Risk Identification

September 2013

Definition: *Risk identification is the process of determining risks that could potentially prevent the program, enterprise, or investment from achieving its objectives. It includes documenting and communicating the concern.*

Keywords: *risk, risk identification, risk management*

MITRE SE Roles & Expectations: MITRE systems engineers (SEs) working on government programs are expected to identify risks that could impact the project and program. They are expected to write and review risk statements that are clear, unambiguous, and supported by evidence [1].

Background

Risk identification is the critical first step of the risk management process depicted in Figure 1.

Figure 1. Fundamental Steps of Risk Management [2]

The objective of risk identification is the early and continuous identification of

events that, if they occur, will have negative impacts on the project's ability to achieve performance or capability outcome goals. They may come from within the project or from external sources.

There are multiple types of risk assessments, including program risk assessments, risk assessments to support an investment decision, analysis of alternatives, and assessments of operational or cost uncertainty. Risk identification needs to match the type of assessment required to support risk-informed decision making. For an acquisition program, the first step is to identify the program goals and objectives, thus fostering a common understanding across the team of what is needed for program success. This gives context and bounds the scope by which risks are identified and assessed.

Identifying Risks in the Systems Engineering Program

There are multiple sources of risk. For risk identification, the project team should review the program scope, cost estimates, schedule (to include evaluation of the critical path), technical maturity, key performance parameters, performance challenges, stakeholder expectations vs. current plan, external and internal dependencies, implementation challenges, integration, interoperability, supportability, supply-chain vulnerabilities, ability to handle threats, cost deviations, test event expectations, safety, security, and more. In addition, historical data from similar projects, stakeholder interviews, and risk lists provide valuable insight into areas for consideration of risk.

Risk identification is an iterative process. As the program progresses, more information will be gained about the program (e.g., specific design), and the risk statement will be adjusted to reflect the current understanding. New risks will be identified as the project progresses through the life cycle.

Best Practices and Lessons Learned

Operational Risk. Understand the operational nature of the capabilities you are supporting and the risk to the end users, their missions, and their operations of the capabilities. Understanding of the operational need/mission (see the Concept Development topic of the Systems Engineering Guide) will help you appreciate the gravity of risks and the impact they could have to the end users. This is a critical part of risk analysisrealizing the real-world impact that can occur if a risk arises during operational use. Typically operational users are willing to accept some level of risk if they are able to accomplish their mission (e.g., mission assurance), but you need to help users to understand the risks they are accepting and to assess the options, balances,

and alternatives available.

Technical maturity. Work with and leverage industry and academia to understand the technologies being considered for an effort and the likely transition of the technology over time. One approach is to work with vendors under a non-disclosure agreement to understand the capabilities and where they are going, so that the risk can be assessed.

Non-Developmental Items (NDI). NDI includes commercial-off-the-shelf and government-off-the-shelf items. To manage risk, consider the viability of the NDI provider. Does the provider have market share? Does the provider have appropriate longevity compared to its competitors? How does the provider address capability problems and release fixes, etc.? What is the user base for the particular NDI? Can the provider demonstrate the NDI, preferably in a setting similar to that of your customer? Can the government use the NDI to create a prototype? All of these factors will help assess the risk of the viability of the NDI and the provider. Seek answers to these questions from other MITRE staff that have worked the area or have used the NDI being assessed.

Acquisition drivers. Emphasize critical capability enablers, particularly those that have limited alternatives. Evaluate and determine the primary drivers to an acquisition and emphasize their associated risk in formulating risk mitigation recommendations. If a particular aspect of a capability is not critical to its success, its risk should be assessed, but it need not be the primary focus of risk management. For example, if there is risk to a proposed user interface, but the marketplace has numerous alternatives, the success of the proposed approach is probably less critical to overall success of the capability. On the other hand, an information security feature is likely to be critical. If only one alternative approach satisfies the need, emphasis should be placed on it. Determine the primary success drivers by evaluating needs and designs, and determining the alternatives that exist. Is a unique solution on the critical path to success? Make sure critical path analyses include the entire system engineering cycle and not just development (i.e., system development, per se, may be a "piece of cake," but fielding in an active operational situation may be a major risk).

