6

SYSTEM ENGINEERING PROGRAM PLANNING

The first five chapters of this text have dealt with the system engineering process, the major steps in the process, and some of the available technologies and tools that can be applied. Given the definition of the process as a baseline, the remaining challenge lies with its *implementation.* To meet the objectives described herein requires application of a combination of *technology* and *management skills.* Although it is essential that one completely understand the "technical" process and available tools, this (by itself) will not guarantee success unless the proper organizational "environment" is created where the applicable management skills can be effectively applied in fulfilling the stated goals. As shown in Figure 6.1, there are technology-related activities and there are planning and organization activities that must jointly be applied throughout the top-down life-cycle process that has been the thrust of the earlier chapters.

The key to the successful implementation of any program is *early planning.* Planning for system engineering activities commences at program inception. As the need for a system is identified and feasibility studies are conducted in selecting a technical design approach, requirements are established to define a program structure that can be implemented to bring the system into being. Planning is initiated with the definition of program requirements and the subsequent development of a *Program Management Plan* (PMP), shown in Figure 1.26.[1] This, in turn, leads to the identification of system engineering requirements and the preparation of a detailed *System Engineering Management Plan* (SEMP).

[1]Usually, there is one overall planning document for every program/project, which covers all program requirements at a high level and leads to a variety of lower-level plans that address specific areas of activity. Although the specific nomenclature may vary from one program to the next, the title *Program Management Plan* (PMP) was selected in this instance to represent this top-level plan (refer to Figure 3.2, Chapter 3).

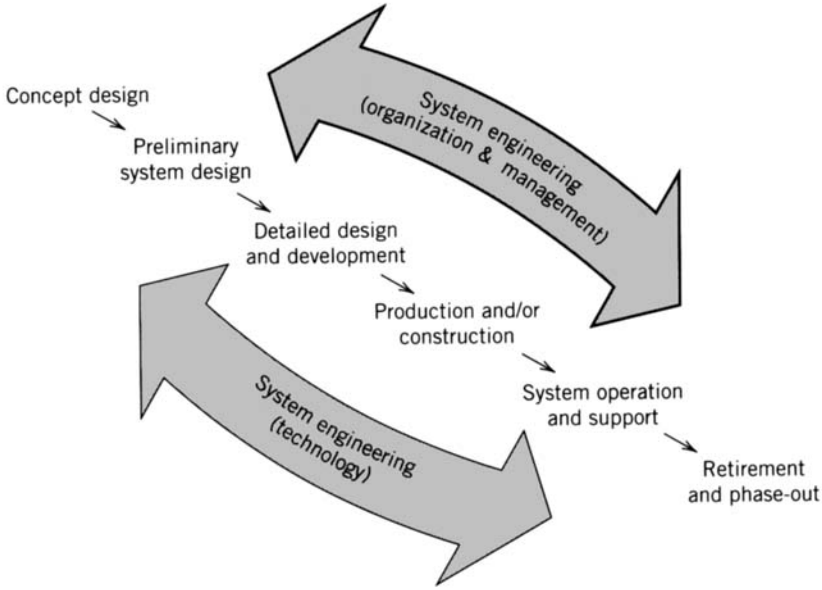

Figure 6.1 Management and technology applied to the system engineering process.

The SEMP is developed during the conceptual design and advanced planning phase, as illustrated in Figure 6.2, and includes a description of the system engineering functions and tasks to be accomplished, work packages and a work breakdown structure (WBS), task schedules and cost projections, an organizational structure and its interfaces, key policies and procedures, documentation and reporting requirements, and so on. The SEMP constitutes the overall planning document, which includes the necessary directives and guidance material for the successful implementation of the requirements described throughout the first five chapters of this text.[2]

This chapter covers system engineering program planning, the first step in system management, illustrated in Figure 6.2. The material presented leads into the discussion of the organization for system engineering (Chapter 7) and system engineering program evaluation (Chapter 8). Implementing the requirements described in the first five chapters is highly dependent on the thoroughness of planning from the beginning and in the follow-on organization, management, and control later on.

[2]In the preparation and implementation of the SEMP, it should be noted that system engineering activities may be implemented by the customer (consumer), by the prime producer (contractor), and/or by a major supplier. In some instances (particularly for large programs), an initial SEMP for the overall *system* may be prepared by the customer, with a lower-level SEMP prepared by the producer. Obviously, the second must evolve from and support the first. In any event, the intent herein is to describe what material might be included in the SEMP overall, and not attempt to differentiate between who does what. It is assumed that the proper "tailoring" will be accomplished, depending on the individual program requirements.

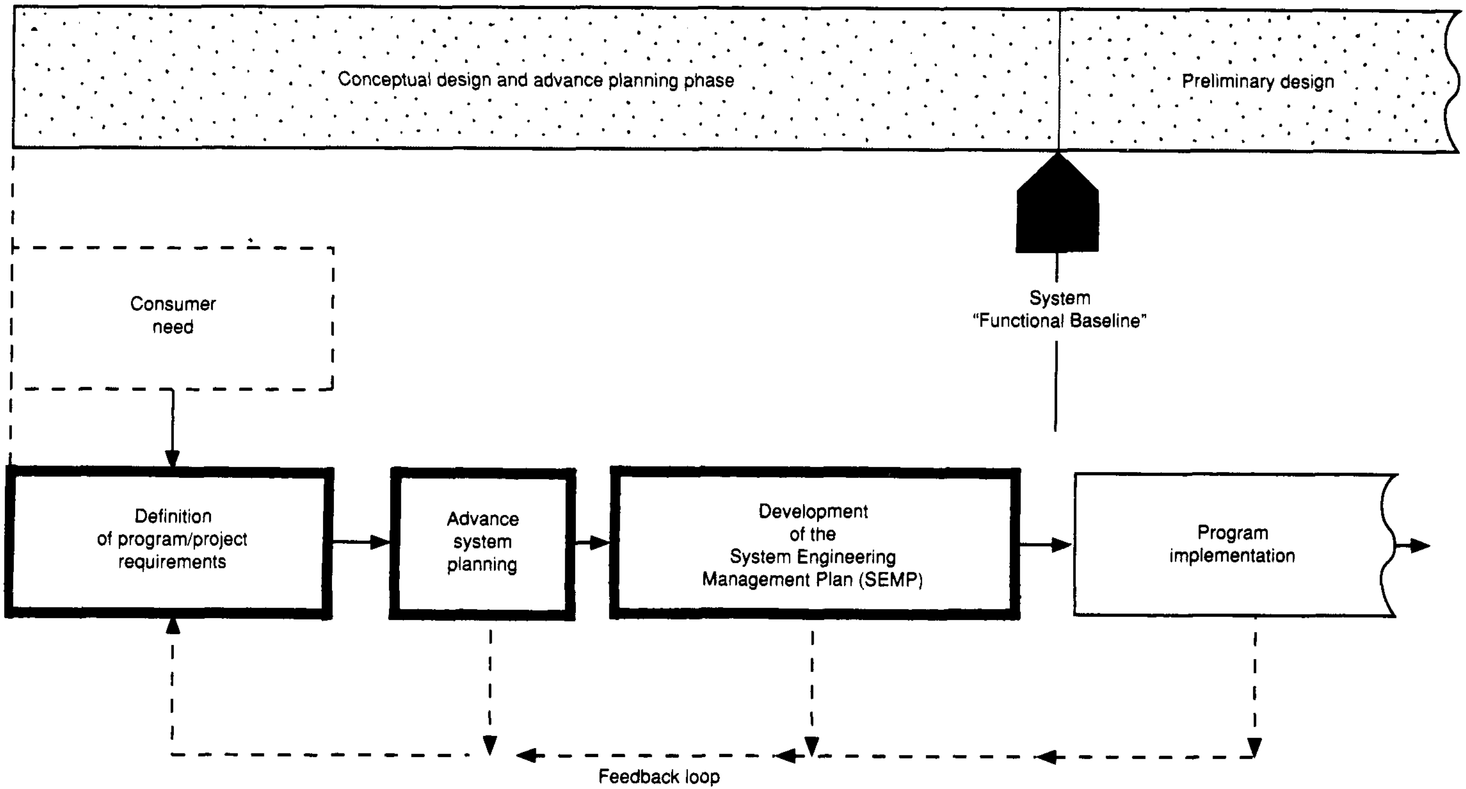

Figure 6.2 System engineering planning.

6.1 SYSTEM ENGINEERING PROGRAM REQUIREMENTS

As shown in Figure 6.2, the first step in the planning process involves the definition of program (or project) requirements. Although this may appear to be rather basic, every program is different and it is essential that system engineering requirements be tailored accordingly. The concepts and methods described throughout this text, however, are applicable to all programs. Only the nature and depth of application may vary from one program to the next.

6.1.1 The Need for Early System Planning

The successful implementation of the concepts and methods of system engineering is highly dependent on (1) employing a top-down approach in the development of a system, (2) integrating early the design and related supporting activities, (3) viewing requirements in terms of the entire system life cycle, and (4) preparing complete *requirements documentation* from the beginning (i.e., applicable specifications and plans). As early system concepts are generated and feasibility studies are conducted to determine possible alternative technical solutions in response to a given design problem, the appropriate level of planning must be initiated to ensure that the ideas generated through analysis are properly consumated and integrated into a final product configuration in a cost-effective manner.

System engineering planning commences at program inception during the needs analysis (described in Section 2.2). Liaison activities with the customer (consumer/user) are required to ensure that the defined need is interpreted correctly and described accurately, and that the translation from the stated need to the definition of system requirements is responsive. This is a critical step in the early definition of system requirements, as there is often a communications gap and the *real* need is not thoroughly understood. This, of course, may result in the development of planning information for a design configuration that will not perform the functions intended.

