Chapter 2. Risk Analysis—Identifying and Prioritizing Needs

with Christopher Alberts and Audrey Dorofee

In This Chapter

Risk management in systems acquisition and development has typically focused exclusively on cost and schedule concerns. Organizations fund desired features and functions selected for implementation based on cost estimates, budget availability, and perceived criticality of need. Organizations closely monitor changes in any of these three areas and make adjustments to planned delivery dates and features based on risk evaluation.

Risk is one of the assurance principles described in **Chapter 1**, "**Cyber Security Engineering: Lifecycle Assurance of Systems and Software**," and effective risk management of software assurance is a competency that is not consistently applied in acquisition and development projects.

This competency considers what could go wrong and establishes how to reduce, mitigate, or avoid the undesirable results that would occur if the risk were realized. Most project participants focus on how to reach success and dismiss those raising the problems that may impede achieving the project's objectives. A successful project needs both perspectives working collaboratively side by side.

Risk can be connected to systems and software from many directions, and organizations must consider all of those connections to effectively manage risk. Acquisition and development are complex, and opportunities for things to go wrong abound. Effective risk analysis for assurance requires, at a minimum, consideration of the following types of risk:

- Development risk

- Acquisition risk

- Mission risk

Development and acquisition risks typically dominate risk management efforts and relate primarily to cost and schedule. These are actually short-term concerns, but they dominate the early stages of the lifecycle. In this chapter we explore ways to consider the software assurance aspects of all three types of risk.

2.1 Risk Management Concepts

For risk to exist in any circumstance, all of the following must be true [Alberts 2002]:

- The potential for loss exists.

- Uncertainty related to the eventual outcome is present.[1]

[1]. Some researchers separate the concepts of certainty (the absence of doubt), risk (where the probabilities of alternative outcomes are known), and uncertainty (where the probabilities of possible outcomes are unknown). However, because uncertainty is a fundamental attribute of risk,

we do not differentiate between decision making under risk and decision making under uncertainty.

- Some choice or decision is required to deal with the uncertainty and potential for loss.

The essence of risk, no matter what the domain, can be succinctly captured by the following definition of risk: *Risk is the probability of suffering harm or loss.*[2]

[2]. This definition is derived from the definition used in *Introduction to the Security Engineering Risk Analysis (SERA) Framework* [Alberts 2014].

Figure 2.1 illustrates the three components of risk:

- **Potential event**—An act, an occurrence, or a happening that alters current conditions and leads to a loss

- **Condition**—The current set of circumstances that leads to or enables risk

- **Consequence**—The loss that results when a potential event occurs; the loss is measured in relationship to the status quo (i.e., current state)

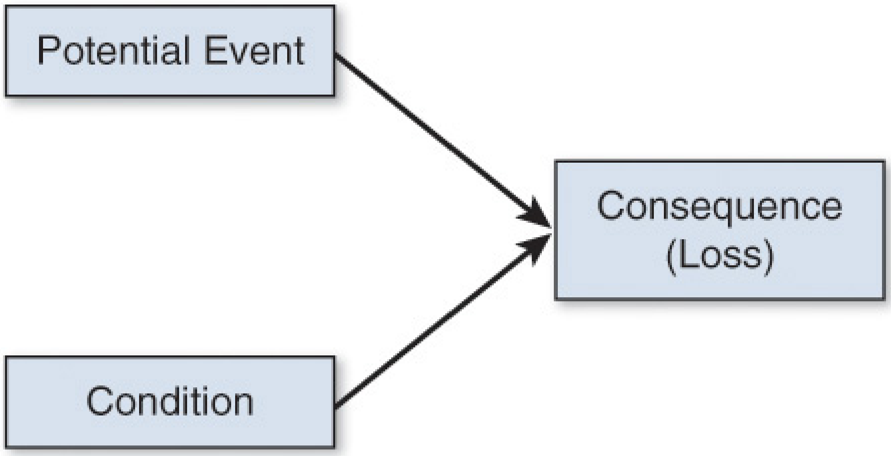


Figure 2.1 *Components of Risk*

From the risk perspective, a condition is a passive element. It exposes an entity[3] (e.g., project, system) to the loss triggered by the occurrence of an event. However, by itself, a risk condition does not cause an entity to suffer a loss or experience an adverse consequence; it makes the entity vulnerable to the effects of an event [Alberts 2012a].

[3] An *entity* is an object affected by risk. The entities of interest in this chapter are interactively complex, software-reliant systems. Examples include projects, programs, business processes, and networked technologies.

Consider the following scenario: A project team is developing a software system for a customer. The team has enough people with the right skills to perform its tasks and complete its next milestone on time and within budget (status quo). However, the team does not have redundancy among team members' skills and abilities (condition). If the team loses people with certain key skills (potential event), then it will not be able to complete its assigned tasks (consequence/loss). This puts the next milestone in jeopardy, which is a loss when measured in relationship to the status quo (on track to achieve the next milestone).

However, if none of the team members leaves or is reassigned (the event does not occur), then the project should suffer no adverse consequences. Here, the condition enables the event to produce an adverse consequence or loss.

When a risk occurs, an adverse consequence (a loss) is realized. This consequence ultimately changes the current set of conditions confronting the entity (project or system). In this example, a realized risk means that the project team has lost people and no longer has enough people to complete its assigned tasks. The project now faces a problem that must be resolved. Put another way, the risk has become an issue/problem (a condition that directly results in a loss or adverse consequence).

Three measures are associated with a risk: probability, impact, and risk exposure.[4] The basic relationships between probability and impact and the components of risk are shown in Figure 2.2.[5] In this context, *probability* is defined as a measure of the likelihood that an event will occur, while *impact* is defined as a measure of the loss that occurs when a risk is realized. Risk exposure provides a measure of the magnitude of a risk based on current values of probability and impact.

[4] A fourth measure, time frame, is sometimes used to measure the length of time before a risk is realized or the length of time in which action can be taken to prevent a risk.

[5] The relationships between probability and impact and the components of risk depicted in Figure 2.2 are based on the simplifying assumption that the loss resulting from the occurrence of an event is known with certainty. In many cases, a range of adverse outcomes might be possible. For example, consider a project team that is worried about the consequence of losing team members. The magnitude of the loss will depend on a number of factors, such as which team member leaves the project, whether anyone is available to take the team member's place, the skills and experience of potential replacements, and so forth. The consequence could be minor if an experienced person is available to step in and contribute right away. On the other hand, the consequence could be severe if no one is available to step in and contribute. A range of probable outcomes is thus possible. When multiple outcomes are possible, probabilities are associated with the potential outcomes. As a result, risk analysts must consider two probabilities—one associated with the potential event and another associated with the consequence. However, basic risk assessments assume that the loss is known with relative certainty (or they only focus on the most likely consequence), and only the probability associated with the event is considered.

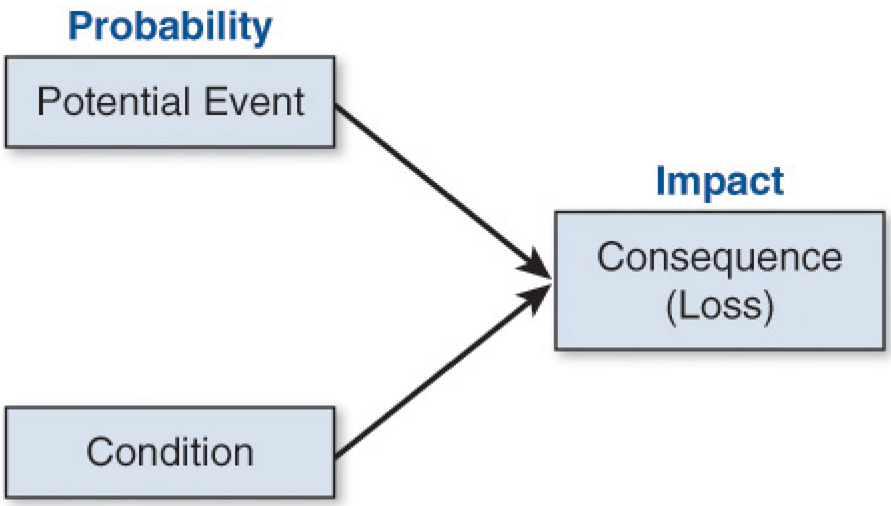


Figure 2.2 *Risk Measures and the Components of Risk (Simplified View)*

Risk management is a systematic approach for minimizing exposure to potential losses. It provides a disciplined environment for the following:

- Continuously assessing what could go wrong (i.e., assessing risks)

- Determining which risks to address (i.e., setting mitigation priorities)

- Implementing actions to address high-priority risks through avoidance or mitigation

Figure 2.3 illustrates the three core risk management activities:

- **Assess risk**—Assessment involves transforming concerns people have into distinct, tangible risks that are explicitly documented and analyzed.

