Chapter **3**

System Life Cycle

PATRICK J. DRISCOLL, Ph.D.
PAUL KUCIK, Ph.D.

> There seems to be a kind of order in the universe, in the movement of the stars and the turning of the earth and the changing of the seasons, and even in the cycle of human life. But human life itself is almost pure chaos.
>
> —Katherine Anne Porter (1890–1980)

> A mental note is worth all the paper it is written on.
>
> —Marie Samples, OCME, New York

3.1 INTRODUCTION

All systems have a useful lifetime during which they serve the purpose for which they were created. Just like a human lifetime, the degree to which a system achieves this purpose typically varies with age. New systems start out by hopefully meeting their performance targets. After entry in service, system elements and processes may begin to degrade. Degradation that occurs during a system's useful years motivates a host of specialized maintenance activities, some planned and some unplanned, intended to restore the system to as close to its original state as possible. Creating written lists, calendars, and other memory enhancement techniques are examples of maintenance items we use to restore memory functionality as close as possible to earlier periods of peak performance.

Decision Making in Systems Engineering and Management, Second Edition
Edited by Gregory S. Parnell, Patrick J. Driscoll, Dale L. Henderson

Eventually, most systems degrade to a point where they are no longer effectively meeting consumer needs and are retired. At the retirement decision, the cost of further maintaining a system could be exceeding the cost of replacing the system. Or, perhaps the system is operating as intended but it can no longer provide value to its stakeholders due to changes in the environment within which it exists. The 8Track technology for audio recording and playback is an example of a system that was retired, because it lost its ability to compete in a consumer environment where "smaller is better" drove demand. A similar competition is ongoing between high-definition movie format and Blu-ray™ technology.

In a similar fashion to human physiology, it is useful to think of a system as progressing through a succession of stages known as a *life cycle*. For living systems, this cycle consists of four stages simply described: birth, growth, deterioration, and death [1]. From a systems engineering perspective, there are at least three major reasons why a life cycle structure is an effective metaphor. We can:

1. Organize system development activities in a logical fashion that recognizes some activities must be accomplished prior to others.
2. Identify the specific activities needed to be accomplished in each stage to successfully move to the next stage.
3. Effectively consider the impact that early decisions have on later stages of the system life cycle, especially with regard to cost and risk (the likelihood and consequences of system problems).

As illustrated in Figure 3.1, the system life cycle activities referred to in the first reason have a logical sequencing. They align with the transition of a system from its conceptual birth to eventual retirement. Notice that these activities are not the same as those described in the systems decision process (SDP). The SDP is a cycle of phases to support major systems decisions repeated at critical decision points (gates) typically encountered once in each stage of the life cycle. The four phases of the SDP are described in Part III of this book.

The second reason reinforces that as members of multidisciplinary teams (see Chapter 7), systems engineers maintain an appropriate focus on what needs to be done and when it needs to be done. The specifics of what, when, how, and why associated with these needs are dictated by the life cycle stage of the system. For example, the list of alternative solutions generated in a systems decision process concerning a recently deployed system would consist of process and/or product modifications and enhancements designed to aid the system to better achieve the purpose for which it was created. Later in the stage, the list of alternative solutions might focus on new systems to replace the current system.

Of the three reasons stated as to why a system life cycle metaphor is effective, the third is the most important. Some decisions made during early stages of system development are irreversible; once there is commitment to a particular system concept (e.g., airplane) and a detailed design, consumer needs and resource limitations (including time) usually prevent the design team from switching to alternative

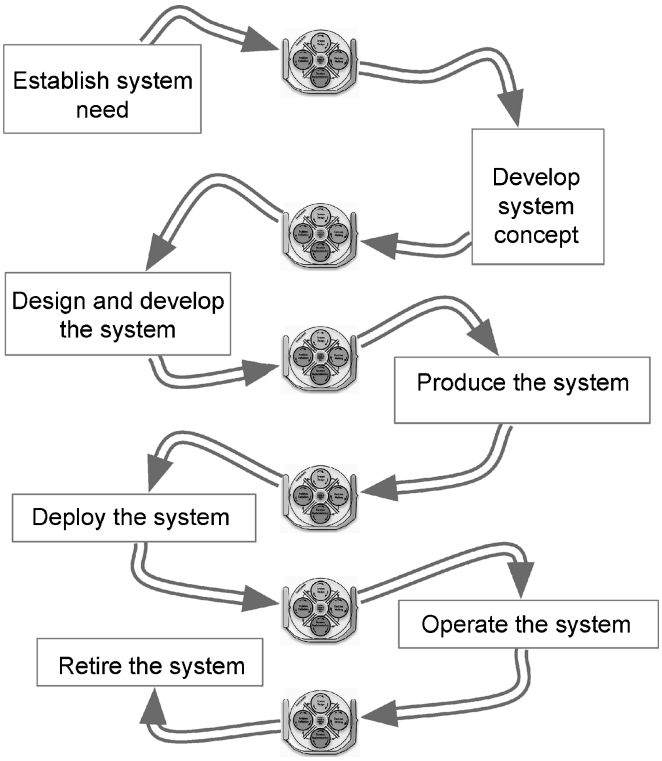

Figure 3.1 Systems decision processes occurring in each system life cycle stage.

design solutions. Other minor design decisions (e.g., paint color) can be altered readily without significant impact on project planning and execution.

However, all decisions have immediate and delayed costs associated with them. These costs can consist of a blend of financial, risk, environmental, technological, legal, and moral factors of direct concern to stakeholders, many of which may not be realized until later stages in the life cycle.

Not considering life cycle costs when making decisions early in the system life cycle could prove to be disastrous to the long term system viability and survivability. The principle underlying this idea goes back to an idea originating during the Scottish enlightenment known as the Law of Unintended Consequences, more recently stated [1] succinctly as "[w]hether or not what you do has the effect you want, it will have three at least that you never expected, and one of those usually unpleasant." The least expensive and most effective hot water pipe insulating material to use in the northeast United States might be asbestos, but deciding to use asbestos without seriously considering the cost factors associated with the operational and retirement stages would not be wise.

This principle reminds us to be careful not to commit to a "whatever is best for right now" solution without examining the degree to which such a solution remains

optimal for later life cycle stages as well. Hidden system costs often occur because someone failed to employ systems thinking over a complete system life cycle.

In practice, life cycles are driven by the system under development. They serve to describe a system as it matures over time with regard to its functioning. A life cycle also guides professionals involved with sustaining the value delivered by the system to consumers during this maturation. The life cycle stages need to contain sufficient detail to enable systems engineers, systems managers, operations researchers, production engineers, and so on, to identify where their skill set tasks fit into the overall effort. At the same time, the stages need to be described broadly enough to accommodate related activities in a natural way.

System life cycles are structured and documented for success. Among the myriad of life cycle models in existence, two fundamental classifications arise: predictive and adaptive [2]. Predictive life cycle models favor optimization over adaptability. Adaptive life cycle models accept and embrace change during the development process and resist detailed planning. Once a life cycle for a particular systems decision problem is defined and documented, it is then possible to structure the management activities that will be used to support each stage of the life cycle. This provides the data that are necessary to support major decision gates to move to the next stage. An effective management system prevents system development from occurring in a piecemeal or disjoint basis that has a tendency to increase risk.

The life cycle model we use in this text has the advantage of being able to simply represent stages in a system's lifetime along with the activities within each stage. The structured process used to define and support systems engineering activities within these stages, the systems decision process (SDP), is naturally cyclic, thereby providing a constant feedback mechanism that encourages revision consistent with changes in the system environment, all the while taking full advantage of opportunities to capture and deliver value to the stakeholders. The SDP is typically used at least once during each life cycle stage to determine if the system should advance to the next stage.

As a consequence of separating the system life cycle from the SDP, the SDP provides the essential information for systems decision makers independent of the life cycle stage the system is in. Admittedly, each application of the SDP is tailored to the system and the life cycle stage. Some elements of the SDP may be truncated, while others may be amplified for some systems in some stages. This adaptability feature is perhaps one of the SDPs greatest attributes. How to tailor the SDP is described in Chapter 9.

