5

SYSTEMS ENGINEERING MANAGEMENT

5.1 MANAGING SYSTEM DEVELOPMENT AND RISKS

As noted in the first chapter, systems engineering is an integral part of the management of a system development project. The part that systems engineering plays in the project management function is pictured in the Venn diagram of Figure 5.1. The ovals in the diagram represent the domain of *project management* and those of its principal constituents: *systems engineering* and *project planning and control*. It is seen that both constituents are wholly contained within the project management domain, with technical guidance being the province of systems engineering, while program, financial, and contract guidance are the province of project planning and control. The allocation of resources and the definition of tasks are necessarily shared functions.

To better understand the many different functions of systems engineering, this chapter describes some of the main features of the project management framework, such as the work breakdown structure (WBS), project organization, and the systems engineering management plan (SEMP). It also discusses the subject of risk management, the organization of systems engineering effort, and the capability maturity model integrated as it applies to systems engineering.

Systems Engineering Principles and Practice, Second Edition. Alexander Kossiakoff, William N. Sweet, Samuel J. Seymour, and Steven M. Biemer

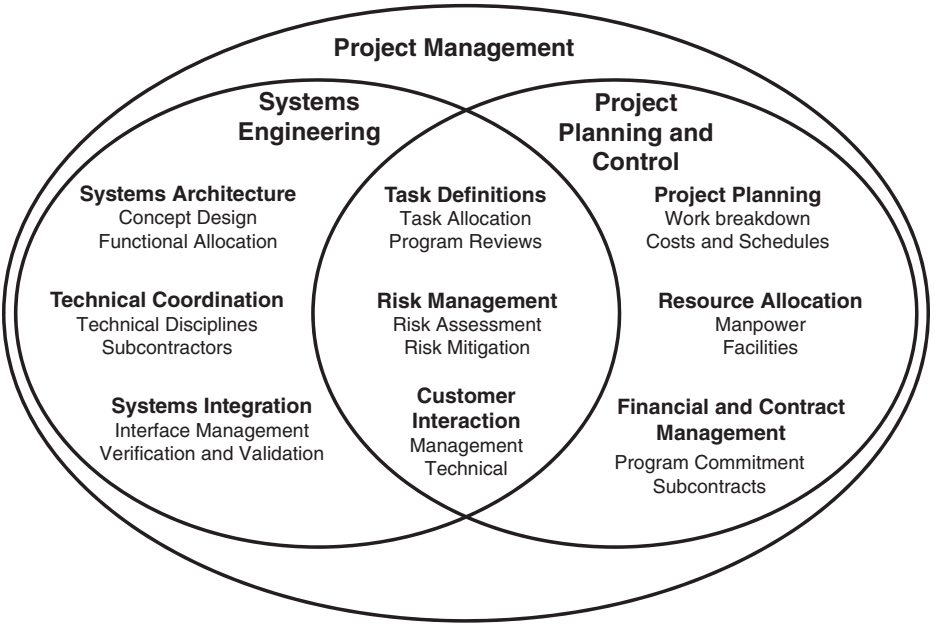


Figure 5.1. Systems engineering as a part of project management.

The engineering of a complex system requires the performance of a multitude of interrelated tasks by dozens or hundreds of people and a number of contractors or other organizational entities. These tasks include not only the entire development process but also usually everything needed to support system operation, such as maintenance, documentation, training, and so on, which must be provided for. Test equipment, facilities, and transportation have to be developed and acquired. The tasks involved in project management and systems engineering, including planning, scheduling, costing, and configuration control, need to be explicitly dealt with.

The sections in this chapter are intended to apply to the management of all systems engineering activities for all types of complex systems. However, in the management of software-intensive systems, in which essentially all of the functionality is performed by software, there are a number of special characteristics that need to be considered. These are noted in Chapter 11, in particular, in the section Software Engineering Management.

Proposal Development and Statement of Work (SOW)

System development often starts with someone who has a need, a customer, who requests support often in the form of a request for proposal (RFP) when in a competitive environment. Following a corporate decision to respond to the RFP, a program manager or a professional proposal team is assigned to generate the proposal. While a systems engineer may not be officially assigned to the team, it is essential that the

technical concepts and implied design and interfaces are feasible. Hence, even in the early phases of a project, the integration of systems engineering with project management is evident.

A critical element of the proposal is the SOW. This is a narrative description of the work that is needed to develop the system to meet the customer needs. The systems engineer concerns will focus on the product to be developed; ensuring the scope of work in the SOW includes all the products and services needed to complete the effort. Specifically, the systems engineer focuses on being responsive to the customer needs, ensures the SOW is based on a credible concept of operations, reviews the implied design for the use of legacy components and their availability, and examines to see if the proposed system integrates commercial off-the-shelf (COTS) components and determines the technology readiness levels for the important subsystems envisioned in the preliminary system design. This early planning sets the stage for the work the technical contributors will have "to live with" throughout the life of the project.

5.2 WBS

The successful management of the system development effort requires special techniques to ensure that all essential tasks are properly defined, assigned, scheduled, and controlled. One of the most important techniques is the systematic organization of project tasks into a form called the *WBS* or, less commonly, the project or system breakdown structure. It defines all of the tasks in terms of goods and services to be accomplished during the project in terms of a hierarchical structure. Its formulation begins early in the concept definition phase to serve as a point of reference for concept trade-off studies. It is then more fully articulated in the latter stages to serve as a basis for system life cycle costing. The WBS is often a contractual requirement in competitive system developments.

The WBS typically defines the whole system to be developed, produced, tested, deployed, and supported, including hardware, software, services, and data. It defines a skeleton or framework on which the project is to be implemented.

Elements of a Typical WBS

The WBS format is generally tailored to the specific project in hand, but always follows a hierarchical tree structure designed to ensure a specific place for every significant portion of work under the project. For purposes of illustration, the following paragraphs describe the main elements of a typical system WBS.

With the system project at level 1 in the hierarchy (some WBS structures begin at Level 0), the level 2 categories may be broken down as follows:

1.1. system product,
1.2. system support,
1.3. system testing,

1.4. project management, and
1.5. systems engineering.

