Chapter 5: IT Project Risk Management

Project risk is the possibility of unpredictable and unwanted loss that can occur anytime and anywhere during the project's life cycle. Risk is associated with all aspects of the project, such as requirements, architecture, design, implementation, team members, management, and work breakdown structure (WBS). Risk can be due to cost overruns, schedule delays, manpower turnaround, design constraints, influence of new technology, hardware defects, software bugs, communication problems, and network and Internet failure. The project manager continually assesses, identifies, monitors, and manages risk before it becomes a problem. The manager establishes an industry best practices risk management plan to identify and control performance, cost, and schedule risk.

What is Project Risk?

Project risk is a subjective assessment that is made on the probability of *not* achieving a specific objective within the time, costs, and allocated resources. In addition, project risk is the possibility of suffering any loss during the project's life cycle. In a development project, the loss describes the influence to the project in the form of diminished quality of the end product, increased costs, delayed completion, or failure of the project.

A loss associated with a surprise event can take place during the project's system development and maintenance phases. This event creates loss of time, quality, money, control, and understanding among the developers, customers, users, stakeholders, and project manager. For example, if requirements change frequently, the project can suffer from loss of control and understanding. This event can start ripple effects, affecting cost, schedule, and morale of practitioners associated with the project.

Such an event is always likely to affect the project drastically. For example, a project is developed on a host computer to be ported to a target computer when fully tested. Suppose the target computer model is not of the same version as the host computer (i.e., risk probability). The risk probability is measured from 0 (impossible) to 1 (certain). When the risk probability is 1, the risk is called a problem (the probability of failing to achieve a specific outcome and the consequences of failing to achieve that outcome).

The project manager examines each identified risk, refines the description of the risk, isolates the cause, and determines the influence of the risk on the project. The risk influence analysis includes the probability of risk occurrence, its consequences, and its relationship to other risk areas and processes.

Risk Factors

The IT project manager should carefully calculate risk factors. The manager creates and maintains a list of the risk factors at regular intervals until the completion of the project. Risk factors can be due to changes in the project requirements, environment, technology, organization, personnel, and unforeseen events. The three levels of risk factors are low, moderate, and high. The manager prioritizes and resolves high−level risk factors first.

System hazards and safety are related to the system risk factor. A hazard is a set of conditions that can lead to an accident in certain environmental conditions. Safety is defined in terms of hazards. For example, software can contribute to system hazards, and they must be eliminated or controlled to reduce or prevent accidents. Decreasing any or all three of the following risk factors can reduce risk:

1. Likelihood of a hazard occurring
2. Likelihood that the hazard will lead to an accident

3. Worst possible potential loss associated with such an accident

During its life cycle, the system can potentially contribute to risk through the effect of the software on system hazards. Therefore the manager should ensure that the software executes within the IT system context without resulting in unacceptable risk. System risk safety involves such factors as identifying hazards, assessing hazards, and controlling hazards risk.

Identification of hazards risk should begin at the earliest stages of system development during concept and requirements definition. The manager should make an initial risk assessment that identifies the safety−critical areas and functions, identifies and evaluates hazards, and identifies the safety design criteria that will be used in the remainder of the system development phases.

Hazard assessment can be viewed as falling along a continuum in terms of severity. One approach establishes a cutoff point on this continuum.

Only the hazards above this point are considered in further system safety procedures. Another approach uses several cutoff points that establish categories of hazards. These categories can be negligible, marginal, serious, or critical when different levels of time and effort are applied.

The manager can introduce hazard control during the system requirements analysis and design techniques to attempt to prevent or minimize the occurrence of a hazard. The objective of system safety requirements analysis is to ensure that the requirements' functionality is specified and implemented and that they are consistent with the safety constraints as shown in Figure 5−1. Ideally, the manager accomplishes this by verifying that the system requirements satisfy the safety constraints and the system correctly implements the requirements.