3.2 SYSTEM LIFE CYCLE MODEL

The life cycle stages listed in Table 3.1 are broadly defined so as to apply to as many systems as possible. As can be seen in the sections that follow, various other life cycle models exist that have specific types of system development models or systems engineering applications in mind. For example, the spiral design model illustrated in Figure 3.3 is frequently used in software system development with an eye toward highlighting risk management throughout the various life cycle stages.

TABLE 3.1 System Life Cycle Model for Systems Engineering and Management

System Life Cycle Stage	Typical Stage Activities
Establish system need	Define the problem
	Identify stakeholder needs
	Identify preliminary requirements
	Identify risk factors and initial risk management plan
Develop system concept	Refine system requirements
	Explore system concepts
	Propose feasible system concepts
	Refine risk factors
	Assess initial performance, schedule, and technology risks
Design and develop system	Develop preliminary design
	Develop final design
	Assess initial development cost, market, and business risks
	Perform development tests to reduce risk
	Refine performance, schedule, and technology risk assessments; include mitigation steps in risk management plan
	Build development system(s) for test and evaluation
	Verify and validate design
	Test for integration, robustness, effectiveness
	Includes production scheduling, economic analysis, reliability assessments, maintainability, and spiral design implementation considerations, among others
Produce system	Produce system according to design specifications and production schedule
	Apply Lean Six Sigma as appropriate
	Refine development cost, market, and business risks; include mitigation steps in risk management plan
	Monitor, measure and mitigate performance, schedule, and technology risk
	Assess initial operational risk to the system
Deploy system	Refine operational risk management plan
	Develop a deployment plan
	Complete training of users and consumers
Operate system	Operate system to satisfy consumer and user needs
	Monitor, measure and mitigate operational risks
	Identify opportunities for enhanced system performance
	Provide sustained system capability through maintenance, updates, or planned spiral developed enhancements
Retire system	Develop retirement plan
	Store, archive, or dispose of the system

3.2.1 Establish System Need

Establishing a clear system need is a critical first step in system management. Successfully addressing the purposes associated with this stage of the life cycle increases the likelihood of a match between the system that is truly needed and the one that is developed. The client facing the problem that generates a system need is typically dealing with the problem's symptoms and resulting effects on a day-to-day basis. It is not unusual for a customer in this situation to communicate an initial system need that is focused on treating these symptoms.

In the short term, treating the symptoms might improve the working conditions associated with the problem but it does not help to resolve the underlying cause(s) of the symptoms, which is much more difficult to uncover. The true cause(s) of observed symptoms typically emerges from a process of intensive interaction with stakeholders using techniques introduced in Chapter 10.

The first stage of the life cycle is about exploration, discovery, and refining key ingredients necessary for a project to get off to a good start. This consumes a large amount of time and effort. Once the actual problem, stakeholder needs, and preliminary requirements are successfully defined, the beginning steps are taken toward effective risk management and the system transitions into the next life cycle stage.

3.2.2 Develop System Concept

The life cycle stage of developing a system concept is centered on applying techniques designed to inspire creative thought contributing to effective, efficient systems for delivering maximum value to the stakeholders. These techniques both generate novel system possibilities that meet stakeholder needs and eliminate those system concepts that are infeasible. Conceptual system models involving graphical illustrations, tables, and charts comparing and contrasting system characteristics, along with a multitude of linked hierarchical diagrams, are used to identify possible system concepts that can meet stakeholder needs.

In light of this initial set of feasible system concepts, various dimensions of system risk are identified and refined, such as performance (will the technology work?), schedule (can it be provided when needed?), and cost (is the system affordable?). The life cycle transitions to the next stage only when a sufficient number of feasible system concepts are identified that possess acceptable levels of risk.

3.2.3 Design and Develop System

The design and development stage involves designing, developing, testing, and documenting the performance of the chosen system concept. Quite often the models produced during this stage take the form of simulations, prototype code modules, mathematical programs, and reduced and full-scale physical prototypes, among others. One must be careful to adhere to professional best practices when testing and analyzing the performance of competitive designs using these models and

simulations (see Chapter 4), especially when it comes to verifying and validating model results.

A feasible concept along with a system model enables the system team to develop estimates of program costs along with market and business risks. Of course, the accuracy of these risk estimates depends strongly upon the assumptions made concerning the system deployment environment that will occur in the future. It is wise under these conditions to carefully develop a set of use case scenarios for testing the performance of the system design using models and simulations developed for this purpose. These use cases should reasonably reflect the full span of "What if?" possibilities. This is the only way of identifying system design problems and limitations short of building, deploying and operating a full-scale system. Engineering managers are fully engaged in the system at this point in the life cycle as well, developing plans to address all the dimensions of implementation noted. When satisfactorily completed, the decision gate naturally supports the system going forward into a production stage.

3.2.4 Produce System

Realizing success in the system production life cycle stage is as far from a foregone conclusion as one might imagine. This is a period in the system life cycle that stresses the management team charged with producing a system that meets all of its intended purposes, meets or exceeds design requirements reflecting an acceptable risk across all risk dimensions, and does all this in an effective and efficient manner to achieve competitive advantage for the consumer of system products and services.

Ultimately, the system must deliver value to the stakeholders or it will fail. However, systems can fail through no fault of their own simply because of changing environmental conditions. Remember that throughout the life cycle and all the efforts that have gone into making a concept a reality, time continues to evolve, raising the very real possibility that substantial threats to system success that were not present earlier in the life cycle could exist upon deployment. Thus, during this stage the systems team tries to identify and assess the types and levels of operational risks the deployed system will face.

Depending on the type of system to be delivered—product-focused or service-focused—Lean Six Sigma [3] and other process improvement methods are applied so that the production processes used to make the designed system a reality are as effective and efficient as possible.

A portion of the ongoing risk analysis engages in monitoring, measuring, and mitigating risks identified in the previous life cycle stages. Moreover, a good deal of effort goes into maintaining vigilance for any new risks that might emerge as a result of changes in the environment outside of the control of the team. This sensitivity naturally leads to considering those external factors that could present risk to the system once it is placed into operation.

Operational risk [4] is emerging to be one of the least quantified, less understood, yet potentially largest impact areas of risk for systems engineering. While no

single definition is predominant currently, operational risk is generally understood to mean the loss resulting from inadequate or failed internal processes, people, and support systems or from environmental events. During this stage of the life cycle, brainstorming and other ideation techniques are again used to identify an initial list of operational risks that might threaten program success once the system is fielded and in the hands of users and consumers.

3.2.5 Deploy System

Successfully deploying a new or reengineered system is the direct result of executing a well-thought-out deployment plan that provides detailed planning for the activities necessary to place the system into an operational environment. The plan includes, as a minimum, a description of the assumptions supporting the current capabilities and intended use of the system, the dependencies that can affect the deployment of the system, and any factors that limit the ability to deploy the system.

In close coordination with the system users, consumers, and owners, many other deployment details are specified in the deployment plan as well. These include information concerning deployment locations, site preparation, database conversions or creation, and phased rollout sequencing (if appropriate). Training programs and system documentation are created during this life cycle stage. Specific training plans for system users and maintainers play a critical role in achieving a successful system deployment.

The systems team itself transitions into a support role during this life cycle stage as they begin to disengage from primary contact with the system and transfer system functionality to the client. Additional resource requirements for the design team are identified, support procedures designed, and a host of transition activities along with management roles and responsibilities become part of the deployment plan. Operations and maintenance plans are created and specified as well in order to provide explicit guidance to system consumers and users as to how they might capture the greatest value return consistent with the intended design goals.

Finally, contingency plans are developed during this stage as well, some of which are included in the deployment plan while others are maintained internally by the systems engineering and deployment teams in case some of the risks identified and planned for in previous stages become reality. The operational risks identified previously are refined and updated as forecasted information concerning the system environment used to develop an initial list of potential threats to program success becomes reality.

3.2.6 Operate System

The most visible life cycle stage is that of operating the system in the mode it was intended. System users operate systems to provide products and services to system consumers. When we recognize the existence of systems in our environment, we are

observing them in this stage of their life cycle. Some everyday systems that fall into this characterization include transportation networks, communications networks, electricity grid supply networks, emergency services, tourism, law enforcement, national security, politics, organized crime, and so on.