Given the identification and description of the need, the next step involves the accomplishment of feasibility studies (refer to Section 2.3). Future technological opportunities are identified and alternative approaches are investigated for possible design application. The feasibility of each of the alternatives being considered is a function not only of meeting the necessary performance requirements, but also of being responsive to the following questions:

1. Have the resource requirements associated with each alternative been defined (i.e., human, material, equipment, software, and data needs)? Have the sources of supply been identified? Will the necessary resources be available when required?
2. Does the alternative being considered for selection reflect a cost-effective approach based on a life-cycle analysis?
3. Has an *impact analysis* been accomplished to determine whether there are possible secondary and/or tertiary effects as a result of selecting a given alterna-

tive? It is hoped that the selection of a specific alternative will not have a detrimental impact relative to the environment; that is, in regard to social, political, or ecological concerns as presented in Figure 3.36 (Section 3.4.11). In addition, interactions with other systems should be minimized.

It is at this stage in the life cycle when early system engineering planning is important. The analysis effort is directed toward the *system* level, potential suppliers are identified, the overall system integration process commences, the interaction effects (both internal and external) are assessed, and potential areas of risk are identified. As system definition continues through the development of operational requirements, the maintenance concept, and the prioritization of technical performance measures (TPMs), the planning process evolves through another series of iterations. The requirements for system integration are greater in terms of both the *technical* integration of the various elements of the system and the integration of the many and varied organizational entities participating in the system development effort.

System planning is continuous, commencing with the definition of a need and extending through the development of the System Engineering Management Plan (SEMP). As system-level requirements are defined, the planning process leads to the identification of those activities that must be accomplished in order to provide a system configuration that will fulfill these requirements. Design and management decisions at this stage in the system life cycle have a great impact on program activities later on. Thus, it is imperative that a complete and well-integrated planning effort be implemented from the beginning.

6.1.2 Determination of Program Requirements

Although the concepts, methods, and processes describing system engineering are generally applicable to all categories of systems, they must be tailored for each individual application. Further, the applications are numerous and varied and include:

1. Large-scale systems with many different components such as a space system, an urban transportation system, a hydroelectric power generating system, and so on.
2. Small systems with relatively few components, such as a local communications system, a computer system, a hydraulic system, a mechanical braking system, and so on.
3. Systems in which there is a great deal of new design and development effort required; that is, the application of new technologies.
4. Systems in which the design is based primarily on the utilization of existing standard off-the-shelf components.
5. Systems that are highly equipment-intensive, software-intensive, facilities-intensive, or human-intensive; for example, a production system versus a ground command and control system, versus a data distribution system, versus a maintenance capability.

6. Systems in which there are a large number of suppliers involved in the design and development process, at both the national and international levels.
7. Systems in which there are a number of different organizations involved in the design and development process.
8. Systems being designed and developed for utilization in the government sector, the private sector, and so on.

Although this text basically addresses only a few of the major categories of systems described in Chapter 1 (i.e., the man-made, open-loop, dynamic system), there are still a wide variety of applications, as illustrated in Figure 6.3.

In each individual situation, the system engineering process described in Chapter 2 is applicable. Although the extent and depth of effort will vary, the steps required for bringing a system into being are basically the same. A needs analysis and feasibility analysis are accomplished, operational requirements and the system maintenance concept are defined, and functional analysis and requirements allocations are completed. Even though one may be dealing with a relatively simple case, such as a small system made up of standard off-the-shelf components, there is still a need to accomplish a top-down requirements analysis, a functional analysis and allocation, and so on. In other words, there is a *system* design requirement, even though new design may not be required at the subsystem or component level.

Following the general steps reflected in Figure 1.12 (Chapter 1) is a good overall approach to the design and development of any new system. As one progresses from the needs analysis through the accomplishment of feasibility studies and the definition of system-level requirements, the process evolves from the identification of the "WHATs" to the "HOWs"; that is, *what* is required in terms of a system functional

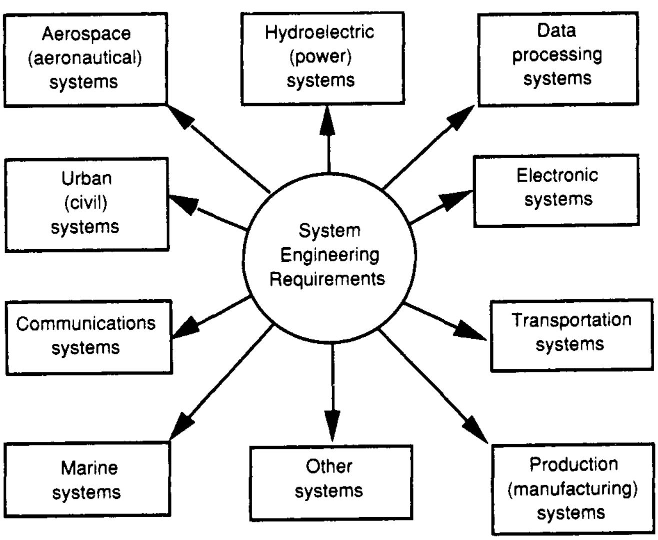

Figure 6.3 The application of system engineering requirements.

capability, and *how* is this to be accomplished from the standpoint of a technical design approach. Evaluation of the early responses to the proposed technical design approach leads to the identification of specific *program* (or *project*) requirements.

Program requirements, in this context, refer to the management approach and the steps to be followed in the procurement and/or acquisition of the system in response to a stated need, along with the identification of the resources required to fulfill this objective. A program structure should be established that will enable the design and development, production and/or construction, and delivery of the system to the consumer in a cost-effective manner. This includes the identification of program functions and detailed tasks, the development of an organizational structure, the development of a work breakdown structure (WBS), the preparation of program schedules and cost projections, the implementation of a program evaluation and control capability, and so on. This information, presented in the form of a program plan, provides the necessary day-to-day management guidance required in the realization of any technical objective.

In fulfilling the system engineering objectives described in the earlier chapters of this text, a System Engineering Management Plan (SEMP) is developed as part of the early planning requirements for each program (refer to Figure 6.2). Although the detailed content may vary from one instance to the next, some of the major features of the SEMP are noted in Section 6.2.

6.2 SYSTEM ENGINEERING MANAGEMENT PLAN (SEMP)[3]

As shown in Figure 3.2, the SEMP is developed based on the Program Management Plan (PMP) and covers all management functions associated with the performance of system engineering activities for a given program. The SEMP constitutes the chief engineer's plan for identifying and integrating all major engineering activities; i.e., the top *technical* plan that allows the integration of the many more subordinate plans, such as the mechanical engineering design plan, the software engineering plan, the reliability and maintainability plan(s), the human-factors and safety program plan(s), and so on.

Preparation of the SEMP is the responsibility of the "system manager" and may be accomplished by the customer (consumer/user) or by a major contractor (producer), depending on the program. The relationships between the customer, prime contractors or major producers, subcontractors, suppliers, and so on, particularly for large-scale systems, may take the form illustrated in Figure 6.4. In such instances, the customer/user is the system manager and is responsible for the SEMP, but may delegate the overall system integration and management responsibility to a prime contractor (i.e., Contractor A or Contractor B in the figure).

In the event that the customer prepares the SEMP, then Contractor A and Con-

[3]Three additional sources that include coverage of the SEMP are (1) A. P. Sage, and J. E. Armstrong, *Introduction to Systems Engineering* (New York: John Wiley & Sons, Inc., 2000); (2) EIA/IS-632, *Processes for Engineering a System* (Washington, DC: Electronic Industries Association); and (3) IEEE-1220, *Standard for Application and Management of the Systems Engineering Process* (New York: Institute of Electrical and Electronics Engineers, IEEE).

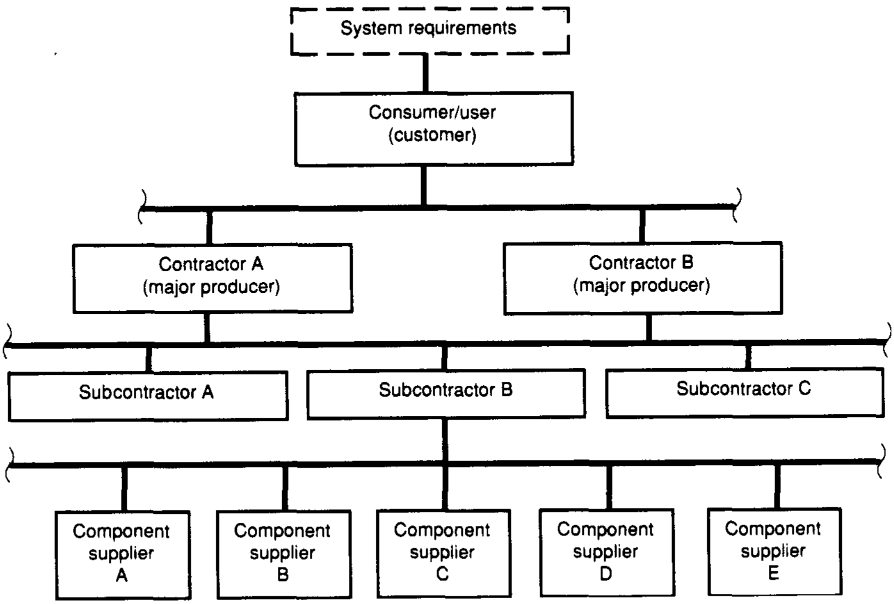

Figure 6.4 Consumer, producer, and supplier interfaces.

tractor B must each prepare a SEMP covering their respective system engineering activities, each being responsive to the higher-level SEMP. On the other hand, if the system integration and management responsibility is delegated to Contractor A (for example), then the responsibility for preparation of the SEMP, and for implementing the tasks defined therein, will be at this level.[4]

The process discussed here may appear to be self-evident. However, it is important to be clear that if the SEMP is to be meaningful and accomplish its objectives, it must be developed directly from the top-level Program Management Plan (PMP). Further, the responsibility for the SEMP, and for the accomplishment of the functions described within, must be clearly defined and supported by the program manager (or program director). When system management responsibility is delegated to Contractor A (in Figure 6.4), then Contractor A must be given both the *responsibility* and the *authority* to perform *all* system-level functions (described in the SEMP) on behalf of Contractor B, as well as for all subcontractors and applicable suppliers.[5] Finally, the SEMP must be appropriately identified as the key top-level design engineering plan in the overall program documentation tree structure.

