Chapter VIII

Risk Planning and Management

He who is not courageous enough to take risks will accomplish nothing in life.
(Muhammad Ali)

Success today in the business world (whether as an employee, manager, executive, or self-employed businessperson) involves taking some risks. The systems that are changing the world today are very risky, but the payback is enormous (DeMarco & Lister, 2003). One needs to know how to manage risk, however, including how to identify risk sources, quantify risk parameters, and develop plans to handle risk; these are the topics covered in this chapter.

Risks are inevitable in projects (particularly IT projects), *and if a PM does not practice sound risk management, that PM may constantly be in a crisis-management mode*. The high failure rate of modern *large* IT projects, such as those involving EAI/ERP, CRM, and SCM, is largely due to senior management and project management's failure to assess risks up front and to mitigate the causes of the greatest risks at the start of the project (Gibson, 2003). An adequate analysis of potential risks can significantly increase the likelihood of success for a project and can justify dollar amounts set aside for management reserves. "Risk management is increasingly seen as one of the main jobs of project managers" (Sommerville, 2003).

Project Risks and Opportunities

Risk is the possibility of suffering loss. In IT, the loss may involve increased costs, longer completion times, reduced scope, reduced quality, reduced realization of proposed

benefits, or reduced stakeholder satisfaction. *Risk and opportunity are different sides of the same coin. Some IT projects advance the state of the art, and as such are more risky than those that do not. The opportunity for significant advancement cannot be done without significant risk.* "Risk in itself is not bad; risk is essential to progress, and failure is often a key part of learning. But we must learn to balance the possible negative consequences of risk against the potential benefits of its associated opportunity" (Van Scoy, 1992).

Many organizations around the world are involved with general project risk management and/or IT project risk management. Barry Boehm of the U.S. Air Force, while working with the Software Engineering Institute (SEI), set the stage for modern IT risk management (Boehm, 1991). According to the SEI, risk management is a software engineering practice with the following processes: assess continuously what can go wrong (risks), determine what risks are important to deal with, and implement strategies to deal with those risks. The SEI model for risk management is shown in Figure 8.1.

The ISO/IEC 17799-1:2000 Code of Practice provides a sequencing of the risk management process into subprocesses for context identification, risk identification, risk analysis, risk evaluation, and risk treatment. The IEEE 1540 standard on software risk management is being merged with the corresponding ISO/IEC standard. Figure 8.2 shows the IEEE 1540:2001 overall risk management process.

The Project Management Institute (PMI; 2000) Risk Management Processes are:

- Risk identification
- Risk quantification
- Risk response development
- Risk response control

This book will mainly follow the PMI process definitions. All projects have some degree of risk, and most IT projects have *considerable* risk. Risk can, however, be reduced (studies have found that risk can be reduced up to 90% [PMI, 2000]). A PM should be somewhat risk averse (avoids taking unnecessary risks), but to significantly reduce risks, one must start with thorough risk planning.

Figure 8.1. SEI risk management model

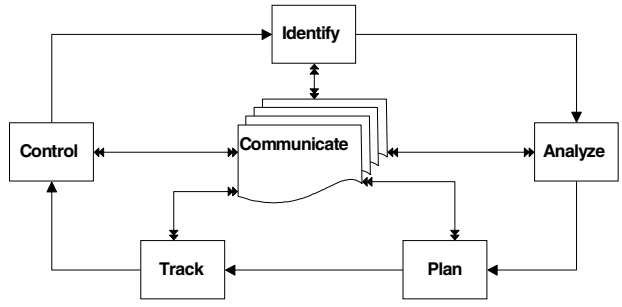
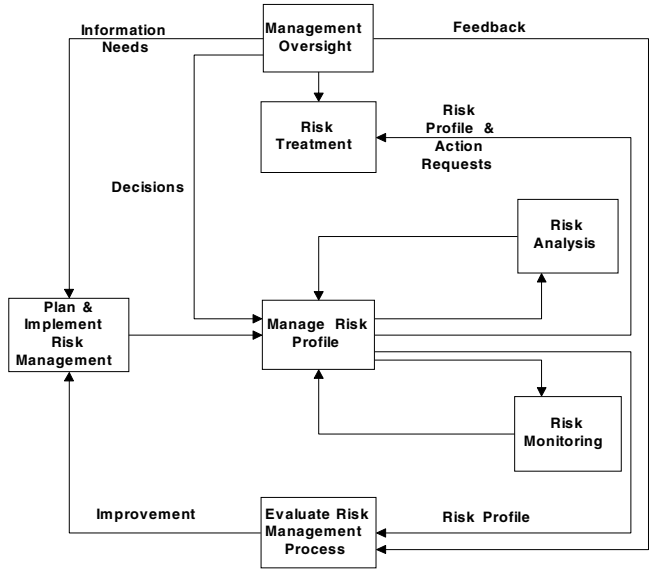


Figure 8.2. IEEE 1540:2001 risk management process



Risk management involves making plans and decisions in the face of uncertainty, and uncertainty is a state of nature, typified by the absence of information on a desired outcome. A *risk event* is a particular occurrence that could affect a project for good or bad. *Risk analysis* is the combination of risk identification and risk quantification. Therefore, *risk management* is the processes involved with identifying, analyzing, and handling risk. It includes maximizing the results of positive risk events and minimizing the consequences of adverse events.

Risk *context identification* is a step in the ISO risk-management processes. This context identification establishes the strategic, organizational, and risk-management environment.

Context identification is made through familiar business models like:

- *SWOT:* Strengths, weaknesses, opportunities, threats
- *Context:* Describes the system capabilities, as well as its goals and objectives and the strategies that are in place to achieve them
- *Target:* Describes the goals, objectives, strategies, scope and parameters of the activity, or system to which the risk management processes being applied
- *Assets:* Describes the identified assets and their dependencies
- *Security Requirements:* Describes the security requirements needed to preserve the identified assets

Risk context identification is typically focused on a particular management level: the entire organization, the IT organization, or the programs and projects,. Risk management for the entire organization involves threats and hazards to the organization as a whole. Risk management for the IT organization usually goes under the name "disaster planning" and involves threats and hazards to the IT assets, including hardware, software, and people. This chapter focuses on project-level threats and hazards that may affect the completion criteria or the satisfaction criteria identified earlier in this book. Most IT projects create or acquire software for deployment across one or a few servers and affect only part of an organization (i.e., few stakeholders). In EAI/ERM-type projects, however, the entire organization is affected (numerous stakeholders) and there are many more risk sources, including complex political, business value, expectations (requirements validation), security, workflow, and support issues (Maverick, 2003).