Note that these categories are not parallel in content or scope, but collectively, they are designed to encompass all the work under the system project.

1.1. *System Product* is the total effort required to develop, produce, and integrate the system itself, together with any auxiliary equipment required for its operation. Table 5.1 shows an example of the WBS breakdown of the system product. The level 3 entries are seen to be the several subsystems, as well as the equipment required for their integration (assembly equipment), and other auxiliary equipment used by more than one subsystem. The figure also shows an example of the level 4 and 5 breakdown of one of the subsystems into its

TABLE 5.1. System Product WBS Partial Breakdown Structure

Level 1	Level 2	Level 3	Level 4	Level 5
1. System product				
	1.1 System product			
		1.1.1 Subsystem A		
			1.1.1.1 Component A_1	
				1.1.1.1.1 Functional design
				1.1.1.1.2 Engineering design
				1.1.1.1.3 Fabrication
				1.1.1.1.4 Unit text
				1.1.1.1.5 Documentation
			1.1.1.2 Component A_2	
				1.1.1.2.1 Functional design … (etc.)
		1.1.1 Subsystem B		
			1.1.2.1 Component B_1	
				1.1.2.1.1 Functional design … (etc.)
		1.1.3 Subsystem C		
		1.1.4 Assembly equipment		
		1.1.5 Assembly equipment		

constituent components, which represent definable products of development, engineering, and production effort. It is preferred that integration and test of hardware and software component is done separately for each subsystem, and then the tested subsystems are integrated in the final system for testing (1.3 below). Finally, for cost allocation and control purposes, each component is further broken down at level 5 into work packages that define the several steps of the component's design, development, and test. From this level and below the WBS, elements are often expressed with action words, for example, purchase, design, integrate, and test.

1.2. *System Support* (or integrated logistic support) provides equipment, facilities, and services necessary for the development and operation of the system product. These items can be categorized (level 3 categories) under six headings:

1.2.1. Supply support

1.2.2. Test equipment

1.2.3. Transport and handling

1.2.4. Documentation

1.2.5. Facilities

1.2.6. Personnel and training

Each of the system support categories applies to both the development process and system operation, which may involve quite different activities.

1.3. *System Testing* begins after the design of the individual components has been validated via component tests. A very significant fraction of the total test effort is usually allocated to system-level testing, which involves four categories of tests as follows:

1.3.1. *Integration Testing.* This category supports the stepwise integration of components and subsystems to achieve a total system.

1.3.2. *System Testing.* This category provides for overall system tests and the evaluation of test results.

1.3.3. *Acceptance Testing.* This category provides for factory and installation tests of delivered systems.

1.3.4. *Operational Testing and Evaluation.* This category tests the effectiveness of the entire system in a realistic operational environment.

Individual tests to be performed at each level are prescribed in a series of separate test plans and procedures. However, an overall description of test objectives and content and a listing of the individual tests to be performed should also be set forth in an integrated test planning and management document, the "test and evaluation management plan" (TEMP) in defense acquisition terminology. Chapter 13 is devoted to the subject of system integration and evaluation.

1.4 *Project Management* tasks include all activities associated with project planning and control, including the management of the WBS, costing, scheduling,

performance measurement, project reviews and reports, and associated activities.

1.5 *Systems Engineering* tasks include the activities of the systems engineering staff in guiding the engineering of the system through all its conceptual and engineering phases. This specifically includes activities such as requirements analysis, trade-off studies (analysis of alternatives), technical reviews, test requirements and evaluations, system design requirements, configuration management, and so on, which are identified in the SEMP. Another important activity is the integration of specialty engineering into the early phases of the engineering effort, in other words, concurrent engineering.

The WBS is structured so that every task is identified at the appropriate place within the WBS hierarchy. Systems engineering plays an important role in helping the project manager to structure the WBS so as to achieve this objective. The use of the WBS as a project-organizing framework generally begins in the concept exploration phase. In the concept definition phase, the WBS is defined in detail as the basis for organizing, costing, and scheduling. At this point, the subsystems have been defined and their constituent components identified. Also, decisions have been made, at least tentatively, regarding outside procurement of elements of the system. Accordingly, the level down to which the WBS needs to be defined in detail should have been established.

It is, of course, to be expected that the details of the WBS evolve and change as the system is further engineered. However, its main outline should remain stable.

Cost Control and Estimating

The WBS is the heart of the project cost control and estimating system. Its organization is arranged so that the lowest indenture work packages correspond to cost allocation items. Thus, at the beginning of the project, the target cost is distributed among the identified work packages and is partitioned downward as lower-level packages are defined. Project cost control is then exercised by comparing actual reported costs against estimated costs, identifying and focusing attention on those work packages that deviate seriously from initial estimates.

The collection of project costs down to the component level and their distribution among the principal phases of project development, engineering, and fabrication is essential also for contributing to a database, which is used by the organization for estimating the costs of future projects. For new components, cost estimates must be developed by adapting the previously experienced costs of items directly comparable to those in the projected system, at the lowest level of aggregation for which cost figures are available. At higher levels, departures from one system to the next become too large to reliably use data derived from previous experience without major correction.

It should not be expected that the lowest indenture level would be uniform throughout the various subsystems and their components. For example, if a subsystem is being obtained on a fixed price subcontract, it may well be appropriate to terminate the lowest

indenture in the WBS at that subsystem. In general, program control, including costing, is exercised at the level at which detailed specifications, interface definitions, and work assignments are available, representing in effect a contract between the project and the organization charged with the responsibility for developing, engineering, or fabricating given elements of the system.

Critical Path Method (CPM)

Network scheduling techniques are often used in project management to aid in the planning and control of the project. Networks are composed of events and activities needed to carry out the project. Events are equivalent to a milestone indicating when an activity starts and finishes. Activities represent the element of work or task, usually derived from the WBS that needs to be accomplished. Critical path analysis is an essential project management tool that traces each major element of the system back through the engineering of its constituent parts. Estimates are made of not only the size but also the duration of effort required for each step. The particular path that is estimated to require the longest time to complete its constituent activities is called the "critical path." The differences between this time and the times required for other paths are called "slack" for those paths. The resulting critical path network is a direct application of the WBS. The systems engineer uses the CPM to understand the dependences of task activities, to help prioritize the work of the technical teams, and to communicate graphically the work of the entire program.