Condition monitoring of a system while it is in operation is an activity that has traditionally consisted of use-based measures such as the number of hours of operation, number of spot welds performed, number of patients treated, and so on. Measures such as these have dominated reliability analysis for systems whose wear and tear during periods of nonuse is so small as to render its impact insignificant.

A good example of this can be seen in aircraft systems scheduled maintenance planning, which is based on flight hours of the aircraft in operation and not on the amount of time passed since the aircraft was placed into operation. Many systems have switched to condition-based maintenance for example, in fleet aircraft transportation operations, recognizing that not all pilots fly aircraft in the same manner [5]. For military aircraft, condition-based maintenance assumes that an hour flying routine missions imposes a significantly different level of system stress than does an hour flying in combat missions—at night, in bad weather, amidst hostile fire. Thus, system maintenance planning, which used to consist of executing routine tasks on a preset schedule, is evolving to the point where real-time monitoring of system condition indicators is becoming more commonplace.

The operational risks due to external influences on system elements, services, and performance to meet goals identified in previous life cycle stages are closely monitored during system operation. However, management focus during this life cycle stage is not exclusively centered on potential bad things that might occur. They also maintain a heightened awareness for possible opportunities to enhance system performance that could add value to the stakeholders or increase the competitive advantage of consumers and users.

In fact, when systems engineers are called upon to engage a system during one of its operational life cycle stages, the underlying motivation of the user organization is centered on this very principle of exacting increased performance value out of the existing system. This could mean reengineering the system or its processes, applying optimization techniques to increase the efficiency of some dimensions of the system operation, using reliability methods to better understand and reduce the overall maintenance costs, or perhaps generating new ideas for system replacement that leverage recent developments in technology, knowledge, or the competitive landscape. Some of these advancements or changes in the operating environment may have been predicted and planned for during earlier life cycle stages, in which case system enhancements would be applied using principles of spiral development as well [6] .

3.2.7 Retire System

Finally, when users determine that it is no longer in their best interest to continue operating the system, it is retired from service. While the activities during this stage might be as simple as donating the existing system to a nonprofit organization or

placing the system in storage, this life cycle stage can actually be quite complicated. One needs only consider the intricate requirements associated with taking a nuclear power plant offline in order to gain an appreciation for how complex this system retirement stage could be [7].

3.3 OTHER MAJOR SYSTEM LIFE CYCLE MODELS

As mentioned previously, there are several life cycle models in common use today. All of these life cycle models, including the one we use, are based on sets of professionally agreed standards, the primary ones being listed in Table 3.2 [8]. Despite The International Council on Systems Engineering (INCOSE) has taken a lead on setting a common standard for professional practice and education against which programs and businesses can compare their quality levels.

The last standard listed, ISO/IEC 15288, represents a modern interpretation of a system life cycle relative to the complex and heavily stakeholder dependent nature of the typical systems addressed in professional practice today. It represents the evolving standard of systems engineering practice in the United Kingdom. In this vein, the ISO/IEC 15288 system life cycle model is serving a purpose closely aligned with the life cycle model we use, except that, as can be seen in Table 3.3, they combine a life cycle model with various steps in a systems decision process. In this text, we separate the two processes into the system life cycle model discussed earlier and the SDP that supports decision making during all the stages of the system life cycle model.

Given the sequential representation shown in Table 3.3 [9], the ISO/IEC 15288 system life cycle implies that one stage is completed before transitioning into the next. As a result, this life cycle model has been criticized for its lack of robustness

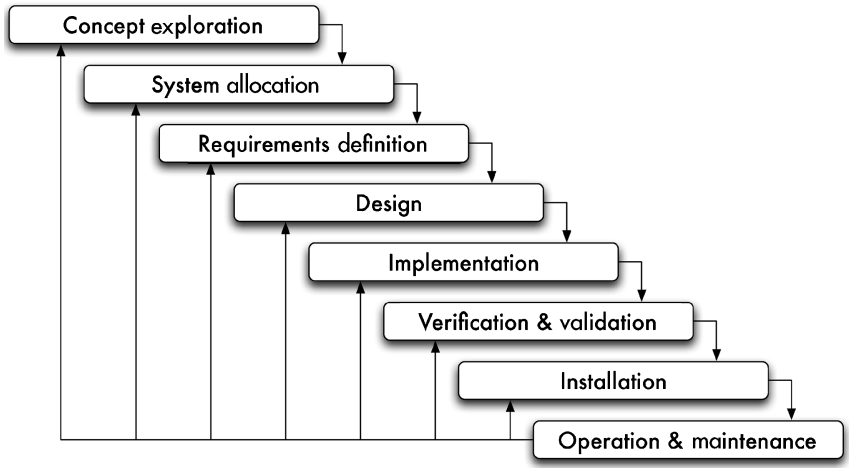

Figure 3.2 Waterfall system life cycle model.

TABLE 3.2 A Comparison of Standards-Driven Life Cycle Models

Standard	Description	System Life Cycle Stages
MIL/-STD/-499B	Focuses on the development of defense systems.	• Preconcept • Concept exploration and definition • Demonstration and validation • Engineering and manufacturing development • Production and deployment • Operations and support
EIA.IS 632	A demilitarized version of MILSTD499B	• Market requirements • Concept definition and feasibility • Concept validation • Engineering and manufacturing development • Production and deployment • Operations and support
IEEE 1220	Introduces the interdisciplinary nature of the tasks involved in transforming client needs, requirements, and constraints into a system solution.	• System definition • Subsystem definition • Preliminary design • Detailed design • Fabrication, assembly, integration, and test • Production • Customer support
EIA 632	Focus is on defining processes that can be applied in any enterprise-based life cycle phase to engineer or reengineer a system.	• Assessment of opportunities • Solicitation and contract award • System concept development • Subsystem design and predeployment • Deployment, installation, operations, and support
ISO/IEC 15288	Includes both systems engineering and management processes at a high level of abstraction.	• Concept process • Development process • Production process • Utilization process • Support process • Retirement or disposal process

TABLE 3.3 The ISO/IEC 15288 Systems Engineering Life Cycle Model

Life Cycle Stage	Purpose	Decision Gates
Concept	• Identify stakeholder needs • Explore concepts • Propose feasible solutions	Execute next stage
Development	• Refine system requirements • Create solution description • Build system • Verify and validate	Continue current stage
Production	• Mass produce system • Inspect and test	Go to previous stage
Utilization	• Operate system to satisfy user needs	Hold project activity
Support	• Provide sustained system capability	Terminate project
Retirement	• Store, archive, or dispose of system	

in dealing with a wide variety of system problems. The waterfall model (Figure 3.2) is more robust because system problems arising in any stage can lead the systems team to recycle back through earlier stages in order to resolve them.

In contrast to the waterfall, the spiral life cycle model shown in Figure 3.3 formalizes the notion of repeated cycling through a development process. Each spiral produces increasingly more complex prototypes leading to a full-scale system deployment. In essence, the spiral model executes a series of waterfall models for each prototype development.

One attractive feature of the spiral model is the explicit recognition of the important role that risk plays in system development. This same consideration is intentionally incorporated in the system life cycle model we use. In both life cycle models, various types of risks are identified during each prototype development cycle—for example, investment risk, performance risk, schedule risk, and so on. If these risks are successfully mitigated, the systems team evaluates the results of the current cycle, presents the results and conclusions in support of the decision gate, and, if approved, proceeds to enhance the prototype in the spiral model or moves to another stage in our life cycle model. Failing to resolve important risks can cause the program to terminate during any stage.