[4]In such instances, the SEMP is usually prepared and included as part of the contractor's proposal to the customer.

[5]Quite often, contractors are held responsible for the development and implementation of a system engineering program (and the SEMP), but are not delegated the authority to successfully complete the task. Because the requirements for system engineering must start at the top, it is essential that the proper level of authority be delegated along with the responsibility.

In terms of material content, the SEMP must be *tailored* to the system requirements, the program size and complexity, and the nature of the procurement and acquisition process. To indicate the nature of the information that may be included in a SEMP, a proposed outline is presented in Figure 6.5. Although this outline is certainly not "fixed" (as there are likely to be all kinds of variations in use today), it will serve as a guide for further discussion of some of the content. Not every topic in the detailed SEMP outline is discussed here, but a few selected areas deserve additional coverage.[6]

6.2.1 Statement of Work

The Statement of Work (SOW) is a narrative description of the work required for a given project. In regard to the SEMP, it must be developed from the overall project SOW described in the PMP and should include the following:

1. A summary statement of the tasks to be accomplished. An identification of the major system engineering tasks is presented in Section 6.2.2. These, in turn, must be supported by elements of work included in the work breakdown structure (WBS) discussed in Section 6.2.4.
2. An identification of the input requirements from other tasks. These may include the results from other tasks accomplished within the project, tasks completed by the customer, and/or tasks accomplished by a supplier.
3. References to applicable specifications (including the System "A" Specification), standards, procedures, and related documentation as necessary for the completion of the defined scope of work. These references should be identified as key requirements in the documentation tree described in Section 6.2.5.
4. A description of the specific results to be achieved. This may include deliverable equipment, software, design data, reports, and/or related documentation, along with the proposed schedule of delivery as presented in Section 6.2.7.

In preparing the SOW, the following general guidelines are considered appropriate.

1. The SOW should be relatively short and to the point (not to exceed two to three pages), and must be written in a clear and precise manner.
2. Every effort must be made to avoid ambiguity and the possibility of misinterpretation by the reader.
3. Describe the requirements in sufficient detail to ensure clarity, considering both practical applications and possible legal interpretations. Do not *underspecify* or *overspecify!*
4. Avoid unnecessary repetition and the incorporation of extraneous material and requirements. This can result in unnecessary cost.
5. Do not repeat detailed specifications and requirements that are already covered in the applicable referenced documentation.

[6]Table 6.1 (p. 483) in A. P. Sage and J. E. Armstrong, *Introduction to Systems Engineering* (New York: John Wiley & Sons, Inc., 2000) offers a different SEMP outline.

System Engineering Management Plan (SEMP)

1.0 Overview
2.0 Applicable Documents
3.0 General Description of System Architecture
4.0 System Engineering Process
  4.1 System Operational Requirements
  4.2 Maintenance Concept
  4.3 Technical Performance Measures (TPMs)
  4.4 Functional Analysis (System Level)
  4.5 Allocation of Requirements
  4.6 System Synthesis, Analysis, and Design Optimization
  4.7 System Test and Evaluation
  4.8 Construction/Production Requirements
  4.9 System Utilization and Sustaining Support
  4.10 System Retirement and Material Recycling/Disposal
5.0 Technical Program Planning, Implementation, and Control
  5.1 Program Requirements/Statement of Work
  5.2 Organization (Customer/Producer/Supplier Structure and Interrelationships)
    5.2.1 Producer/Contractor Organization (Project/Functional/Matrix)
    5.2.2 System Engineering Organization
    5.2.3 Program Tasks
    5.2.4 Supplier Requirements
  5.3 Key Organizational Interfaces
  5.4 Work Breakdown Structure (WBS)
  5.5 Project Schedule and Milestone Charts
  5.6 Technical Performance Measurement (TPM) "Tracking"
  5.7 Program Cost (Projections/Reporting)
  5.8 Technical Communications (Program Reports/Documentation)
  5.9 Program Monitoring and Control
6.0 Engineering Specialty Integration (Identification of Key Engineering Specialties, How They Relate to System Engineering, and Their Interrelationships with Each Other)
  6.1 "Functional" Engineering (e.g., Electrical, Mechanical, Structural, Industrial, etc.)
  6.2 Software Engineering
  6.3 Reliability Engineering
  6.4 Maintainability Engineering
  6.5 Human Factors Engineering
  6.6 Safety Engineering
  6.7 Security Engineering
  6.8 Manufacturing and Production Engineering
  6.9 Logistics and Supportability Engineering
  6.10 Disposability Engineering
  6.11 Quality Engineering
  6.12 Environmental Engineering
  6.13 Value/Cost Engineering
  6.14 Other Engineering Disciplines (as appropriate)
7.0 Configuration Management (CM)
8.0 Data Management (DM)
9.0 Program Technology Requirements (Computer-Aided methods, EC/EDI/IT Applications etc.)
10.0 Risk Management
11.0 References (Specifications, Standards, Plans, Procedures, Pertinent Documentation)

Figure 6.5 System Engineering Management Plan (SEMP) outline.

The SOW will be read by many different individuals with a variety of backgrounds (e.g., engineers, accountants, contract managers, schedulers, lawyers), and there must be no unanswered questions as to the scope of work desired. This statement forms a basis for the definition and costing of detailed tasks, for the establishment of subcontractor and supplier requirements, and so on.

6.2.2 Definition of System Engineering Functions and Tasks

System engineering, as defined throughout this text, covers a broad spectrum of activity. It may even appear that the "systems engineer," or the system engineering organization, does everything! Although this is not practical, the fulfillment of system engineering objectives does require some involvement, either directly or indirectly, in almost every facet of program activity. The challenge is to identify those functions (or tasks) that deal with the overall *system* as an entity and, when successfully completed, will have a positive impact on the many related and subordinate tasks that must be accomplished.

For the purpose of identifying a select number of *key* tasks for system engineering, the process described in Chapter 2 can be considered a framework for further discussion. As a start, a review of some of the overall basic goals is appropriate. The objectives of system engineering are the following:

1. Ensure that the requirements for system design and development, test and evaluation, production, operation, and support are developed in a timely manner through a top-down, iterative requirements analysis.
2. Ensure that system design alternatives are properly evaluated against meaningful, quantifiable criteria that relate to all of the desired characteristics; for example, performance factors, effectiveness factors, reliability and maintainability characteristics, human factors and safety factors, supportability characteristics, and life-cycle cost.
3. Ensure that all applicable design disciplines and related specialty areas are appropriately integrated into the *total engineering effort* in a timely and effective manner.
4. Ensure that the overall system development effort progresses in a logical manner with established configuration baselines, formal design reviews, the proper documentation supporting design decisions, and the necessary provisions for corrective action as required.
5. Ensure that the various elements (or components) of the system are compatible with each other and are combined to provide an entity that will perform its required functions in an effective and efficient manner.

A review of these general goals leads to the question, What detailed program/project tasks should be performed in order to successfully meet the objectives of system engineering? Although each individual program is different and activities must be tailored accordingly, the tasks presented in Figure 6.6 and discussed in the following paragraphs are considered applicable in most instances.

System Engineering Tasks	Task Input Requirements	Task Output Requirements
1. Perform needs analysis and conduct feasibility studies.	Consumer/customer requirements documentation; technical information reports covering technology applications; selected research reports; trade-off study reports supporting design approach.	Feasibility study reports; trade-off study reports justifying system-level design decisions.
2. Define operational requirements and the system maintenance concept.	Consumer/customer requirements documentation; customer specifications and standards; feasibility study reports; trade-off study reports supporting design approach.	System requirements documentation (operational requirements and maintenance concept); trade-off study reports justifying system-level design decisions; list of prioritized TPMs; functional analysis (system-level).
3. Prepare the system Type "A" specification.	Technical information reports covering technology applications; feasibility study reports; system requirements documentation (operational requirements and maintenance concept); trade-off study reports justifying system -level design decisions; list of prioritized TPMs; functional analysis (system-level).	System Type "A" specification.
4. Prepare the Test and Evaluation Master Plan (TEMP).	System Type "A" specification; customer test specification and standard; test requirements sheets (individual discipline test requirements).	Test and Evaluation Master Plan (TEMP).
5. Prepare the System Engineering Management Plan (SEMP).	Consumer/customer requirements documentation; customer program specifications and standards; system requirements documentation (operational requirements and maintenance concept); system Type "A" specification; Test and Evaluation Master Plan (TEMP); advance system planning information ; Program Management Plan (PMP).	System Engineering Management Plan (SEMP).
6. Accomplish functional analysis and the allocation of requirements.	System requirements documentation (operational requirements and maintenance concept); system specification; trade-off study reports justifying system-level design decisions.	Functional analysis reports–functional flow diagrams (operational and maintenance functions), timeline analysis sheets, requirements allocation sheets (RASs), trade-off study reports, test requirements sheets, design criteria sheets.
7. Accomplish system analysis, synthesis, and system integration.	Consumer/customer requirements documentation; customer specifications and standards; functional analysis reports; system Type "A" specification; System Engineering Management Plan (SEMP); Test and Evaluation Master Plan (TEMP); individual design discipline program planning requirements.	Selected design data; system integration reports; supplier data and reports; trade-off study reports justifying design decisions; selected design discipline reports (predictions and analyses).
8. Plan, coordinate, and conduct formal design review meetings.	Program Management Plan (PMP); System Engineering Management Plan (SEMP); applicable design data (drawings, parts and material lists, reports, databases); trade-off study reports justifying design decisions;individual design discipline reports (predictions, analyses, etc.).	Design review meeting minutes; action-item lists with designated responsibilities; approved/ released design data and supporting documentation.
9. Monitor and review system test and evaluation activities.	Test and Evaluation Master Plan (TEMP); System Engineering Management Plan (SEMP); individual test data and reports.	System test and evaluation report(s).
10. Plan, coordinate, implement, and control design changes.	Configuration management data and reports (description of design baseline); proposed engineering change proposals; change control requirements and actions.	Change implementation plans, change verification data/reports.
11. Initiate and maintain production/ construction liaison; supplier liaison; and customer service activities.	System design data; production/construction requirements; approved design changes; system operating and maintenance procedures; consumer/customer operations and system utilization requirements; field data and failure reports.	Field data and failure reports; customer service reports on field operations.