- **Plan for controlling risk**—Planning involves determining an approach for addressing each risk and producing a plan for implementing the approach.

- **Control risk**—Controlling risk involves dealing with each risk by implementing its defined control plan and tracking the plan to completion.

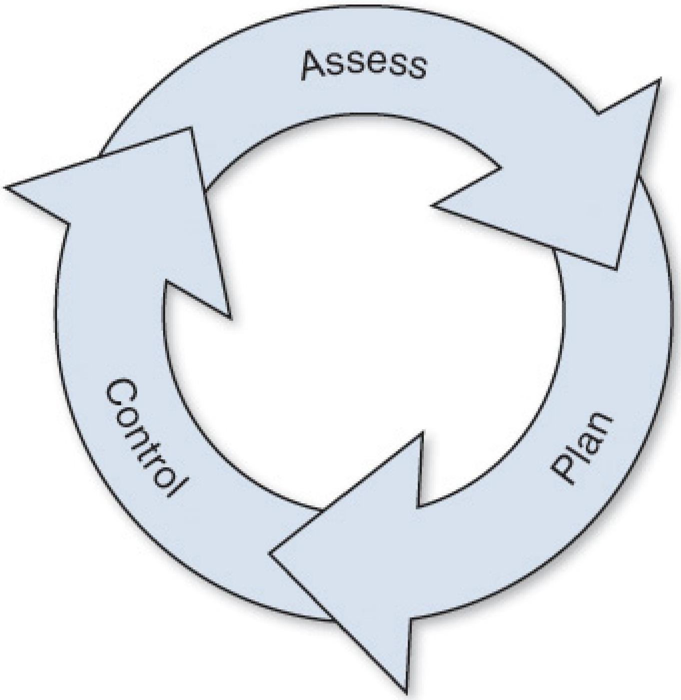


Figure 2.3 *Risk Management Activities*

When you consider the subactivities under the three main activities, the connection to the well-known "Plan, Do, Check, Act" (PDCA) model is apparent:

- Individuals and interactions over processes and tools

- Working software over comprehensive documentation

- Attributes

- Responding to change over following a plan

- Activity 2.1 Assess risk

- 2.1.1 Identify risk

- 2.1.2 Analyze risk

- 2.1.3 Develop risk profile

- Activity 2.2 Plan for risk control

- 2.2.1 Determine control approach

- 2.2.2 Develop control plan

- Activity 2.3 Control risk

- 2.3.1 Implement control plan

- 2.3.2 Track control plan

- 2.3.3 Make tracking decision

The mapping to PDCA is

- **Plan**—2.2.2 Develop control plan

- **Do**—2.3.1 Implement control plan

- **Check**—2.3.2 Track control plan

- **Act**—2.3.3 Make tracking decision

Everything before subactivity 2.2.2 (risk identification, risk analysis, risk prioritization/risk profile, and control approach) prepares risk management personnel to be able to implement the PDCA cycle. The same type of mapping could be done for the OODA (Observe, Orient, Decide, and Act) decision-making framework.

One of the fundamental conditions of risk is uncertainty regarding its occurrence. A risk, by definition, might or might not occur. With an issue, no uncertainty exists—the condition exists and is having a negative effect on performance.[6] Issues can also lead to (or contribute to) risks by

- Creating a circumstance that enables an event to trigger additional loss
- Making an existing event more likely to occur
- Aggravating the consequences of existing risks

Figure 2.4 illustrates the two components of an issue or a problem:

- **Condition**—The current set of circumstances that produces a loss or an adverse consequence
- **Consequence**—The loss that is triggered by an underlying condition that is present

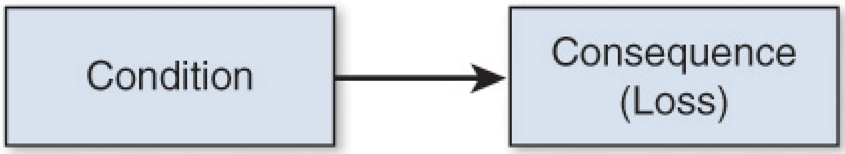


Figure 2.4 *Components of an Issue/Problem*

From the issue perspective, a condition directly causes an entity (e.g., project,- system) to suffer a loss or experience an adverse consequence. Unlike a risk, an issue does not need an event to occur to produce a loss or an adverse consequence.

[6] Many of the same tools and techniques can be applied to both issue and risk management.

2.2 Mission Risk

From the mission perspective, *risk* is defined as the probability of mission failure (i.e., not achieving key objectives). *Mission risk* aggregates the effects of multiple conditions and events on a system's ability to achieve its mission.

Mission risk analysis is based on systems theory.[7] The underlying principle of systems theory is to analyze a system as a whole rather than decompose it into individual components and then analyze each component separately [Charette 1990]. In fact, some properties of a system are best analyzed by considering the entire system, including the following:

- Influences of environmental factors
- Feedback and nonlinearity among causal factors
- Systemic causes of failure (as opposed to proximate causes)
- Emergent properties

2.3 Mission Risk Analysis

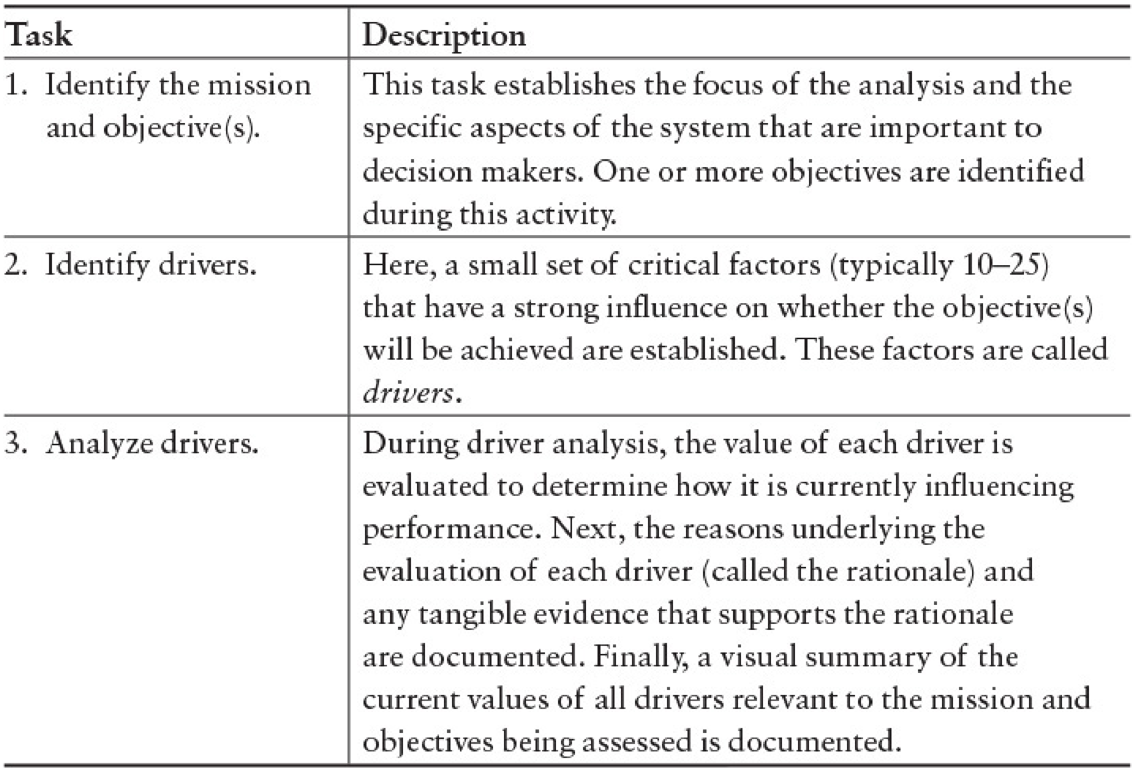
The goal of mission risk analysis is to gauge the extent to which a system is in a position to achieve its mission and objective(s). This type of risk analysis provides a top-down view of how well a system is addressing risks.