5.3 SEMP

In the development of a complex system, it is essential that all of the key participants in the system development process not only know their own responsibilities but also know how they interface with one another. Just as special documentation is required to control system interfaces, so the interfacing of responsibilities and authority within the project must also be defined and controlled. This is usually accomplished through the preparation and dissemination of a SEMP or its equivalent. The primary responsibility of creating such a plan for guiding the engineering effort is that of the systems engineering component of project management.

The importance of having formalized plans for managing the engineering effort has been recognized in defense acquisition programs by requiring the contractor to prepare a SEMP as part of the concept definition effort. The most important function of the SEMP is to ensure that all of the many active participants (subsystem managers, component design engineers, test engineers, systems analysts, specialty engineers, subcontractors, etc.) know their responsibilities to one another. This is an exact analogue of the component interface function of systems engineering defining the interactions among all parts of the system so that they fit together and operate smoothly. It also serves as a reference for the procedures that are to be followed in carrying out the numerous systems engineering tasks. The place of the SEMP in the program management planning is shown in Figure 5.2.

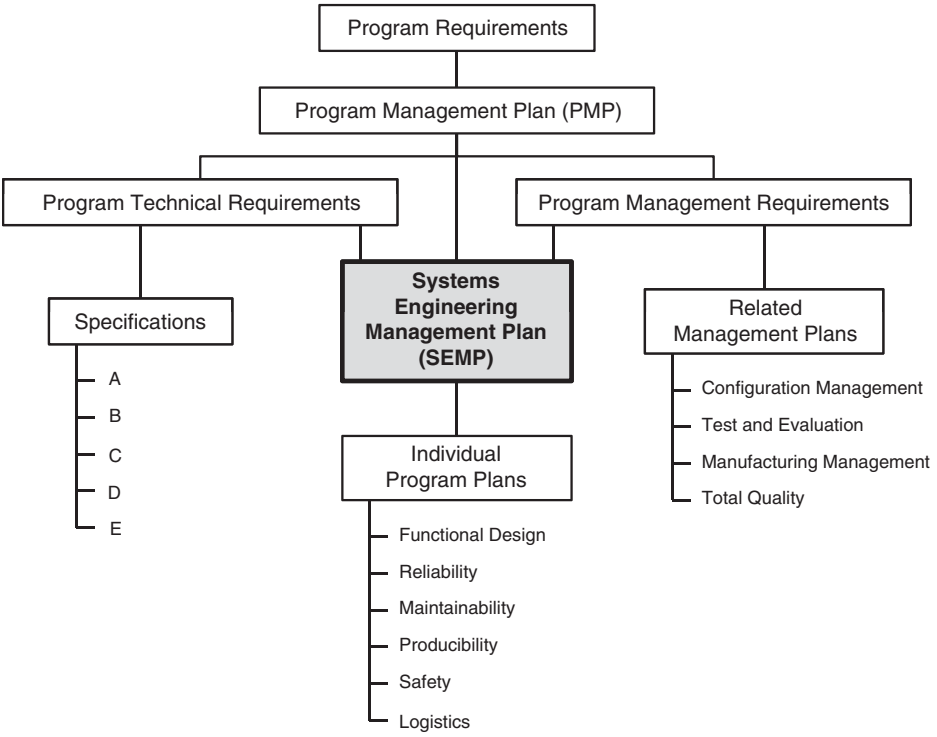


Figure 5.2. Place of SEMP in program management plans.

The SEMP is intended to be a living document, starting as an outline, and being elaborated and otherwise updated as the system development process goes on. Having a formal SEMP also provides a control instrument for comparing the planned tasks with those accomplished.

Elements of a Typical SEMP

The SEMP contains a detailed statement of how the systems engineering functions are to be carried out in the course of system development. It can be considered to consist of three types of activity:

1. *Development Program Planning and Control:* describes the systems engineering tasks that must be implemented in managing the development program, including
   - statements of work;
   - organization;
   - scheduling;

- program, design, and test readiness reviews;
- technical performance measurement; and
- risk management.

2. *Systems Engineering Process:* describes the systems engineering process as it applies to the development of the system, including
   - operational requirements,
   - functional analysis,
   - systems analysis and trade-off strategy, and
   - system test and evaluation strategy.
3. *Engineering Specialty Integration:* describes how the areas of specialty engineering are to be integrated into the primary system design and development, including
   - reliability, maintainability, availability (RMA) engineering;
   - producibility engineering;
   - safety engineering; and
   - human factors engineering.

A typical SEMP outline is tailored to the development system but could include the following:

Introduction
  Scope, Purpose, Overview, Applicable Documents
Program Planning and Control
  Organizational Structure
  Responsibilities, Procedures, Authorities
  WBS, Milestones, Schedules
  Program Events
  Program, Technical, Test Readiness Reviews
  Technical and Schedule Performance Metrics
  Engineering Program Integration, Interface Plans
Systems Engineering Process
  Mission, System Overview Graphic
  Requirements and Functional Analysis
  Trade Studies (Analysis of Alternatives)
  Technical Interface Analysis/Planning
  Specification Tree/Specifications
  Modeling and Simulation
  Test Planning
  Logistic Support Analysis
  Systems Engineering Tools

Engineering Integration
- Integration Design/Plans
- Specialty Engineering
- Compatibility/Interference Analysis
- Producibility Studies

5.4 RISK MANAGEMENT

The development of a new complex system by its nature requires acquiring knowledge about advanced but not fully developed devices and processes so as to wisely guide the system design to a product that performs its intended mission reliably and at an affordable cost. At every step, however, unpredictable outcomes can be encountered that pose risks of performance shortfalls, environmental susceptibility, unsuitability for production, or a host of other unacceptable consequences that may require a change in course with impacts on program cost and schedule. One of the greatest challenges to systems engineering is to steer a course that poses minimum risks while still achieving maximum results.

