Unit -5-Risk Management

Introduction:

Risk management is the process of identifying, evaluating, and managing the effects of uncertainty on objectives. It applies to both positive and negative outcomes, such as opportunities and threats1. Risk management can be applied to various domains, such as finance, law, strategy, and security. The goal of risk management is to minimize, monitor, and control the potential losses or maximize the potential gains from investing or operating.

The term risk is defined as the potential future harm that may arise due to some present actions as explained in Wikipedia. Risk management in software engineering is related to the various future harms that could be possible on the software due to some minor or non-noticeable mistakes in software development project or process. "Software projects have a high probability of failure so effective software development means dealing with risks adequately. Risk management is the most important issue involved in software project development. This issue is generally managed by Software Project Management (SPM). During the life cycle of software projects, various risks are associated with them. These risks in the software project were identified and managed by software risk management which is a part of SPM. Some of the important aspects of risk management in software engineering are software risk management, risk classification and strategies for risk management.

Risk management process:

All risk management processes follow the same basic steps, although sometimes different jargon is used to describe these steps. Together these 5 risk management process steps combine to deliver a simple and effective risk management process.

Step 1: Identify the Risk. You and your team uncover, recognize, and describe risks that might affect your project or its outcomes. There are several techniques you can use to find project risks. During this step you start to prepare your Project Risk Register.

Step 2: Analyze the risk. Once risks are identified you determine the likelihood and consequence of each risk. You develop an understanding of the nature of the risk and its potential to affect project goals and objectives. This information is also input to your Project Risk Register.

Step 3: Evaluate or Rank the Risk. You evaluate or rank the risk by determining the risk magnitude, which is the combination of likelihood and consequence. You make decisions about whether the risk is acceptable or whether it is serious enough to warrant treatment. These risk rankings are also added to your Project Risk Register.

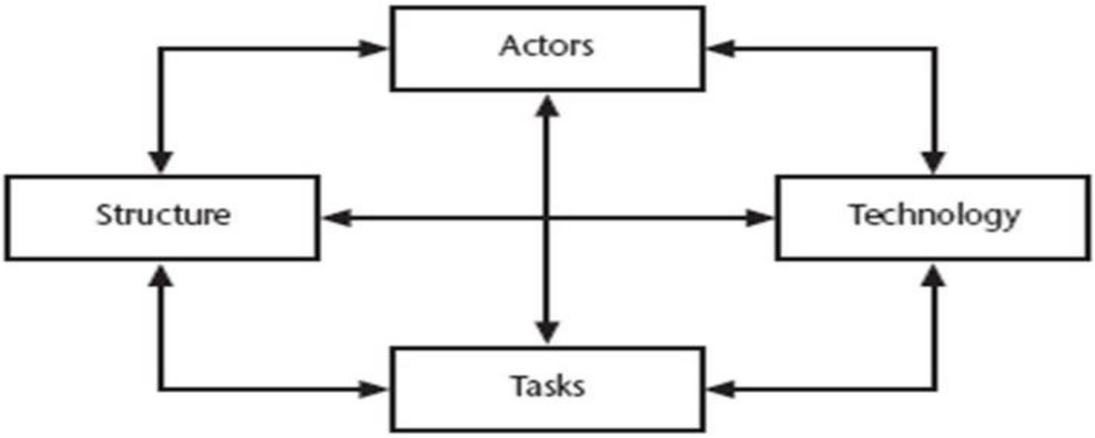
Step 4: Treat the Risk. This is also referred to as Risk Response Planning. During this step you assess your highest ranked risks and set out a plan to treat or modify these risks to achieve acceptable risk levels. How can you minimize the probability of negative risks as well as enhancing the opportunities? You create risk mitigation strategies, preventive plans, and contingency plans in this step. And you add the risk treatment measures for the highest ranking or most serious risks to your Project Risk Register.

Step 5: Monitor and review the risk. This is the step where you take your Project Risk Register and use it to monitor, track and review risks.