Good risk management is not a substitute for poor project planning! Risk management looks for things that might be a problem, whereas project planning (scope, cost, time, quality, etc.) identifies things that will definitely be problems and plans to make sure they do not happen (Young, 2003).

Risk Identification

The risk identification process normally considers the product description, scope, WBS, planning documents, historical information, and industry information to determine sources of risk, potential risk events, and risk symptoms. One must determine which risks are likely to affect the project (and the product that is the subject of the project) and the key characteristics of each risk. Ideally, risk identification should start during project initiation and should finish during project planning. In practice for IT projects, however, a risk analysis is typically done after project planning and before the final costing of the project. Risk identification cannot be fully completed until the WBS is created and most work, staff, and procurements have been specified. Then risks are further identified as the project proceeds and as change orders come in.

Project critical success criteria and factors were discussed previously in this book. In a broader sense, risk identification should start with these critical success factors of the project. These factors can be used to identify critical sources of risks that may arise from our satisfaction criteria and completion criteria. The critical success factors were determined by considering all stakeholders for a project, and risk source identification should also consider all stakeholders. For large IT projects that will create products that will significantly change organizations (such as how business processes are performed and the "balance of power" within an organization), the major risks may involve satisfaction criteria more than the completion criteria. "Business based project failures come from such things as not having new workflow processes [to go with the new product], not adapting the structure of the organization to the new ways of working, not revising incentives and rewards to emphasize the new goals, and keeping the old cultural practices in place even when they impede the new ways of working" (Gibson, 2003).

Risk identification is carried out by finding potential hazards and threats in different risk sources. What is the difference between a hazard and a threat?

- The primary volcanic hazards are pyroclastic flows and surges, airborne fallout such as tephra clasts, ash (heavy accumulations and persistent), and lightning.
- *The* threats posed to people *by these hazards are death or injury from inundation, fire, heat, missile impact, lightning strike, and ash ingestion.*
- *The* threats to infrastructure, lifelines, and property *are destruction, damage, route obstruction, structural collapse, electrical malfunction, and pollution.*

For example, a board with a nail sticking out or it is a hazard that is more of a threat to people if the board is lying on the beach rather than lying in the street. Lying in the street this hazard may be more of a threat to cars than that same board lying on the beach. Risks/ hazards are often separated by such categories as:

- *Business Risk:* Risk of a gain or loss
- *Pure (Insurable) Risk:* Risk of a loss only

The sources of risk are further classified as:

- *Internal:* Project variables (including managing the "normal" trade-offs in the project schedule, cost, quality, scope) and other factors inside an organization
- *Technical:* Technology uncertainty or change
- *External:* Factors outside of the organization
- *Unforeseeable:* Only 10% of risks fall into this category

Internal and technical risks are often quantified at the WBS level for projects, whereas external and unforeseeable risks are quantified at the overall project level. A PM is generally responsible for internal and technical types of risk events. Sometimes issues dealing with the customer are classified as external, and sometimes they are classified as internal depending upon whether the customer (benefiting organization) is internal or external to the company. The same situation may be true of procurement, that it may be classified as internal or external; outsourcing risks are usually considered internal. Procurement and outsourcing are discussed in detail in a later chapter of this book. Some indicators of potential internal risks would be related to:

- *Investment Size:* Size of project cost versus budget of benefiting or performing department (i.e., IT department)
- *Project Size:* Time length of project compared to "cycle" time in that industry

- *Impact Analysis:* How broadly project results may impact organization, customers, industry
- *Business Risks:* New corporate organization, merger, new employees, new vendors/contractors
- *Political Risks:* Who internally cares about the project and their corporate influence and power
- *Performing Organization Risks:* Staff and management uncertainties

Some technical risks in IT projects would possibly be:

- New type of project
- New area of application
- New methodology
- New technology (platforms, languages, tools, algorithms, methods, etc.)
- New standards
- "Going where no project in this company has gone before"

External and unforeseeable risks are not usually the responsibility of the PM. Unforeseeable risks include natural hazards (such as weather events, earthquakes, etc.), market fluctuations, riots, fires, crime, war, and the like. Only about 10% of risks are unforeseeable (PMI, 2000). Some indicators of external risks would be related to:

- *Benefiting Organization (Customer) Risks:* Management and contact uncertainties
- *Procurement Risks:* Vendor issues
- *Political Risks:* Those who externally cares about the project and their political power
- *Compatibility Risks:* Alignment to current and new standards
- *Economic Risks:* Flexibility to changes in local, national, and global economic factors

One good way to start to identify risks is with a standard industry checklist or questionnaire. One such questionnaire from Pearlson and Saunders (2004) is:

- Are we doing the right things?
  - Are project objectives clear?
  - Will the proposed solution support business activities?
  - What changes should be considered?

- Are we doing it the best way?
  - Have alternative ways been explored?
  - Are there new or emerging ways we should consider?
  - What changes would increase the likelihood of success?
- How do we know how we are doing?
  - What are the performance standards?
  - Is there regular progress reporting?
  - How will the staff give feedback?
- What impacts are we having on the business?
  - To what extent have project objectives been achieved?
  - Are the project clients satisfied?
  - Is satisfaction improving or declining?
  - Is support for the project improving, stable, or declining?
- Is the project cost effective?
  - What significant business costs are influenced by this project?
  - What is the trend of these costs?
  - What significant variances from budget have occurred?
- Is there clear accountability for the project?
  - Are the right people involved?
  - Are lines of responsibility clear?
  - Is senior management supportive?
  - Is performance monitored and on track?
  - Do all those involved with the project understand their roles?
- Are key assets protected?
  - Will the IT infrastructure handle the deployment of this application?
  - Is IT security adequate?
  - Are risks identified and monitored?
  - How are incidents reported and analyzed?"