The Mission Risk Diagnostic (MRD) [Alberts 2006] is one method that can be used to address this type of analysis. The first step in this type of risk analysis is to establish the objectives that must be achieved. The objectives define the desired outcome, or "picture of success," for a system. Next, systemic factors that have a strong influence on the outcome (i.e., whether the objectives will be achieved) are identified. These systemic factors, called *drivers* in this chapter, are important because they define a

[7] Because mission risk analysis is based on system theory, the term *systemic risk* can be used synonymously with *mission risk*. The term *mission risk* is used throughout this chapter.

small set of factors that can be used to assess a system's performance and gauge whether the system is on track to achieve its key objectives. The drivers are then analyzed to enable decision makers to gauge the overall risk to the system's mission.

Table 2.1 presents a summary of the three core tasks that form the basis of the MRD. The MRD comprises 13 tasks that must be completed. (A description of all MRD tasks is provided in Section 5 of the *Mission Risk Diagnostic (MRD) Method Description* [Alberts 2006].)



Task	Description
1. Identify the mission and objective(s).	This task establishes the focus of the analysis and the specific aspects of the system that are important to decision makers. One or more objectives are identified during this activity.
2. Identify drivers.	Here, a small set of critical factors (typically 10–25) that have a strong influence on whether the objective(s) will be achieved are established. These factors are called *drivers*.
3. Analyze drivers.	During driver analysis, the value of each driver is evaluated to determine how it is currently influencing performance. Next, the reasons underlying the evaluation of each driver (called the rationale) and any tangible evidence that supports the rationale are documented. Finally, a visual summary of the current values of all drivers relevant to the mission and objectives being assessed is documented.

Table 2.1 *Core Tasks of the MRD*

We describe how to address each of these core tasks in the following sections.

2.3.1 Task 1: Identify the Mission and Objective(s)

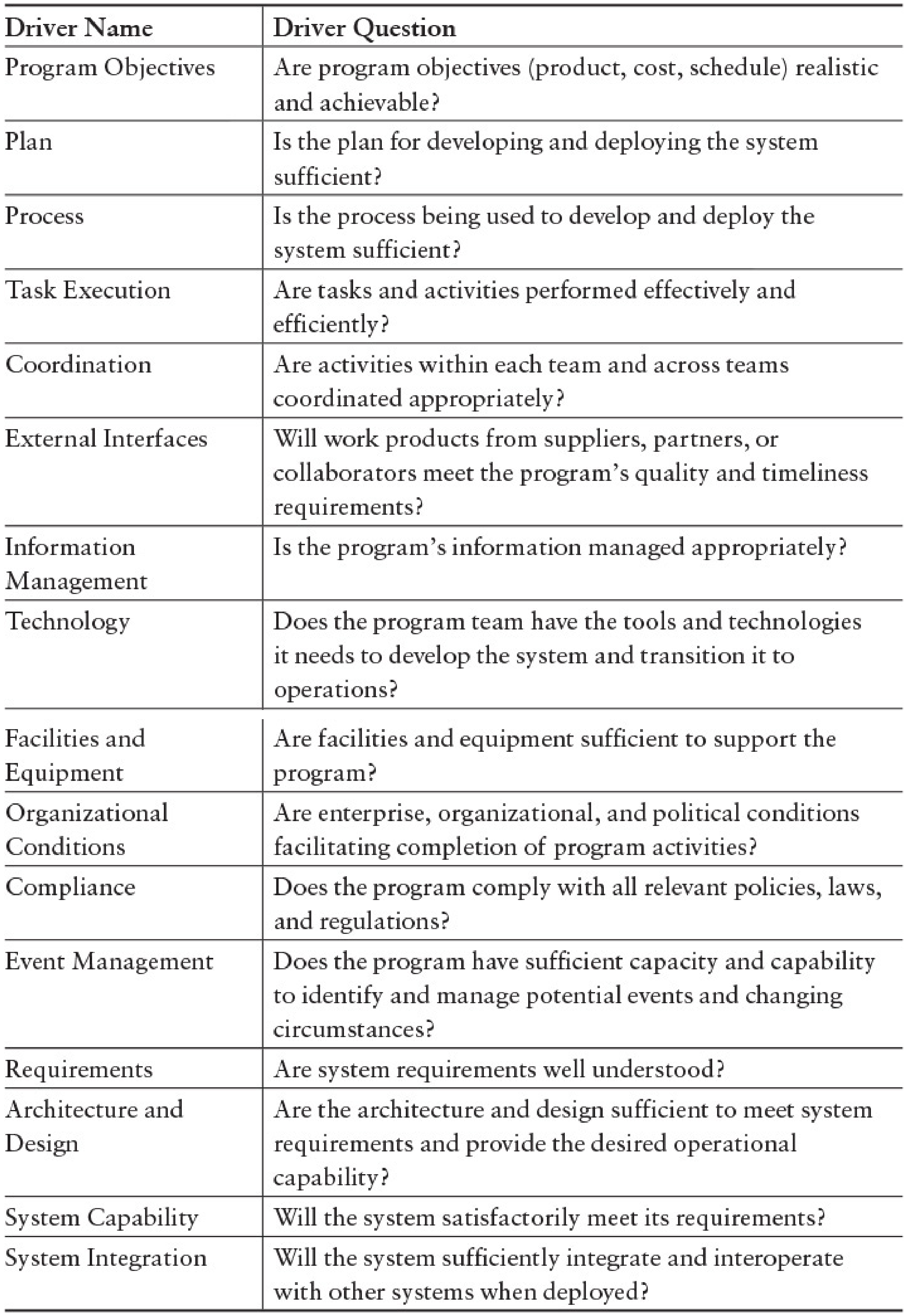
The overarching goals when identifying the mission and objective(s) are to (1) define the fundamental purpose, or mission, of the system that is being examined and (2) establish the specific aspects of the mission that are important to decision makers. Once they have been established, the mission and objective(s) provide the foundation for conducting the assessment.

The mission statement is important because it defines the target, or focus, of the analysis effort. Each mission typically comprises multiple objectives. When assessing a system, analysts must select which specific objective(s) will be evaluated during the assessment. Selecting objectives refines the scope of the assessment to address the specific aspects of the mission that are important to decision makers.

While decision makers have a tacit understanding of their objectives, they often cannot precisely articulate or express the objectives in a way that addresses the criteria. If a program's objectives are not clearly articulated, decision makers may have trouble assessing whether the program is on track for success.

2.3.2 Task 2: Identify Drivers

The main goal of driver identification is to establish a set of systemic factors, called *drivers*, that has a strong influence on the eventual outcome or result to be used to measure performance in relation to a program's mission and objectives. Knowledge within the organization can be tapped to review and refine the prototype set of drivers provided in **Table 2.2**. Once the set of drivers is established, analysts can evaluate each driver in the set to gain insight into the likelihood of achieving the mission and objectives. To measure performance effectively, analysts must ensure that the set of drivers conveys sufficient information about the mission and objective(s) being assessed.



Driver Name	Driver Question
Program Objectives	Are program objectives (product, cost, schedule) realistic and achievable?
Plan	Is the plan for developing and deploying the system sufficient?
Process	Is the process being used to develop and deploy the system sufficient?
Task Execution	Are tasks and activities performed effectively and efficiently?
Coordination	Are activities within each team and across teams coordinated appropriately?
External Interfaces	Will work products from suppliers, partners, or collaborators meet the program's quality and timeliness requirements?
Information Management	Is the program's information managed appropriately?
Technology	Does the program team have the tools and technologies it needs to develop the system and transition it to operations?
Facilities and Equipment	Are facilities and equipment sufficient to support the program?
Organizational Conditions	Are enterprise, organizational, and political conditions facilitating completion of program activities?
Compliance	Does the program comply with all relevant policies, laws, and regulations?
Event Management	Does the program have sufficient capacity and capability to identify and manage potential events and changing circumstances?
Requirements	Are system requirements well understood?
Architecture and Design	Are the architecture and design sufficient to meet system requirements and provide the desired operational capability?
System Capability	Will the system satisfactorily meet its requirements?
System Integration	Will the system sufficiently integrate and interoperate with other systems when deployed?

Table 2.2 *Prototype Set of Driver Questions for Software Acquisition and Development Programs*

Each driver has two possible states: a success state and a failure state. The *success state* means that the program's processes are helping to guide the program toward a successful outcome (i.e., achieving the objective[s] being evaluated). In contrast, the failure state signifies that the program's processes are driving the program toward a failed outcome (i.e., not achieving the objective[s] being evaluated).

