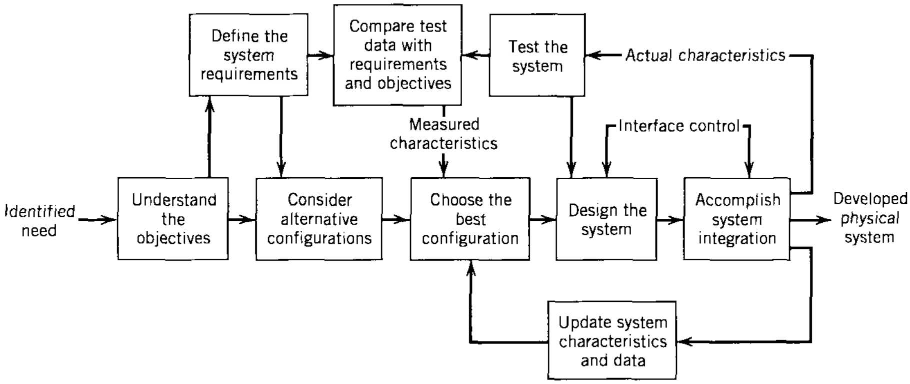


Figure 1.15 "Feedback" in the system engineering process.

perimental investigation, and theoretical explanation of facts, physical laws, interrelationships, and so on, associated with natural phenomena. Science deals with basic concepts and principles that help to explain how the physical world behaves. In the sense that they are applied sciences, the disciplines of biology, chemistry, and physics cover many of these relationships. In any event, system engineering includes the application of scientific principles throughout the system design and development process.[15]

1.2.7 System Analysis

Inherent within the system engineering process is an ongoing analytical effort. In a somewhat puristic sense, analysis refers to a separation of the whole into its component parts, an examination of these parts and their interrelationships, and a follow-on decision relative to a future course of action.

More specifically, throughout system design and development there are many different alternatives (or trade-offs) requiring an evaluation effort in some form. For instance, there are alternative system operational scenarios, alternative maintenance and support concepts, alternative equipment packaging schemes, alternative diagnostic routines, alternative manual versus automation applications, and so on. The process of investigating these alternatives, and the evaluation of each in terms of certain criteria, constitute an ongoing analytical effort.

To accomplish this activity effectively, the engineer (or analyst) relies on the use of available analytical techniques/tools to include operations research methods such as simulation, linear and dynamic programming, integer programming, optimization

[15]Systems science is a major subject by itself, and adequate coverage is not included here. Three excellent references are R. L. Ackoff, S. K. Gupta, and J. S. Minas, *Scientific Method: Optimizing Applied Research Decisions* (New York: John Wiley & Sons, Inc., 1962); G. M. Sandquist, *Introduction to System Science* (Upper Saddle River, NJ: Prentice-Hall, 1985); and L. Von Bertalanffy, *General Systems Theory* (New York: George Braziller, 1968).

(constrained and unconstrained), and queuing theory to help solve problems. Further, mathematical models are used to help facilitate the quantitative analysis process.

In essence, *system analysis* includes that ongoing analytical process of evaluating various system design alternatives, employing the application of mathematical models and associated analytical tools as appropriate. Analytical methods and models are discussed further in Chapter 4.[16]

1.2.8 Some Additional System Models

In the early 1980s, when the makeup of systems became more *software intensive,* there were a number of models developed with the objective of portraying the system life cycle. The "waterfall model" is probably the oldest and most widely used of the system development models in this category at the time.[17] This model, shown in Figure 1.16, is based on a top-down approach for software development and includes the steps of initiation, requirements analysis, design, testing, and so on. Often, in its implementation, the steps were viewed as being relatively independent from one another and were to be executed in a strict sequence, and the feedback effects were not emphasized. In addition, the required interfaces with the other elements of the system (e.g., hardware, the human factor, facilities, data) were not usually considered.

In the mid-1980s a generic "spiral model" was developed for software-intensive systems.[18] In this method, the analyst continually examines objectives, strategies, design alternatives, and validation methods. System development results through several iterations of this model. Figure 1.17 illustrates a modified version of the original generic approach, evolving from a prototype model. Note that rapid prototyping is used in each cycle and that the model emphasizes risk analysis. This approach is particularly useful in high-risk developments because design sometimes evolves as detailed requirements emerge.