Several other specialized models are used for system development, although none as prevalent as the two already mentioned. Rapid applications development,

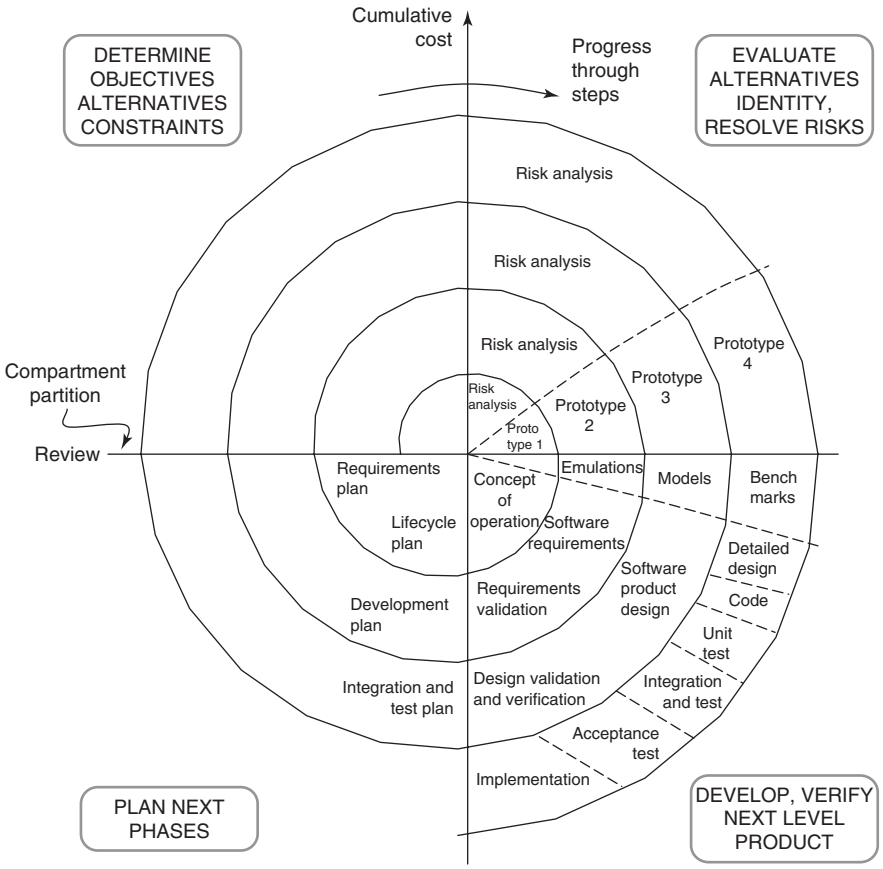

Figure 3.3 Spiral life cycle model with embedded risk assessment [10].

a methodology created to respond to the need to develop software systems very fast, strives to deploy an 80% system solution in 20% of the time that would be required to produce a total solution [11], and agile life cycle models [12] are among these specialized models.

3.4 RISK MANAGEMENT IN THE SYSTEM LIFE CYCLE

As the complexity of systems and their environment increases, the number, type, likelihood, and impact of events that can and might occur to threaten the well-being of systems becomes increasingly more difficult to identify.

A *risk* is a probabilistic event that, if it occurs, will cause unwanted change in the cost, schedule, or value return (e.g., technical performance) of an engineering system [13]. The goal of risk management is to identify and assess risks in order to enact policy and take action to reduce the risk-induced variance of system technical

performance, cost, and schedule estimates over the entire system life cycle. In other words, risk management describes a collection of conscience and deliberate actions to protect the system from the adverse effects of specific events that have a nonzero probability of occurring in the future.

System complexity works against accomplishing this goal in an easy manner because by increasing the number and type of interconnections, vested interests, and uncertainty, it becomes more and more difficult to effectively apply risk management. This is especially true if the risk management activities lack formal, repeatable organization. The situation today is that it is simply impossible to "fly by intuition" in this regard. Moreover, while risks associated with specific components, so-called *nonsystemic* risks, might be identifiable by ad hoc procedures based on experience alone, the more subtle and elusive *systemic* risks, those inherent in the entire system (shared across components), will routinely avoid detection without some organized, repeatable process. This latter group can, if left unattended to, take down an entire financial, communications, transportation, or other system when they occur [14].

Risk management, which is comprised of three main activities—risk identification, risk assessment, and risk mitigation—is an ongoing process applied throughout the life cycle of a systems engineering project. In this section, we take a broad view of risk management [15], focusing on core principles and concepts that set the stage for a more in-depth exploration in later chapters.

3.4.1 Risk Identification

The process of identifying risks consists of determining any sources of risk and the scenarios under which they may occur. Risk identification seeks to discover and categorize uncertain events or conditions whose occurrence will have a negative impact on system cost, schedule, value, technical performance, or safety. The focus of this effort often changes during a systems decision problem. A team could be initially concerned about the risk associated with having the project proposal approved, shifting then to possible risk impediments to the SDP effort and, finally, shifting to addressing threats to the successful implementation and sustained health of the selected system solution. Ideally then, techniques used to identify risks need to be flexible or general enough to apply throughout the life of a systems decision problem and its resulting solution. In what follows, we refer to the systems decision problem effort as "the project."

Two convenient techniques for identifying possible risks to systems are prompt lists and brainstorming. Both techniques involve extensive interaction with the system stakeholders. Their unique insights arising from their extensive knowledge of the operating environment of the needed system are critical information elements needed to develop comprehensive risk categories.

A *prompt list* is simply a listing of possible categories of risks that are particular to the current systems decision problem. They function as word recall prompts during stakeholder interviews, helping participants to think of as many risks to its

success as possible. As a technique, a prompt list can be used on an individual basis or in a group setting with a facilitator from the systems team in control.

For example, when identifying risk elements during a life cycle stage that focuses on establishing the need for a system, the team could use a prompt list consisting of the SDP environmental factors—technological, health & safety, social, moral/ethical, security, cultural, ecological, historical, organizational, political, economic, and legal—in order to develop a preliminary list of major risks associated with developing a system to meet the needs of stakeholders. The risk elements emerge in subsequent discussions as the details required to document risks in a risk register are identified. Executive board objections to the overall decision support, potential financial problems with funding the effort to completion, knowledge deficiencies due to venturing into new competitive territory, political backlash from government administrators or the general public, and so on, are examples of the types of risk that arise.

Brainstorming (see Chapter 11) is a technique that works much in the same manner as a prompt list except that a neutral human facilitator from the systems team serves a similar purpose as the prompt list: to elicit without judgment from the stakeholders any possible risks to successful project completion they might identify from their experience. Brainstorming is also performed almost exclusively in a group setting. The facilitator might employ project schedules, graphical illustrations, data tables, or even a prompt list to help the participants identify risks.

A successful brainstorming session depends heavily on the participation of key stakeholders (see Chapter 10). As with many senior leaders of organizations who have constant demands on their time, these key stakeholders may not be able to assemble as a single group for any significant length of time. When stakeholder access is limited, prompt lists are a better technique to use for risk identification because they allow decentralized participation in the risk identification process while maintaining a common frame-of-reference provided by the logical structure of the list. In either instance, a good practice is to plan on at least two complete iterations of stakeholder interviews so that the results of the first set of interviews might be leveraged as prompts for all stakeholders during the second session.

There are six common questions [16] that are commonly used to capture various dimensions of risk to the system during brainstorming sessions with key stakeholders. The answers to these questions provide the data needed to begin to analyze risks and plan for their mitigation during a systems decision problem. The six questions are:

1. What can go wrong?
2. What is the likelihood of something going wrong?
3. What are the consequences?
4. What can be done and what options are available?
5. What are the tradeoffs in terms of risk, costs, and benefits?
6. What are the impacts of current decisions on future options?

TABLE 3.4 Techniques for the Identification of Potential Sources of Risk in the NSTS Program

Hazard analysis	Design and engineering studies
Development and acceptance testing	Safety studies and analysis
FMEAs, CILs, and EIFA	Certification test and analysis
Sneak circuit analyses	Milestone reviews
Failure investigations	Waivers and deviations
Walk-down inspections	Mission planning activities
Software reviews	Astronaut debriefings and concerns
OMRSD/OMI	Flight anomalies
Flight rules development	Aerospace safety advisory panel
Lessons-learned	Alerts
Critical functions assessment	Individual concerns
Hot line	Panel meetings
Software hazard analysis	Faulty tree analysis
Inspections	Change evaluation
Review of manufacturing process	Human factors analysis
Simulations	Payload hazard reports
Real-time operations	Payload interfaces

To make an important point clear: Identifying project risks is a demanding task that consumes a good deal of time, effort, and brainpower to do it right. Using a structured, repeatable method that is easy to understand is a key ingredient to success. As an example of how important this process is to successful systems decision problems and how systems engineers attempt to address this concern as comprehensively as possible, consider the listing of techniques used by NASA scientists and risk specialists to identify risks to the National Space Transportation System (NSTS) [17] shown in Table 3.4. These tasks represent thousands of work hours by a host of people across a broad range of system stakeholders.