A widely used checklist was given by Wideman (1992) who listed common general project risks and then specific risks by category: external unpredictable, external predictable, internal nontechnical, technical, and legal. Some of the items from his common general lists that are typical in IT projects are:

- This project is very different from previous ones we have done.
- The project scope, objectives, and deliverables are not clearly defined.
- Some or all technical data are lacking.

- The technical process is immature.
- Standards for performance are unrealistic.
- Design lacks engineering input.
- Prototype of key elements are missing.
- There is a higher than usual R&D component
- Other similar projects have not been successful.
- A wide variation in bids for support products and services.
- Some key subsystems are sole source bids.

Sommerville (2003) identified common risks to software development: staff turnover, management change, hardware unavailability, requirements change, specification delays, size underestimation, CASE tool underperformance, technology change, and product competition.

Another way to identify risk is via a framework. One such framework, defined by Marchewka (2003), began by examining risks involving project scope, quality, or budget. It then viewed risk influences for these items in terms of people, legal, process, and so forth. It then considers whether the risk is internal or external, what is known about the risk (frequency and impact), and where in the project timeline the risk will occur:

Scope
Quality
Schedule
Budget
People
Legal
Process
Environment
Technology
Organization
Product
Other
Internal
External
Known risks (frequency and impact known)
K-U risks (frequency known, impact unknown)
Unknown risks (frequency and impact unknown)
Project conceptualization
Charter and plan

Execute and control
Closeout
Evaluation

The IEEE framework (taxonomy) is based upon identifying risk in three areas: product engineering, development engineering, and program constraints. Each category has subcategories, and each subcategory has specific areas (Carr, 1993):

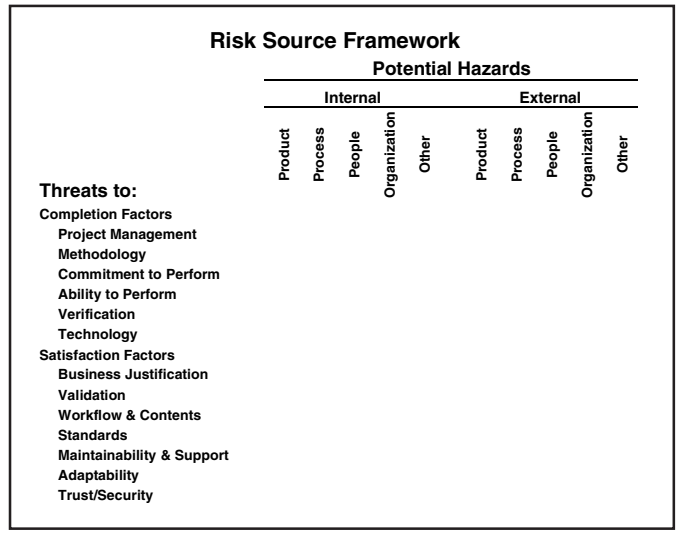
- Product engineering
  - Requirements
  - Design
  - Code and unit test
  - Integration and test
  - Engineering specialties
- Development engineering
  - Development process
  - Development system
  - Management process
  - Management methods
  - Work environment
- Program constraints
  - Resources
  - Contract
  - Program interfaces

This framework is very specific; for example, under product engineering/requirements, the areas are stability, completeness, clarity, validity, feasibility, precedent, and scale. Next questionnaires are developed based upon the three framework areas. The aforementioned IEEE report has specific questionnaires, and other related IEEE documents prescribe meeting and interview techniques to administer the questionnaires.

A more general framework suggested herein for IT projects is based upon the critical success criteria and factors introduced early in this book. Figure 8.3 shows a template for this risk source identification template.

Each cell in our framework table (intersection of a hazard and a threat) is a risk source arena. There may be more than one risk source in each cell. In practice, each cell is analyzed to identify sources of risk, and for each risk identified, a set of specific symptoms is listed. These symptoms are early warning signals that a risk event may be about to occur. Thus, the PM, project team, and line management can watch for these symptoms during the project execution.

Figure 8.3. Critical success-factor–based risk framework



Risk Quantification

After the risks and their symptoms are identified, those risks need to be quantified and the stakeholders' risk tolerance levels determined. The risk quantification process will result in a list of opportunities to pursue, threats to respond to, opportunities to ignore, and threats to accept. Risk quantification may utilize several models, including

- *Hazard Frequencies:* Describes frequency estimates for the identified hazards
- *Threat Frequencies:* Describes frequency estimates for the identified threats
- *Consequence Estimates:* Describes consequence estimates for the identified hazards

The formal analysis of risk includes the following risk factors:

- The probability that the risk event(s) will occur
- The economic impact (money at stake)
- When the risk event(s) may occur (timing)
- How often are they likely to occur (frequency)

The first step in the analysis is determining the probability and impact. The two methods are commonly used are

- *Qualitative:* Expert opinion, project historical data, educated guess
- *Quantitative:* Parametric formulas, simulation, industry and/or application statistical data

Quantitative methods are certainly better (less subjective and more accurate) when the data are available and the use of that data is appropriate. Monte Carlo simulations using the network diagram and PERT estimates are sometimes used to simulate risk involving time and cost (gives a percentage probability that each task will be on the critical path).

The most common quantitative method is the expected monetary value (EMV) calculation. Probability and impact are used to calculate the EMV as:

EMV = Probability * Impact

EMV units may be dollars or some other loss scale. *EMV is very important to a PM in that it helps the PM prove his or her need for reserves!* The impact is typically in money or person hours. A single impact number may be used in the previous formula, or the impact may be calculated from a maximum impact (total loss) times the probability of that maximum value (Pi):

Impact = maxImpact * Pi

EMV = Pe * Pi * maxImpact (where Pe is the event probability)

Calculation of management reserves involves summing up the EMV for all the identified threats and opportunities. This is can be done in tabular or spreadsheet form, as shown in Figure 8.4.