Use capability evolution to manage risk. If particular requirements are driving implementation of capabilities that are high risk due to unique development, edge-of-the-envelope performance needs, etc., the requirements should be discussed with the users for their criticality. It may be that the need could be postponed, and the development community should assess when it might be satisfied in the future. Help users and developers gauge how much risk (and schedule and cost impact) a particular capability should assume against the requirements to receive less risky capabilities sooner. In developing your recommendations, consider technical feasibility and knowledge of related

implementation successes and failures to assess the risk of implementing now instead of the future. In deferring capabilities, take care not to fall into the trap of postponing ultimate failure by trading near-term easy successes for a future of multiple high-risk requirements that may be essential to overall success.

Key Performance Parameters (KPPs). Work closely with the users to establish KPPs. Overall risk of program cancelation can be centered on failure to meet KPPs. Work with the users to ensure the parameters are responsive to mission needs and technically feasible. The parameters should not be so lenient that they can easily be met, but not meet the mission need; nor should they be so stringent that they cannot be met without an extensive effort or pushing technologyeither of which can put a program at risk. Seek results of past operations, experiments, performance assessments, and industry implementations to help determine performance feasibility.

External and internal dependencies. Having an enterprise perspective can help acquirers, program managers, developers, integrators, and users appreciate risk from dependencies of a development effort. With the emergence of service-oriented approaches, a program will become more dependent on the availability and operation of services provided by others that they intend to use in their program's development efforts. Work with the developers of services to ensure quality services are being created, and thought has been put into the maintenance and evolution of those services. Work with the development program staff to assess the services that are available, their quality, and the risk that a program has in using and relying upon the service. Likewise, there is a risk associated with creating the service and not using services from another enterprise effort. Help determine the risks and potential benefits of creating a service internal to the development with possibly a transition to the enterprise service at some future time.

Integration and Interoperability (I&I). I&I will almost always be a major risk factor. They are forms of dependencies in which the value of integrating or interoperating has been judged to override their inherent risks. Techniques such as enterprise federation architecting, composable capabilities on demand, and design patterns can help the government plan and execute a route to navigate I&I risks. Refer to the Enterprise Engineering section of the Systems Engineering Guide for articles on techniques for addressing I&I associated risks.

Information security. Information security is a risk in nearly every development. Some of this is due to the uniqueness of government needs and requirements in this area. Some of this is because of the inherent difficulties in countering cyber attacks. Creating defensive capabilities to cover the spectrum of attacks is challenging and risky. Help the government develop resiliency approaches (e.g., contingency plans, backup/recovery, etc.).

Enabling information sharing across systems in coalition operations with international partners presents technical challenges and policy issues that translate into development risk. The information security issues associated with supply chain management is so broad and complex that even maintaining rudimentary awareness of the threats is a tremendous challenge.

Skill level. The skill or experience level of the developers, integrators, government, and other stakeholders can lead to risks. Be on the lookout for insufficient skills and reach across the corporation to fill any gaps. In doing so, help educate government team members at the same time you are bringing corporate skills and experience to bear.

Cost risks. Programs will typically create a government's cost estimate that considers risk. As you develop and refine the program's technical and other risks, the associated cost estimates should evolve, as well. Cost estimation is not a one-time activity.

Historical information as a guide to risk identification. Historical information from similar government programs can provide valuable insight into future risks. Seek out information about operational challenges and risks in various operation lessons learned, after action reports, exercise summaries, and experimentation results. Customers often have repositories of these to access. Government leaders normally will communicate their strategic needs and challenges. Understand and factor these into your assessment of the most important capabilities needed by your customer and as a basis for risk assessments.

Historical data to help assess risk is frequently available from the past performance assessments and lessons learned of acquisition programs and contractors. In many cases, MITRE staff will assist the government in preparing performance information for a particular acquisition. The AF has a Past Performance Evaluation Guide (PPEG) that identifies the type of information to capture that can be used for future government source selections [3]. This repository of information can help provide background information of previous challenges and where they might arise againboth for the particular type of development activity as well as with the particular contractors.

There are numerous technical assessments for vendor products that can be accessed to determine the risk and viability of various products. One MITRE repository of evaluations of tools is the Analysis Toolshed that contains guidance on and experience with analytical tools. Using resources like these and seeking others who have tried products and techniques in prototypes and experiments can help assess the risks for a particular effort.

