

A large, light gray, curved graphic element that starts from the top left and curves towards the bottom right, framing the central text.

The Fear Factor!

Risk Management



Authors:

Ravi Sinha, Quality Process Analyst in Emirates Group IT with more than 10 years of extensive experience in Quality Assurance, Quality System, Process Improvement & Man Management.

Srividhya Rangarajan, Quality Process Analyst in Emirates Group IT with more than 10 years of extensive experience in Quality Assurance, Quality System, Process Improvement & Man Management.



Abstract

There is a popular television program that appeals to the brave of heart called “Fear Factor.” During this program, contestants compete against each other in events and activities that cause considerable fear for most people. Although it is considered reality TV, the things the contestants are required to eat, drink, touch, dive into, or navigate through are not considered reality by the average person. Contestants are eliminated with each fear-defying competition, until a single winner remains. In the *reality world* of software development and/or acquisition, there are circumstances, events, activities, etc. that instill fear in the hearts of software developers, acquirers, and managers. Too often these fear factors are ignored and have devastating impacts on software projects. From our experiences, we have identified several issues that are common to many software- projects. These issues, *risk factors*, are keys to either the success or failure of any software- projects..



What Is Software Risk Management?

Risk is defined as, “A possible future event that, if it occurs, will lead to an undesirable outcome” [1]. In the context of software,, risk management would logically imply the managing of possible future events that could have undesirable effects on software projects. That sounds simple enough. What future events could possibly negatively impact my software project? Risk management is simply defined as a generalized process for managing risks [2]. However, to be an effective risk management process, it has to be both accurate and usable; that is, it must provide results to a manager in an adequate timeframe to enable the manager to make informed decisions that allow risks to be mitigated or avoided.

Every development project has risks. The risks can range from the common, “We might not be able to find a JAVA programming language expert by next month,” to the uncommon, “A hurricane might destroy the primary data centre.” The way that you approach these risks is what is important.

Risk Management Problem

One practical problem that we face in any organization is the acknowledgement and acceptance of risk. Rather than meeting them head on, and despite increased efforts to promote sound risk management practises; it is common practice to tend to be in denial about whether risks exist on our projects and, if they do, how they should be addressed. Figure 1, adapted from the Software Program Manager’s Network (SPMN) “The Little Book of Bad Excuses” [SPMN, 1998] highlights some of the most common traps that all too many projects have fallen into over the years, and continue to fall into today. As a result, symptoms that organizations are not effectively practicing risk management continue to flourish, including a continual state of project instability, constant fire-fighting, multiple schedule slippages because of recurring surprise factors, and constantly operating in a high-stress, crisis management environment.