2.3.3 Task 3: Analyze Drivers

Analysis of a driver requires determining how it is currently acting (i.e., its current state) by examining the effects of conditions and potential events on that driver. The goal is to determine whether the driver is

- Almost certainly in its success state

- Most likely in its success state

- Equally likely to be in its success or failure states

- Most likely in its failure state

- Almost certainly in its failure state

This list can be used to define a qualitative scale for driver analysis.

As illustrated in Figure 2.5, a relationship exists between a driver's success state (as depicted in a driver profile) and mission risk. A driver profile shows the probability that drivers are in their success states. Thus, a driver with a high probability of being in its success state (i.e., a high degree of momentum toward the mission) translates to a low degree of mission risk. Likewise, a driver with a low probability of being in its success state (i.e., a high probability of being in its failure state) translates to a high degree of mission risk.

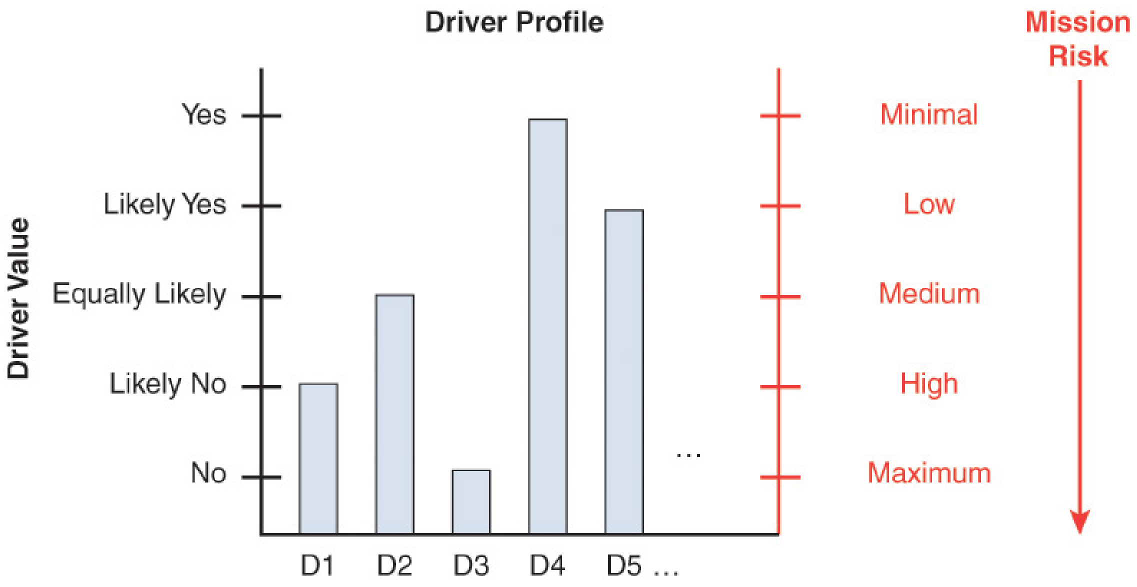


Figure 2.5 *The Relationship Between Driver Value and Mission Risk*

The driver profile thus helps decision makers understand how a system is performing against potential mission risks.

2.4 Security Risk

Security risk is a measure of (1) the likelihood that a threat will exploit a vulnerability to produce an adverse consequence or loss and (2) the magnitude of the loss. Figure 2.6 illustrates the three core components of security risk:

- **Threat**—A cyber act, occurrence, or event that exploits one or more vulnerabilities and leads to an adverse consequence or loss

- **Vulnerability**—A weakness in an information system, system security procedures, internal controls, or implementation that a threat could exploit to produce an adverse consequence or loss; a current condition that leads to or enables security risk

- **Consequence**—The loss that results when a threat exploits one or more vulnerabilities; the loss is measured in relationship to the status quo (i.e., current state)

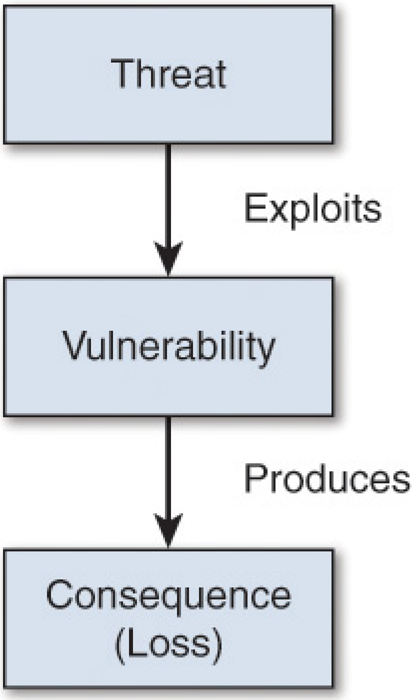


Figure 2.6 *Components of Security Risk*

From the security perspective, a *vulnerability* is the passive element of risk. It exposes cyber technologies (e.g., software application, software-reliant system) to threats and the losses that those threats can produce.

However, by itself, a vulnerability does not cause an entity to suffer a loss or experience an adverse consequence; rather, the vulnerability makes the entity susceptible to the effects of a threat.[8]

[8] Adapted from the book *Managing Information Security Risks: The OCTAVE Approach* [Alberts 2002].

The strategy for controlling a risk is based on the measures of the risk (i.e., probability, impact, and risk exposure), which are established during the risk assessment. Decision-making criteria (e.g., for prioritizing risks or deciding when to escalate risks within an organization) can help determine the appropriate strategy for controlling a risk. Common control approaches include the following:

- **Accept**—If a risk occurs, its consequences will be tolerated; no proactive action to address the risk will be taken. When a risk is accepted, the rationale for doing so is documented.

- **Transfer**—Risk mitigation is shifted to another party (e.g., through insurance or outsourcing). The system owner always retains responsibility for managing the risk, even if it is transferred.

- **Avoid**—Activities are restructured to eliminate the possibility of a risk occurring.

- **Mitigate**—Actions are implemented in an attempt to reduce or contain a risk.

For any security risk that is not accepted, a security analyst should develop and document a control plan for that risk. A control plan defines a set of actions for implementing the selected control approach. For risks that are being mitigated, plans can include actions from the following categories:

- **Recognize and respond**—Monitor the threat and take action when it is detected.

- **Resist**—Implement protection measures to reduce vulnerability to the threat and minimize any consequences that might occur.

- **Recover**—Recover from the risk if the consequences or losses are realized.

In order to fully address a security risk, it is important to understand the environment in which it resides. The focal point of the environment is the threat actor. A common goal of many threat actors is to inflict harm or loss on a mission's stakeholders. To accomplish that goal, a threat actor first targets data used to support a workflow or mission thread.[9] To access targeted mission data, a threat actor must navigate through the complex network of people, processes, and technologies, looking for weaknesses to exploit in organizational security practices and vulnerabilities in software-reliant systems. Getting to the mission data can be difficult. A threat actor may need to jump from one targeted computer to another when attempting to achieve the goal of the attack. In many cases, an actor may target computers that are owned and maintained by trusted partners and third-party collaborators when conducting a cyberattack.

[9]. A workflow is a collection of interrelated work tasks that achieves a specific result [Leveson 2004]. A workflow includes all tasks, procedures, organizations, people, technologies, tools, data, inputs, and outputs required to achieve the desired objectives. The business literature uses several terms synonymously with *workflow*, including *work process*, *business process*, and *process*. *Mission thread* is essentially the term the military uses in place of *workflow*. A mission thread is a sequence of end-to-end activities and events that takes place to accomplish the execution of a military operation.

The threat actor is ultimately looking to violate the security attributes of mission data, with the hope of causing a range of indirect, negative consequences for mission stakeholders. Data have three basic security attributes: confidentiality, integrity, and availability.[10] For a given risk, a threat actor generally tries to produce one or more of the following outcomes:

[10]. *Confidentiality* is defined as keeping proprietary, sensitive, or personal information private and inaccessible to anyone who is not authorized to see it. *Integrity* is defined as the authenticity, accuracy, and completeness

of data. *Availability* is defined as the extent to which, or frequency with which, data must be present or ready for use. These definitions are adapted from the book *Managing Information Security Risks: The OCTAVE Approach* [Alberts 2002].