At the outset of the development, there are uncertainties and hence risks in every aspect. Are the perceived operational requirements realistic? Will they remain valid throughout the new system's operational life? Will the resources required to develop and produce the system be available when needed? Will the advanced technology necessary to achieve the required operational goals perform as expected? Will the anticipated advances in production automation materialize? Will the development organization be free from work stoppages?

It is the special task of systems engineering to be aware of such possibilities and to guide the development so as to minimize (mitigate) their impact if and when they may occur. The methodology that is employed to identify and minimize risk in system development is called *risk management*. It has to begin at the outset of the system development and progress throughout its duration.

Risk Reduction through the System Life Cycle

Reducing program risks is a continual process throughout the life cycle. For example, the needs analysis phase reduces the risk of embarking on the development of a system that does not address vital operational needs. The concept exploration phase reduces the risk of deriving irrelevant or unrealistic system performance requirements. And the system definition phase selects a system concept that utilizes technical approaches that are neither excessively immature nor unaffordable, but rather one that has the best chance of meeting all system goals.

Figure 5.3 is a schematic representation of how the program risk of a hypothetical system development (in arbitrary units) decreases as the development progresses through the phases of the life cycle. The abscissa is time, sectioned into the phases of system development. In the same figure is plotted a curve of the typical relative effort expended during each phase.

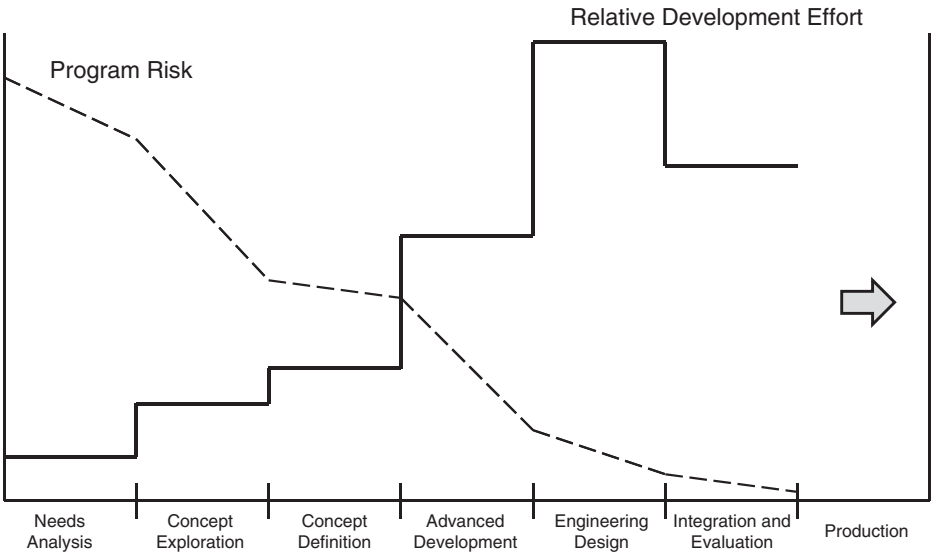


Figure 5.3. Variation of program risk and effort throughout system development.

The descending risk curve conveys the fact that as development progresses, uncertainties (unknowns), which constitute risks of unforeseen adverse events, are systematically eliminated or reduced by analysis, experiment, test, or change in course. A variant of this curve is referred to as the "risk mitigation waterfall" (Figure 5.4). The ascending effort curve represents the stepwise increases in the costs of succeeding phases of system development, showing the progression of activity from conceptual to engineering to integration and evaluation.

Figure 5.3 is intended to illustrate several key principles:

1. As the development progresses, the investment in program effort typically rises steeply. To maintain program support, the risk of failure must be correspondingly reduced so as to maintain the financial risk at reasonable levels.
2. The initial stages in the program produce major reductions in risk, when the basic decisions are made regarding the system requirements and the system concept. This demonstrates the importance of investing adequate effort in the formative phases.
3. The two phases that typically produce the greatest risk reduction are *concept exploration* and *advanced development*. Concept exploration provides a solid conceptual basis for the system approach and architecture. Advanced development matures new advanced technologies to insure their meeting performance goals.
4. By the time the development is complete and the system is ready for production and distribution, the residual level of risk must be extremely low if the system is to be successful.

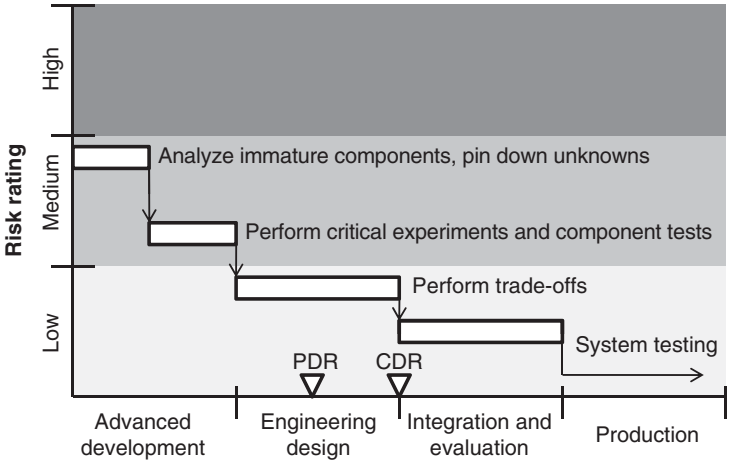


Figure 5.4. Example of a risk mitigation waterfall chart. PDR, Preliminary Design Review; CDR, Critical Design Review.

Components of Risk Management

Risk management is formally recognized in systems engineering standards, and especially in government acquisition programs. Each program is expected to prepare a risk management plan. Risk management for a major system is expected to have its own organization, staffing, database, reporting, and independent review, and to extend to all phase of program development, production, operation, and support. A detailed description of risk management as defined by the DoD is contained in the *Risk Management Guide for DoD Acquisition* published by the Defense Acquisition University.

The *Risk Management Guide* divides the subject of risk management into risk planning, risk assessment, risk prioritization, risk handling, and risk monitoring. The discussion to follow will combine these into two categories: *risk assessment*, which will include risk planning and prioritization, and *risk mitigation*, which will include risk handling and monitoring. The subject of risk planning is addressed by the risk management plan, which is part of the SEMP.