A less precise quantitative method uses the base formula but expresses both impact and event probability on a scale (such as from 1 to 10) or as a rough percentage. This method, though less precise, may be more applicable, particularly in IT projects in which impact and probability are harder to estimate in an absolute sense. The impact for each risk is the fraction of the project overall budget that is directly affected by that risk. A relative EMV is calculated for each risk by multiplying the probability of the risk by the impact

Figure 8.4. EMV determination

Risk/Opportunity	Impact	Probability	EMV
Threat 1			
…			
Threat N			
Opportunity 1			
…			
Opportunity N			
Total	-----	-----	xxx

(amount of the budget at risk). The relative EMVs may be summed to calculate a management reserve for risk mitigation:

$$\text{Management Reserve} = \sum \text{Probability}_i * \text{Impact}_i$$

For example, if there are two risks, and the first risk affects 50% of the project budget with a probability of 20%, and the second risk impacts 30% of the project budget, with an probability of 15%, then he management reserve would be 14.5%. Another similar method is based upon a risk matrix, which gives a grade to the intersection of risk probabilities and impacts. This is shown in Figure 8.5. Each risk is assigned a grade and then is ranked accordingly.

Tolerance levels (the amount of risk that is acceptable) should be determined for each stakeholder (or each type of stakeholder). Those tolerance levels are later used in developing the risk management plan, specifically what types of risk events may be accepted versus those risk events that need to be averted, deflected, or mitigated. A project's stakeholders may be daredevils, or they may be timid. If they are daredevils, they will assume risks and will not be too upset if a risky decision proves to be the wrong choice; however, if the stakeholders are timid, they will be upset about decisions that involve any significant degree of risk. The R-Variables control the stakeholders view of risk: Regret/Resent [pain], and Rejoice [rejoice] (Piney, 2003). By knowing the stakeholders well, or by interviewing them, one can determine risk tolerance. A formal utility function can also be developed, as is illustrated in Figure 8.6.

Piney (2003) defined different zones for the utility function graph that indicate different ways risks may need to be analyzed to determine stakeholder tolerance. The dead zone indicates threats and opportunities for which no response is developed. A table or spreadsheet may be prepared listing threats and opportunities for which responses need to be developed, as is shown in Figure 8.7.

As stated previously, internal and technical risks are often quantified at the WBS level for projects, whereas external and unforeseeable risks are quantified at the overall project level. Figure 8.8 shows a screen from the FiveAndDime system that provides for project-level risk factors for internal and technical risks (the project risk factor—how risky is this work) and for external risk (the customer—how risky [difficult] is this customer to work with). These risk factors are applied to cost estimates. Further on in this chapter, a specific example is given to illustrate one way to calculate these factors. Figure 8.9 also shows a screen from the FiveAndDime system for a WBS item in which there is a risk factor at the WBS level (how risky is this item of work).

Figure 8.5. Risk grading

		Impact	
Probability	**Low**	**Medium**	**High**
Low	1	2	3
Medium	2	3	4
High	3	4	5

Figure 8.6. Risk utility function

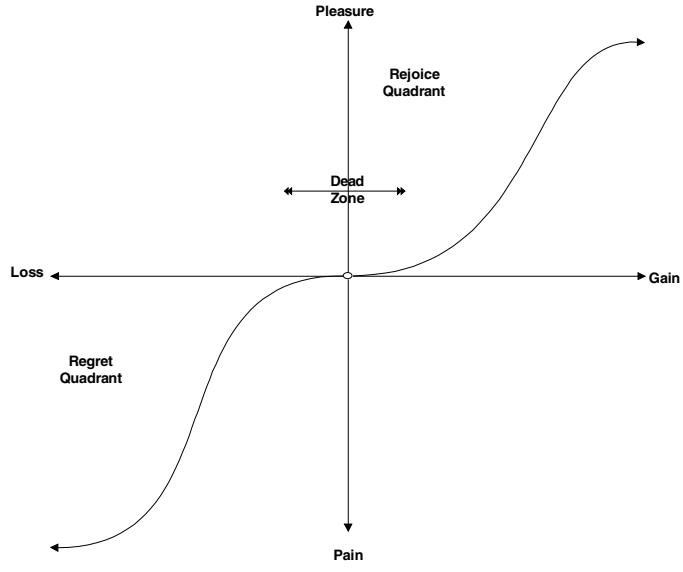


Figure 8.7. Response matrix

Risk/Opportunity	EMV	Priority	Respond ?
Threat 1			
…			
Threat N			
Opportunity 1			
…			
Opportunity N			
Total	**xxx**	-----	-----

Risk Response Development

Once the threats and opportunities have been categorized (opportunities to pursue, threats to respond to, opportunities to ignore, and threats to accept), the risk responses are formulated and the risk management plan is completed. The risk management plan usually specifies the overall management reserve. This is illustrated in Figure 8.10.

Figure 8.8. Project form (showing project risk factors)

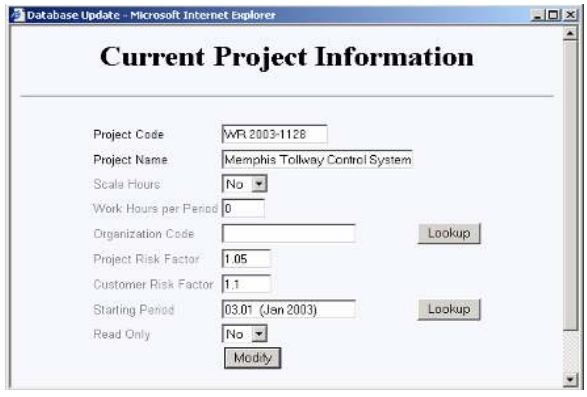


Figure 8.9. WBS form (showing WBS risk factor)