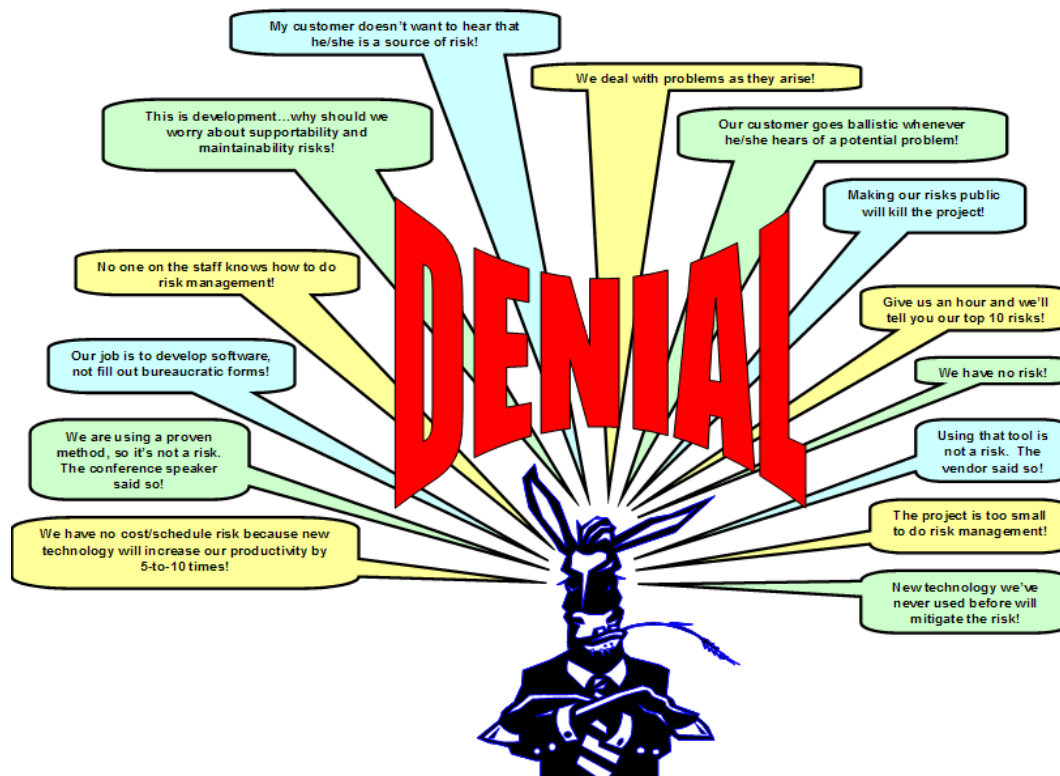




Figure 1

The Stubborn Face of Risk Denial (adapted from The Little Book of Bad Excuses) [SPMN, 1998]

Effective risk management requires establishing and following a rigorous process. It involves the entire project team, as well as requiring help from outside experts in critical risk areas (e.g., technology, manufacturing, logistics, etc.). Risks will be found in all areas of the project and will often be interrelated, therefore risk management should include hardware, software, integration issues, and of course the human element.



A Framework for Formal Software Risk Management

The conceptual view of the Risk Management Model is presented in Figure 2. The map helps to first lay the foundation by building the capability needed for the next stage, second ease the transition between phases by using an incremental strategy, and third persevere by basing risk management activities on a detailed plan that evolves over time.

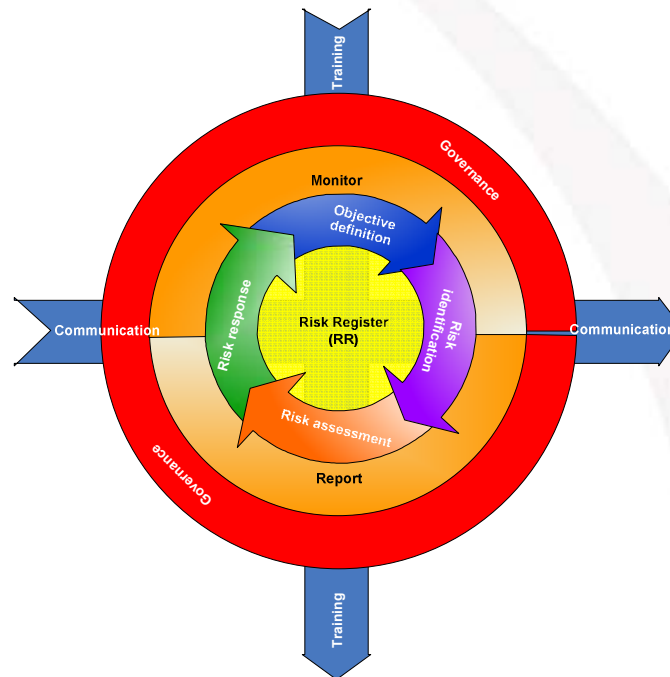


Figure2
Risk Management Model

While the risk management approach should be tailored to specific project requirements, the process is iterative and should have all the components shown in Figure 2. Note that while planning appears as the first step, there is a feedback loop from the monitoring activity that allows planning and the other activities to be redone or controlled by actual results, providing continual updates to the risk management strategy. In essence, the process is a standard approach to problem solving:

1. Plan or define the problem-solving process.
2. Define the problem.
3. Work out solutions for those problems.
4. Track the progress and success of the solutions

The following sections expand upon the approach to risk management..



RISK IDENTIFICATION

One of the more formal tools to help in risk identification is the SEI Taxonomy-Based Risk Identification report [Carr, 1993]. The report defines three major areas that need to be considered for risk management in a software project. These areas are:

- ❖ Product Engineering
 - Subdivided into Requirements; Design; Code and Unit Test; Integration and Test; and Engineering Specialties (e.g., reliability, maintainability, etc.)
- ❖ Development Environment
 - Subdivided into Development Process Development System; Management Process; Management Methods; and Work Environment
- ❖ Program Constraints
 - Subdivided into Resources; Contract; and Program Interfaces

Appendix B of the SEI report includes a Taxonomy-Based Questionnaire which can be put into checklist format to ensure that the appropriate elements of software risk are covered on a specific project.