Figure 5−1: Safety risk constraints

An advantage of safety analysis steps is that the errors are caught earlier and thus are easier and less costly to fix. The safety analysis also mitigates the risk factor. The information from early verification activities can help the manager design safety features into code and provide leverage for the final code verification effort. The verification effort is distributed throughout the system (hardware and software) development process instead of being concentrated at the end. Ideally, each step merely requires that newly added detail does not violate the safety verification of the higher−level abstraction at the previous step. Each level is consistent with the safety−related assumptions made in the analysis at the next higher level. These verification activities may have both formal and informal aspects; static analysis uses formal proofs, technical walk−through, and technical reviews. Dynamics analysis involves various types of testing that provide confidence in the models and the assumptions used in the static analysis.

The first step in any safety verification procedure verifies that the system requirements are consistent with or satisfy the safety constraints. This analysis is important so that the code ensures and satisfies that these requirements will be safe. The analysis also identifies important conflicts and tradeoffs early, before the

manager makes design decisions. The manager must make decisions about tradeoffs between safety and reliability or other system and software qualities, and between safety and functionality, for each project on the basis of potential risk, acceptable risk, liability issues, and requirements.

Risk Management

Risk management is a practice of controlling risk. The risk management practice consists of processes, methods, and tools for managing risks in an IT project before they become problems. Figure 5−2 illustrates a risk management process. Risk management consists of actions taken to identify, assess, and eliminate or reduce risk to an acceptable level in such areas as cost, schedule, technical, and products. The risk management process provides a disciplined environment for the IT project manager to make decisions for those areas. The process includes the following factors:

- Continuously assess what could go wrong with risks.
- Determine which risks are important to deal with.
- Implement strategies to deal with those risks.

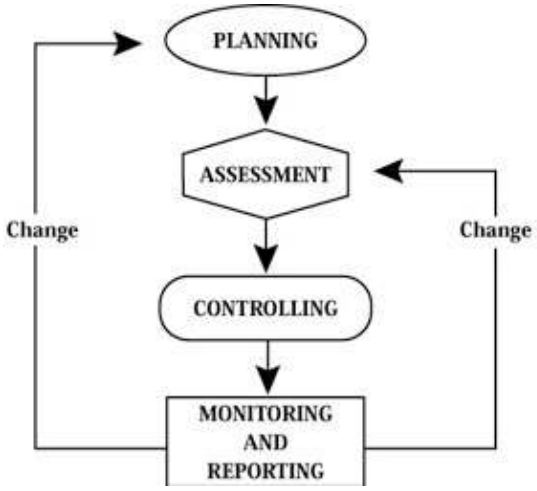


Figure 5−2: Risk management process

In addition, the pressure to reduce costs and improve IT project performance, time to market, and management practices is driving organizations to avoid expensive problems, hence to more effectively manage risk.

The risk management process includes the following four major activities:

1. Risk management plan
2. Risk assessment plan
3. Risk mitigation plan
4. Risk monitoring and reporting plan

Private Risk Management Plan

The risk management plan is an organized, well−structured, comprehensive, and iterative approach for managing risk. The risk management plan records the results of the risk−planning process. The plan includes a set of functions that are identified as continuous activities throughout the life cycle of the project. Each risk

goes through these functions sequentially, but the activity occurs continuously and concurrently. Risks are tracked in parallel while new risks are identified and analyzed. A mitigation plan for one risk may yield another risk throughout the project's life cycle.

The risk management plan includes planning, identification, assessment, mitigation, and continuous tracking with control procedures that feed back to the project manager for making decisions as shown in Figure 5−3. Every management action has some type of risk factor involved. A risk management worksheet is a part of the risk management plan and should be designed to assist the project manger in identifying the overall risk level of the phase or project. The worksheet should provide the following:

- Visibility for the subjective factors that may influence the management in the performance
- Identification and measurement of the potential risk areas early in the commitment process
- Identification of the areas that need special emphasis in planning and control
- Estimation of the degree of risk, which should be conveyed to those concerned