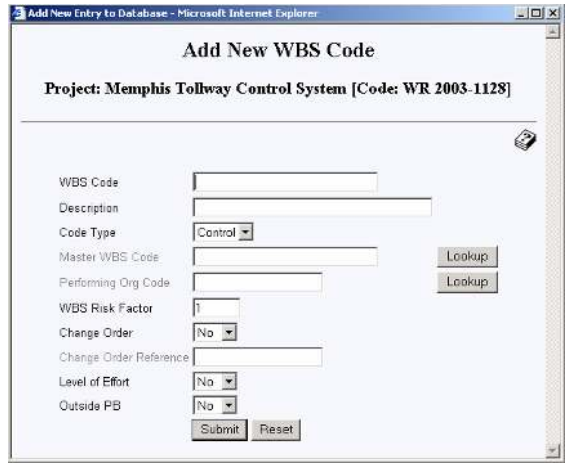
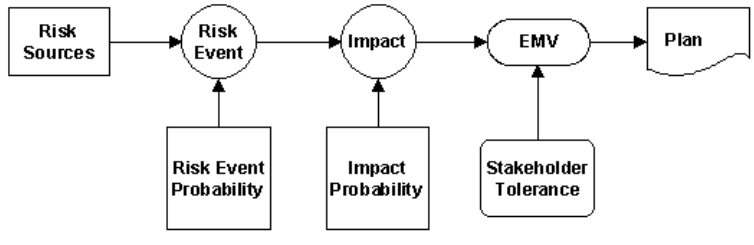


Figure 8.10. Risk plan development



There are several types of risk response plans (often called "mitigation strategies"):

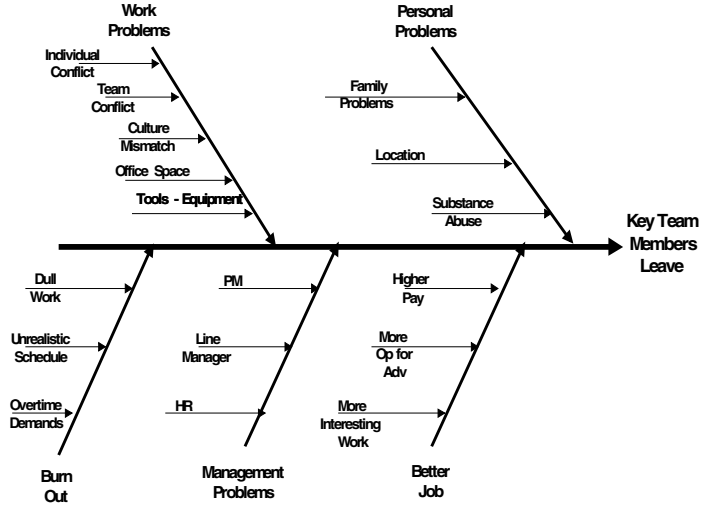
- *Avoidance:* Eliminate the cause of the event(s) or reduce the EMV via reducing probability
- *Mitigation:* Reduce the EMV via reducing the impact of the risk event
- *Acceptance:* Accept the risk (take no preventive action)
- *Deflection:* Assign (transfer) the risk to another party

Avoidance reduces the EMV by reducing (or setting to zero) the probability. The main methods used here are safety- and prevention-related techniques, which are employed in the early stages of a project. Prevention measures are available for almost all IT risks and include such methods as employee retention and motivation incentives, buying parts of a system instead of building all of it, use of contract labor for nonconfidential parts of the system, parallel design and construction of alternative algorithms, platform independent implementation techniques, use of open source software, using reusable components, and using object-oriented techniques. These methods have been discussed in earlier chapters of this book. For avoidance to be effective, one must identify the root cause of potential risk problems. One method is the Ishikawa Diagram (commonly called the *fish bone* or *cause-and-effect* diagram), which is illustrated in Figure 8.11.

Mitigation reduces the EMV by reducing the impact; these methods are:

- **Contingency Plans:** "Planned mitigation"; alternative means to do something should a certain risk event occur; "contingency reserves"
- **Workarounds:** A method devised to handle risk when the risk event happens ("unplanned mitigation")

Figure 8.11. Cause and effect diagram



A bullet analogy (extended here), concerning being the target of a gunshot, was presented by Wideman (1992):

Mitigation (workarounds):

Move out of the way

Deflect the bullet

Repair the damage done by the bullet

Mitigation (planned):

Wear a bulletproof vest

Avoidance (reduces the probability of being shot at):

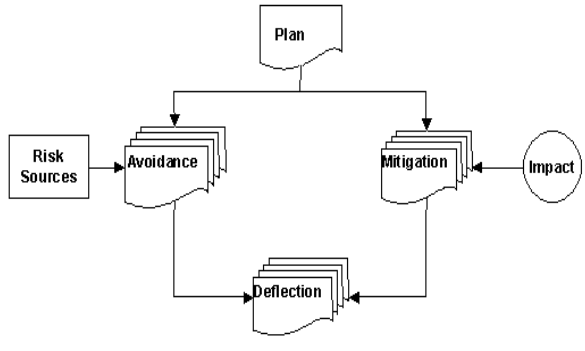
Carry a visible gun yourself

"Play dead"

Take steps to avoid being confronted by someone with a gun

Avoidance is usually a better measure than mitigation, but it is not always cost effective or possible. One can chose mitigation, avoidance, or both; however, one is not in control of the bullet. One could also ignore the risk by assuming that the likelihood is very low or that the impact is very low (bullets cannot hurt me). Mitigation techniques may be planned but are not typically implemented until the risk event is imminent or has occurred. This is illustrated in Figure 8.12. Well-planned projects have more contingency plans than workarounds; the opposite is the case for poorly planned projects. Avoidance and mitigation usually cost money to implement, and reserves are formulated. Reserves are the amount of time and cost added to a project to account for risk, also called management reserve. PMI recommends at least 10% (PMI, 2000). Using PERT and Monte Carlo simulations, one can calculate the reserve needed at the task level and then total these amounts for the entire project. However, in practice, less-precise methods are typically used because the data available for the analysis are less precise.