Hall [Hall, 1997] and others view software risk as being comprised of two key areas (management and technical), and categorized into project, process and product (see Figure 3). Software process risk includes both management and technical work procedures. In management procedures, process risk may be found in activities such as planning, staffing, tracking, quality assurance and configuration management. In technical processes, it may be found in activities such as requirements analysis, design, code and test. Planning is the management process risk that is most often reported. The most reported technical process risk is the software development process itself. [Hall, 1997]

Software product risk contains intermediate and final work product characteristics. Primarily a technical responsibility, product risk may be found in requirements stability, design performance, code complexity and test specifications. Product risk is difficult to manage because software requirements are often perceived as flexible. Requirements are the most significant product risks reported in risk assessments. [Hall, 1997]



Figure 3

Software Risk Classifications [Hall, 1997]

Beyond the technical and management sources of risk, there may also be sources external to the project or organization (e.g., the nature of the marketplace, the business culture, etc.). The base of the pyramid in Figure 4 highlights what some of these external risk sources might be.

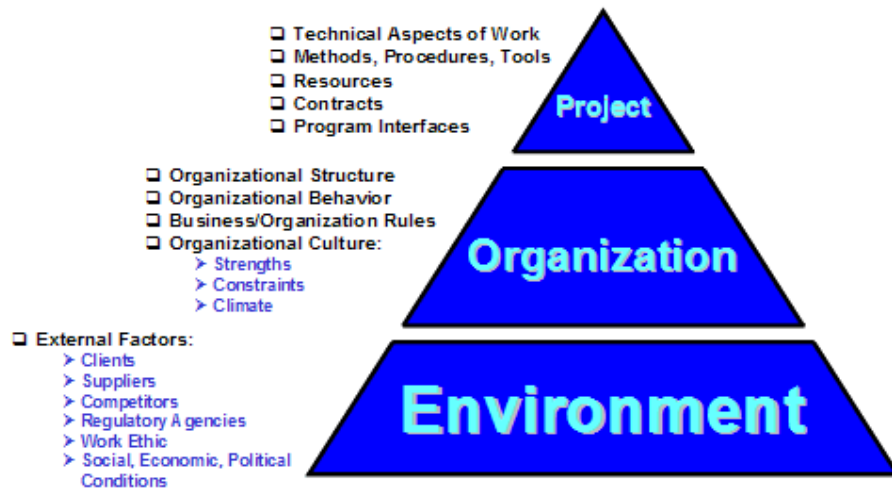


Figure 4
Sources of Project Risk

RISK ANALYSIS

After risks are identified, they should be partitioned into categories such as technical, cost, schedule, management, etc. Note that some risks may fall into multiple categories.

Why do risks need to be partitioned? First of all, some risks are more important than others. Also, different stakeholders may be concerned about different risks, or different personnel may bear responsibility for tracking/monitoring different risks. Finally, different risk types may require different mitigation strategies.

The initial activity in risk analysis is to identify contributing factors, then establish a hierarchy of those contributing factors. Figure 5 illustrates a hierarchy of how a project might fail, given the contributing factors of staffing, funding, performance failures, and so on. The staffing factor is further broken down to show, first, how staffing may become a contributing factor to project failure and second, what the contributing factors might be that result in insufficient staffing (subsequently leading to project failure). All contributing factors defined within the hierarchy would be broken down to a correspondingly meaningful level of detail.