Figure 6.6 System engineering tasks.

1. Perform a needs analysis and conduct feasibility studies (refer to Sections 2.1, 2.2, and 2.3). These activities should be the responsibility of the system engineering organization, because they deal with the system as an entity and are fundamental in the initial interpretation and subsequent definition of system requirements.

2. Define system operational requirements, the system maintenance concept, and the technical performance measures (TPMs) (refer to Sections 2.4, 2.5, and 2.6). The results of these activities are included in the overall definition of system-level requirements and are the basis for top-down system design.

3. Prepare the System Type "A" Specification (refer to Section 3.2). This is the top technical document for system design, and fulfilling the objectives of system engineering is dependent on the completeness and comprehensiveness of this specification. The "B," "C," "D," and "E" Specifications are based on the requirements of the "A" Specification.

4. Prepare the Test and Evaluation Master Plan (refer to Section 2.11). This document reflects the approach, methods, and procedures that are to be followed in the overall evaluation of the system in terms of compliance with the initially specified requirements. Although there are many different and relatively small facets of testing, it is the compilation of these that provides an overall evaluation of the system as an entity.

5. Prepare the System Engineering Management Plan (refer to Sections 1.4 and 6.2). This, of course, is the top management document for all system engineering program activities.

6. Accomplish functional analysis and the allocation of requirements (refer to Sections 2.7 and 2.8). Functional analysis, which is the process of translating system-level requirements into detailed design criteria, provides the foundation for the development of many different individual design disciplinary tasks (refer to Section 2.7.4). The allocation process defines the specific design requirements for different components of the system, whether developed through supplier activities or procured off the shelf. In any event, the responsibility for this effort is appropriate because it facilitates the necessary design integration effort by providing a common baseline definition of the system in functional terms.

7. Accomplish system synthesis, analysis, and design integration functions on a continuing basis throughout the overall design and development process (refer to Sections 2.9 and 2.10 and Chapter 3). System integration is iterative by nature and includes both the technical considerations dealing with the physical and functional interfaces of equipment, software, personnel, facilities, and so on, and the management considerations pertaining to organizational interfaces. From the management perspective, the system engineering organization is responsible for ensuring that (a) all program design-related functions/tasks are initially defined, (b) appropriate responsibilities and working relationships are established, (c) organization and communication channels are identified, and (d) program requirements are completed in a satisfactory manner. As noted in Chapter 3, the system engineering organization is responsible for ensuring that the proper level of communications, coordination, and integration exists between the various design disciplines as applicable. Of particular

interest are the task requirements for reliability engineering (Figure 3.15), maintainability engineering (Figure 3.21), human-factors engineering (Figure 3.27), safety engineering (Figure 3.30), logistics (Figure 3.34), software engineering (Section 3.4.1), manufacturing and production engineering (Section 3.4.7), quality engineering (Section 3.4.10), value/cost engineering (Section 3.4.12), and environmental engineering (Section 3.4.11).

8. Plan, coordinate, and conduct formal design review meetings; for example, conceptual design review, system design reviews, equipment/software design reviews, and critical design review (refer to Chapter 5). The system engineering organization is responsible for ensuring that an ongoing design evaluation is performed. This is partially accomplished through the scheduling of periodic design review meetings. The conductance of these meetings must be accomplished by a unbiased individual, and the overall results must be supportive of *system-level* design objectives.

9. Monitor and review system test and evaluation activities (refer to Section 2.11). It is essential that the system engineering organization be involved from the standpoint of interpreting and integrating individual test results into the evaluation of the system as a whole.

10. Plan, coordinate, implement, and control design changes as they evolve from engineering change proposals (ECPs) initiated from either the informal day-to-day review activity or as a result of formal design reviews (refer to Section 5.4). The system engineering organization is responsible for establishing and maintaining system "baselines" through the design and development process; for example, "functional" baseline, "allocated" baseline, and the "product" baseline in Figure 1.12. System engineering is essentially responsible for configuration management as the system evolves through its planned life cycle.

11. Initiate and maintain production/construction liaison, supplier liaison, and customer service activities. As the system configuration progresses from the design and development phase into production and/or construction, and subsequently into operational use, there is a requirement for a specified level of engineering support. The purpose is to provide some engineering assistance relative to training and the understanding of system design, the incorporation of approved engineering changes into the system, and the acquisition of data from production activities and consumer operations in the field. The system engineering organization must be able to track the system throughout its planned life cycle.

The 11 basic program tasks just described constitute an example of what might be appropriate for a typical program, although the specific requirements may vary from one program to the next. The goal is to identify tasks that are oriented to the *system,* and that are *critical* relative to meeting the five major system engineering objectives stated earlier. More specifically, it is essential that an overall *system's* approach be followed from the initial establishment of requirements. As design progresses, it is essential that the system configuration being developed includes the desired characteristics. Finally, it is essential that the product output be validated in terms of meeting the initially established requirements.

In accomplishing this, there are requirements definition tasks, there are design review and approval tasks, there are configuration control tasks, and there are final test and evaluation tasks. These activities are undertaken through the combination of providing key documentation (specifications, plans, and reports), conducting carefully scheduled design reviews with the appropriate feedback provisions, and providing the necessary ongoing coordination and integration efforts. These activities must address all *system* functions accomplished throughout the various levels depicted in Figure 6.4.

To provide a more in-depth understanding of the 11 tasks, Figure 6.6 presents a summary, listing these tasks and showing typical input and output requirements. Although the majority of the input-output requirements are self-explanatory, through a review of the appropriate sections of this text, some additional discussion is necessary in support of the output requirements of Task 6, dealing with functional analysis and allocation.

Functional analysis encompasses the process of translating system-level requirements into detailed design criteria and results in the complete definition of the system configuration in functional terms (refer to Section 2.7). The accomplishment of functional analysis is facilitated through the development of functional flow block diagrams, described in Section 2.7.1. Based on these diagrams, the system engineer may wish to evaluate the various functions further from the standpoint of series-parallel relationships, time durations, and, ultimately, the identification of major resource requirements. In addition, specific functional requirements need to be communicated to program/project personnel through time line analysis sheets, requirements allocation sheets (RASs), trade-off study reports, test requirements sheets, design criteria sheets, and the like. Time line analysis sheets and requirements allocation sheets are briefly discussed in the following paragraphs.

1. *Time line analysis sheets:* Although the functional block diagrams convey general series-parallel relationships, these requirements may be developed further through the use of time line analysis sheets. Time line analysis adds considerable detail in defining the durations of various functions. Concurrency, overlap, and the sequential relationships of functions/tasks can be projected. Moreover, time-critical functions can be readily identified; that is, those functions that directly affect system availability, operating time, and maintenance downtime. An example of a time line analysis sheet format is presented in Figure 6.7.

2. *Requirements allocation sheets:* A requirements allocation sheet (RAS) is often used as the primary document for the identification of specific design requirements based on the functional analysis. The RAS is developed for each block in the functional flow block diagram. Performance requirements are described, which include (a) the purpose of the function, (b) the detailed performance characteristics that the function must accomplish, (c) the criticality of the function, and (d) applicable design constraints. Performance requirements must address design characteristics such as size, weight, volume, output, throughput, reliability maintainability, human factors, safety, supportability, economic factors, and so on. Both qualitative and quanti-

System:		Subsystem:	Description of requirement:												
Source (functional flow block diagram)		Function number	Location												
Task number	Task description	Personnel	Elapsed time (hours)												Total time
			0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0	5.5	6.0	

Figure 6.7 Time line analysis sheet.

tative performance requirements resulting from an analysis of the function are identified by the RAS. These requirements are expanded in sufficient detail to allow for the synthesizing and evaluation of alternative concepts for satisfying each functional need, employing a combination of resources in terms of equipment, personnel, software, and facilities. An initial definition of these resource requirements is included in the RAS. Figure 6.8, which is an extension of Figure 2.20, presents an example of a requirements allocation sheet (RAS) format.

The specific output from Task 6 (Figure 6.6) will vary in structure and format, depending on the type of system and the stage of design and development. For large-scale systems involving many different interfaces, the relationships illustrated in Figure 6.9 may exist. On the other hand, for smaller systems in which the design is relatively simple, the utilization of all of these data outputs may not be feasible.

6.2.3 System Engineering Organization

One of the most important sections of the SEMP describes the organizational structure that is being proposed for implementation of the objectives and the tasks defined in Section 6.2.2. A specific and individual system engineering department or group, by itself, will not be able to complete all of the work required for the 11 tasks presented in Figure 6.6. However, it is not the intent here to justify a large organization for working out the details. Yet the system engineering organization, through its system-level technical expertise and its leadership abilities, must take the lead and ensure that these task requirements are completed in an effective, efficient, and timely manner. In other words, the system engineering organization must be able to work with, influence, and inspire many other groups within (and external to) the project if the specified tasks are to be successfully completed. The system engineering organization must have the respect and cooperation of the other required functions in order for the proper integration to occur.