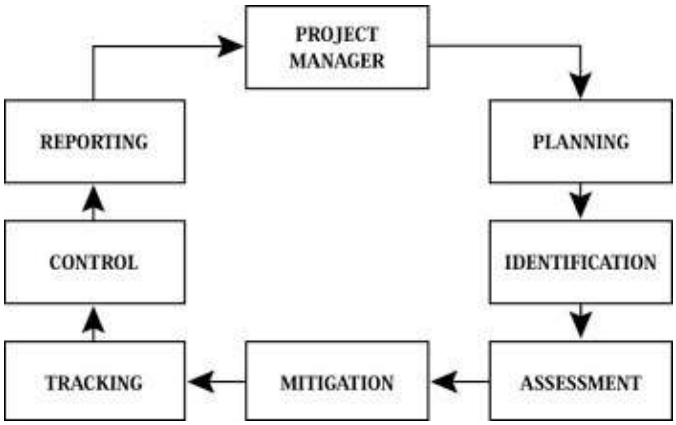


Figure 5−3: Risk management plan activities

The risk management plan includes the following for reducing risk:

- Efficient project management techniques
- Effective communication
- Good control

Larger IT projects (those involving 20 or more practitioners) are at a greater risk of failure than are smaller projects. For example, larger projects need more management involvement, a more structured approach, sufficient decision points and phases, and effective controls. The use of a more structured approach helps the manager review the project after each phase. The larger the project, the greater the need to subdivide it into manageable subprojects, which reduces risk factors. Smaller projects are more manageable because they include more effective management involvement.

Risk Assessment Plan

The risk assessment plan is a process of identifying and analyzing risk and analyzing probability and consequences of identified risks. Risk assessment determines the influence, sensitivity, and relationships of risk and integrates the technical, schedule, and cost assessment. The risk assessment plan starts at the beginning of the project and continues during all phases of the system development and maintenance.

Risk identification is a process of identifying the known and unknown risk events in the project. The location of the risk event can be at various stages in the project. The location can be within WBS and a functional area.

Risk analysis refines the description of an identified risk event. The manager isolates the cause of risk in risk analysis, determines the influence of risk, and determines the type of metrics for high, medium, and low risk as shown in Table 5−1. The manager computes quantitative measures of probability and influence on cost, schedule, and performance.

Table 5−1: Risk Analysis Criteria

High	Moderate	High	High
Probability of occurrence	Low	Moderate	High
Low	Low	Low	Moderate
	Low	**Severity of consequences**	**High**

To measure risk is to quantify risk. If the manager cannot measure risk, he or she cannot manage it. The manager establishes a process to measure risk and manage it. The success of an IT project relies on a measure of risk for budget and scheduling, system development and testing phases, and work force.

The ability to quantify risk is essential to the process of budgeting and scheduling. During the process of completing specified tasks, the project manager must be able to verify the estimates and make sound judgments on the risks of cost overruns and time delays. The project manager should discuss the following questions before making sound decisions:

- Do developers with little experience overestimate or underestimate the complexity of the task because of their experience, assumptions they make, models they select, and how they define the model parameters?
- What are the sources of risk associated with project cost estimation? How can such risk be quantified?

Risk measurement process should include the following:

- Constructing the probability density functions
- Probing the sources of risks and uncertainties
- Analyzing and regarding the likelihood of technical and nontechnical risks
- Drawing conclusions on the basis of the accumulated evidence and ultimately selecting the contractors most likely to complete the project without major cost overruns or time delays

Risk Mitigation Plan

The risk mitigation plan determines the project's success or failure. This plan includes various techniques, technology demonstration, prototyping, and test and evaluation. A good mitigation plan includes the following:

- Search for and locate risks before they become problems.
- Transform risk data into decision−making information.
- Evaluate influence, probability, and time frame.
- Classify and prioritize risks.
- Translate risk information into decisions and mitigating actions, both present and future, and implement those actions.
- Monitor risk indicators and mitigation actions.
- Mitigate high and moderate risks.

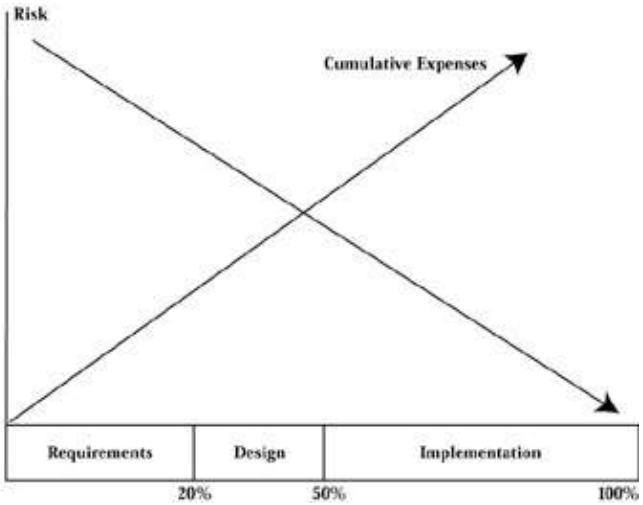
The risks are greater in developing a new IT system than in maintaining or converting an exiting system. In new system development, minimization of risks consists of the following:

- Identification of the critical risk elements of a project
- Development of an action plan to anticipate potential problems

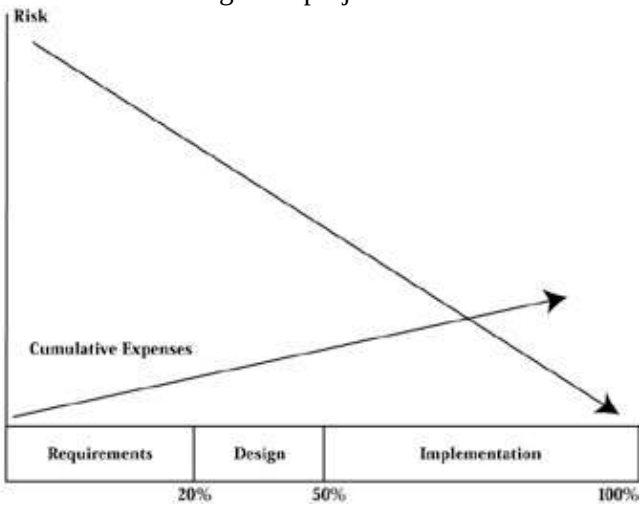
The mitigation plan contains plan−of−action guidelines as follows:

- Emphasize the selection of a methodology.
- Plan the construction of a prototype.
- Provide proper technical training.
- Include tasks and activities.
- Implement the result in each phase.

The risk level decreases as the project moves on in each phase. Figure 5−4, *A,* shows a well−managed project if the risk and the cumulative cost intersect at the middle during the design phase. An unsuccessful project is illustrated in Figure 5−4, *B,* which shows that the intersection has gone beyond the implementation phase; this project is at high risk of failure. The project manager should take command and bring the project back to the successful path by identifying, resolving, and controlling risk factors.



5−4 A: A well−managed IT project



5−4 B: A poorly managed IT project

In addition, the manager needs to create and implement a continuous process for the effective management of risk. The process includes all interested parties in the project (e.g., developers, customers, users, and stakeholders).

The risk mitigation plan contains metrics as follows to reduce and control risk:

- Planned versus actual computer resources used
- Planned versus actual work force used
- Planned versus actual practitioner turnover
- Number of requirement changes versus original requirements
- System progress showing number of units designed, reused, tested, and integrated
- Cost and schedule (budget versus actual)

Risk Monitoring and Reporting Plan

The risk monitoring and reporting plan consists of systematically tracking and evaluating performance of identified risk areas and events against established metrics. This plan includes cost, schedule, and performance measurement reports. Figure 5−5 shows the risk monitoring process.

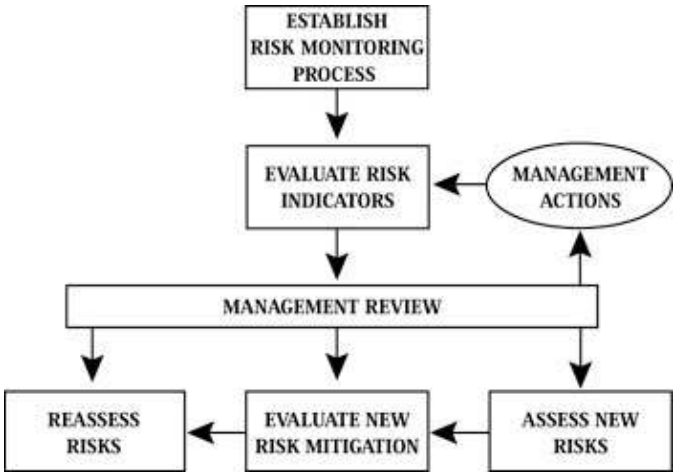


Figure 5−5: Risk−monitoring process

Risk monitoring results and reporting are properly documented for future reference. This document also contains records of risk assessments, risk mitigation, risk analysis, rationale for decisions, and updates.