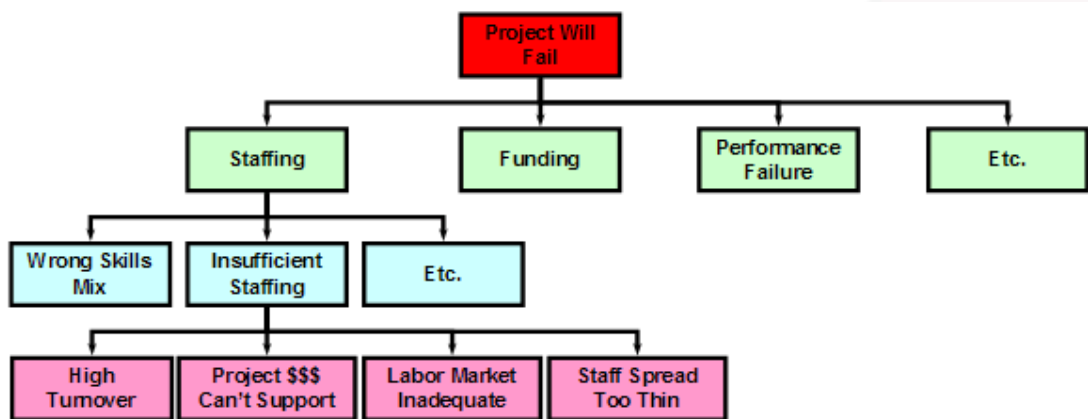


Figure 5

A Sample Hierarchy of Contributing Factors



Similarly, for positive risk, or to put this in another way, opportunity, a hierarchy of contributing factors could also be created, this time highlighting those elements for which risk is being undertaken in order to leverage a perceived opportunity for the project, such as “Schedule completion will be early”.

There are any number of ways that risk can be partitioned, analyzed and quantified. The approach taken and method(s) used should always be tailored to meet the needs of the business, the customer and the project.

Once you have successfully partitioned and quantified project risk, prioritization of risk becomes the next logical activity.

RISK PRIORITIZATION

Risk prioritization is a critical characteristic of the formal risk management process, as it provides the opportunity to apply what are typically limited project resources to those risks having the largest potential impact on the project. For many risk prioritization approaches, risks are ranked and prioritized based on some combination of probability (i.e., how likely is it that the risk will occur) and impact (i.e., what are the consequences if the risk does occur). This can be done qualitatively in a risk prioritization matrix or quantitatively using some type of composite probability-impact score. This basic approach has been illustrated in Figure 6, RISK HEAT MAP

RISK HEAT MAP

L I K E L I H O O D	Almost Certain 5	Medium 5	Medium 1	Significant 1	Significant 2	High 2
	Likely 4	Medium 4	Medium 8	Medium 1	Significant 1	Significant 2
	Possible 3	Moderate 3	Medium 6	Medium 9	Medium 1	Significant 1
	Remote 2	Moderate 2	Moderate 4	Medium 6	Medium 8	Medium 1
	Unlikely 1	Low 1	Moderate 2	Moderate 3	Medium 4	Medium 5
		Negligible 1	Minor 2	Average 3	Major 4	Extreme 5
		IMPACT				



Likelihood Description	Estimated Probability	Impact Description	Financial Loss (AED)	Schedule Delay
Almost Certain(5)	> 90% probability	Extreme (5)	> 5M	Tasks delayed by upto 20% beyond the agreed project plan.
Likely (4)	65 – 90% probability	Major (4)	1M – 5M	Tasks delayed by upto 15% beyond the agreed project plan.
Possible (3)	35 – 64% probability	Average (3)	100K – 1M	Tasks delayed by upto 10% beyond the agreed project plan.
Remote (2)	10 - 34% probability	Minor (2)	50K -100 K	Tasks delayed by upto 5% beyond the agreed project plan.
Unlikely (1)	< 10% probability	Negligible (1)	< 50k	Tasks delayed beyond the committed date, but still within agreed overall project plan.

Figure 6
RISK HEAT MAP for Risk Prioritization

Note that the qualitative matrix has some basis in quantitative values for frequency and impact, although these can be subjectively defined and tailored to suit specific sets of circumstances. The highest priority risks would be those falling in the red (intolerable) region.

RISK MANAGEMENT PLANNING

Risk management planning provides the basis for the identification of the monitoring procedures that should be put in place for each risk, including how to tell if a risk is going to manifest as a real problem, and how frequently each identified risk should be monitored. Risk planning also takes into account risk aversion planning (i.e., what actions will be taken to mitigate risk before it occurs) and contingency planning (i.e., how to react if a risk actually manifests).

The purpose of a Risk Management Plan is to define how risk management activities are implemented and supported during a project. It is a key output of the planning process, and serves as the mechanism for implementing software risk management.

A Risk Action Request can be created to serve as a mechanism by which risk information can be captured and communicated to the stakeholders. An effective risk management process requires the creation of risk action requests for any risks that exceed their quantified risk threshold value.

Finally, a Risk Treatment Plan can be used to define how risks that are found to be unacceptable are to be treated. It serves as the mechanism for implementing a selected recommended alternative defined within a risk action request. Note that “treatment” and “mitigation” are considered synonymous.



RISK RESOLUTION

The “Software Acquisition Risk Management Guidebook” [Gallagher, 1997] contains a software risk planning decision flowchart (Figure 10) that, among other things, highlights the major options to be considered when developing response plans for threats, namely “Keep”, “Delegate” and “Transfer” if personal risk responsibility is appropriate; and “Research”, “Accept”, “Mitigate” and “Watch” (or “Monitor”) if further involvement in risk resolution is warranted.