Figure 6.10 presents an abbreviated illustration to show how a system engineering department/group might fit within and relate to other major functions within the overall organization for a large contractor. In delving further into the subject of "organization," it is clear that there are many different structures and approaches that may apply. For example, the primary system engineering organization may be contained within the customer's organization, with various responding subgroups within the contractor's organization. In a contractor's organization, the basic structure may constitute a *functional* approach, a *project/product line* approach, a *matrix* approach, or various combinations thereof. There are advantages and disadvantages associated with each of these approaches, which are essential to recognize if the system engineering organization is to work effectively within the structure provided. Further, there are the external interactions involving subcontractors and suppliers, which, in turn, may be critical for the accomplishment of the work required.

The subject of organization—the development of organizational structures, the staffing of an organization, and the "organization for system engineering"—is covered in detail in Chapter 7. However, at this point, it should be emphasized that a complete

System:	Subsystem:		Description of requirement:
Source (functional flow block diagram)	Function number	Location	

Functional performance and design requirements	Personnel requirements				Equipment requirements		Software/data requirements		Facility requirements	
	Tasks	Task time	Perfromance requirements	Training	Nomenclature	Specification	Nomenclature	Specification	Nomenclature	Specification

Figure 6.8 Requirements allocation sheet.

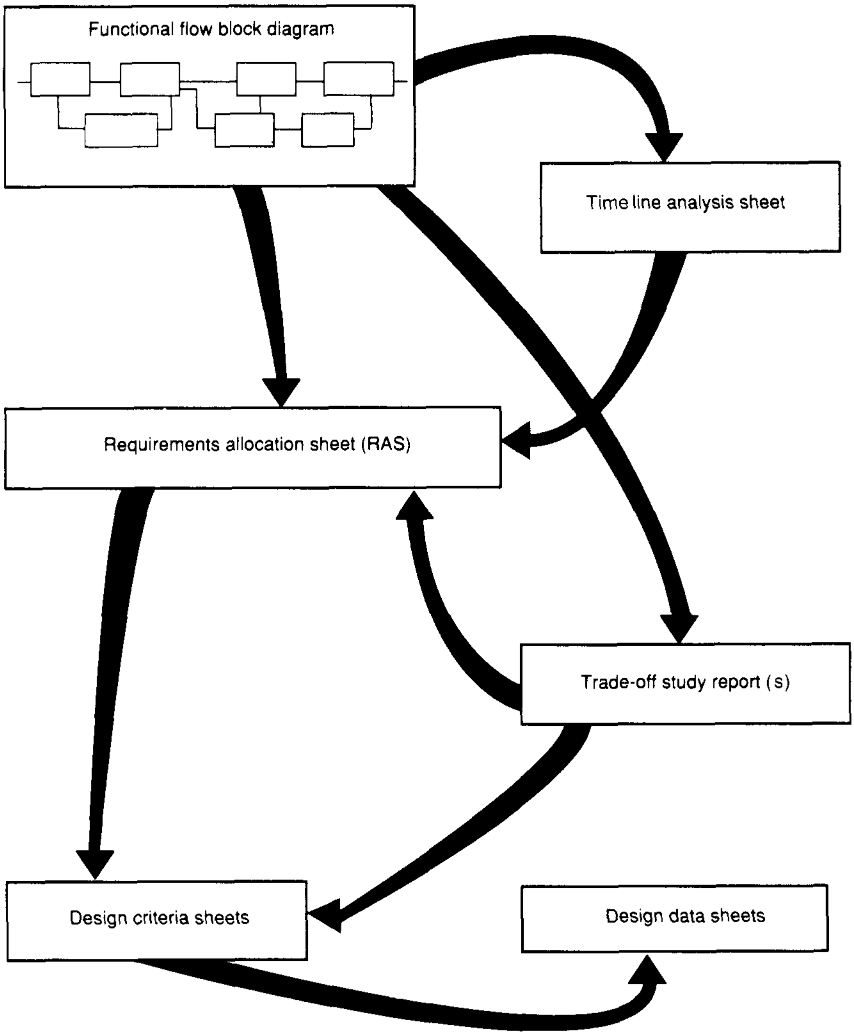

Figure 6.9 System engineering documentation.

and thorough discussion of the organizational approach to be implemented for the project in question must be included in the SEMP for the system being developed or modified. Of particular interest are the numerous interfaces that must exist, across the board, if the objectives described throughout this text are to be met. Effective communication links, represented by the dotted lines in Figure 6.10, must be in place and functioning from the beginning. Although the organizational "makeup" within the system engineering block (in the figure) may look great on paper, it will not work unless the many noted interfaces are operational on a day-to-day basis.

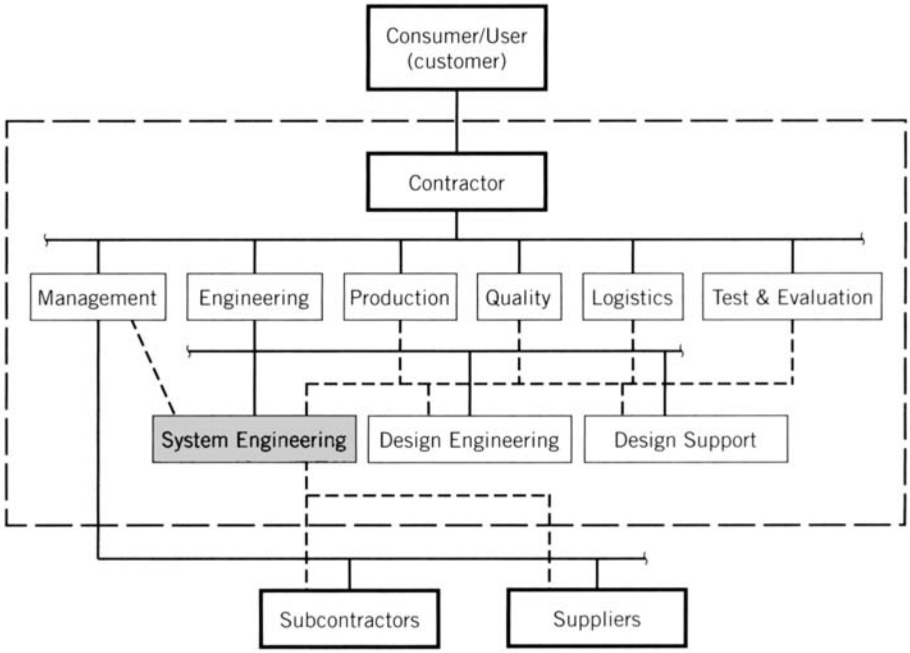

Figure 6.10 System engineering organization and interfaces.

6.2.4 Development of a Work Breakdown Structure (WBS)

One of the initial steps in program planning, after the generation of the Statement of Work (SOW) and identification of the organizational structure, is the development of the work breakdown structure (WBS).[7] The WBS is a product-oriented tree that leads to the identification of the activities, functions, tasks, subtasks, work packages, and so on, that must be performed for the completion of a given program. It displays and defines the system (or product) to be developed and portrays all of the elements of work to be accomplished. The WBS is *not* an organizational chart in terms of project personnel assigned and responsibilities, but does represent an organization of work packages prepared for the purposes of program planning, budgeting, contracting, and reporting.[8]

Figure 6.11 illustrates an approach to the development of the WBS. During the early stages of system planning, a Summary Work Breakdown Structure (SWBS) is

[7]WBS and work packaging are covered in most texts dealing with project management. A good reference is H. Kerzner, *Project Management: A Systems Approach to Planning, Scheduling, and Controlling,* 7th ed., (New York: John Wiley & Sons, Inc., 2000).

[8]Although the 11 tasks described in Section 6.2.2 reflect a *generic* approach for a system engineering organization in general, the development of the WBS will lead to the identification of the *specific* task requirements for the system being covered by the SEMP. A *tailored* approach is required.