- Disclosure of data (violation of the confidentiality attribute)

- Modification of data (violation of the integrity attribute)

- Insertion of false data (violation of the integrity attribute)

- Destruction of data (violation of the availability attribute)

- Interruption of access to data (violation of the availability attribute)

- System destruction, destabilization, or degradation (violation of the availability attribute)

Each outcome maps to a security attribute of the data. As indicated in Figure 2.7, the violation of a security attribute has an impact on the workflow/mission thread and the organization's ability to achieve its mission successfully.

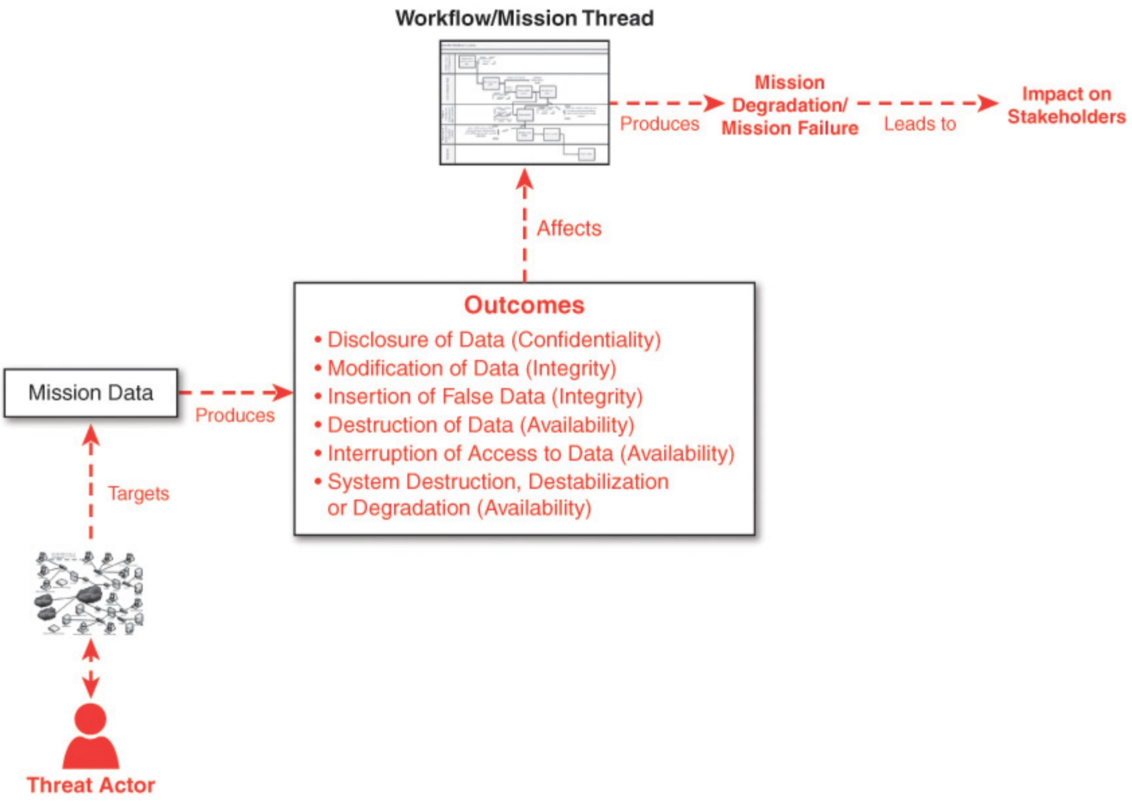

Workflow/Mission Thread
Produces
Mission Degradation/ Mission Failure
Leads to
Impact on Stakeholders
Affects
Outcomes
• Disclosure of Data (Confidentiality)
• Modification of Data (Integrity)
• Insertion of False Data (Integrity)
• Destruction of Data (Availability)
• Interruption of Access to Data (Availability)
• System Destruction, Destabilization or Degradation (Availability)
Mission Data
Produces
Targets
Threat Actor

Figure 2.7 *Security Risk Environment*

The final basic element of the security risk environment is the impact on mission stakeholders.[11] When a threat actor produces mission degradation or mission failure, the consequence can have a negative impact on various stakeholder groups.

[11] A *stakeholder* is defined as a person or group with an interest in a workflow/mission thread and the products it produces or the services it provides.

2.5 Security Risk Analysis[12]

[12] The material in this section comes from Microsoft [Microsoft 2013].

System and software security risk can be evaluated using the Security Engineering Risk Analysis (SERA) framework [Alberts 2014]. SERA differs from many other risk-identification methods that are based on brainstorming techniques. When brainstorming is used, participants describe risks based on their tacit understanding of the operational environment. For security risk-identification methods, people tend to identify threats with which they have some familiarity. They also tend to describe consequences based on their personal knowledge of organizational workflows and associated stakeholders. In lieu of brainstorming, SERA implements a detailed analysis that employs a multi-model approach for establishing operational content. The SERA evaluation is not limited to the knowledge of the active participants.

The SERA framework defines an approach for analyzing security risk in software-reliant systems and systems of systems across the software lifecycle. Traditional security-risk analysis methods are based on a simplified view of security risk, where a single threat actor exploits a single vulnerability in a single system to cause an adverse consequence. However, in reality, multiple actors exploit multiple vulnerabilities in multiple systems as part of a complex chain of events.

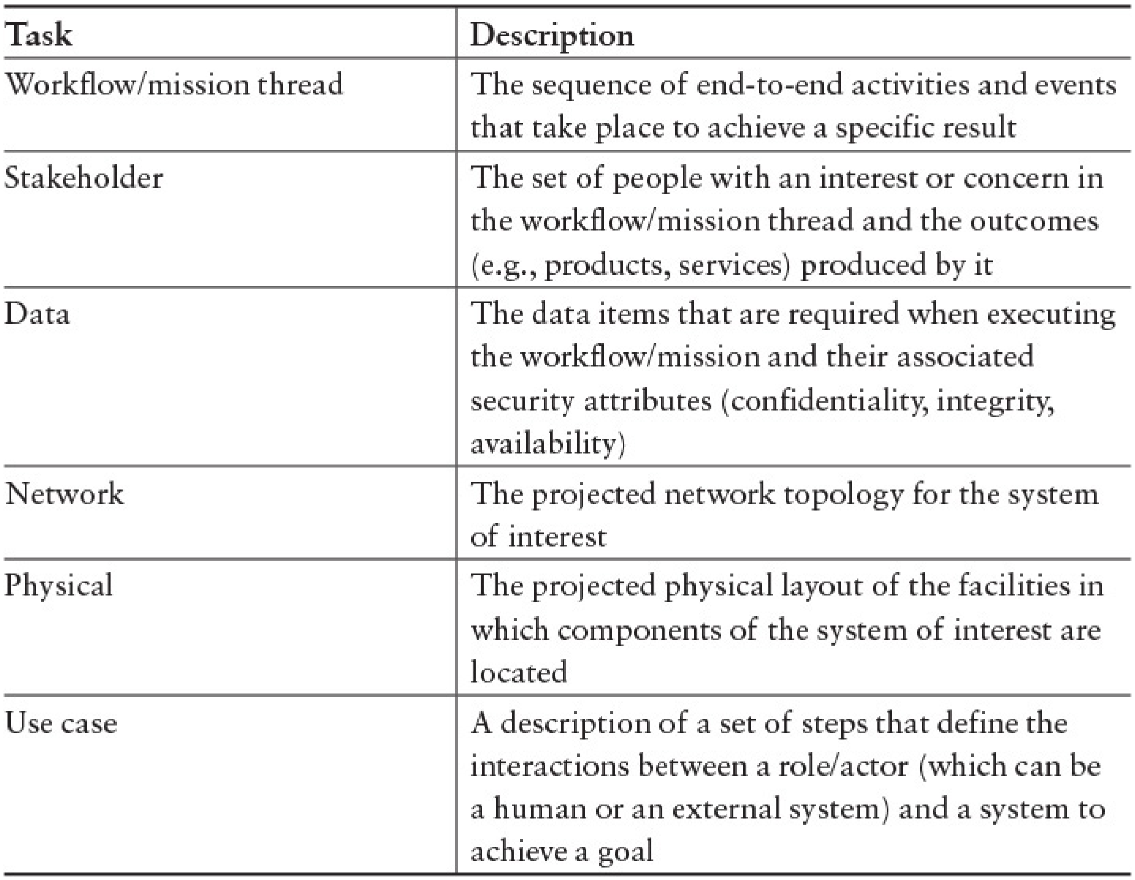
For SERA, a shared understanding of the system in its operational or production environment is assembled using multiple models that represent various aspects of the system that are important to security. If the system is still in development, the development environment is the targeted environment.