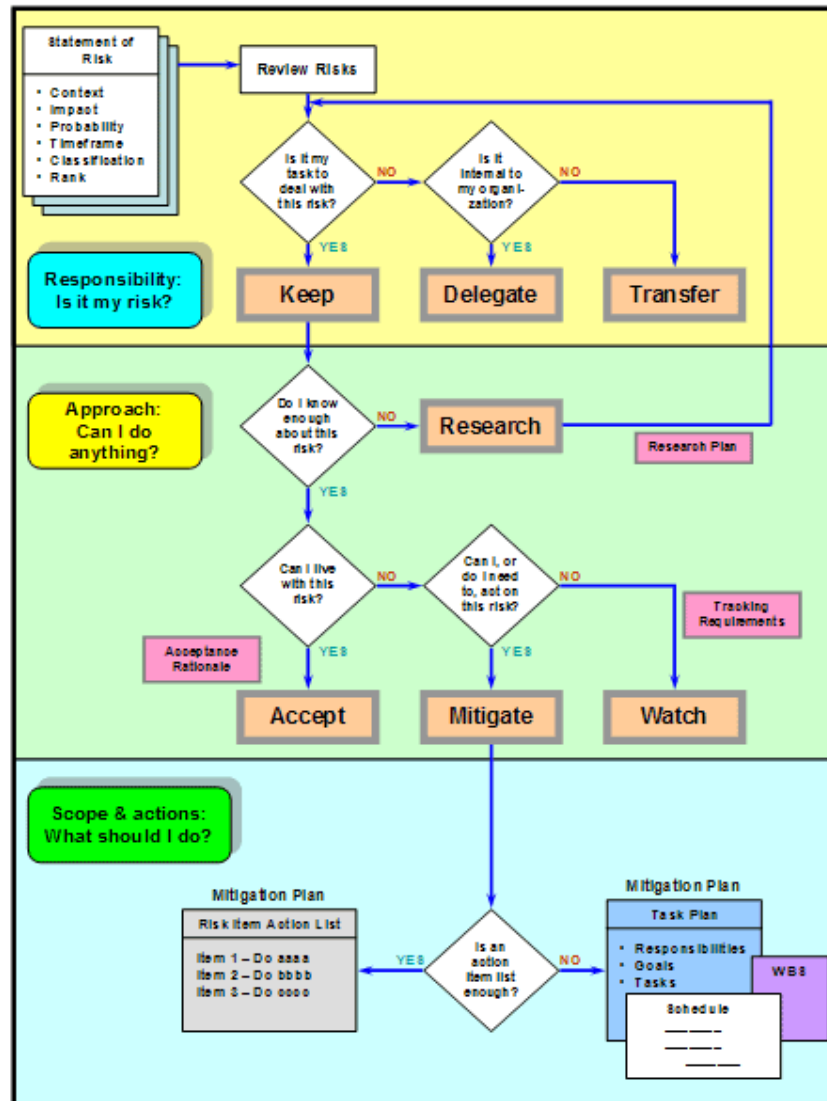


Figure 7

Planning Risk Decision Flowchart [Gallagher, 1997]

There are a number of key questions that can help facilitate the effectiveness of risk allocation and risk transfer approaches.



RISK MONITORING

There are various measurement and metrics tools that can be used to monitor and report status of project risks. Table 1 presents some of the basic measurements and metrics that can be considered to meet these needs.

Table 1
Important Risk Measurements/Metrics

Risk Measurements/Metrics	
·	Number of Identified Risks
·	Number of Active Risks
·	Number of Risks Assessed “High”, “Medium” and “Low”
·	Total “Probability X Impact” (PI) Score for All Risks
·	Average PI Score
·	Expected Value (= Probability x Cost Impact)
·	Percent Likelihood of Meeting Target Schedule and Budget
·	Overall Project Criticality Index
·	Overall Project Cruciality Index
·	Relative Risk Exposure Index
·	Risk Reduction Leverage (= Reduction in Cost Impact divided by Cost of Response)

A useful metric for assessing project risk is the percent likelihood of meeting the target schedule and/or budget. Any significant negative deviation from this metric would not bode well for the success of the project.