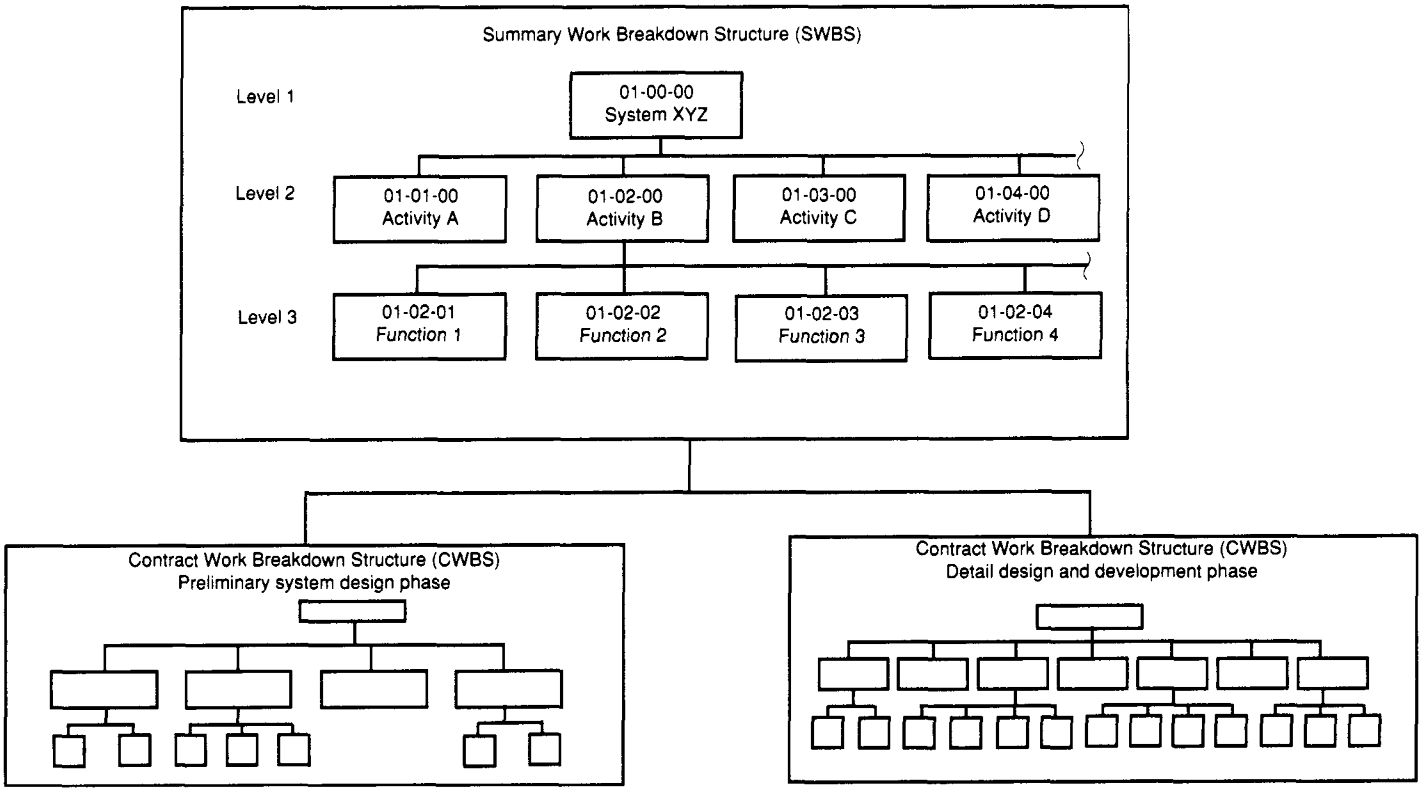

Figure 6.11 Partial work breakdown structure development.

usually prepared by the customer and included in a Request for Proposal (RFP) or an Invitation for Bid (IFB). This structure, developed from the top down, primarily for budgetary and reporting purposes, covers *all* programs functions and generally includes three levels of activity:

1. *Level 1:* Identifies the total program scope of work, or the system to be developed, produced, and delivered to the customer. Level 1 is the basis for the authorization and "go-ahead" (or release) for all program work.
2. *Level 2:* Identifies the various projects, or categories of activity, that must be completed in response to program requirements. It may also include major elements of the system and/or significant project activities; for example, subsystems, equipment, software, elements of support, program management, and system test and evaluation. Program budgets are usually prepared at this level.
3. *Level 3:* Identifies the activities, functions, major tasks, and/or components of the system that are directly subordinate to the Level 2 items. Program schedules are generally prepared at this level.

As program planning progresses and individual contract negotiations are consumated, the SWBS is developed further and adapted to a particular contract or procurement action, resulting in a Contract Work Breakdown Structure (CWBS). Referring to Figure 6.4, for example, the customer may develop the SWBS with the objective of initiating program work activity. This structure will usually reflect the integrated efforts of all organizational entities assigned to the project and should not be related to any single department, group, or section. The SWBS, included in the customer's RFP, is the basis for the definition of all internal and contracted work to be performed on a given program. Through the subsequent preparation of proposals, contract negotiations, and related processes, Contractor A is selected to accomplish all work associated with the preliminary system design phase, and Contractor B is selected to complete all work associated with the detail design and development phase. From the definition of individual statements of work, a CWBS is developed to identify the elements of work for each program phase. The CWBS is tailored to a specific contract (or procurement action) and may be applicable to prime contractors, subcontractors, and/or suppliers, as shown in Figure 6.4.

The WBS constitutes a top-down hierarchical breakout of project activities that can be further divided into functions, functions into tasks, tasks into subtasks, subtasks into levels of effort, and so on. Conversely, detailed tasks (with defined starting and ending dates) can be combined into work packages, and work packages can be integrated into functions and activities, with the accumulation of all work being reflected at the top program or system level.

In developing a WBS, care must be exercised to ensure that (1) a continuous flow of work-related information is provided from the top down, (2) all applicable work is represented, (3) enough levels are provided to allow the identification of well-defined work packages for cost/schedule control purposes, and (4) the duplication of work effort is eliminated. If the WBS does not contain enough levels, then management visibility and the integration of work packages may prove to be difficult. On the other hand, if too many levels exist, too much time may be wasted in performing program review and control actions.

Figure 6.12 presents an example of a Summary Work Breakdown Structure (SWBS) covering the development of a large system. As program requirements are defined through a contractual (or procurement) arrangement, the SWBS can be readily converted into a CWBS to reflect the actual work required under the contract. The CWBS, as it appears in a contractual document, is also presented at three levels in order to provide a good baseline for planning purposes while allowing for some flexibility within the contractor's organization. An expansion of the CWBS can be accomplished as necessary to provide for internal cost/schedule controls.

Figure 6.13 shows an expansion of the system engineering activities to the fifth level; that is, those work packages under 3B1100 in Figure 6.12. The purposes are to recognize the major system engineering tasks presented in Figure 6.6 and to provide a breakout of these tasks in a CWBS format to the extent necessary for proper cost/schedule visibility. Note that Figure 6.13 includes two different CWBSs, one covering the work to be performed during the conceptual design and advance planning phase and the other directed toward the work required during the preliminary system design phase. Each individual CWBS is derived from the SWBS and the overall program CWBS. Further, there must be a close tie between the two, as the CWBS for preliminary design must reflect the activities that evolve directly from the earlier phase.

The elements of the WBS may include an identifiable item of equipment or software, a deliverable data package, an element of logistic support, a human service, or a combination thereof. WBS elements should be selected to permit the initial structuring of budgets and the subsequent tracking of technical performance measures (TPMs) against cost. Thus, in expanding the WBS to successively lower levels, the requirements for day-to-day task management must be balanced against the overall reporting requirements for the program. In essence, program activities are broken down to the lowest level that can be associated with both an organization and a cost account, as illustrated in Figure 6.14. From this, schedules are developed, cost estimates are generated, accounts are established, and program activities are monitored for purposes of schedule/cost control.

In developing the WBS, it is essential that a good comprehensive "WBS dictionary" be prepared. This is a document containing the terminology and definition of each element of the WBS. Traceability must be maintained from the top down, and all applicable work must be included. This is facilitated by assigning a number to each work package in the WBS. In Figure 6.11, the total program is represented by 01-00-00, and the numbers are broken down for activities, functions, tasks, subtasks, and so on. In Figure 6.12, a slightly different numbering system is used. Although the numbering systems will vary for different programs (and with different contractors), it is important to ensure that both activities and budgets/costs can be traced, both upward and downward. In the initial generation of a CWBS by a contractor during the preparation of a proposal, budgets may be allocated downward to specific tasks. After contract award, as tasks are being accomplished, costs are being incurred and charged to the appropriate cost account. These costs are then collected upward for reporting

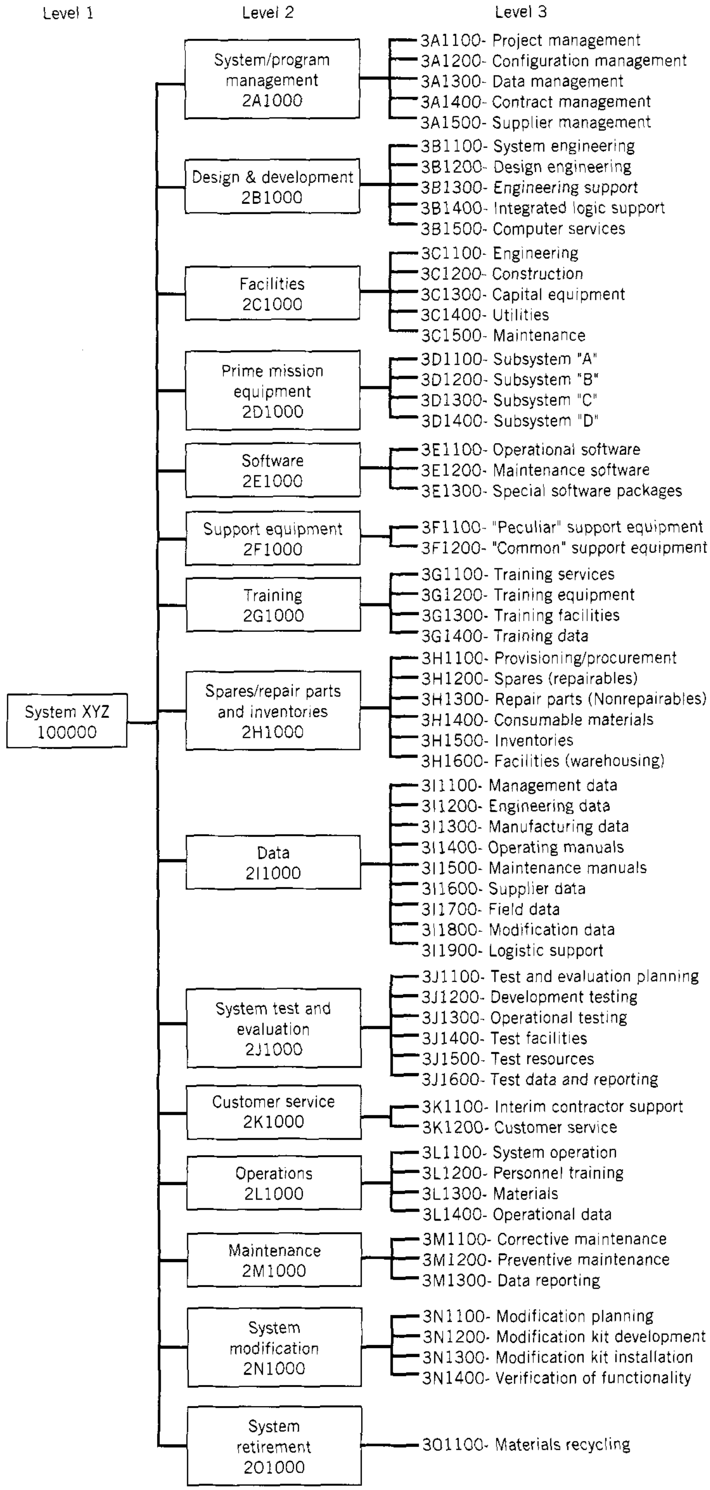

Figure 6.12 Example summary work breakdown structure (SWBS).