As each risk is identified, it is categorized, to ensure that risks are not double-counted and that the identification of risks is comprehensive. The intent is to group risks into mutually exclusive and collectively exhaustive categories. INCOSE recognizes four categories of risk that must be considered during a system decision problem: technical risk, cost risk, schedule risk, and programmatic risk [18, 19].

Technical risk is concerned with the possibility that a requirement of the system will not be achieved, such as a functional requirement or a specific technical performance objective, because of a problem associated with the technology incorporated into the system, used by the system, or interfacing with system input and output. One component of the SDP described in Chapter 10 is a functional analysis, which is used to develop a functional hierarchy that illustrates what any feasible system solution must do to be considered successful. For a host of modern systems, technology is the main driver of these functions. By considering the risk to accomplishing each function, a comprehensive treatment of technical risk ensues.

Cost risk is the possibility of exceeding the planned design, development, production, or operating budgets in whole or in part. For any system, estimates of

future life cycle costs are subject to varying degrees of uncertainty due to uncontrollable environmental factors, time, and the source of information used to develop these estimates. The further forward in time these costs are anticipated to occur, the more uncertainty is associated with their estimates. While objective cost data with similar systems decision problems is desirable, subjective expert opinion is often used to create cost estimates for items less familiar to the project team and stakeholders. This injects additional uncertainty that must be taken into account, as we show in Chapter 5. Cost risk planning is more complicated than simply accounting for program spending and balancing any remaining budget. Cost risk extends over the entire system life cycle. Decisions made throughout the system life cycle are assessed for their downstream impact on the total system life cycle costs. It becomes necessary to identify major cost drivers whose variability can cause the projet to "break the budget" rapidly, thus causing a termination of the effort in the worst case. Properly eliciting the information needed to model and analyze cost uncertainty requires careful thought and consideration [20].

Schedule risk is the possibility that a project will fail to achieve key milestones agreed upon with the client. Scheduling individual tasks, duration, and their interrelationships is critical to sound project planning. Doing so directly identifies those system activities that lie on a critical path to project success (see Chapter 13). Systems engineers and program managers should focus a large amount of their effort on these critical path tasks because when these tasks fail to achieve on-time start and completion times, the overall project schedule and delivery dates are directly effected. While the more common method of identifying critical path activities is deterministic, recent developments have demonstrated significantly improved benefits for analyzing cost, schedule, and risk *simultaneously* via Monte Carlo simulation [21].

Programmatic risk arises from the recognition that any systems decision problem takes place within a larger environmental context. Thus, it is an assessment of how and to what degree external effects and decisions imposed on the projet threaten successful system development and deployment. This last form of risk is closely related to the concept of operational risk emerging from the banking industry [22]. Increased levels of critical suppliers, outsourcing specific engineering tasks, budget reductions, personnel reassignments, and so on, are all examples of programmatic risk.

The INCOSE risk categories provide a useful framework for facilitating risk identification and ensuring a comprehensive treatment of risks. It should be noted that the above risk categories interact with each other throughout a system life cycle. While standard in a systems engineering environment, these are not the only grouping categories that are used. Commercial banks, for example, divide their risk categories into financial, operational, and, more recently, systematic risks in order to track the most common undesirable future events they face.

The systemic risk category is worth emphasizing because of its recent realization in global securities markets. The Counterpolicy Risk Management Group [23] suggests an effective definition for our use. A *systemic risk* is the potential loss or damage to an entire system as contrasted with the loss to a single unit of that

system. Systemic risks are exacerbated by interdependencies among the units often because of weak links in the system. These risks can be triggered by sudden events or built up over time with the impact often being large and possibly catastrophic.

Systemic risk is an interesting phenomenon gaining growing attention across all risk concerns with systems. Recently, the impact of unmitigated systemic risk events occurring within the financial markets was felt across the globe. The U.S. Congressional Research Service (CRS) describes systemic risk in the following manner:

> All financial market participants face risk—without it, financial intermediation would not occur. Some risks, such as the failure of a specific firm or change in a specific interest rate, can be protected against through diversification, insurance, or financial instruments such as derivatives. One definition of systemic risk is risk that can potentially cause instability for large parts of the financial system. Often, systemic risk will be caused by risks that individual firms cannot protect themselves against; some economists distinguish these types of risks as a subset of systemic risks called systematic risks. Systemic risk can come from within or outside of the financial system. An example of systemic risk that came from outside of the financial system were fears (that largely proved unfounded in hindsight) that the September 11, 2001 terrorist attacks on the nation's financial center would lead to widespread disruption to financial flows because of the destruction of physical infrastructure and death of highly specialized industry professionals. Systemic risk within the financial system is often characterized as contagion, meaning that problems with certain firms or parts of the system spill over to other firms and parts of the system [24].

The CRS report emphasizes several characteristics of systemic risk that all systems experience: shared risk due to system interconnectivity of people, organizations, equipment, policy, and so on. Systems engineering teams should be aware that systemic risks loom large on complicated projects. As the system solution structure grows, so does the likelihood that the activities supporting its development within the SDP will be subdivided among groups of the team with specialized knowledge and experience. While both effective and efficient, the project manager (PM) must maintain an integrated, holistic perspective of the overall project. Without this perspective and sensitivity to systemic risk, the project could be doomed to failure. A recently release report of the World Economic Forum strongly emphasized this point by bringing together a wide range of systems thinking experts to assist the financial services industry to develop just such a perspective. During the financial crisis of 2007 and 2008, no one regulatory authority or organization in the financial services industry had system-wide oversight that might have identified the rising systemic risk of over-leveraging that occurred [14].

A common and effective means of documenting and tracking risks once they are identified is through the use of a *risk register*. A risk register holds a list of key risks that need to be monitored and managed. When used properly, it is reviewed and updated regularly and should be a permanent item on any project meeting agenda. Figure 3.4 shows an example risk register for the rocket problem using several of the risk categories noted earlier. The values shown in the impact, likelihood, and risk level columns are developed using the techniques described in what follows.

Risk	Category	Impact	Likelihood	Risk Level	Current	Mitigation	Risk Owner
Government failure to set aside contingency funds	Financial	Medium	Low	Amber	None	Monthly monitoring of contingency funds by design team	Client
Breach of legislation	Legal	Medium	Medium	Amber	Compliance audit	Peer review by legal advisors	Team internal legal
Substandard composite material used in multiple component housings	Systemic	High	Low	Green	Periodic material sampling	Material engineering review during IPRs	Project lead engineer

Figure 3.4 Example risk register used during the SDP.

3.4.2 Risk Assessment

Once risks have been identified and categorized, the next challenge is to determine those risks that pose the greatest threat to the system. This *risk assessment* process involves assessing each hazard in terms of the potential, magnitude, and consequences of any loss from or to a system. When there exists historical data on these losses or the rate of occurrence for the risk event, the risk analysis is directly measured from the statistics of the loss. Otherwise, the risk event is modeled and predicted using probabilistic risk analysis (PRA) techniques [25]. This latter option has become the norm in modern risk analysis because for complex systems, especially those involving new or innovative technologies, such historical loss data rarely exists. Because some of the hazards to the system may involve rare events that have never occurred, estimates of the probability of occurrence can be difficult to assess and often must be based on a subjective estimate derived from expert opinion. When this occurs, techniques such as partitioned multiobjective risk method (PMRM) that use conditional risk functions to properly model and analyze these extreme events are employed [26].