There are several indexes that can be used to measure risk, including project criticality, project cruciality, and risk exposure. The Criticality Index identifies which resources (or resource types) are most constrained and how those constraints may impact the project schedule. Activities with a high criticality index are likely to determine the overall project duration. The Cruciality Index, which is similar to the Criticality Index, factors in how critical the resources are to meeting the project schedule constraints. Activities with a high cruciality index have a direct bearing on the variability of the overall schedule duration. The relative Risk Exposure Index, which we discussed earlier, is the probability of unexpected loss (or risk) and the size of the loss. Finally, the Risk Reduction Leverage refers to the corresponding reduction in cost impact or cost of response as a result of reducing the risk level in other areas, such as performance, reliability, schedule, etc. Its value is determined using the equation below:

$$\text{Risk Reduction Leverage} = (\text{Risk Exposure Before} - \text{Risk Exposure After}) \text{ divided by Risk Reduction Cost}$$

There are also a number of risk management audit measurements / metrics that are appropriate to monitor and track progress on controlling project risk and relate well to SEI CMM concepts. These include factors such as (1) the number and ratio of scheduled and actual risk management audits, (2) the effort required to support risk management audits, (3) the total number of risk-related problem reports (and the number of open and closed reports) that are measured in each risk management audit, and (4) the number and ratio of actual and deferred corrections that are reported in each risk management audit. This number of corrections can be further broken down into categories of major and minor corrections and, as with the number of actual corrections, can be assessed against the amount of effort required to implement them.

Some recommendations about risk monitoring and tracking:

- The frequency with which risks are measured should be dependent on the risk priority (higher priority means measure more often)
- At a minimum, risk measurements should be made/communicated at all key project milestones
- Resist the temptation to perform measurements too precisely or too frequently
- Track all actions to closure (and with demonstrated verification, if possible)



ANTICIPATED BENEFITS OF IMPLEMENTATION:

Effective implementation of formal risk management is based on the following set of benefits resulting from the process:

- Appropriately tailored risk management strategies are defined and implemented
- Potential problems (risks) that could impact project success are identified
- The likelihood and consequences of these risks are understood
- A priority order in which risks should be addressed is established
- Mitigation alternatives appropriate for each potential problem are carefully considered based on project circumstances
- Optimized mitigation techniques for all risks above their thresholds are selected
- Contingency plans in case the risk mitigates are developed proactively, rather than as a result of fire-fighting
- Information to improve risk management policies is captured, analyzed and acted upon
- Risk management processes/procedures are systematically and periodically reviewed and improved to further reduce risk

A necessary part of any approach to ensuring adequate software security is the definition and use of a continuous risk management process.

Conclusion

Project managers recognize and accept the fact that risk is inherent in any project. The most successful project managers choose to deal proactively with risk. They carefully analyze future project events and past projects to identify potential risks. Once risks are identified, managers take steps to reduce their probability or reduce the impact associated with them by establishing and following a entire project team as well as outside experts. Risk management should include hardware, software, integration issues, and the human element. A risk management process includes planning, assessment, handling, monitoring, and documentation. Risk is a product of the uncertainty of future events and is a part of all activity. Learning to balance its possible negative consequences with its potential benefits is the key to successful risk management.



References:

1. Leishman, T., and J. VanBuren. "The Risk of Not Being Risk Conscious: Software Risk-Management Basics." STSC Seminar Series, Hill AFB, UT., 2003.
2. Hall, Elaine M. Managing Risk. Addison- Wesley, 1998.
3. Data and Analysis Center for Software (DACS)
4. Software Engineering Institute (SEI)
5. NASA – Goddard Continuous Risk Management (CRM) Area
6. Software Technology Support Center. "Life Cycle Software Project Management." Project Initiation. Hill Air Force Base, UT, 9 Oct. 2001.
7. Department of Defense. "Risk Management Guide for DoD Acquisition." Washington, D.C.: DoD Feb. 2001: Chap. 2 <www.dsmc.dsm.mil/pubs/gdbks/risk_management.htm>.
8. Higuera, Ron, and Yacov Haimes. "Software Risk Management." Pittsburgh, PA: Software Engineering Institute, 28 June 1996 <www.sei.cmu.edu/publications/documents/96.reports/96.tr.012.html>.
9. Department of Defense. "Risk Management Guide for DoD Acquisition." Washington, D.C.: DoD, Feb. 2001: Appendix B <www.dsmc.dsm.mil/pubs/gdbks/risk_management.htm>. a