Principal Best Practices

Successful system software projects use principal best practices. The Airline Software Council identified the following nine principal best practices:

1. Formal risk management
2. Agreement on interfaces
3. Formal inspection
4. Metric−based scheduling and management
5. Binary quality gates at the inch−pebble level
6. Project−wide visibility of progress versus plan
7. Defect tracking against quality targets
8. Configuration management
9. People−aware management accountability

Formal Risk Management

Formal risk management is vital to the success of an IT project. A formal risk management process requires that an organization accept risk as a major consideration for system software project management, commitment of resources, and methods for identifying, monitoring, and managing risk.

The project manager assigns duties to a practitioner as a risk officer and sets up a risk database. The risk officer reports to the project manager if a risk assessment has an influence on the project plan and decision. He or she updates the risk assessment consistent with decision updates during the project. The project manager establishes objective criteria to identify, evaluate, and manage risk. The risk officer verifies that the information flow pattern and reward criteria within the organization support the identification of risk by all project practitioners.

The project manager observes that the integrity of information is properly managed and controlled. The risk officer draws up a risk profile for each risk and regularly updates the risk's probability of occurrence, consequences, severity, and delay. The risk officer confirms that the risk management plan contains explicit provisions to alert the project manager of imminent risks. The project manager encourages all personnel to identify risks and report them to the risk officer.

Agreement on Interfaces

The project manager must complete agreement on interfaces before the implementation phases and maintain such user interfaces as an integral part of the system specification. The Airline Software Council proposed that for those projects developing both hardware and software, the manager must write a separate software specification with an explicit and complete interface description.

The project manager ensures a complete census of input and outputs, which are defined down to the data element level. The interfaces are stable and include hardware, software, users, and major software components. The system specification includes a separate software specification to show the hardware interfaces. The users should verify the interface descriptions as they develop.

Formal Inspection

The formal inspection process identifies and implements the quality of the system products. The project manager integrates the formal inspection process in the project schedule. The manager should conduct formal inspection during all phases of system development and maintenance, including the requirement, architecture, design, implementation, and testing phases.

The project manager establishes procedures, standards, and rules for the conduct of inspections. He or she uses metrics to gauge the effectiveness of inspections. A documented process exists to conduct inspection. The manager properly documents and tracks defects found during inspections and ensures that all deliveries are inspected for quality before they are released for project use.

Metric−Based Scheduling and Management

Metric−based scheduling and management includes statistical quality control of costs and schedule. This requires early calculation of size metrics, projection of costs and schedules from empirical patterns, and tracking of project status through the use of captured result metrics.

The project manager checks that the cost and schedule performance track against the initial baseline and the latest baseline. The manager tracks the number of changes to the initial cost and schedule baseline and refines the project estimates continuously as the project proceeds. The productivity levels and schedule deadlines are evaluated against past performance and reflected in the risk assessment. The manager monitors the planned versus actual cost and planned versus actual schedule.

Binary Quality Gates at the Inch−Pebble Level

Binary quality gates at the inch−pebble level is the completion of each task, and the phase in the lowest level activity network needs to be defined by an objective binary indication. These completion events should be in the form of gates that assess either the quality of the products produced or the adequacy and completeness of the finished process. Gates may take the form of technical reviews or completion of a specific set of tests, which integrate or qualify system software components, demonstrations, or project audits. The binary indication is meeting a predefined performance standard. The manager can set the standard as a defect density of less than 4% function point. Activities are closed only upon satisfying the standard, with no partial credit given. The manager can apply quality gates at any time during the project.