The "Vee model," introduced in the early 1990s, reflects a top-down and bottom-up approach to system development.[19] In Figure 1.18, the left side of the Vee represents the evolution of user requirements into preliminary and detail design, and the right side represents the integration and verification of system components through

[16]System analysis is covered further in a number of the references listed in the bibliography in Appendix A. Some of the operations research tools utilized in accomplishing systems analyses are included in (a) B. S. Blanchard, and W. J. Fabrycky, *Systems Engineering and Analysis,* 3d ed., Part III (Upper Saddle River, NJ: Prentice-Hall, 1998); (b) F. S. Hillier and G. J. Lieberman, *Introduction to Operations Research,* 6th ed. (New York, McGraw Hill, 1995); and (c) H. A. Taha, *Operations Research: An Introduction,* 6th ed. (Upper Saddle River, NJ: Prentice-Hall, 1996.)

[17]B. W. Boehm, *Software Engineering Economics* (Prentice Hall, 1981), p. 36.

[18]The generic spiral model was presented by B. W. Boehm, "A Spiral Model of Software Development," in *Software Engineering Project Management,* R. H. Thayer and M. Dorfman, eds. (Washington, DC: IEEE Computer Society Press, 1988). This was modified in Figure 1.17 and is included in A. P. Sage, *Systems Engineering* (New York: John Wiley & Sons, Inc., 1992) pp. 53–54.

[19]K. Forsberg and H. Mooz, "The Relationship of System Engineering to the Project Life Cycle," *Proceedings of the 4th Annual Symposium* (Seattle, WA: International Council on Systems Engineering, INCOSE, 1991), p. 289.

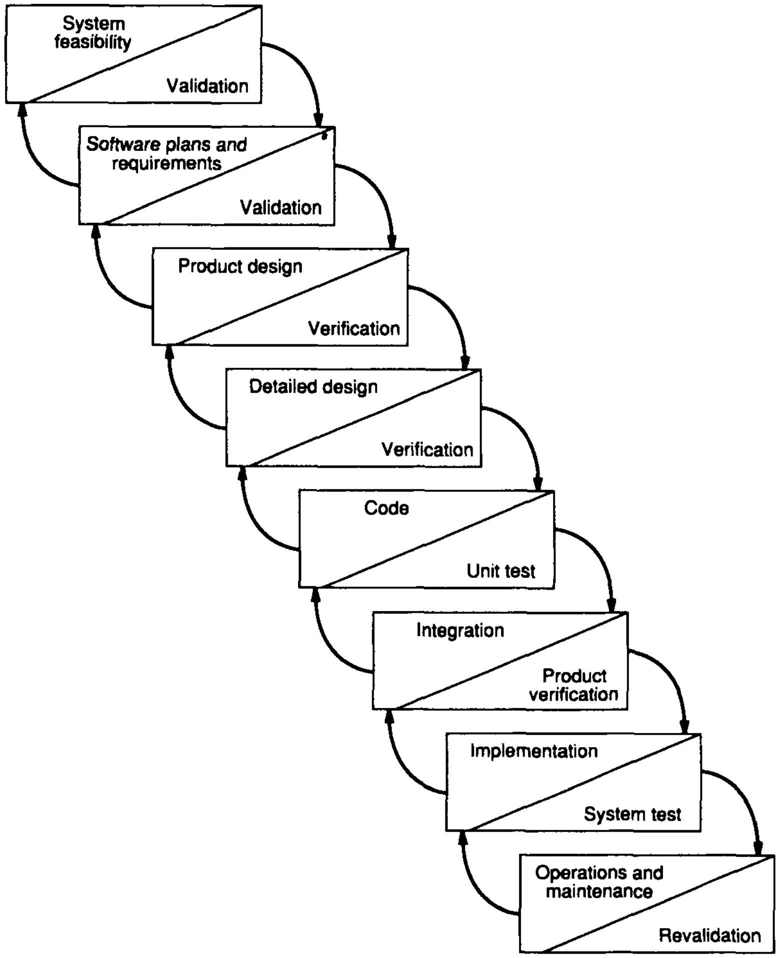
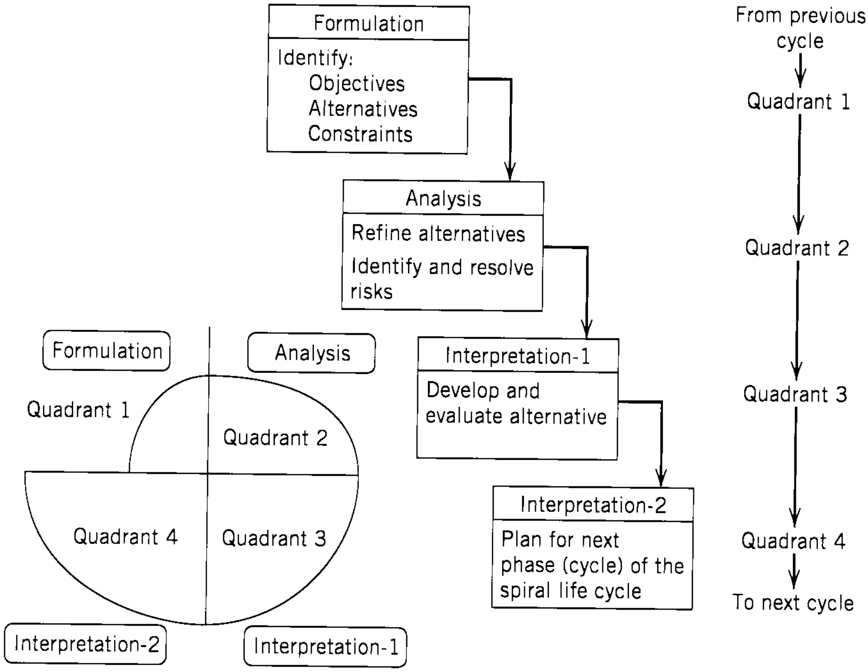