Risk Assessment

The general process of risk assessment is inherent in all decisions involving prospective uncertainty. As will be described in Chapter 10, risk assessment is used to eliminate alternative concepts that are overly dependent on immature technologies, unproven technical approaches, or other ambitious advances that do not appear to be warranted by their projected benefits compared to the uncertainty of their realization. Some of the more common sources of program risk are listed in Chapter 12.

In the advanced development phase, risk assessment will be seen to be a useful approach to the identification and characterization of proposed design features that represent a sufficient development risk (i.e., likelihood of failing to meet requirements)

and a significant program impact to warrant analysis and, if necessary, development and test. Thus, risk assessment identifies the weakest and most uncertain features of the design and focuses attention on means for eliminating the possibility that these features will present complications and will require design changes during the subsequent phases of development.

Once the system components possessing questionable design features have been identified, the task of systems engineering is to define a program of analysis, development, and test to eliminate these weaknesses or to take other actions to reduce their potential danger to the program to an acceptable level. In this process, the method of risk assessment can be of further value by providing a means for determining how to best allocate available time and effort among the identified areas of risk. For this purpose, risk assessment can be applied to judge the relative risks posed by the design features in question.

To compare the potential importance of different sources of program risk, it is necessary to consider two risk components: the *likelihood* that a given component will fail to meet its goals and the *impact* or *criticality* of such a failure to the success of the program. Thus, if the impact of a given failure would be catastrophic, even a low likelihood of its occurring cannot be tolerated. Alternatively, if the likelihood of failure of a given approach is high, it is usually prudent to take a different approach even if its impact may be low but significant.

These risk components are often displayed in the form of a "risk cube" typically of three or five dimensions. The five-dimension cube is shown in Figure 5.5, and the three-dimension cube is discussed below. Since the probabilities are usually qualitative in nature, experienced judgment is needed to develop an informed assignment of risk. The relative nature is also important to understand since work in foundational research areas is naturally more risky than work that is developing a system to well-defined specifications. The risk tolerance of customers will also vary by domain and experience.

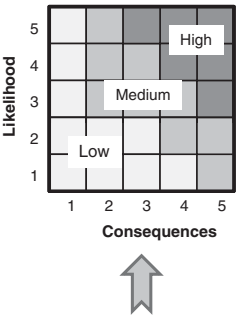
Risk Likelihood: Probability of Failure. There are too many uncertainties to be able to compute a numerical value for the likelihood that a specific program goal will be achieved, and hence it is not useful to attempt to quantify risks beyond a relatively rough measure to assist in their relative prioritization.

In the case of unproven technology, it is possible to estimate very roughly the relative degree of maturity from the engineering status of the technology. This may be carried out by identifying one or more cases where the technology is used in connection with a similar functional application and by determining its level of development (e.g., in the range from a laboratory design to an experimental prototype to a qualified production component). High, medium, and low risk is about as fine a scale as is normally useful. Beyond that, it is good practice to rank order the parts of the system that appear to be risky and to concentrate on the few that are judged to be most immature and complex. If the candidates are numerous, it may be a sign that the entire system design approach is too ambitious and should be reconsidered.

Risks associated with highly complex components and interfaces are even more difficult to quantify than those using advanced technology. Interfaces always require

Derived from
Risk Management—A Process Overview by Bob Skalamera

Likelihood	What is the likelihood the risk will happen?		
	Level		Your approach and processes …
	1	Not likely	… Will effectively avoid or mitigate this risk based on standard practices
	2	Low likelihood	… Have usually mitigated this type of risk with minimal oversight in similar cases
	3	Likely	… May mitigate this risk, but work-arounds will be required
	4	Highly likely	… Cannot mitigate this risk, but a different approach might
	5	Near certainty	… Cannot mitigate this type of risk; no known processes or work-arounds are available



Consequences	Given the risk is realized, what would be the magnitude of the impact?		
Level	Technical	Schedule	Cost
1	Minimal or no impact	Minimal or no impact	Minimal or no impact
2	Minor performance shortfall, same approach retained	Additional activities required, able to meet key dates	Budget increase or unit production cost increase <1%
3	Moderate performance shortfall, but work-arounds available	Minor schedule slip, will miss needed dates	Budget increase or unit production cost increase <5%
4	Unacceptable, but work-arounds available	Project critical path affected	Budget increase or unit production cost increase <10%
5	Unacceptable; no alternatives exist	Cannot achieve key project milestones	Budget increase or unit production cost increase >10%

Figure 5.5. An example of a risk cube display.

special attention, especially in human–machine interactions. The latter always warrant early prototyping and testing. Here again, rank ordering of the relative complexity is an effective way of prioritizing the effort required for risk management.

The prioritization of software risks is again a matter of judgment. Real-time programs with numerous external interrupts always require special attention, as do concurrent processes. New or significantly altered operating systems can be particularly complicated. Programs with high logic content tend to be more likely to malfunction as a consequence of undetected faults than those that are largely computational.

Table 5.2 lists some of the considerations discussed above in arriving at a general prioritization of risk probabilities.

Risk Criticality: Impact of Failure. It was stated earlier that the seriousness of the risk of a particular failure might be considered in terms of two factors—the likelihood that a failure will occur and the criticality of its impact on the success of the program. In a semiquantitative sense, the seriousness of the risk can be thought of as a combination of those two factors.

As in the case of risk likelihood, there is no accepted numerical scale for risk criticality, and one may consider the same relative levels as those for likelihood: high, medium, or low. Some agreed-upon definitions need to be assigned to these levels, such as those listed in Table 5.3.

TABLE 5.2. Risk Likelihood

Risk likelihood	Design status
High	• Significant extension from past designs • Multiple new and untried components • Complex components and/or interfaces • Marginal analytical tools and data
Medium	• Moderate extension from past designs • Components complex but not highly stressed • Analytical tools available
Low	• Application of qualified components • Components of medium complexity • Mature technologies and tools

TABLE 5.3. Risk Criticality

Criticality	System impact	Program impact
High	• Major degradation in performance (50–90%) • Serious safety problem	• Major increase in cost and/or schedule (30–70%) • Production cutbacks
Medium	• Significant degradation in performance (10–50%) • Short losses of operability • Costly operational support	• Significant increases in cost and/or schedule (10–30%) • Intense reviews and oversight • Production delays
Low	• Minor degradation in performance (>10%) • Occasional brief delays • Increased maintenance	• Minor increase in cost and/or schedule (<10%) • Vigorous reviews and oversight

The middle column of the table lists expected impacts on system operation if the system component at risk failed to perform its function. The right column lists the types of impacts on the overall program that could be expected if the system component was discovered to be faulty late in development and indicates the likely effects on the program.