The consequence imposed on system success when a risk event does transpire can involve increased cost, degradation of system technical performance, schedule delays, loss of life, and a number of other undesirable effects. With complex systems, the full consequence of a risk may not be immediately apparent as it might take time for the effects to propagate across the multitude of interconnections. These "downstream" effects, often referred to second- third-, and higher-order effects, are very difficult to identify and assess, and can easily be of higher consequence than the immediate ones. The risk of the Tacoma-Narrows bridge on Highway 16 in Seattle collapsing can be assessed from structural engineering information and historical data existing from its previous collapse in 1940. However, suppose that when this bridge collapses, the express delivery van carrying human transplant organs does not make it to the regional hospital in time to save the patient. The patient happens to be a U.S. senator who is the current champion of a new bill to Congress authorizing direct loans to Washington State residents suffering under the collapse of the mortgage industry. The bill fails to pass and thousands of people lose their homes, and so on. The middle English poet John Gower captured this domino effect in his poem *For Want of a Nail*, the modern nursery rhyme version of which goes:

For want of a nail, the shoe was lost;
For want of the shoe, the horse was lost;
For want of the horse, the rider was lost;
For want of a rider, the battle was lost;
For want of the battle, the kingdom was lost;
And all for the want of a horseshoe nail.

Probability–impact (P–I) tables [27], also known as probability–consequences tables, are a straightforward tool that can be used both to differentiate between and help prioritize upon the various risks identified and to provide clarifying summary information concerning specific risks. In concept, P–I tables are similar to the matrix procedure described in Military Standard (MIL-STD) 882 [28], elsewhere adapted to become a bicriteria filtering and ranking method [18].

P–I tables are attractive for use early in the system life cycle because as a qualitative technique they can be applied using only stakeholder input. Later, as risk mitigation costs become available, a third dimension representing the mitigation cost range can be imposed on the P–I table, thereby completing the trade space involved with risk management. Stakeholders are asked to select their assessed level of likelihood and impact of risks using a constructed qualitative scale such as very low, low, medium, high, and very high. If the actual probability intervals are difficult to assess at an early stage, a similar constructed scale can be used to solicit stakeholder input as to the likelihood of risks: unlikely, seldom, occasional, likely, and frequent [18]. The point is to start the risk management process early in the system life cycle and not to delay risk consideration until sufficient data are available to quantify risk assessments.

Typically, each of these qualitative labels is defined with a range specific of outcomes for the risks that helps the stakeholder to distinguish between levels. Using ranges, such as those illustrated in Figure 3.5 for five qualitative labels, helps normalize estimates among stakeholders. Ideally, what one stakeholder considers very high impact should correspond to what all stakeholders consider very high impact. When this is not possible to achieve, other methods such as swing weighting (see Chapter 10) become useful.

It is very important to understand what can go awry with subjective approaches such as that used in the P–I table approach, and nearly all of these considerations

			Impact on project		
	Scale	Prob	Schedule delay	Cost increase	Performance
Value ranges	Very high	40–50	>6 Days	>20%	Multiple major failures
	High	30–40	3–5 Days	15–20%	Limited major failures
	Medium	20–30	2–3 Days	10–15%	Single major failure
	Low	10–20	1–2 Days	5–10%	Multiple minor failures
	Very low	0–10	<1 Day	<5%	Limited minor failures

Figure 3.5 Example of constructed value range scales.

are based on the fact that stakeholders are involved [29]. Among these, three are important to highlight: Stakeholders can have very different perceptions of risk and uncertainty [30]; qualitative descriptions of likelihood are understood and used very differently by different stakeholders; and numerical scoring schemes can introduce their own source of errors. Straightforward techniques such as calibration tests [29] can help move stakeholders to a common scale while helping the systems team translate stakeholder input for use in risk modeling and assessment. The swing weighting technique introduced in later chapters can easily be modified and used for eliciting reasonably accurate stakeholder numerical scores. Its basis in decision analysis mitigates many of the scoring error concerns noted in the literature.

Since each risk element is characterized in terms of its likelihood of occurrence and subsequent impact should the event occur, a two-dimensional P–I table as shown in Figure 3.6 can be used to categorically match each risk with its pairwise correlation to the two characteristics. The resulting table enables the systems team to prioritize its risk management efforts appropriate to the threat level posed by specific risk elements.

For example, risk 5 has been estimated by stakeholders to have a very low likelihood of occurring and, if it does occur, will have very low impact on the system. Although it would continue to be monitored and measured throughout the system life cycle stages in which it was present, it more than likely would receive very little mitigation effort on the part of the systems team. Risk 4 on the other hand, has been estimated by stakeholders to have a high likelihood of occurring and, if it does occur, will have a high (and serious) impact on the success of the project, would command a good degree of attention throughout the life cycle of the system.

Figure 3.7 shows that the stakeholders consider risk element 3 to have three different impacts on the system: schedule (S), technical performance (P), and cost ($), each with varying estimations on their likelihood of occurring and their potential impact should they occur. In this example, the likelihood of violating cost limits for the program is estimated to be very low, but if it does occur it has the possibility of potentially terminating the program because of its very high impact. This is an example of an extreme event described earlier. Its low probability of occurrence does very little to allay the fears associated with this risk, should it occur.

P–I tables provide an important perspective on the anticipated risks that a system or project will face. To form a comprehensive understanding, P–I table

Probability impact table for project x risk elelments						
Impact	Very high	3				
	High				4	
	Medium					
	Low		1,2	6		
	Very low	5				
		Very low	Low	Medium	High	Very high
		Probability of occurrence				

Figure 3.6 Example P–I table for six risk elements.

Probability impact table for risk element 3						
Impact	Very high	$		5		
	High					
	Medium					
	Low		P			
	Very Low					
		Very Low	Low	Medium	High	Very High
		Probability of occurrence				

Figure 3.7 Specific P–I table for risk element 3.

results should be combined with other methods as appropriate. These include: capturing the frequency of occurrence, estimating correlation to other risks, estimating "time to impact" if a risk were to come to fruition, using decision analysis, and incorporating simulation experiments to assess the dynamic effects associated with risk. Generally, the analysis proceeds through increasing levels of risk quantification, beginning with a qualitative identification of the risk, followed by an understanding of the plausible range of each parameter, a "best estimate" of each parameter, and finally an estimate of the probability distribution of each parameter and the effect on the overall program. The size of the system, the severity of the risks, and the time available will determine the appropriate degree of quantification.

Assessment of technical risk, which involves the possibility that a requirement of the system will not be achieved, is enabled by functional analysis, introduced in Chapter 2, and further discussed in Chapter 10. Through functional analysis, the systems engineer defines the functions that the system must perform to be successful. Technical risk is assessed by considering each function and the likelihood and consequence of hazards to that function.

It is important to consider any required interactions between functions and risks to these interactions. Being sensitive to the connections between all elements of a systems solution forces a systems team to pay attention to a common source of failure in complex systems: *the seams of a system*. These seams afford interface compatibility and sharing protocols between systems. They are system boundaries rather than system components. Because of this, they are easily overlooked during risk identification sessions with stakeholders who have not internalized a systems thinking perspective of their operational environment. Accounting for the importance of each function and the degree of performance required enables a prioritization and comprehensive treatment of technical risk.

Assessment of cost risk, which involves the possibility of exceeding the design, development, production, or operating budgets in whole or in part, consists of examining (a) the various costs associated with a system solution or project, (b) their uncertainties, and (c) any possible risks and opportunities that may affect these costs. The risks and opportunities of interest are those that could potentially increase or decrease the estimated costs of the project, this includes decisions made

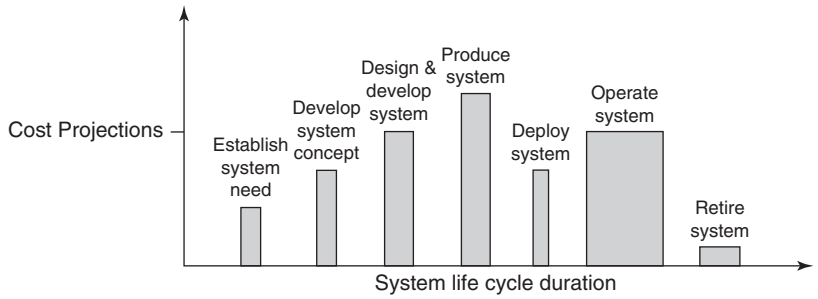

Figure 3.8 Example system life cycle cost profile.

throughout the system life cycle, which may have downstream effects on the total system life cycle costs. These risks are projected for all stages of the life cycle of a system project. Assessment of cost risk is enabled by an understanding of the uncertainty involved with the major cost drivers. The resulting analysis produces a projected system life cycle cost profile as shown in Figure 3.8. This profile varies by the type of system. A software program, for example, has high design and development costs, but generally lower production and deployment costs. These estimates are less certain and more likely to vary the more into the future they occur. Chapter 5 discusses life cycle costing in detail and describes the use of Monte Carlo simulation analysis to assess cost risk.