Figure 1.16 The "waterfall model" of the software life cycle. *Source:* SOFTWARE ENGINEERING ECONOMICS by Boehm, B.W., © Reprinted by permission of Pearson Education, Inc., Upper Saddle River, NJ.

subsystem and system testing. This model most nearly reflects the approach conveyed in Figure 1.11 (Section 1.2.4).

Figure 1.19 represents an extension of the Vee model concept.[20] Of particular note is the interface between the "system" and the "software subsystem." Quite often, individuals refer to "software systems." Although software may be predominant within the structure of a system, it is *not* the "system" per se. It does not fulfill a functional requirement by itself. Software requirements are identified through functional analysis

[20] B. G. Downward, "A Brave New World: Melding Systems and Software Engineering," *Proceedings of the 4th Annual Symposium* (Seattle, WA: International Council on Systems Engineering, INCOSE, 1991), p. 157.



Flow of activity in spiral life cycle

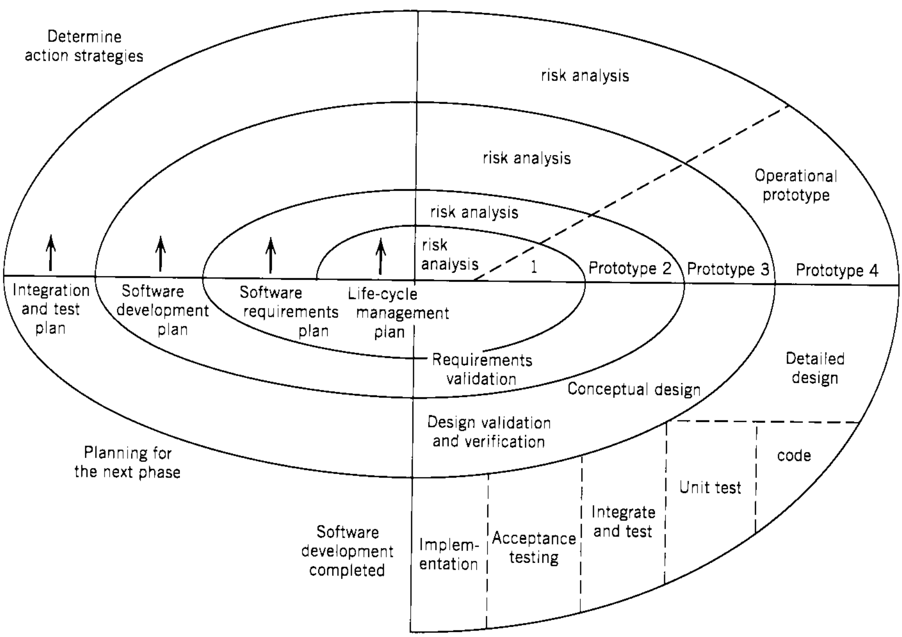


Figure 1.17 The spiral model for the software life cycle. *Source:* A. P. Sage, *Systems Engineering* (New York: John Wiley & Sons, Inc., 1992). This material is used with permission.

Next Page

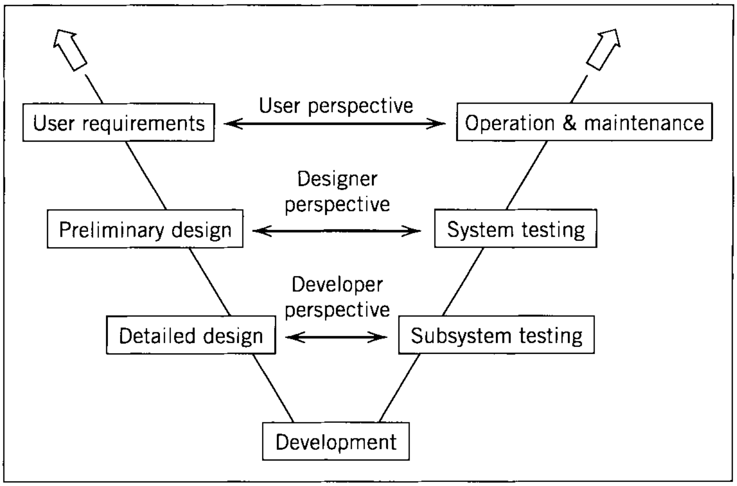


Figure 1.18 The generic "Vee" developmental model.

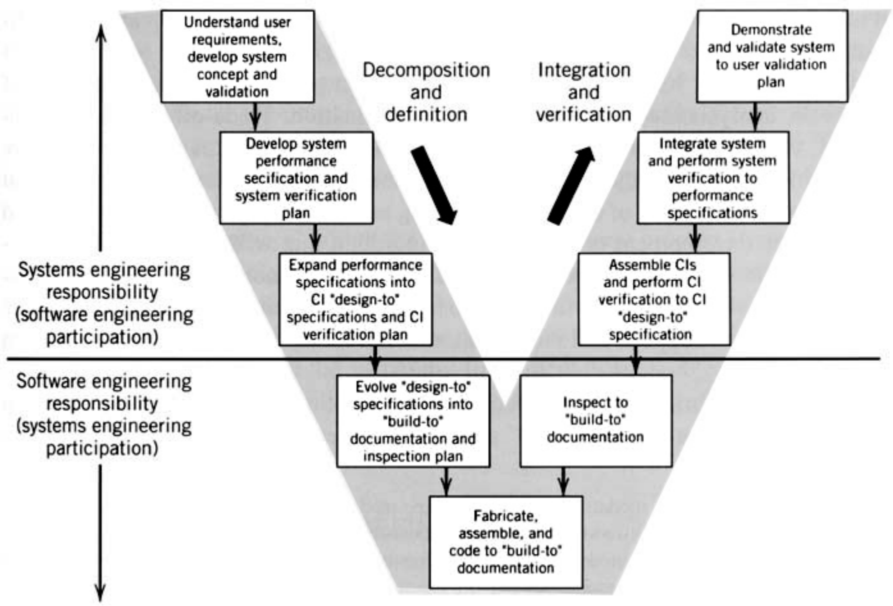


Figure 1.19 The systems versus software engineering boundary. *Source:* B. G. Downward, "A Brave New World: Molding Systems and Software Engineering," *Proceedings of the Symposium of the International Council on Systems Engineering* (Seattle, WA: INCOSE, 1991), 157.