While some systems engineering textbooks advocate the derivation of an overall risk factor by assigning numerical values to the estimates of risk likelihood and risk criticality and taking their product, the disadvantages of this practice are believed to outweigh the presumed advantage of a seemingly simple single risk factor. In the first place, assignment of numerical estimates creates the illusion of quantitative knowledge, which has no real basis. In the second place, combining the two indices into one has the effect of diminishing the net information content, as was noted in connection with combining figures of merit of individual parts of the system into a single score. Accordingly, it is recommended that the individual ratings be retained as abstractions,

such as high, medium, and low, and the two components retain their identity, such as medium–low, and so on.

In connection with the criticality scale, the highest level of criticality listed in Table 5.3 stops short of including the case of near total loss in system performance resulting in mission failure. Such an eventuality would likely risk program cancellation, and as such would be considered unacceptable. This implies that design risks of this degree of criticality would not be considered as feasible options.

Role of Systems Engineering. The task of risk assessment (and the subsequent task of risk management) is clearly the responsibility of systems engineering. This is because the judgments that are involved require a breadth of knowledge of system characteristics and the constituent technologies beyond that possessed by design specialists, and also because judgments of risk criticality are at the system and program levels. The process of risk assessment thus helps the systems engineer to identify the system features that need to be most thoroughly understood and raised to a level of design maturity suitable for full-scale engineering.

Risk Mitigation

The most common methods of dealing with identified program risks are the following, listed in order of increasing seriousness of the perceived risk:

1. intensified technical and management reviews of the engineering process,
2. special oversight of designated component engineering,
3. special analysis and testing of critical design items,
4. rapid prototyping and test feedback,
5. consideration of relieving critical design requirements, and
6. initiation of fallback parallel developments.

Each of the above methods is briefly described below.

Technical and Management Reviews. Formal design reviews may address entire subsystems, but the depth of coverage is mainly on design aspects considered of greatest importance. It is the responsibility of systems engineering to ensure that the significant risk items are fully presented and discussed so that special management attention and resources may be directed to issues warranting additional effort. The aim should be to resolve problems at the earliest possible time, so full disclosure of experienced or anticipated difficulties is essential. The process of design reviews is further described in the Component Design section of Chapter 12 (Section 12.4).

Oversight of Designated Component Engineering. Regularly scheduled design reviews are neither frequent enough nor detailed enough to provide adequate oversight of known design problem areas. Each designated problem area should be assigned a special status, subjected to appropriately frequent reviews, and overseen by

designated senior design and systems engineers. Where appropriate, outside consultants should be engaged in the process. A risk mitigation plan should be prepared and tracked until the problem areas are resolved.

Special Analysis and Testing. For components whose design involves issues not resolved in the advanced technology phase, additional analysis and, if necessary, fabrication and test should be carried out to obtain sufficient design data to validate the technical approach. This will require assigning additional resources and modifying the engineering schedule to accommodate the results of such analysis and testing.

Rapid Prototyping. For unproven components for which analysis and limited testing cannot adequately validate the design, it may be necessary to construct and test prototypes to ensure their validity. Such action would normally be taken in the advanced technology phase, but sometimes, the problem is not recognized at that time, and in other cases, the action fails to resolve the problem.

Relief of Excessive Requirements. Experience has shown that attempting to meet all initially posed requirements often fails to achieve a practical overall solution and requires an adjustment of some performance or compatibility requirement. This possibility should be explored whenever efforts to meet fully a requirement result in a solution that is inordinately complex, costly, unreliable, or otherwise undesirable from a practical standpoint. This problem is uniquely a task for systems engineering since all factors of performance, cost, and schedule need to be considered together. It is an option that should be invoked only in exceptional cases, but neither should it be put off until excessive resources and time have been committed to vain efforts to fulfill the requirement.

Fallback Alternatives. The development of alternative design approaches is most appropriate for components using new technology whose successful development cannot be fully assured. In such cases, adequate alternative approaches should be established during the advanced development phase to serve as fallbacks in the event that the new designs do not fulfill expectations. Such fallback alternatives almost always result in reduced performance, greater cost, or some other perceived deficiency compared to the selected approach, but are more conservative in their design and hence are more certain to succeed.

It happens not infrequently that the engineering design phase begins before a clear resolution is reached as to the ultimate success of a given technical approach, and hence before a final decision as to whether or not to fall back to a more conservative alternative. In such cases, an expedited program to reach such a decision by further development, analysis, and test must be invoked. Again the decision is one for systems engineering. Often the choice also involves reexamination of the initial requirements, as discussed in the previous paragraphs.

The above methods may be applied singly, but most often work best in combination. Their oversight is a program manager's responsibility, and their planning and direction are a systems engineering function.

TABLE 5.4. Sample Risk Plan Worksheet

Risk title:	Project name:
Risk owner:	Last updated:
Team:	
Date submitted:	

Description of risk:

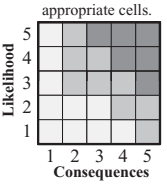
Statement of basic cause:

Consequence if risk is realized:

Risk type:

☐ Technical
☐ Schedule
☐ Cost
☐ Other

Place X, 1, 2, … in the appropriate cells.



Risk reduction plan

Action/milestone event.	Date	Success criteria	Risk level if successful L	C	Comments
1.					
2.					
3.					
4.					

Risk Management Plan

The importance of the actions described above to the overall success of a system development requires that it be part of the overall program management process. To this end, a formal risk management plan should be developed and progressively updated, in which mitigation is a major part.