Figure 8.12. Risk plan components



Deflection attempts to transfer the risk (part or all) to another party via:

- *Insurance:* Exchanges most of a risk of a probabilistic event(s) for a certain fixed cost
- *Outsourcing:* Let someone more capable or experienced do the work
- *Procurement/Contracts:* Buy/rent the needed expertise, equipment, material, software, and so forth

Deflection transfers and reduces the risk, but does not eliminate it. As discussed in a later chapter of this book on procurement and outsourcing, there are considerable risks in outsourcing. Hallows (1998) lists such subcontractor risks as:

Technical
- Competent resources not assigned
- Lack of familiarity with project or product
- Methodology not proven
- Poor project management techniques
- Technical disconnects due to distance of relationship

Operational
- Subcontractor staff goes out on strike
- Subcontractor lands a higher priority project
- Distance leads to business and operational disconnects
- Transportation causes problems
- Customs causes problems

Financial
- Subcontractor goes bankrupt
- Subcontract holds up deliverables due to contract or payment disagreements
- Subcontractor uses your schedule to extract extras
- Subcontractor reduces quality

Jones (1994) lists the major risk factors (and a percentage of projects at risk) in contracting and outsourcing arrangements as:

- High maintenance costs (60%)
- Friction between contractor and client personnel (50%)
- Creeping user requirements (45%)

- Unanticipated acceptance criteria (30%)
- Legal ownership of software and deliverables (20%)

These risks can be minimized by thoroughly qualifying the vendor, requiring the vendor to have the proper certifications, having the proper contract (with regard to terms, legal language, length, etc.), handling security issues, and requiring the vendor to report costs and progress using earned value methods (discussed later in this book).

The table in Figure 8.13 lists common risks for IT projects and possible avoidance and mitigation measures.

Figure 8.13. Common IT project risks

Risk	Avoidance	Mitigation
Incomplete requirements; Insufficient user involvement	Document clearly all requirements and get customer approval; utilize prototypes; make sure your analysts are talking to the right customer personnel; establish formal change control system	Involve users in documenting and approving requirements; use prototypes to flush out requirements; involve users with testing and documentation
Customer is difficult to work with	Assign higher risks and increase reserve; request customer management contact for conflict resolution; determine if the problem is with your personnel or customer personnel; have very good legal contracts	Document all customer interaction; frequently involve customer with requirement analysis, prototype review, design review, and test review
Lack of standard architecture	Obtain software engineering expertise; adopt standard architecture; adopt relevant IT standards; consider open source software	Depending upon depth into project, adopt and enforce relevant standards; use more prototypes
Inaccurate task estimating; Unrealistic task estimates	Use parametric estimation technique and compare with historical data and estimates by those who will do the work	Re-estimate remaining work if original estimate were not done in a quantitative manner and/or if multiple estimation techniques were not used; see cost overruns below
Inexperienced or poor PM	Set up apprentice PM program in organization; require PMI (or equivalent) certification for all PM's (for projects over a certain size); set up a PMO in organization	Project plans, controls, and issues reviewed by internal or external certified PM consultants; upper management review of PM choice
Insufficient staff; recruiting problems, staff illness	Prioritize requirements and phase project; use contract labor; outsource part of work; buy components	Use contract labor; outsource part of work; request extension from customer
Dependency on key team member(s)	Special recognition, position, incentives for these key persons; identify backup employees or contractors	Additional incentives to motivate the key members to stay thru project completion

Figure 8.13. Common IT project risks (cont.)

Risk	Avoidance	Mitigation
Scope creep, requirements changes	Have user sign off on requirements and change order plan; contingency funds for unforeseen changes; more use of prototypes	Document all change requests; Prioritize requirements and phase project; Charge customer for changes and develop new baseline schedule and cost plan
Vendor problems (lateness, quality issues, etc.)	Have a formal procurement process that results in qualified vendors, good legal contracts, and a "win-win" situation for both buyer and seller	Negotiate issues with vendor and use whatever measures are available within your contract (see book chapter on procurement)
Cost overruns (not due to scope creep)	Use earned value metrics (see book chapter on performance measurement); employ multiple estimation techniques, employ PERT estimation	Voluntary uncompensated overtime, scope reduction, project phasing, buy components
Lateness (not due to scope creep)	Use earned value metrics	Crashing, fast tracking, contract resources, scope reduction, project phasing
Quality problems	Carefully set and enforce standards, utilize modern object oriented architectures, use proven technology, plan for thorough testing; use Quality Function Deployment (QFD) to involve customer; use extensive prototyping	Increase prototyping and testing, verify standards adherence, use QFD
Team problems: low productivity, burn out, low morale	HR to interview backups, identify contractors; "team building" measures	"Team building" measures; re-assign people to different tasks or projects; utilize backup personnel or contractors
Weak upper management support	Strong quantified business justification for project; thorough project charter signed off at high level in organization	Revisit business justification with upper management; seek other support in organization; regular reporting of project progress and cost

The final risk-management plan documents information about the identified risks and how each risk is be handled. It may also include information about noncritical risks (risks not needing a response development) so that they can be revisited (during project execution) if necessary. An excellent risk management plan is a sign of an experienced PM versus an inexperienced PM.

Risk Plan Example

As a very simple example, consider the case of a company that plans to develop a software system, and has the following steps in its risk management plan:

- Identify the risks via the success factor framework, its lessons learned and industry historical data
- Grade each risk on a scale of 1 (*low*) to 10 (*high*), based on the probability of occurring
- For each risk, list the symptoms
- Grade each risk on its impact (percentage of budget affected) from 0 (*no impact*) to 10 (*total*); a value of 10 in this instance means that the entire project budget is at risk
- Calculate the relative EMV (impact times probability divided by 100)
- Identify the response to be taken for each risk (deflection, avoidance, contingency plans, workarounds)

In identifying the risks, the PM and his or her team review the published lists of the most common reasons why projects fail, both in general and for IT projects in particular. Jones (1994) studied software risks in detail, and his list of the most serious software risks are:

- Inaccurate metrics
- Inadequate measurement (of software development costs)
- Excessive schedule pressure
- Management malpractice (PM experience)
- Inaccurate cost estimation
- Silver bullet syndrome
- Creeping user requirements
- Low quality
- Low productivity
- Canceled projects

Another well-known list for software risks in IT projects is called "CHAOS," from the Standish Group (2004), which has surveys for several years from 1994 to 2004:

- Lack of executive management support
- Insufficient user involvement
- Inexperienced project manager
- Business objectives not clear
- Minimization and compromise of scope
- Lack of standard software architecture/infrastructure
- Lack of clear statement of requirements
- Lack of formal methodology

- Poor estimates
- Lack of proper planning
- Unrealistic expectations
- Scope minimized
- Lack of project "ownership" by team
- Team not hard working and focused
- Vision and objectives unclear
- Incompetent staff
- Improper setting of milestones

This Standish risk list represents the ranking of the problems in their 2000 report, although other years had a different order to the issues, for example in 2004 user involvement was first and executive support was second. Here is another list of risks from the ETP Group (O'Connell, 2002):

- The goal of the project is not defined properly
- The goal of the project is defined properly, but then changes to it are not controlled
- The project is not planned properly
- The project is not led properly
- The project is planned properly, but then is not resourced as planned
- The project is planned such that it has no contingency
- The expectations of the project participants are not managed
- The project is planned properly but then progress against that plan is not monitored and controlled properly
- Project reporting is inadequate or nonexistent
- When projects get in trouble, people believe the problem can be solved by some simple actions (e.g., work harder, extend the deadline, or add more resources)

Here is yet another list, from Tennant (2002):

- Poor planning
- Lack of resources (money and people)
- Constant reorganization and scope changes
- Lack of management support
- Poor communications
- Too much infighting and disputes
- No clear definition of roles and responsibilities

- Lack of clear objectives or scope
- Failure to recognize warning signs
- Unrealistic expectations

Another list is available from PCI Global (2002):

- Lack of timely approvals
- Delay in funding
- Surprise audits
- Defective materials
- Mistakes trigger rework
- Vendors do not deliver on time
- Key staff member is pulled off project
- Management institutes a "hiring freeze"
- Member of the team is absent too much
- Member of the team resigns
- Changes in specifications

This example in Figure 8.14 is a new and difficult type of project for the company, and the project team identifies the following risks: employee burnout, poor project management, insufficient resources, general employee turnover, key programmers leave, scope creep, overly low task estimates, and nonfeasable choice in technology (i.e., "immaturity"). Using these identified risks, the project team prepares a risk-management plan in

Figure 8.14. Example risk analysis

Risk	Prob.	Budget Impact	Relative EMV	Symptoms
Employee "burnout"	2	1	.02	Low Morale, lateness
Poor project management	1	5	.05	Lateness, cost overrun, earned value issues
Insufficient resources available	3	1	.03	Lateness, staffing problems
Employee turnover	2	1	.02	People leaving
Key programmers leave	2	4	.04	Key people leave
Scope "creep"	3	1	.03	Lateness (project level); additional scope
Task estimates are too low	2	3	.06	Lateness (task level), earned value issues (task levels)
Poor IT architecture choice	1	2	.02	Prototype time lengthens

Figure 8.15. Example risk plan

Risk	Response
Task estimates are too low	Closely monitor against actual costs to see if project needs to be phased or scope reduced
Poor project management	PM and team address specific issues, Upper Mgmt. involvement
Key programmers leave	Provide added incentives to key people to at least stay until project completion
Insufficient resources available	Phase project or request more $
Scope "creep"	PM steps in to "phase" project and deal with customer
Employee turnover	HR to interview backups, identify potential contractors
Employee burnout	Re-assign people to different tasks or projects
Poor IT architecture choice	Re-evaluate architecture choice, use more prototyping

a tablular format. The columns include information about probability, impact, EMV, and the symptoms for each risk.

The team next ranks the risks by EMV and then determines the response (in this example there is a need to respond to each threat); this is shown in Figure 8.15. The management reserve is calculated at about 27% (sum of relative EMVs). To complete their risk plan, the following information is recorded for each risk:

- Complete definition of the risk
- Why the risk is important to the project
- The impact, probability, and EMV
- The planned response(s)
- Who is responsible for recognizing and tracking the symptom(s)
- Who is responsible for the response(s) and recording results thereof
- What resources are needed for the response(s)

Risk Response Control

Risks need to be monitored continuously during the execution of a project by looking at the risk symptoms and seeing if any risk events have occurred or are about to occur. In fact, one of the most important items to address during project team meetings is risk. If any planned risk events occurred, then the risk management plan must implement the called-for response (i.e., contingency plans); or, if any unplanned risk events occur, then a workaround must be found. New risks or risk events may have also surfaced, or there may be a change in the risk ranking due to changes in probability or impact amount; some previous noncritical risks now become important. In Chapter IX, I review corrective

actions and workarounds for projects that have schedule and/or cost problems, including such methods as fast tracking and crashing.

When risk events happen, they should be recorded in a project risk log, which describes the risk, the circumstances of its occurrence, the risk response taken, the degree of success of the response, the estimated cost of the event, and how the team thinks such risks could be better handled in the future. This log becomes part of the lessons-learned documentation at project closeout (project closeout is discussed in Chapter XI).