Models representing the views listed in Table 2.3 can be analyzed to establish the following key aspects of a threat:

- **Critical data**—(subset of the Data view) Important information highlighted in workflow/mission thread, use case, and network diagrams. By examining these models, analysts can identify which data elements are most critical to the workflow/mission thread and its associated mission.

- **Access path**—(connecting Workflow and Network views) How a threat actor can gain access to data and violate its security attributes (i.e., create breaches of data confidentiality, integrity, and availability). The network and physical models provide insights into potential cyber and physical access paths for an attack.

- **Threat outcome**—(identification of Workflow view failures that impact Critical data) The direct consequence caused by the threat. A direct consequence describes which security attributes of critical data have been breached. Examples of outcomes include data disclosure, data modification, insertion of false data, destruction of data, and interruption of access to data. The data model is used to identify the immediate consequence of a threat.



Task	Description
Workflow/mission thread	The sequence of end-to-end activities and events that take place to achieve a specific result
Stakeholder	The set of people with an interest or concern in the workflow/mission thread and the outcomes (e.g., products, services) produced by it
Data	The data items that are required when executing the workflow/mission and their associated security attributes (confidentiality, integrity, availability)
Network	The projected network topology for the system of interest
Physical	The projected physical layout of the facilities in which components of the system of interest are located
Use case	A description of a set of steps that define the interactions between a role/actor (which can be a human or an external system) and a system to achieve a goal

Table 2.3 *Views Used to Assemble an Operational System Model*

A threat ends with a description of its direct consequence or outcome. However, a security risk analysis must also account for indirect consequences triggered by the occurrence of a threat. The indirect consequences are used to (1) measure the impact of a security risk and (2) establish a risk's priority for decision makers. Analysts determine indirect consequences using models that represent the workflow/mission thread and stakeholder views. Mission thread analysis, unlike other techniques, allows consideration of the people and their interactions with technology in addition to the functioning of a system itself.

Using the shared operational model, plausible threat scenarios can be developed and analyzed. The SERA framework requires the following data to be recorded for each security risk:

- Security risk scenario
- Risk statement
- Threat components
- Threat sequence

- Workflow consequences

- Stakeholder consequences

- Enablers

The SERA framework comprises the following four tasks:

1. Establish the operational context

2. Identify risk

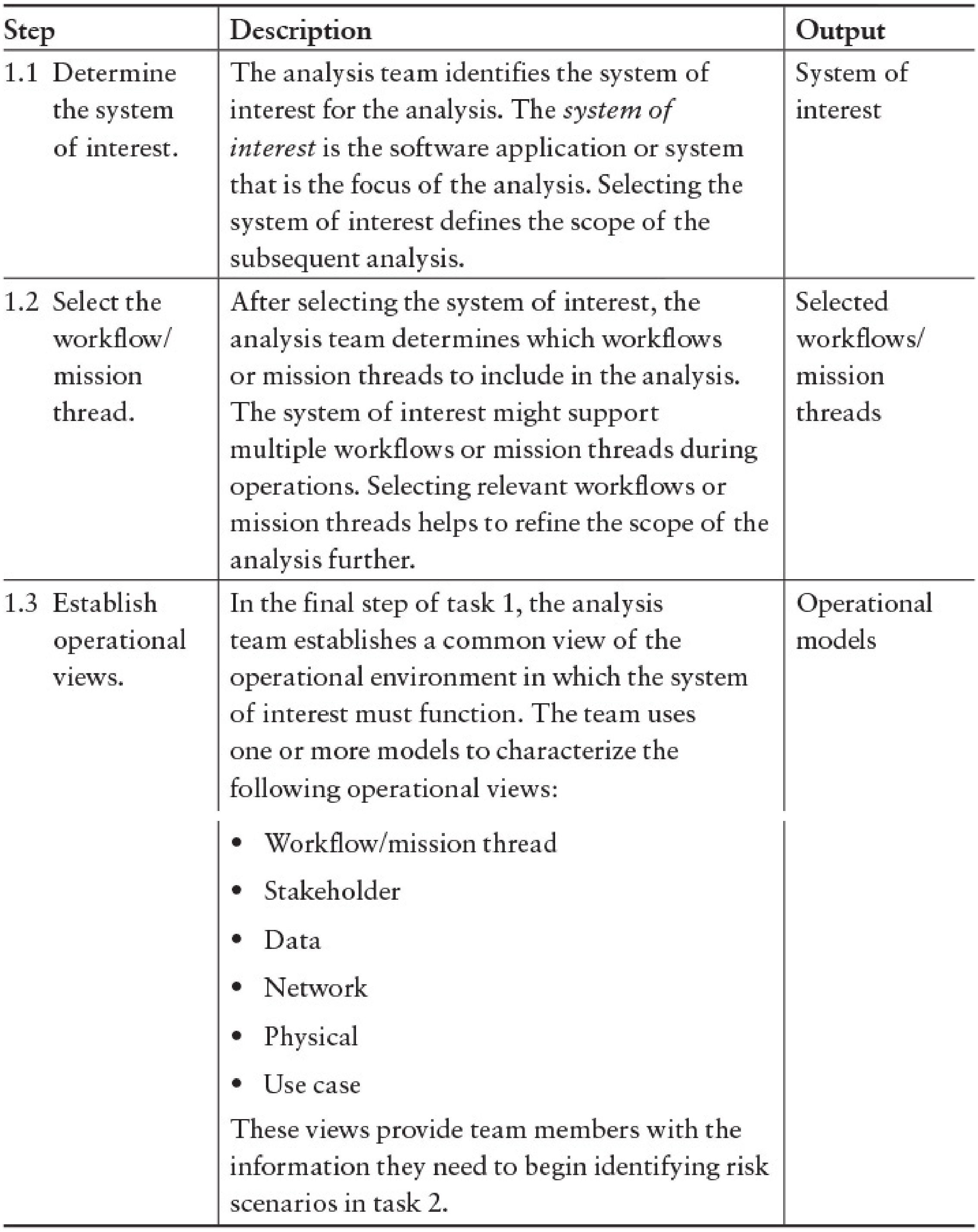
3. Analyze risk

4. Develop a control plan

The SERA framework can be self-applied by the person or group that is responsible for acquiring and developing a software-reliant system or facilitated by external parties on behalf of the responsible person or group.[13] In either case, a small analysis team of approximately three to five people is needed to implement the framework and report findings to stakeholders.

[13]. A facilitated assessment still requires participation from groups that are responsible for acquiring and developing the system of interest. The person facilitating the assessment has expertise in conducting security risk analysis. The facilitator includes others on the team with skills and experience in other areas, such as systems engineering, software engineering, operational cyber security, and physical/facility security.