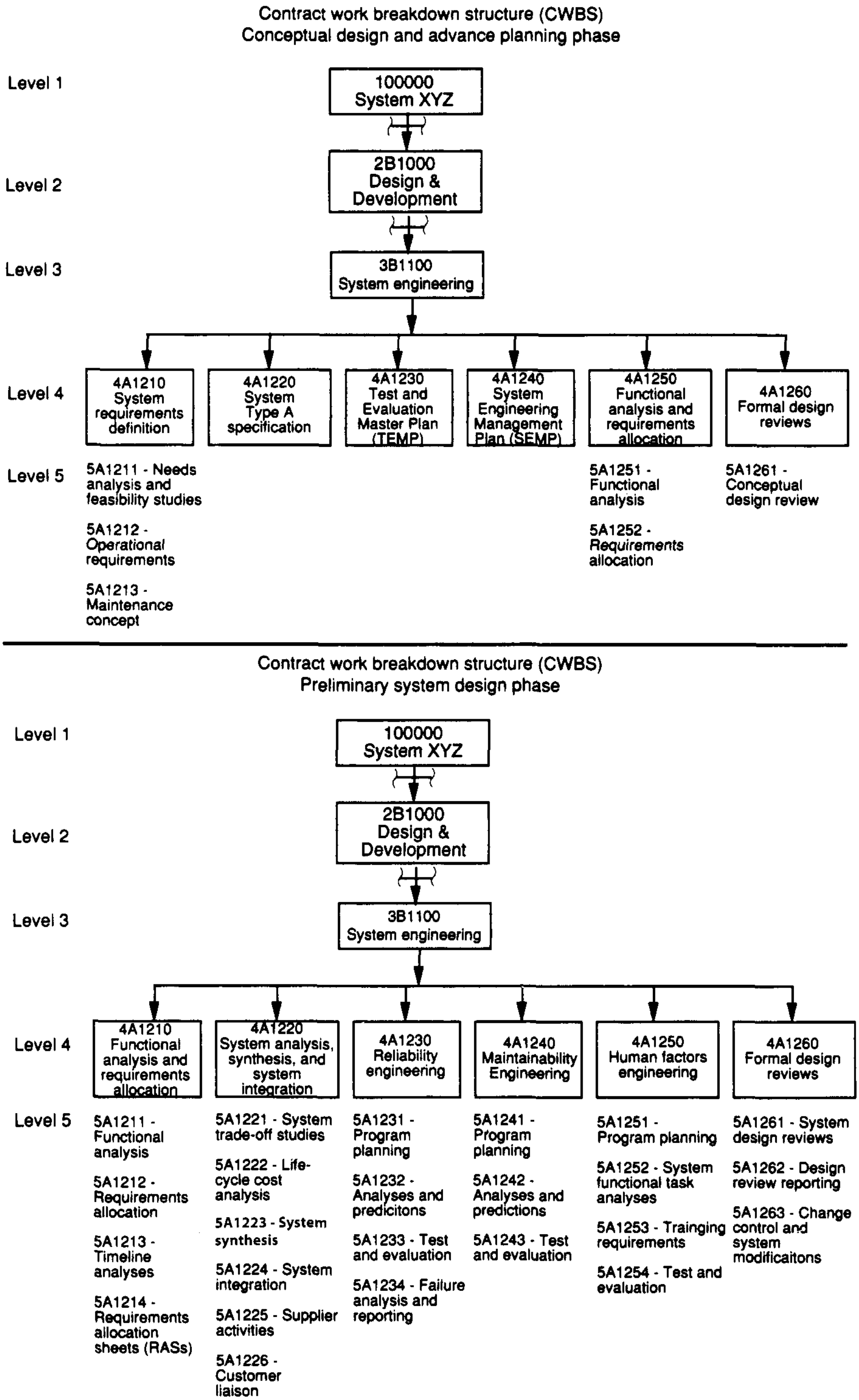

Figure 6.13 CWBS expansion showing system engineering activities.

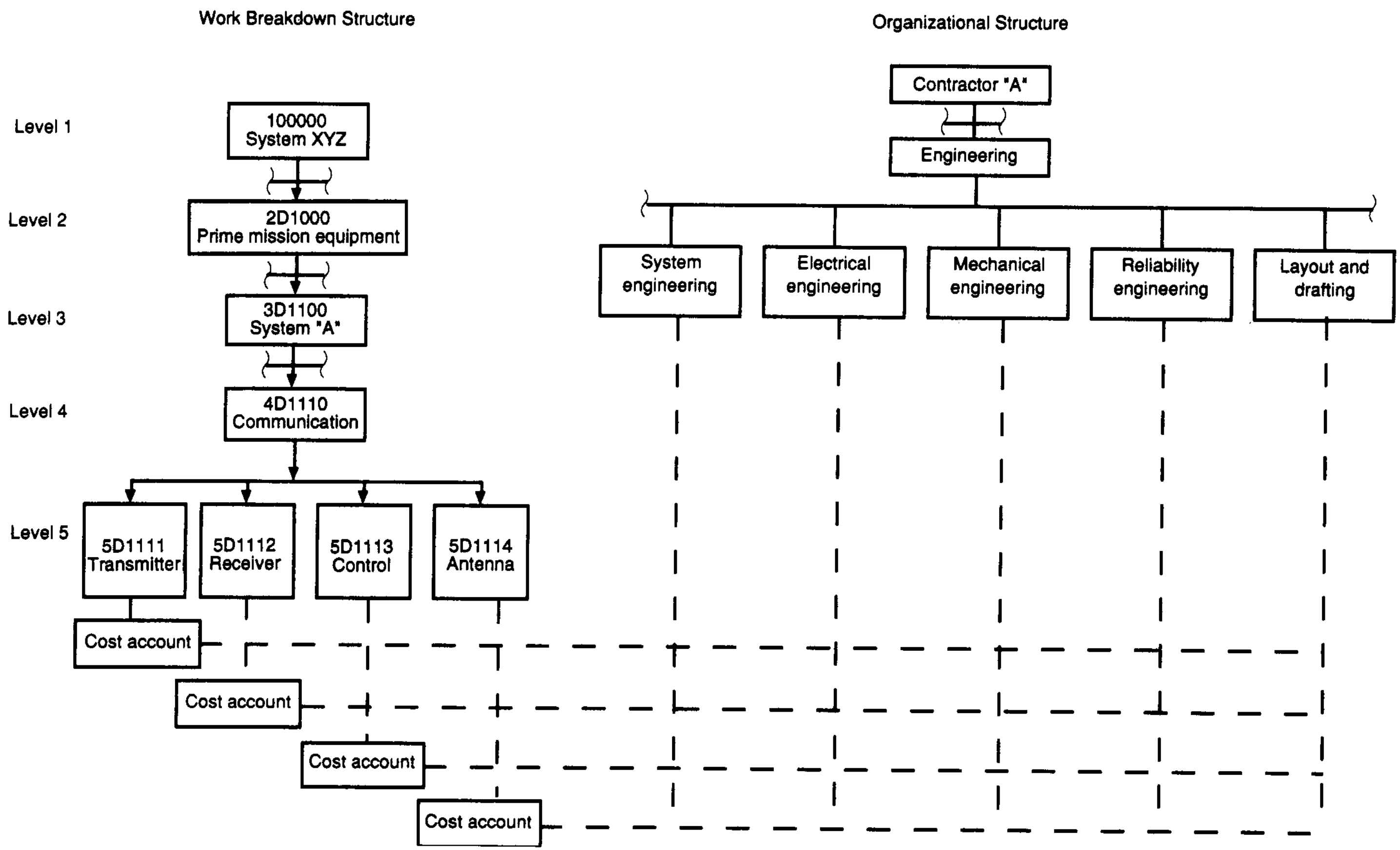

Figure 6.14 Organizational integration with CWBS.

purposes. The WBS provides the vehicle for measuring work package progress in terms of schedule and cost.

In summary, the work breakdown structure (WBS) provides many benefits:

1. The total program, or system, can be easily described through the logical breakout of its elements into nicely definable work packages.
2. The discipline associated with the development of the WBS provides a greater probability that every program activity will be accounted for.
3. The WBS is an excellent vehicle for linking program objectives and activities with available resources.
4. The WBS facilitates the initial allocation of budgets and the subsequent collection and reporting of costs.
5. The WBS provides an excellent matrix for the assignment of tasks and work packages to various organizational departments, groups, and/or sections. Responsibility assignments can be readily identified.
6. The WBS is an excellent vehicle for the reporting of system technical performance measures (TPMs) against schedule and cost.

Finally, the WBS is an excellent tool for the promotion of program communications at various levels. As such, it must be updated to reflect program/system changes, consistent with configuration management actions. Maintaining currency is essential in meeting system engineering objectives.

6.2.5 Specification/Documentation Tree

In Section 3.1 (Figures 3.2 and 3.3), *specifications* are utilized primarily for acquiring items and/or for some element of work; that is, contracting for the design and development of a new item, the procurement of a commercial off-the-shelf (COTS) item, the testing and verification of a product, the construction of a new facility, the production of x quantity of items, and so on. Specifications may be applied on contracts and imposed on major contractors, subcontractors, and the suppliers of goods and services. They are *requirements*-oriented and *performance*-oriented, and they should state the "WHATs" (i.e., *what* to do versus *how* to do it) and must be written in a clear and concise manner. Vague, redundant, nebulous, and ambiguous language should be eliminated. Requirements should be quantifiable and verifiable, and the need to use judgment for interpretation should be avoided; that is, the use of phrases such as "best design practices" and "good workmanship" should be avoided. Specifications establish requirements relative to both design and performance characteristics. Management information, statements of work, procedural data, schedules and cost projections, and so on, should not be included.