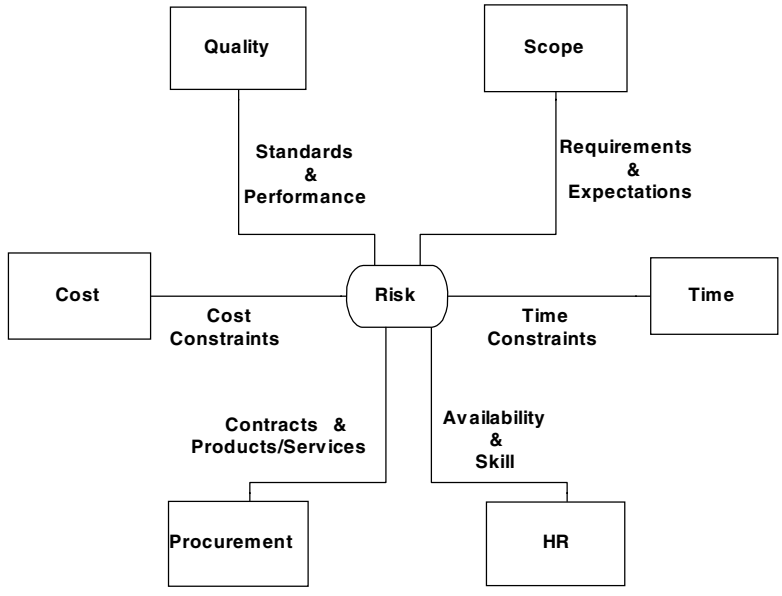
Rita Mulcahy (2003) listed the most common stumbling blocks in risk management:

- *Risk identification is completed without knowing enough about the project.*
- *Project risk is evaluated using only questionnaire, interview, or Monte Carlo techniques and thus does not provide a detailed, per task analysis of risk.*
- *Risk identification ends too soon resulting in a brief list (about 20) rather than an extensive list (hundreds) of risks.*
- *Risk identification and risk quantification are blended resulting in risks that are evaluated or judged when they come to light. This decreases the number of total risks identified and causes people to stop participating in risk identification.*
- *The risks identified are general rather than specific (e.g. communication rather than poor communication of customers needs regarding installation of system xxx caused two weeks of rework).*
- *Whole categories of risks are missed such as technology, cultural, or marketplace.*
- *Only one method is used to identify risk rather than a combination of methods. A combination helps ensure that more risks are identified.*
- *The first risk response strategy identified is selected without looking at other options and finding the best option or combination of options.*
- *Risks are not given enough attention during the project execution stage.* (Mulcahy, 2003)

Chapter Summary

Risk management and the planning thereof have been discussed in this chapter. The project risk-management process interacts with the other project management processes (as defined by PMI, 2003) in such a manner as to create or minimize risks. This is illustrated in Figure 8.16. Procurement and outsourcing activities in any phase of an IT project introduce additional risk, and these topics are covered more extensively in Chapter XII. *Thus, how well the other project management processes are done affects the risk introduced into the total process.*

Figure 8.16. Planning influence on risk



The project management office (PMO; discussed in detail in Chapter XVI), should be the organizational focal point for risk policies, procedures, frameworks, templates, and checklists. Risk factors (probability and impact) need to be evaluated not only initially but throughout the project. Earlier in this book, success criteria and stage gates were discussed. The stage gate evaluation process should include an updated analysis of risk; this overall process was shown in Figure 3.16 in Chapter III. Example forms for this stage gate evaluation were also illustrated earlier in the book, and based on that form each risk was reexamined.

For another excellent analysis of IT and software-related risks, see *Assessment and Control of Software Risks* (Jones, 1994). Jones itemizes the most common and most serious software risks and for each risk discusses: definition, severity, frequency, occurrence, susceptibility and resistance, root causes, associated problems, cost impact, methods of prevention, methods of control, support (product, consulting, education, periodical, standards, and professional associations), effectiveness of known therapies, cost of therapies, and long-range prognosis. Some of the details in this book are outdated at this point in time, although the general observations and recommendations are still very relevant. PMI also has a risk management special interest group (SIG; www.risksig.com/). The PMI Risk Management SIG offers forums for the exchange of ideas on topics in this area. Products like FiveAndDime have risk management built into the total PM system, but there are software products that are used only for risk management and interface to various PM software tools and or spreadsheet programs: Pertmaster (www.pertmaster.com), RiskTrak (www.risktrak.com), Crystalball

(www.decisioneering.com), Primavera's P3-MonteCarlo (www.primavera.com/products), Decision Products' RiskDriver(www.riskdriver.com/), Palisade's @RISK (www.palisade.com), and Risk+ and RiskRadar (www.iceincUSA.com).

References

Boehm, B. (1991). *Software risk management*. Upper Saddle River, NJ: IEEE.

Carr, M. (1993). *Taxonomy-based risk identification* (Tech. Rep. No. 93-TE-6). Pittsburgh, PA: CMU/SEI.

DeMarco, T., & Lister, T. (2003). *Waltzing with bears: Managing risks on software projects.* New York: Dorset House.

Gibson, C. (2003). IT enabled business change: An approach to understanding and managing risk. MIS *Quarterly Executive, 2*(2), 104-115.

Hallows, J. (1998). *Information systems project management*. American Management Association.

Jones, C. (1994). *Assessment and control of software risks.* Englewood Cliffs, NJ: Yourdon Press Computing Series.

Marchewka, J. (2003). *Information technology project management*. Wiley.

Maverick, G. (2003, November). EAI project management. *Business Integration Journal,* 48-50.

Mulcahy, R. (2003). *Risk management, tricks of the trade for project managers.* Minneapolis, MN: RMC.

O'Connell, F. (2002). *Reasons why projects fail*. Retrieved from www.etpint.com/whyfail

PCI Global. (2002). *Crises events in projects.* Retrieved from www.pciglobal.com

Pearlson, K., & Saunders, C. (2004). *Managing and using information systems.* New York: Wiley.

Piney, C. (2003, September). Applying utility theory to risk management. *Project Management Journal.*

PMI. (2000). *The project management body of knowledge (PMBOK).* Newton Square, PA. ISBN 1-880-410-22-2.

Sommerville, I. (2003). *Software engineering*. Boston: Pearson Addison Wesley.

Standish Group. (2004). *Chaos chronicles.* Retrieved from www.standishgroup.com

Tennant, D. (2002, July 7). *Reasons why projects fail.* PMI.

Van Scoy, R. L. (1992, September). *Software development risk: Opportunity, not problem* (CMU/SEI-92-TR-30, ADA 258743). Pittsburgh, PA: Software Engineering Institute.

Wideman, R. (1992). *Project and program risk management*. Newton Square, PA: PMI Press.

Young, S. (2003). Why IT projects fail. *Computerworld.*