How to write a risk—a best practice [2]. A best-practice protocol for writing a

risk statement is the *Condition-If-Then* construct. This protocol applies to risk management processes designed for almost any environment. It is a recognition that a risk, by its nature is probabilistic and one that, if it occurs, has unwanted consequences.

What is the *Condition-If-Then* construct? The *Condition* reflects what is known today. It is the root cause of the identified risk event. Thus, the *Condition* is an event that has occurred, is presently occurring, or will occur with certainty. Risk events are future events that may occur because of the *Condition* present. Below is an illustration of this protocol.

The *If* is the risk event associated with the *Condition* present. It is critically important to recognize the *If* and the *Condition* as a dual. When examined jointly, there may be ways to directly intervene or remedy the risk event's underlying root (*Condition*) such that the consequences from this event, if it occurs, no longer threaten the project. The *If* is the probabilistic portion of the risk statement.

The *Then* is the consequence, or set of consequences, that will impact the engineering system project if the risk event occurs. An example of a *Condition-If-Then* construct is illustrated in Figure 2.

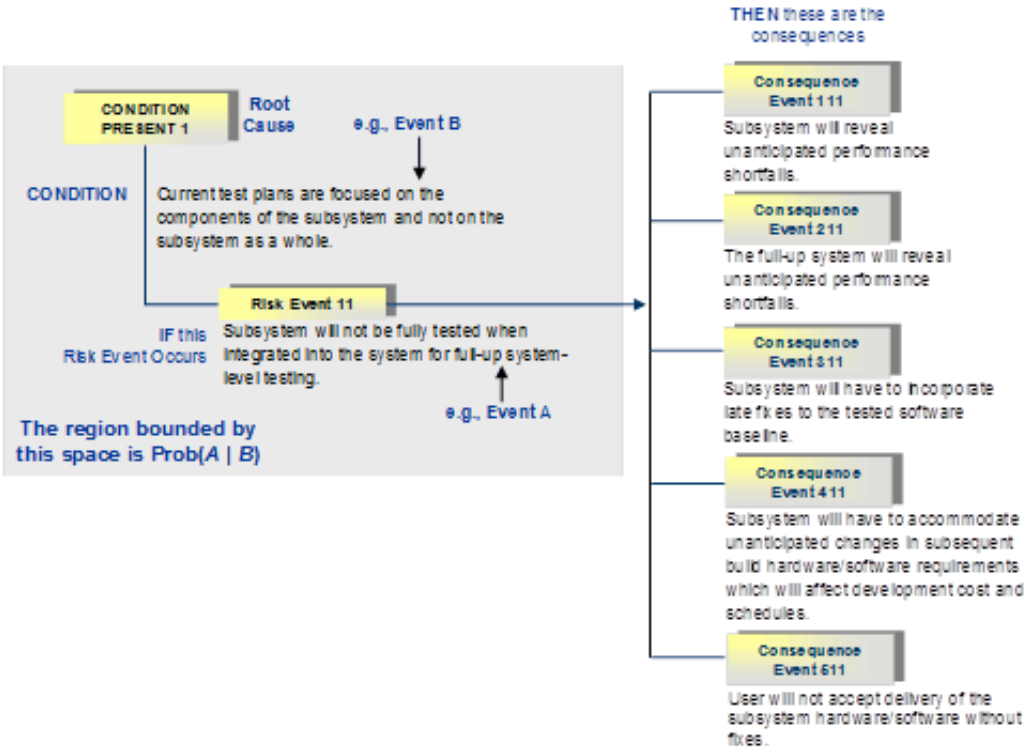

Figure 2. Writing a Risk—The "Condition-If-Then" Best Practice

Encourage teams to identify risks. The culture in some government projects and programs discourages the identification of risks. This may arise because the risk management activities of tracking, monitoring, and mitigating the risks are seen as burdensome and unhelpful. In this situation, it can be useful

to talk to the teams about the benefits of identifying risks and the inability to manage it all in your heads (e.g., determine priority, who needs to be involved, mitigation actions). Assist the government teams in executing a process that balances management investment with value to the outcomes of the project. In general, a good balance is being achieved when the project scope, schedule, and cost targets are being met or successfully mitigated by action plans, and the project team believes risk management activities provide value to the project. Cross-team representation is a *must*; risks should not be identified by an individual, or strictly by the systems engineering team (review sources of risk above).