Relative to applications, there are (1) general specifications, (2) program-related specifications, (3) military specifications and standards, (4) industrial standards,

(5) specific company standards, and (6) international specifications and standards.[9] The different categories of specifications described in Section 3.1 refer primarily to program-unique specifications, or those that are directed toward a particular program requirement and/or a specific system component. In addition, there are specifications and standards that cover components, materials, and processes across the board, independent of application. In any event, there may be a wide variety of specifications and standards applied on a given program.

In applying specifications, extreme care must be exercised to ensure that they are prepared to the proper depth of detail and applied at the appropriate level in the system hierarchy. Specification documents must be detailed to the extent required to impact design in terms of component selection, the utilization of materials, and the identification of processes. On the other hand, applying specifications with too much detail and at a level too low in the system hierarchy can be extremely detrimental. This may not only tend to inhibit innovation and creativity by not allowing for possible trade-offs, but "overspecification" can be quite costly. Applying a detailed specification to a small commercially available off-the-shelf component may result in an overdesign situation, which, in turn, can significantly increase the cost of that component.[10]

Another concern pertaining to the application of specifications involves possible areas of conflict. Experience shows that conflicts (i.e., contradictions in direction) are sometimes introduced with the application of general specifications and standards across the board. These documents are prepared by different individuals, at different times with different applications in mind, and are not necessarily consistent in terms of detailed requirements. Often in the development of program requirements, there is a tendency to follow the most expeditious and easiest approach by attaching a long list of these specifications and standards to a SOW, with an accompanying statement: "The contractor must comply with the attached list of specifications and standards in fulfilling program requirements." This blind application can result in conflicts pertaining to component part selection, manufacturing process variations, test and evaluation parameters, and so on. In such instances, there is a question as to which specification takes precedence. What are the priorities in order of importance?

With the objective of promoting clarification and eliminating the areas of possible conflict, the preparation of a "specification tree" (or documentation tree) is recommended. This is a family tree of specifications and documents that supports the system hierarchy, establishes order of precedence in the event of conflicts, and relates to the elements of work in the work breakdown structure (WBS). Figure 6.15 illustrates a simplified specification tree.

[9]Industrial standards can vary significantly and are developed by organizations such as the American National Standards Institute (ANSI), International Standards Organization (ISO), American Society for Testing and Materials (ASTM), Electronic Industries Association (EIA), Institute of Electrical and Electronics Engineers (IEEE), and the National Standards Association (NSA).

[10]For example, in a number of instances in the defense sector, military specifications and standards have been imposed on the procurement of small components, hand tools, and so on. The "blind" imposition of specifications on commercially available off-the-shelf items can turn out to be quite costly. Not only will an overdesign situation result, but the number of available suppliers will be reduced.

The "tree" in Figure 6.15 is developed from the top down, commencing with the preparation of the system specification (refer to Section 3.2). Subsequently, additional specifications are applied, following the system hierarchy illustrated in Figure 2.21. As one progresses, the application of specifications must be consistent with the work requirements described by the WBS in Figure 6.11. Further, this application must be adapted to the contracting structure between major contractors, subcontractors, and suppliers (refer to Figure 6.4).

The critical task here is the tailoring of specifications to the particular system application. Even though the design requirements may dictate the use of an available off-the-shelf item, the application of that item in this system may be quite different from comparable applications in other systems. Thus, the major components of the system should be described through a series of program-related specifications, as shown in Figure 6.15; for example, development specifications, product specifications, and process specifications. Below this level, it may be appropriate to apply general specifications, as long as they support the overall requirements in system design. When there are a number of different specifications and/or standards applied to the same system component, they must be complementary and mutually supportive. In the event of conflicts in direction or concerns relative to priorities of importance, the specification tree must provide an indication as to which document takes precedence.

The development of design requirements from the top down is critical in meeting the system engineering objectives stated herein. Thus, extreme care must be exercised in the initial identification and application of specifications and standards. Although this function is sometimes viewed as being relatively minor, the results can be rather costly if the proper level of attention is not directed to this area from the beginning. Conflicts, changes in specification requirements resulting in contractual modifications, and so on, can be extremely detrimental to a program. The inclusion of a complete specification tree in the SEMP may assist in avoiding potential problems later.

6.2.6 Technical Performance Measurement (TPM)

In Section 2.6 (Chapter 2), technical performance measures (TPMs) for the system are identified through the development of operational requirements and the maintenance concept, and are prioritized using quality function deployment (QFD) (or equivalent) methods. Figure 2.10 provides an example of the results. As indicated in the figure, *velocity, availability,* and *size* are the top three in priority. Employing the QFD approach will help in developing the criteria and characteristics in design that must be built-in in order to ensure that the velocity, availability, and size requirements are ultimately met.

As the system development effort progresses, periodic design reviews will be conducted, as described in Chapter 5. The known design configuration at that time will be evaluated with the high-priority TPMs in mind. Checklists may be utilized to aid in the evaluation process, identifying those characteristics that have been incorporated and that relate to and directly support the TPM objectives (refer to Figure 5.4). Design parameters and the applicable TPMs will be measured and "tracked," as

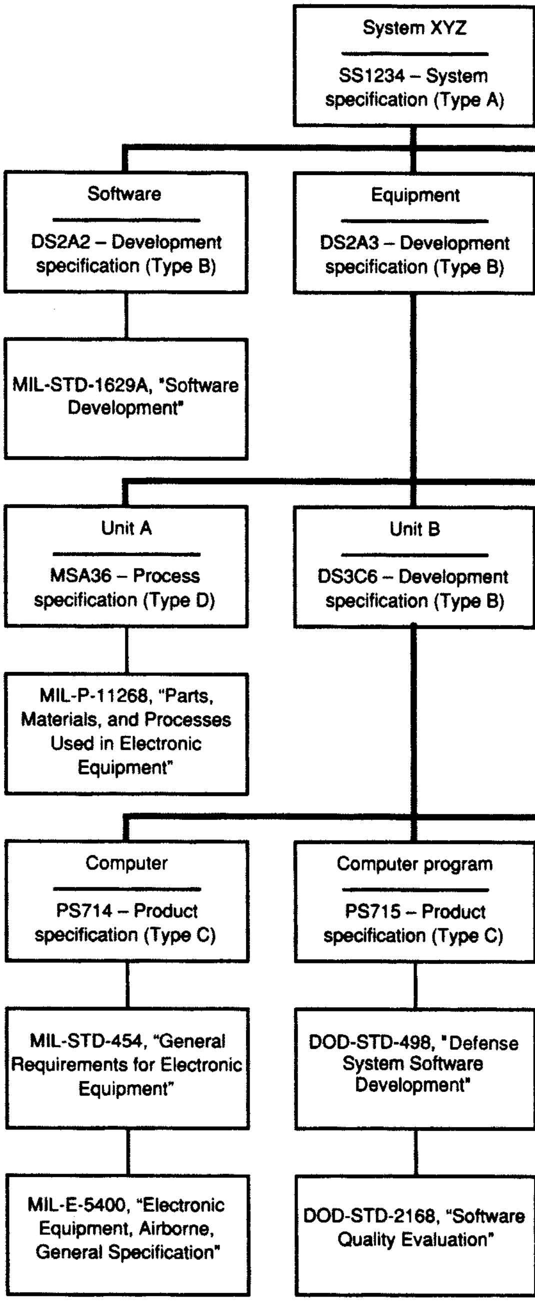

Figure 6.15 Example specification/documentation tree.

Next Page

shown in Figure 5.6. This must be accomplished on a continuing basis, and the results should be included as an inherent part of the regular program management review process. Those high-priority TPMs, assumed to be critical, should receive the most attention in the review and evaluation process.

See Figure 5.6; if there is a deviation from the specified TPM value (either upward or downward), the "causes" for such must be identified and the appropriate corrective action must be initiated accordingly. Inherent within the program management information system structure is the requirement to plot future trends, predict potential deviations, and identify the possible consequences and associated risks in the event that no corrective action is initiated. In essence, technical performance measurement must be built into the regular program management and control process, and the prioritization of TPMs is a necessary input for the risk management plan (refer to 10.0 in the SEMP outline shown in Figure 6.5).

6.2.7 Development of Program Schedules

In line with the Statement of Work (SOW) and the Work Breakdown Structure (WBS), individual program tasks are presented in terms of a time line; that is, a beginning time and an ending time. Schedules are developed to reflect the work requirements throughout all phases of a program.

Schedule planning commences with the identification of major program milestones at the top level and proceeds downward through successively lower levels of detail. A System Engineering Master Schedule (SEMS) is initially prepared, laying out the major program activities on the basis of elapsed time. This serves as the frame of reference for a family of subordinate schedules, developed to cover subdivisions of work as represented by the WBS. Progress against a given schedule is measured at the bottom level, and task status information is related to the appropriate cost account identified by the WBS element and the responsible organization (refer to Figure 6.14).

Program task scheduling may be accomplished using one or a combination of techniques. Some of the more common methods are briefly described in the following paragraphs.

1. *Bar chart:* A simple bar chart presents program activities in terms of sequences and the time span of efforts. Specific milestones and the assignment of resources are not covered. Figure 6.16 illustrates a partial bar chart.

2. *Milestone chart:* A presentation of specific program events (i.e., identifiable outputs) and required start and completion times by calendar date is included. Deliverable items required under contract are noted. Figure 6.17 shows a sample milestone chart.

3. *Combined milestone/bar chart:* The combining of activities and milestones into an overall project schedule is a common approach for many programs. Figure 6.18 presents the primary system engineering tasks, included in Figure 6.6, in a program time line format. This, of course, serves as the basis for the assignment of resources and the development of cost projections.