Assessment of schedule risk, which involved the possibility that the system or project will fail to achieve a scheduled key milestone, examines the time allotted to complete key tasks associated with the project, the interrelationships between these tasks, and the associated risks and opportunities that may affect the timing of task accomplishment. Schedule risk analysis relies on analytical methods such as Pert charts to unveil the sometimes complex logical connections between the tasks. Chapter 13, which describes the Solution Implementation phase of the SDP, addresses scheduling of program tasks, duration, and their interrelationships, as well as identifying system activities that lie on a critical path to project success. These critical path tasks should be a primary focus of schedule risk assessment, because a delay in the completion of any of these tasks will result in a delay in the overall program schedule.

Programmatic risk assessment considers all threats to successful system development and deployment resulting from external effects and decisions imposed on the system or project. This assessment is informed by an understanding of the system and its relation to lateral systems and the metasystem within which it is spatially located. A thorough stakeholder analysis, discussed in Chapter 10, will also enable an assessment of the programmatic risks.

The nature and methods of risk assessment vary somewhat across the risk categories described. In addition to assessing the risks in each category, a systems engineer must consider the seams here as well: possible interactions between risk categories. These interactions can impose correlations that should be included in Monte Carlo simulation models [31]. For example, schedule delays could result

in monetary fines for not meeting agreed-upon contractual deadlines. Also, there may exist correlation between risks, with the occurrence of one risk increasing (or decreasing) the likelihood of other events happening. These dependencies can again be modeled using simulation to analyze the effect of *simultaneous* variation in cost, schedule, and value (technical performance) outcomes. Dependencies between cost elements can be accounted for using correlation coefficients [32] or they can be explicitly modeled [33].

By assessing the relative likelihoods and consequences of each risk across and among each category, risks can be prioritized, policy can be set, and actions can be taken to effectively and efficiently mitigate risks.

3.4.3 Risk Mitigation

With the knowledge gained through risk identification and risk assessment, project managers and systems engineers are equipped to reduce risk through a program of *risk mitigation* designed to monitor, measure, and mitigate risk throughout the system life cycle. Risks should be continuously monitored once identified, even if their assessed threat to the success of the program is minor. Time and situational factors beyond the control of the systems team and stakeholders can dramatically increase (or decrease) the potential threat posed by risk factors. Maintaining a watchful eye on the system environment throughout a systems decision problem helps to identify these risks early, thereby reducing the likelihood of unwelcome surprises.

The goal of risk mitigation is to take action to decrease the risk-based variance on performance, cost, value, and schedule parameters over the entire system life cycle. Figure 3.9 shows a graphical illustration of the variance of a project's total cost

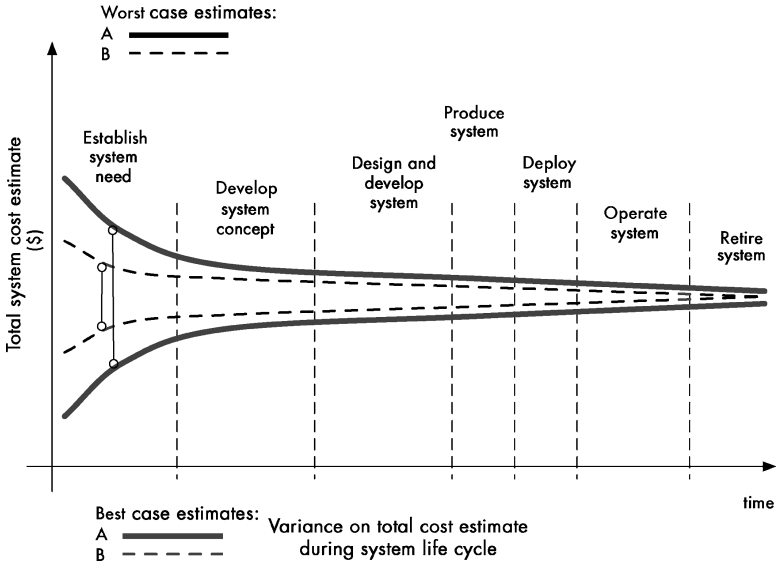

Figure 3.9 Estimate of system cost variance over life cycle.

estimate before effective risk management (A) and after (B). The spread between worst-case and best-case estimates is reduced earlier in the life cycle, yielding more accurate estimates of total system costs and dramatically reducing the threat of cost overruns to project success. Effective risk management has a likewise effect on value, technical performance, and schedule.

Once risks are identified, are actively being monitored, and are being measured, systems teams should be proactive in taking action to mitigate the potential threats to the system or project. Simply being aware of potential system or project risks is insufficient to properly manage or control the degree of their presence or impact. The primary means of deciding how to do this is through a risk management plan that clearly prioritizes the risks in terms of their relative likelihoods and consequences. To be successful, the risk management plan must be supported by organizational and project leadership. By properly aligning incentives, technical expertise, and authority, these leaders can help facilitate the greatest likelihood of overall success.

Once a risk has been identified, assessed, and determined to require mitigation, there are several options available to mitigate system risk. It may be possible to *avoid* the risk, if the organization can take action to reduce the probability of occurrence to zero or completely eliminate the consequences of the risk. It may be appropriate to *transfer* the risk to another organization through a contract; an insurance policy is one example of this approach. An organization may *reduce* risk by taking action to reduce the likelihood of the hazard occurring or reduce the severity of consequences if the hazard does occur. Finally, an organization may choose to *accept* risk if it has little or no control over the risk event and the overall system threat is considered to be very low. Each risk should be considered individually and within the context of the larger system as management decides on the appropriate approach (avoid, transfer, reduce, or accept) and the subsequent actions to take as a result.

All system activities involve risk; therefore, risk management must be a continuous process applied throughout the system life cycle of any systems engineering project. To illustrate this concept as it applies to a large-scale project, Figure 3.10 graphically displays the degree of involvement for major program teams throughout the shuttle life cycle stages identified by the Johnson Space Center. Of particular note is the one category that engages in risk management right from the start: the design and development teams. This just happens to be the principal location of systems engineers for the program.

3.5 SUMMARY

It is impossible to successfully complete systems decision problems without integrating comprehensive planning across an entire system life cycle. Current system life cycle models are based on standards set by professional organizations.