The project manager checks that the project status and planning estimates have been produced based on the inch−pebble quality gates that can be aggregated at any desirable level. The project manager ensures the following:

- All activities have been decomposed into inch−pebbles.
- All near−term work has been decomposed into tasks no longer than 2 weeks in duration.
- Achievable accomplishment criteria have been identified for each task.
- Tasks are based on overall quality goals and criteria for the project.
- The planned tasks are 100% complete before acceptance.
- All reviews are successfully completed.
- The inch−pebble tasks on the critical path are defined, enabling more accurate assessment of schedule risks and contingency plans.
- The set of binary quality gates is compatible with the WBS.

Project−Wide Visibility of Progress Versus Plan

The project participants should know project−wide visibility of progress versus plan. The project manager establishes an anonymous channel feedback and encourages practitioners to provide bad news up and down the project hierarchy. The project manager ensures that all project personnel know the basic project status. The practitioners can report good and bad as well. The project goals, plans, schedules, and risks are available to the members of the project. The anonymous channel feedback is visible to the project members.

Defect Tracking Against Quality Targets

Defect tracking against quality targets is a process of formally tracking the defects at each phase and activity of the project. Configuration management (CM) enables each defect to be recorded and traced through to removal. The manager compares initial quality targets and calculations of remaining or latent defects with counts of defects removed to track progress during testing activities.

The project manager establishes defect targets for the project and defines consequences if a product fails to meet the targets. The project quality targets apply to all products. The project manager defines circumstances under which quality targets are subject to revision and establishes the techniques that are used to project latent defect counts. The manager also develops techniques to confirm the current projected level of removal defects

as adequate to achieve planned quality targets. The test coverage is sufficient to indicate that the latent defect level achieved by the end of testing will be lower than the established quality targets.

The project manager checks that the inspection and test techniques employed during the project are effective in meeting quality targets. All discovered defects undergo CM, and the manager performs accurate counts for defects discovered and defects removed. A closed−loop system links defect actions from when defects are first detected to when they are resolved. The defect information is defined at a level of granularity that supports an objective assessment of resolution on a periodic basis.

Configuration Management

CM is a discipline that is vital for the success of an IT project. CM is an integrated process for identifying, documenting, monitoring, evaluating, controlling, and approving all changes made during the life cycle of the system. The CM information is shared by more than one individual or organization.

The project manager integrates the CM process with the project plan. All versions are controlled by CM. The manager uses configuration control tools for status accounting and configuration identification tracking. The manager checks that periodical reviews and audits are in place to assess the effectiveness of the CM process. All information shared by two or more organizations is placed under CM.

People−Aware Management Accountability

People−aware management accountability is a process that the project manager establishes for staffing qualified people and fostering an environment conducive to high morale and low voluntary staff turnover. Qualified people have domain knowledge and similar experience in previously successful projects.

The project manager ensures that domain experts are available for the project. The team members are fully aware of their roles in the project. The opportunities for professional growth are available to all members of the project. The practitioners believe in the goals of the project and that the schedule is feasible. The project manager ensures that the motivation and retention of the team members are key factors in the success of the project.

Risk Management Checklist

Risk can be due to any changes in the requirements or the occurrence of an unforeseen event. The checklist is summarized as follows:

- Risk assessment

  Risk identification

  Analysis

  Decision

  Risk analysis

  Identification of risk factors

  Cost and performance analysis

Risk prioritization

Computation for risk reduction

- Risk control

Risk reduction

Risk avoidance

Risk transfer

Risk management plan

Process establishment

Preparation of plan to handle risk

Risk resolution

Risk mitigation

Risk monitoring

Risk reporting

Risk evaluation

- Boehm's 10 risk items

Personnel shortfalls

Unrealistic schedules and budgets

Development of the wrong software functions

Development of the wrong user interface

Gold−plating (requirements scrubbing)

Continuing stream of requirements changes

Shortfalls in externally performed tasks

Shortfalls in externally furnished components

Real−time performance shortfall

Straining computer science capabilities

- Head−start factors for risks in an IT project

Significant changes in the organization

Frequent changes in roles and duties of team members

Change of management during the project's development

Significant changes in requirements

Varying requirements during the project's development

Changes in specifications

Lack of senior management support

Shrinking project budget

Changes in project schedule

High practitioner turnover rate

- Risk management best practices

Establish management reserves for risk resolution

Implement metric−based risk decisions

Perform continuous risk management

Formalize risk tracking and review

Manage influence of external dependencies