For every significant risk, there should be a plan that minimizes its potential impact through specific actions to be taken, either concurrently with the engineering or to be invoked should the anticipated risk materialize. The formulation of such a plan must be predicated on the objective of minimizing the total expected program cost, which means that the planned activities to contain program risks must not be more costly than the expected impact of the risks, should they eventuate. For items for which a fallback approach is to be developed, the plan should define the conditions under which the backup will be activated, or if activated at the outset, how far it is to be carried in the absence of evidence that the main approach will prove unsatisfactory. A diagram of a risk mitigation plan known as a "risk mitigation waterfall chart" is shown in Figure 5.4. An example of a risk plan worksheet is pictured in Table 5.4.

5.5 ORGANIZATION OF SYSTEMS ENGINEERING

Despite decades of study, there are many opinions, but no general agreement, on which the organizational form is most effective for a given type of enterprise. For this reason,

the organizations participating in a system development project are likely to employ a variety of different organizational styles. Each individual style has evolved as a result of history, experience, and the personal preferences of upper management. Accordingly, despite its central importance to the success of a given system development project, the systems engineering function will usually need to adapt to preexisting organizational structures.

Virtually all system development projects are managed by a single industrial company. Hence, it is the organizational form of this company that drives the organization of systems engineering. In most cases, this company will develop some subsystems in-house, and contract for other subsystems with subcontractors. We will refer to the first company as the prime contractor or system contractor, and to the collection of participating contractors as the "contractor team." This means that the systems engineering activity must span not only a number of different disciplines but also several independent companies.

The organizational structure of the prime contractor is usually some form of a "matrix" organization. In a matrix organization, most of the engineering staff is organized in discipline- or technology-oriented groups. Major projects are managed by project management teams reporting to a "vice president for project management" or an equivalent. At times, these teams are called integrated product teams (IPTs) (see Chapter 7). A technical staff is assigned to individual projects as required, but employees retain affiliation with their engineering groups.

The main variations in matrix-type organizations relates to whether the bulk of the technical staff assigned to a project are physically relocated to an area dedicated to the project and remain as full-time participants throughout much of the development or whether they remain in their home group areas. A related difference is the degree to which authority for the direction of the technical work assignments is retained by their home group supervisors.

As stated earlier, the organization of the systems engineering function is necessarily dependent on the system contractor's organizational structure. There should be some common practices, however. Referring to Figure 5.1, a major system project should have a single focus of responsibility for the systems engineering function (a project systems engineer), as apart from the project planning and control function. As an integral part of project management, an appropriate title might be "associate (or deputy) project manager for systems engineering" or, more simply, "systems engineering manager." Since the systems engineering function is that of guidance, authority is exercised by establishing goals (requirements and specifications), formulating task assignments, conducting evaluations (design reviews, analyses, and tests), and controlling the configuration.

Effective technical communications are difficult to maintain in any organization for a variety of reasons, many of them inherent in human behavior. They are, nevertheless, absolutely vital to the ultimate success of the development project. Perhaps the single most important task of the project systems engineer is to establish and maintain effective communication among the many individuals and groups, inside and outside the company, whose work needs to interact with others. This is a human interface function corresponding to the system physical interface functions that make the system

elements fit together and operate as one. Since the systems engineer usually works in parallel with rather than through established lines of authority, he or she must exercise extraordinary leadership to bring together those individuals who need to interact.

There are several different means of communication, all of which need to be exercised as appropriate:

1. All key participants need to know what they are expected to do, when, and why: the "what" is expressed in task assignments and WBS; the "when" is contained in schedules, milestones, and critical path networks; and the "why" should be answered in the requirements and specifications. A clear and complete statement of the "why" is essential to ensure that the designers, analysts, and testers understand the objectives and constraints of the task assignments.
2. Participants must be aware of how their portions of the system interact with other key elements and of the nature of their mutual interdependence. Such interactions, and particularly their underlying causes, can never be sufficiently covered in specification documents. This awareness can only be provided by periodic personal communication among the responsible participants and the documentation of any resulting agreements, interface definitions, and so on, however small or tentative. Systems engineering must provide the glue that binds these items of system design through the formation of *interface working groups* and the development of *interface control documents*, and such less formal communications as may be needed in special cases.
3. Subcontractors and other key participants at remote sites must be integrated into the project communication framework. At the management level, this is the task of the system project manager, but at the engineering level, it is the responsibility of the project systems engineering staff. It is essential that the same two coordinating functions described above be provided for the entire contractor team. For this purpose, conventional formal contractual mechanisms never suffice and sometimes hinder. Accordingly, special efforts should be made to integrate the team members effectively into the total system development effort. This needs to be carried out at two levels: (1) periodic program management reviews attended by top-level representatives of the contractor team and (2) frequent technical coordination meetings concerned with specific ongoing aspects of the program.
4. The principal leaders of the system design effort must have a regular and frequent means of communication with one another to keep the program closely coordinated and to react quickly to problems. This is discussed in the following paragraphs.

Systems Analysis Staff

An essential part of any systems engineering organization is a highly competent and experienced analytical staff. Such a staff need not be a single entity, nor does it need to be organizationally colocated with the project staff itself, but it must be part of the

systems engineering organization, at least during the conceptual and early engineering phases of the project. The systems analysis staff must have a deep understanding of the system environment, with respect to both its operational and physical characteristics. In both instances, it must be able to model the system environment, by use of mathematical and computer models, to provide a basis for analyzing the effectiveness of system models. In the concept exploration phase, the systems analysis staff is the source of much of the quantitative data involved in defining the system performance required to meet its operational requirements. In the concept definition phase, the analysis staff is responsible for constructing the system simulations used in the trade-off studies and in the selection of the best system concept. Throughout the engineering development stage of the program, the analysis staff is involved in numerous component trade-off studies. It conducts test analyses to derive quantitative measures of the performance of system prototypes and contributes to defining the quantitative aspects of system design specifications.

While the systems analysis staff must be skilled in mathematical modeling, software design, and other specialized techniques, its members are also required to have a system perspective and a thorough knowledge of the operational requirements of the system under development.