Consider organizational and environmental factors. Organizational, cultural, political, and other environmental factors, such as stakeholder support or organizational priorities, can pose as much or more risk to a project than technical factors alone. These risks should be identified and actively mitigated throughout the life of the project. Mitigation activities could include monitoring legislative mandates or emergency changes that might affect the program or project mission, organizational changes that could affect user requirements or capability usefulness, or changes in political support that could affect funding. In each case, consider the risk to the program and identify action options for discussion with stakeholders. For additional information, see the Risk Mitigation Planning, Implementation, and Progress Monitoring article.

Include stakeholders in risk identification. Projects and programs usually have multiple stakeholders that bring various dimensions of risk to the outcomes. They include operators, who might be overwhelmed with new systems; users, who might not be properly trained or have fears for their jobs; supervisors who might not support a new capability because it appears to diminish their authority; and policy makers, who are concerned with legislative approval and cost. In addition, it is important to include all stakeholders, such as certification and accreditation authorities who, if inadvertently overlooked, can pose major risks later in the program. Stakeholders may be keenly aware of various environmental factors, such as pending legislation or political program support that can pose risks to a project that are unknown to the government or MITRE project team. Include stakeholders in the risk identification process to help surface these risks.

Write clear risk statements. Using the *Condition-If-Then* format helps communicate and evaluate a risk statement and develop a mitigation strategy. The root cause is the underlying *Condition* that has introduced the risk (e.g., a design approach might be the cause), the *If* reflects the probability (e.g., probability assessment that the *If* portion of the risk statement were to occur), and the *Then* communicates the impact to the program (e.g., increased resources to support testing, additional schedule, and concern to meet

performance). The mitigation strategy is almost always better when based on a clearly articulated risk statement.

Expect risk statement modifications as the risk assessment and mitigation strategy is developed. It is common to have risk statements refined once the team evaluates the impact. When assessing and documenting the potential risk impact (cost, schedule, technical, or timeframe), the understanding and statement of the risk might change. For example, when assessing a risk impact of software schedule slip, the risk statement might be refined to include the need-by date, and/or further clarification of impact (e.g., if the xyz software is not delivered by March 2015, then there will not be sufficient time to test the interface exchanges prior to Limited User Test).

Do not include the mitigation statement in the risk statement. Be careful not to fall into the trap of having the mitigation statement introduced into the risk statement. A risk is an uncertainty with potential negative impact. Some jump quickly to the conclusion of mitigation of the risk and, instead of identifying the risk that should be mitigated (with mitigation options identified), they identify the risk as a sub-optimal design approach. For example, a risk statement might be: If the contractor does not use XYZ for test, then the test will fail. The concern is really test sufficiency. If the contractor does not conduct the test with measurable results for analysis, then the program may not pass limited user test. Use of XYZ may be a mitigation option to reduce the test sufficiency risk.

Do not jump to a mitigation strategy before assessing the risk probability and impact. A risk may be refined or changed given further analysis, which might affect what the most efficient/desired mitigation may be. Engineers often jump to the solution; it is best to move to the next step discussed in the Risk Impact Assessment and Prioritization article to decompose and understand the problem first. Ultimately this will lead to a strategy that is closely aligned with the concern.

References & Resources

1. The MITRE Institute, September 1, 2007, *MITRE Systems Engineering (SE) Competency Model*, Version 1, pp. 10, 40-41.
2. Garvey, P.R., 2008, *Analytical Methods for Risk Management: A Systems Engineering Perspective*, Chapman-Hall/CRC-Press, Taylor & Francis Group (UK), Boca Raton, London, New York, ISBN: 1584886374.U.S. Air Force, January 2008, *Air Force Past Performance Evaluation Guide (PPEG),* IG5315.305(a).
3. U.S. Air Force, January 2008, *Air Force Past Performance Evaluation Guide (PPEG)*, IG5315.305(a).

Additional References & Resources

MITRE E520 Risk Analysis and Management Technical Team checklists, Risk Checks, Risk Analysis and Management Documents.

Project Management Institute, *A Guide to the Project Management Body of Knowledge, (PMBOK Guide),* Fourth Edition, ANSI/PMI 99-001-2008, pp. 273-312, accessed March 2, 2010.

SEPO, "Standard Process/Steps of Process, Step 2: Identify Risks & Hazards," MITRE SEPO *Risk Management Toolkit*, accessed May 5, 2010.