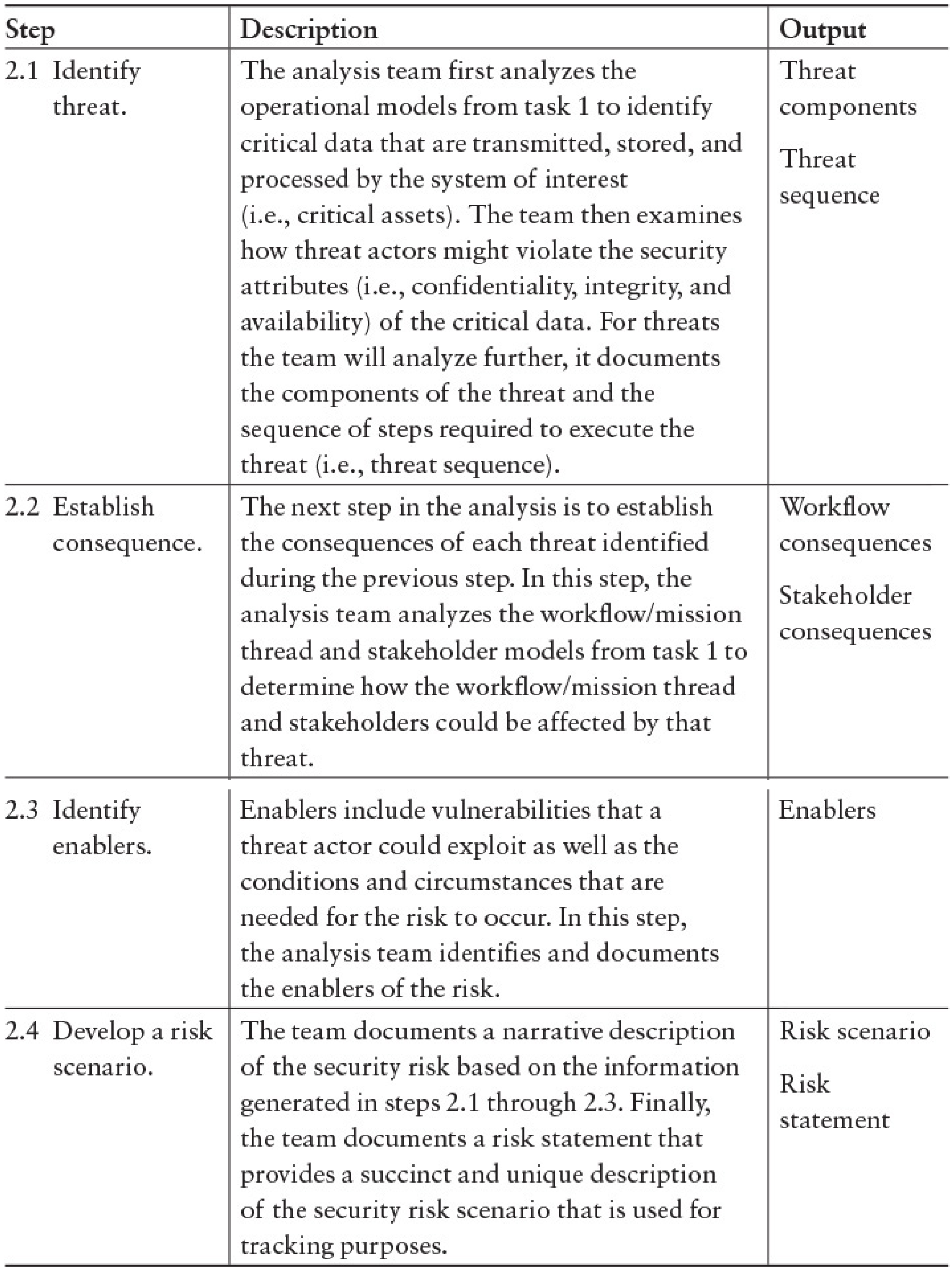
The analysis team should be an interdisciplinary team with members providing diverse skill sets. Examples of skills and experience that should be considered when forming a team include security engineering, risk analysis, systems engineering, software engineering, operational cyber security, and physical/facility security. The exact composition of an analysis team depends on the point in the lifecycle at which the SERA framework is being applied and the nature of the engineering activity being pursued. The analysis team begins its work by focusing on the environment in which a software-reliant system will be deployed. **Table 2.4** lists the steps involved in task 1.



Step	Description	Output
1.1 Determine the system of interest.	The analysis team identifies the system of interest for the analysis. The *system of interest* is the software application or system that is the focus of the analysis. Selecting the system of interest defines the scope of the subsequent analysis.	System of interest
1.2 Select the workflow/ mission thread.	After selecting the system of interest, the analysis team determines which workflows or mission threads to include in the analysis. The system of interest might support multiple workflows or mission threads during operations. Selecting relevant workflows or mission threads helps to refine the scope of the analysis further.	Selected workflows/ mission threads
1.3 Establish operational views.	In the final step of task 1, the analysis team establishes a common view of the operational environment in which the system of interest must function. The team uses one or more models to characterize the following operational views: • Workflow/mission thread • Stakeholder • Data • Network • Physical • Use case These views provide team members with the information they need to begin identifying risk scenarios in task 2.	Operational models

Table 2.4 *Task 1 (Establish the Operational Context) Steps*

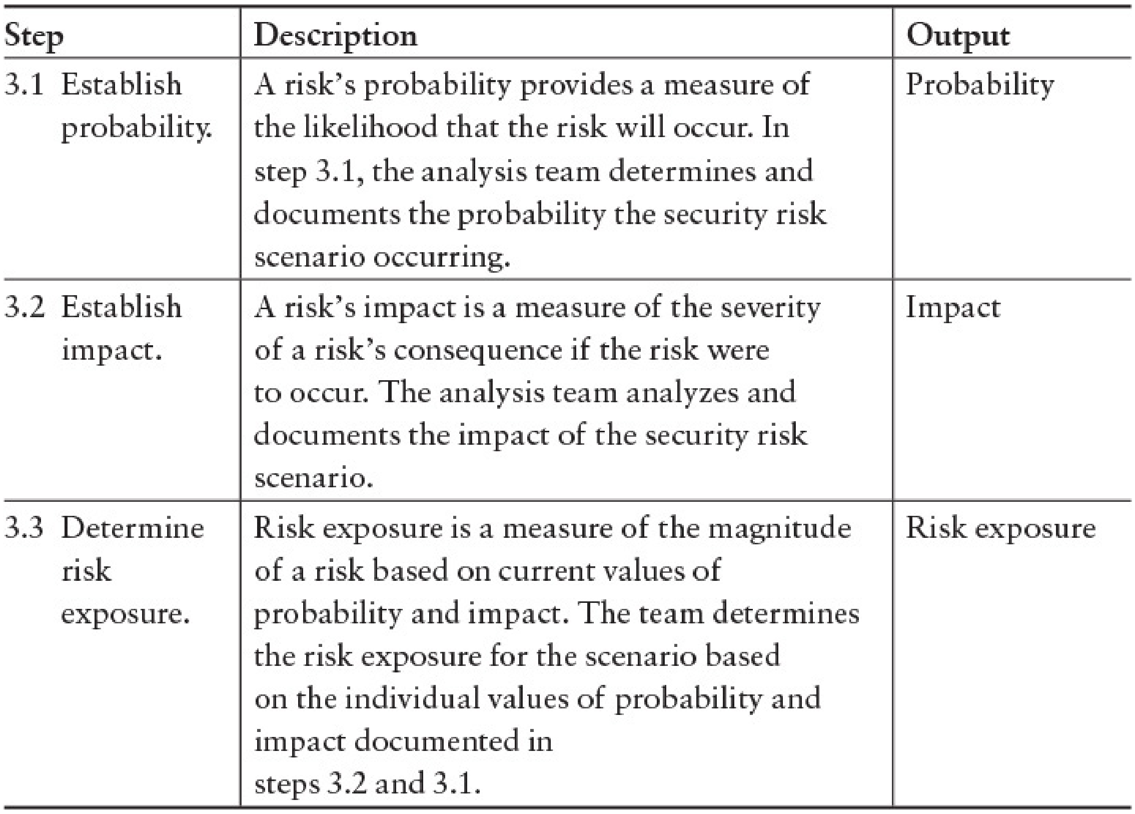
In task 2 the analysis team transforms a security concern into a distinct, tangible risk scenario that can be described and measured. **Table 2.5** lists the steps involved in task 2.



Step	Description	Output
2.1 Identify threat.	The analysis team first analyzes the operational models from task 1 to identify critical data that are transmitted, stored, and processed by the system of interest (i.e., critical assets). The team then examines how threat actors might violate the security attributes (i.e., confidentiality, integrity, and availability) of the critical data. For threats the team will analyze further, it documents the components of the threat and the sequence of steps required to execute the threat (i.e., threat sequence).	Threat components Threat sequence
2.2 Establish consequence.	The next step in the analysis is to establish the consequences of each threat identified during the previous step. In this step, the analysis team analyzes the workflow/mission thread and stakeholder models from task 1 to determine how the workflow/mission thread and stakeholders could be affected by that threat.	Workflow consequences Stakeholder consequences
2.3 Identify enablers.	Enablers include vulnerabilities that a threat actor could exploit as well as the conditions and circumstances that are needed for the risk to occur. In this step, the analysis team identifies and documents the enablers of the risk.	Enablers
2.4 Develop a risk scenario.	The team documents a narrative description of the security risk based on the information generated in steps 2.1 through 2.3. Finally, the team documents a risk statement that provides a succinct and unique description of the security risk scenario that is used for tracking purposes.	Risk scenario Risk statement

Table 2.5 *Task 2 (Identify Risk) Steps*

For task 3, the analysis team evaluates each risk in relationship to predefined criteria to determine the risk's probability, impact, and exposure. Table 2.6 lists the steps involved in task 3.