The system life cycle provides an effective metaphor for structuring the critical activities performed by a systems team during a project. The system life cycle introduced in this chapter consists of seven stages: establish system need, develop

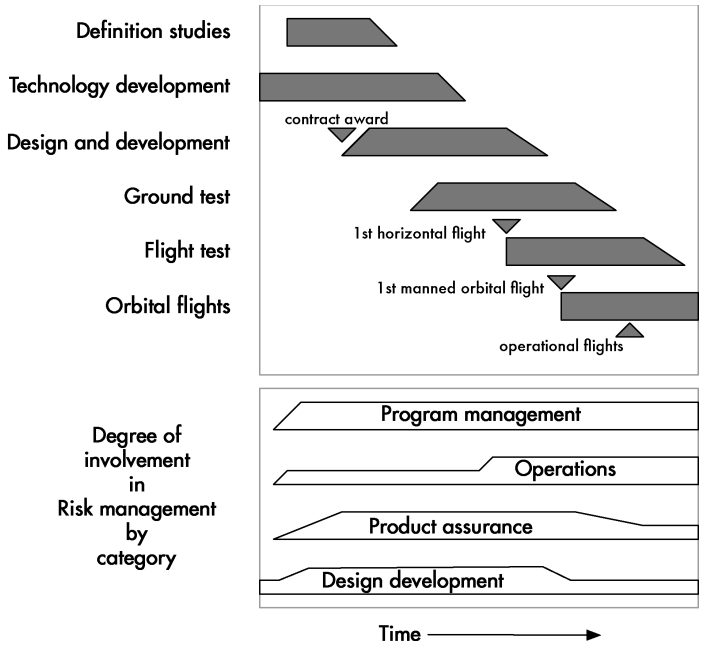

Figure 3.10 Degree of involvement in risk management across program life cycle (NASA JSC).

system concept, design and develop system, produce system, deploy system, operate system, and retire system. As described in Chapter 2, there are six principal stakeholder groups associated with every systems decision problem. All six of these groups are completely imbedded in life cycle stage activities to maximize the potential contributions their expertise has on a program.

The life cycle is separate from the SDP that systems teams execute during each stage of a system life cycle. The SDP is described in Part III.

The system life cycle model structures risk management necessary for continued successful operation. Effective risk management is crucial to success. Identifying risks is a challenging task, further complicated by the need to focus attention on the seams of systems, boundaries at which required systems interactions often fail. Awareness of these seams and the ability to identify and anticipate second-, third-, and higher-order consequences of risk events is enhanced through systems thinking. Failing to adequately manage risks will cause problems for a system. Risk management is an ongoing activity throughout the entire system life cycle.

3.6 EXERCISES

3.1. For each of the following systems, use the system life cycle model introduced in this chapter to identify the life cycle stage you believe the system

to be in based on your current level of knowledge and understanding of the system.

(a) The Vonage internet communications system.

(b) Trade unions in the United States.

(c) Satellite radio in the United States.

(d) Al-Qaeda terrorist network.

(e) The relationship existing between you and your current group of friend.

(f) Cancer research in the United States.

(g) Apple iPhone.

(h) Your parent's automobile(s).

(i) Mono Lake ecosystem in California.

(j) The aquifer system supporting Phoenix, Arizona.

(k) Amtrak rail system in the United States.

(l) MySpace.com.

(m) Legal immigration process.

(n) Eurail pass system.

(o) Professional cricket in Pakistan.

(p) The musical group Gorillaz.

(q) Professional soccer in the United States.

(r) The U.S. interstate highway system.

3.2. Given the stages you identified for each of the systems in the previous question, we now want to consider risk relative to these stages. For each of the systems:

(a) List the risks you believe exist that threaten its successful continued operation.

(b) Construct a probability–impact (P–I) table using the scales shown in Figure 3.6 to place each of the risks.

(c) From the risks that you identified, pick the two highest and two lowest assessed risks and construct individual P-I tables for each of these following the example in Figure 3.7.

(d) For the four risks that you constructed P–I tables, identify two actions that could be taken by system owners or users to manage these risks. Additionally, explain whether these management actions would eliminate, mitigate, or have little effect on the level of risk that you assessed?

(e) Lastly, if the systems associated with these risks advance to their next life cycle stage, will the risks you assesed still be relevant? Explain.

REFERENCES

1. Merton, RK. The unanticipated consequences of purposive social action. *American Sociological Review*, 1936;1(6):894–904.
2. Archibald, RD. Management state of the art. Max's Management Wisdom. Available at http://www.maxwideman.com. Accessed January 20, 2005.
3. George, ML. *Lean Six Sigma*. New York: McGraw-Hill, 2002.
4. Hoffman, DG. *Managing Operational Risk: 20 Firmwide Best Practice Strategies*. New York: John Wiley & Sons, 2002.
5. Amari, SV, McLaughlin, L. Optimal design of a condition-based maintenance model. Proceedings of the Reliability and Maintainability, 2004 Annual Symposium—RAMS, pp. 528–533.
6. A Survey of System Development Process Models, CTG.MFA-003. Center for Technology in Government. New York: University of Albany, 1998.
7. Nuclear Regulatory Legislation, NUREG-0980, 1(6), 107th Congress, 1st Session, Office of the General Council, U.S. Nuclear Regulatory Commission, Washington, DC, June 2002.
8. Sheard, SA, Lake, JG. Systems engineering standards and models compared. Software Productivity Consortium, NFP, 1998. Available at http://www.software.org/pub/externalpapers/9804-2.html.
9. Price, S, John, P. The status of models in defense systems engineering. In: Pidd, M, editor. *Systems Modeling*: *Theory and Practice*. West Sussex, England: John Wiley & Sons, 2004.
10. Boehm, B. *Spiral Development*: *Experience, Principles, and Refinements*. CMU/SEI-2000-SR-008. Pittsburgh, PA: Carnegie Mellon Software Engineering Institute, 2000.
11. Maner, W. 1997: Rapid applications development. Available at http://csweb.cs.bgsu.edu/maner/domains/RAD.htm. Accessed June 7, 2006.
12. The Agile Manifesto. Available at http://www.agilemanifesto.org. Accessed July 8, 2006.
13. Garvey, PR. *Analytical Methods for Risk Management*. Boca Raton, FL: Chapman & Hall/CRC Press, 2009.
14. Rethinking Risk Management in Financial Services. Report of the World Economic Forum, New York. Available at http://www.weforum.org/pdf/FinancialInstitutions/RethinkingRiskManagement.pdf. Accessed April 20, 2010.
15. Vose, D. *Risk Analysis*: *A Quantitative Guide*. West Sussex, England: John Wiley & Sons, 2000.
16. Haimes, YY. Total risk management. *Risk Analysis*, 1991;11(2):169–171.
17. Post-Challenger evaluation of space shuttle risk assessment and management. The Committee on Shuttle Criticality Review and Hazard Analysis Audit, Aeronautics and Space Engineering Board. Washington, DC: National Academy Press, 1988.
18. Haimes, YY. *Risk Modeling, Assessment and Management*. New York: John Wiley & Sons, 1998.
19. INCOSE-TP-2003-016-02. *Systems Engineering Handbook*. Seattle, Washington.

20. Galway, LA. Subjective probability distribution elicitation in cost risk analysis: A review. RAND Techical Report: Project Air Force. Santa Monica, CA: RAND Corporation, 2007.
21. Primavera Risk Analysis, ORACLE Data Sheet. Available at http://www.oracle.com. Accessed April 20. 2010.
22. Operational Risk. Report of the Basel Committee on Banking Supervision Consultative Document, Bank for International Settlements, January 2001.
23. Containing systemic risk: The road to reform. Report of the Counterparty Risk Management Policy Group III. New York, NY: available at http://www.crmpolicygroup.org, August 6, 2008.
24. Labonte, M. Systemic risk and the Federal Reserve. CRS Report for Congress R40877, Washington, DC: October 2009.
25. Modarres, M. *Risk Analysis in Engineering: Techniques, Tools and Trends*, Boca Raton, FL: CRC Press, 2006.
26. Haimes, YY. *Risk Modeling, Assessment, and Management*. Hoboken, NJ: John Wiley & Sons, 2004.
27. Simon, P. *Risk Analysis and Management*. London, England: AIM Group, 1997.
28. Roland, HE, Moriarty, B. *System Safety Engineering and Management*, 2nd ed. New York: John Wiley & Sons, 1990.
29. Hubbard, DW. *The Failure of Risk Management*. Hoboken, NJ: John Wiley & Sons, 2009.
30. Kahneman, D, and Tversky, A. Subjective probability: A judgment of representativeness. *Cognitive Psychology*, 1972;3:430–454.
31. New horizons in predictive modeling and risk analysis. ORACLE White Paper, ORACLE Corporation, Redwood Shores, California, 2008.
32. Book, SA. Estimating probable system cost. *Crosslink*, 2001;2(1):12–21.
33. Garvey, PR. *Probabilistic Methods for Cost Uncertainty Analysis: A Systems Engineering Perspective*. New York: Marcel Decker, 2000.