System Design Team

The exercise of leadership and coordination in any large program requires one or more teams of key individuals working closely together, maintaining a general consensus on the conduct of the engineering program. A system design team for a complex system development project may have the following membership:

- systems engineer,
- lead subsystems engineers,
- software systems engineers,
- support engineers,
- test engineers,
- customer representative, and
- specialty and concurrent engineers.

The customer representative is an advocate for the system requirements. The advantage of the team approach is that it generally increases the esprit de corps and motivation of the participants and broadens their understanding of the status and problems of other related aspects of the system development. This develops a sense of ownership of the team members in the overall system rather than the limitation of responsibility that is the rule in many organizations. It makes the response to unexpected problems and other program changes more effective.

In a particular application, the leadership of the system development needs to be tailored to the prime contractor's organization and to the customer's level of involvement in the process. The most important common denominators are

1. quality of leadership of the team leader,
2. representation of those with key responsibilities, and
3. participation of key technical contributors.

Without energetic leadership, the members of the system design team will flounder or go their separate ways. If for some reason the person designated as the project systems engineer does not have the required personal leadership qualities, either the project engineer or a deputy systems engineer should assume the team leadership role.

The presence of the leaders of the major portion of the development effort is necessary to bring them into the design decision process, as well as to have them available to use their resources to resolve problems. There are usually several senior systems engineers whose experience and knowledge are of great value to the project. Their presence adds a necessary ingredient of wisdom to the design process.

Involvement of the customer in the design process is essential but, in many cases, may be an inhibiting influence on free discussion in team meetings. Frequent but more formal meetings with the customer may be preferred to team membership.

5.6 SUMMARY

Managing System Development and Risks

Systems engineering is a part of project management that provides technical guidelines, system integration, and technical coordination.

WBS

The systems engineer's role also involves contributing to resource allocation, task definition, and customer interaction, with the initial focus on the development of the WBS, a hierarchical task organization that subdivides total effort into successively smaller work elements. This provides the basis for scheduling, costing, and monitoring, and enables cost control and estimating.

One key tool used for program scheduling is the CPM. CPM is based on WBS work elements and creates a network of sequential activities. Analyzing this network enables the systems engineer and program manager to identify paths that take the longest to complete.

SEMP

The SEMP plans the implementation of all systems engineering tasks. In the process, it defines the roles and responsibilities of all participants.

Risk Management

Risk management is a major challenge to systems engineering since all new system developments present uncertainties and risks. Reducing program risks is a continual

process throughout the life cycle; moreover, risk must be reduced as program investment rises.

A risk management plan is important to support risk management. Risk assessment identifies the importance of risk in terms of risk likelihood (probability of occurrence) and risk criticality (impact and consequences of risk realization).

Risk mitigation of a critical area may include one or more of the following: management reviews, special engineering oversight, special analysis and tests, rapid prototyping, retry of rigorous requirements, and/or fallback developments.

Organization of Systems Engineering

The systems engineering organization spans disciplines and participating organizations, but also adapts to the company organizational structure. Therefore, systems engineering must communicate effectively "what, when, and why" to the proper stakeholders and must provide technical reviews for all participants. In large programs, systems engineering is supported by a systems analysis staff.

Large programs will require formal system design teams, which integrate major subsystems and subcontractors, and the products of software systems engineering. These teams contain members from support engineering and the test organization, and typically contain specialty (concurrent engineering) members as appropriate. They may also include user representation when appropriate. A key role for systems engineering involvement in these design teams is to keep their focus on the success of the entire enterprise.

PROBLEMS

5.1 Developing a detailed WBS for a system development project is a basic function of project management. What part should be played by systems engineering in the definition of the WBS in addition to detailing the section named "Systems Engineering"?

5.2 The preparation of a formal SEMP is usually a required portion of a contractor's proposal for a competitive system development program. Since at this time the system design is still in a conceptual state, explain where you would get the information to address the elements of a typical SEMP as listed in this chapter.

5.3 Define the two main components of risk management discussed in this chapter and give two examples of each. Show by an example how you would apply risk management processes to a system development project that proposes to use one or more components that utilize unproven technology.

5.4 One of the methods for estimating the risk likelihood (probability of failure) of a system development is to compare the current design status with comparable situations in existing systems. Table 5.2 shows some basic characteristics that are useful in making these estimates. For each of the first three

conditions associated with high-risk projects, briefly describe an example of such a condition in a real system, or describe a hypothetical example having such characteristics.

5.5 Suppose you are a new systems engineer for a major new system development effort that involves new technology. Obviously, this represents a major technical (if not programmatic) risk area. What activities would you recommend early in the system development effort to mitigate these technical risks? For each mitigation activity, describe whether the activity will lower the likelihood of the risk, or the consequences of the risk, or both.

5.6 There are a number of risk mitigation methods for dealing with program risks. Referring to the description of high and low program impact in Table 5.3, discuss how you would best use risk mitigation methods to reduce their risk criticality.

5.7 Suppose you are a systems engineer on a new system development project in which your design engineers have never developed the subsystems and components required for this new system. Obviously, this represents a major risk area. What activities would you recommend early in the system development effort to mitigate these technical risks? For each mitigation activity, describe whether the activity will lower the likelihood of the risk or the consequences of the risk, or both.

5.8 This chapter presents a method for quantifying risk in two elements, likelihood and criticality, and for plotting these two metrics on a risk matrix. Suppose you wanted to combine these two metrics into a single, combined metric for risk. Suggest three methods for combining likelihood and criticality into a single metric. List the advantages and disadvantages for each.

5.9 Research the building of the tunnel under the English Channel in the late twentieth century.

(a) What risks were present with this project?

(b) What successful activities were undertaken to mitigate these risks that led to the tunnel's completion?

5.10 Describe the general type of the organizational structure in which you work. Discuss instances where this structure has been beneficial and those where it has not been so beneficial to programs you have been involved in or have some knowledge of.

5.11 Discuss the advantages of using the system design team approach for a large development project. List and discuss six requirements that are needed to make this approach successful.

FURTHER READING

B. Blanchard. *System Engineering Management*, Third Edition. John Wiley & Sons, 2004.

B. Blanchard and W. Fabrycky. *System Engineering and Analysis*, Fourth Edition. Prentice Hall, 2006, Chapters 18 and 19.