Step	Description	Output
3.1 Establish probability.	A risk's probability provides a measure of the likelihood that the risk will occur. In step 3.1, the analysis team determines and documents the probability the security risk scenario occurring.	Probability
3.2 Establish impact.	A risk's impact is a measure of the severity of a risk's consequence if the risk were to occur. The analysis team analyzes and documents the impact of the security risk scenario.	Impact
3.3 Determine risk exposure.	Risk exposure is a measure of the magnitude of a risk based on current values of probability and impact. The team determines the risk exposure for the scenario based on the individual values of probability and impact documented in steps 3.2 and 3.1.	Risk exposure

Table 2.6 *Task 3 (Analyze Risk) Steps*

In Task 4, the team establishes a plan for controlling a selected set of risks. First, the analysis team prioritizes the security risk scenarios based on their risk measures (probability and impact). Once priorities have been established, the team determines the basic approach for controlling each risk (i.e., accept or plan[14]), based on predefined criteria and current constraints (e.g., resources and funding available for control activities). For each risk that is not accepted, the analysis team develops a control plan that indicates the following:

[14] The SERA framework examines control approaches in steps 4.2 and 4.3. During step 4.2, the analysis team determines which risks will be accepted and no longer considered and which will have control plans. At this point in applying the framework, the analysis team does not identify specific strategies for transferring, avoiding, and mitigating risks. Those strategies are addressed in step 4.3. Security risk scenarios comprise multiple threat steps (as defined in the threat sequence), many enablers, and a range of indirect consequences. An analysis team might employ multiple strategies for addressing a given security risk scenario. For example, some steps in the threat sequence might be avoided through restructuring the workflow/mission thread or changing the network architecture. Certain financial consequences might be transferred to third parties by purchasing insurance. The probability of occurrence for some steps in the

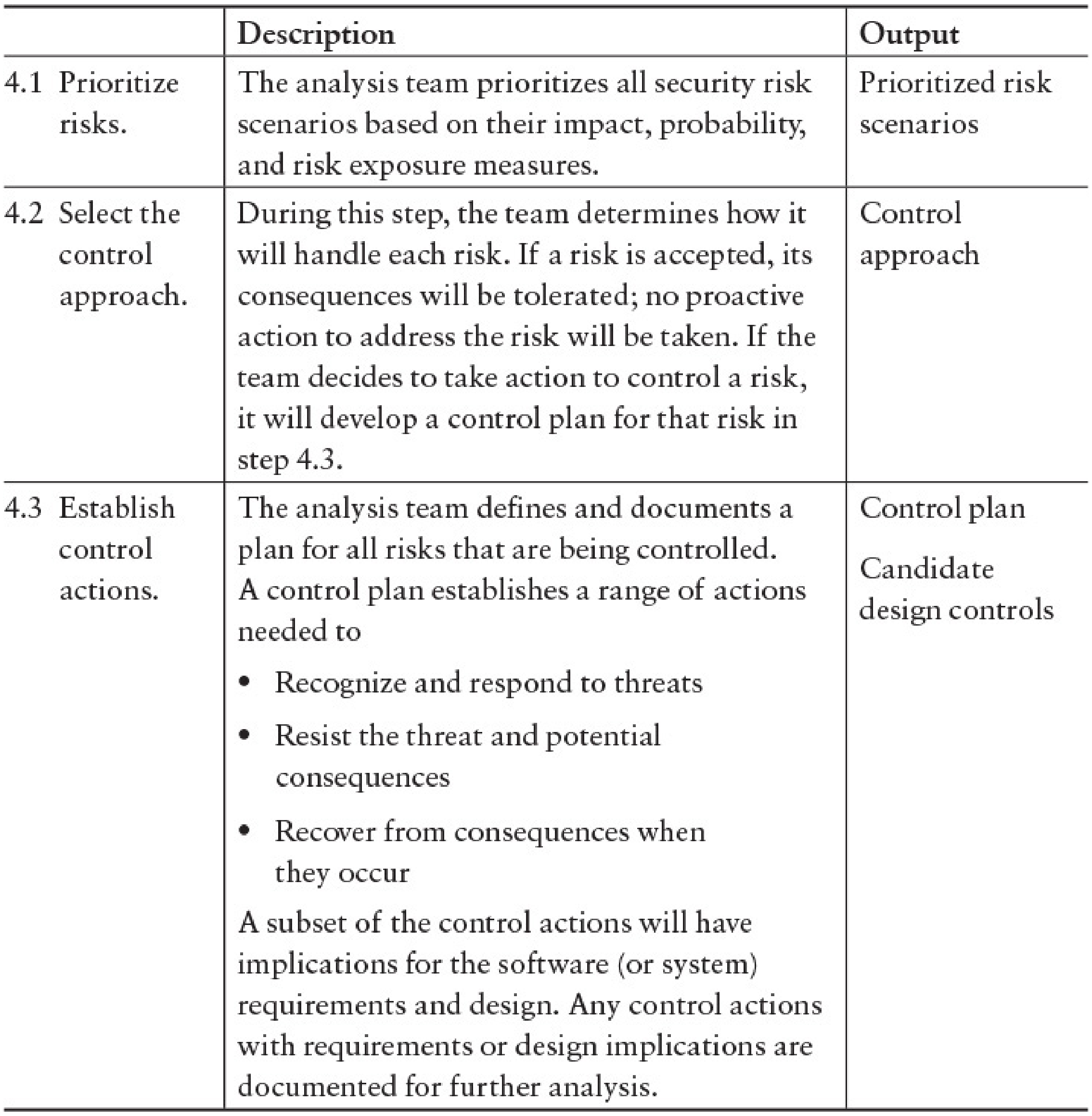
threat sequence or some types of consequences might be reduced by implementing mitigation controls. Specific control strategies (e.g., transfer, avoid, mitigate) are considered when the control plan is being developed.

• How the threat can be monitored and the actions taken when it occurs (recognize and respond)

• Which protection measures can be implemented to reduce vulnerability to the threat and minimize any consequences that might occur (resist)

• How to recover from the risk if the consequences or losses are realized (recover)

Table 2.7 lists the steps involved in task 4.



	Description	Output
4.1 Prioritize risks.	The analysis team prioritizes all security risk scenarios based on their impact, probability, and risk exposure measures.	Prioritized risk scenarios
4.2 Select the control approach.	During this step, the team determines how it will handle each risk. If a risk is accepted, its consequences will be tolerated; no proactive action to address the risk will be taken. If the team decides to take action to control a risk, it will develop a control plan for that risk in step 4.3.	Control approach
4.3 Establish control actions.	The analysis team defines and documents a plan for all risks that are being controlled. A control plan establishes a range of actions needed to • Recognize and respond to threats • Resist the threat and potential consequences • Recover from consequences when they occur A subset of the control actions will have implications for the software (or system) requirements and design. Any control actions with requirements or design implications are documented for further analysis.	Control plan Candidate design controls

Table 2.7 *Task 4 (Develop a Control Plan) Steps*

A case study illustrating the use of SERA framework for the Wireless Emergency Alert (WEA) system can be found in **Appendix A**, "**WEA Case Study: Evaluating Security Risks Using Mission Threads**."

2.6 Operational Risk Analysis—Comparing Planned to Actual

Assessments should be used to confirm that the implemented system meets the expected levels of risk that were planned in acquisition, design, and development and continues to do so over time. If effective security risk analysis is performed as the system is being developed, this knowledge can be leveraged to focus assessments on confirming that expected mitigations are in place and are appropriately addressing the risks.

Data from actual security incidents can be compared to the risks that were anticipated to identify gaps that may indicate the need to revisit the risk analysis activities to factor in the new information and determine whether changes are needed to meet the realities.

The goal of the risk assessment is to say with certainty that the currently deployed set of controls properly addresses the right threats. The assessment should also demonstrate that those controls continue to be effective, given overall business goals.

In addition to assessments, actual incidents should be collected and compared to anticipated risks to identify gaps for improvements in future system releases.

2.7 Summary

Risk management is a critical element of software assurance. Most organizations are focused only on risk to cost and schedule. The MRD can be used to analyze how organizational risks, which can include lack of capability in risk management, impact the ability of a system to meet its objectives. The SERA framework provides a view from each system of the security risks it may be contributing that can negatively affect a mission. The SERA framework is structured to assemble these risks so they can be prioritized along with other system risks.