Boehm's Top 10 Risks

Risk	Area	Risk reduction techniques
1. Personnel Shortfalls	SPM: Managing People	Staffing with top talent; job matching; team building; training and career development; early scheduling of key personnel
2. Unrealistic time and cost estimates	SPM: Effort estimation	Multiple estimation techniques, design to cost, incrementaldevelopment, recording and analysis of past projects, standardization of methods
3. Developing the wrong software functions	Requirements	Improved software evaluation, formal specification methods, user Surveys, prototyping; early user manuals
4. Developing Wrong user interface	Requirements	Prototyping; task analysis; user involvement
5. Gold plating	Requirements, design	Requirements scrubbing, prototyping,design to cost
6. Late changes to requirements	Requirements, development model	Change control, incremental development
7. Shortfalls in externally supplied components	Quality Assurance/ Verification of Resources	Benchmarking, inspections, formal specifications, contractualagreements, quality controls
8. Shortfalls in externally performed task	Quality Assurance, software Process	Quality assurance procedures, competitive design etc.
9. Real time performance problems	Implementation, Verification	Simulation, prototyping, tuning
10. Development technically too difficult	Design, Implementation	Technical analysis, cost-benefit analysis, prototyping, training

Categories of Risk



Actors: Actors refer to all the people involved in the development of the application in question. These include the various development specialists, the different user group, and managers with differing responsibilities.A typical risk in this area is that high staff turnover leads to information of value to the project being lost. For example, if a software developer builds a software component and then leaves before it has been fullytested, the team member taking over the component might find that their lack of familiarity with the software makes diagnosis and correction of faults difficult.

Technology: Technology encompasses both the technology used to implement the application and that embedded in the delivered products. Risk here could relate to the appropriateness of the technologies and to possible faults within them, especiallyif they are novel.

Structure: Structure describes the management structures and systems including those affecting planning and control. For example, the implementation will need the users to carry out some tasks, but the responsibilityfor managing the users' contribution to the project might not have been clearly allocated.

Tasks: The means the work to be carried out. For example, the complexity of the work might lead to delays because of the additional time required to co-ordinate and integrate the large number of different elements.

Types of Risk

The various categories of risks associated with software project management are enumerated below.

1. Schedule / Time-Related / Delivery Related Planning Risks
2. Budget / Financial Risks
3. Operational / Procedural Risks
4. Technical / Functional / Performance Risks
5. Other Unavoidable Risks

1. Schedule / Time-Related / Delivery Related Planning Risks

These risks are related to running behind schedule and are essential time-related risks, which directly impact the delivery of the project.

Some of the reasons for such risks are:

- Incorrect Time Estimation, and consequently an incorrect project schedule
- Improper Resource Allocation
- Underutilization of Resources
- Superficial Understanding of Project Complexities
- Unexpected Expansion of Project Scope
- Incorrect time estimation may occur because activities may have external dependencies such as client approvals, subcontractors etc. and a delay in a critical path activity has a cascading effect on the entire project.
- Resource Allocation may be improper / Underutilization of resources may take place, especially if resources are shared between projects.
- A silo approach of members in various teams in a project may lead to an isolated superficial understanding of project complexities which may result in delays in subsequent stages e.g. when different development teams work on various aspects of a software project and run into issues during system/integration testing.

2. Budget / Financial Risks

These are the monetary risks which are associated with budget overruns. Some of the reasons for such risks are:

- Improper Budget Estimation
- Cost Overruns due to underutilization of resources
- Expansion of Project Scope
- Improper Tracking of Finances
- Underutilization of resources especially happens when resources are shared between

projects because it becomes difficult to effectively manage such resources and a certain amount of productivity may go waste.

- Further, unexpected expansion of project scope (due to addition of features by clients, etc.) may lead to budget overruns as such expansions may not have been factored into the original estimates.
- Delay of projects may also have certain penalty costs associated with it e.g. construction projects.

3. Operational / Procedural Risks

These are risks which are associated with the day-to-day operational activities of the project. These could be due to any of the below reasons:

- Improper Process Implementation
- Silo approach followed by software development teams leading to conflicts.
- Conflicting Priorities
- Lack of conflict resolution / team spirit
- Lack of clarity in responsibilities
- Breakdown in communications
- Lack of sufficient training

Effective team communication is an essential part of project management and in people-intensive projects such as software projects, there is a strong need for an established communication structure, a setup for escalation, a conflict resolution process, established project priorities and above all, the employees need to be trained in making use of these processes within the organization.

4. Technical / Functional / Performance Risks

These are technical risks associated with the functionality of the software or with respect to the software performance.

- To compensate for excessive budget overruns and schedule overruns, companies sometimes reduce the functionality of the software.
- Software testing is a downstream stage in the software development lifecycle and as the project falls behind schedule, downstream activity times are shrunk to meet delivery dates which results in insufficient software testing.
- Further, developers face a constant trade-off between achieving maximum functionality of the software (in terms of software features) and peak performance (maximum speed and quick response time by minimizing and eliminating unnecessary frills from the software)

To maintain the sanctity of the software development process, while simultaneously catering to the customer's needs, a mutually agreed-upon cut-off date should be determined, beyond which "expected software functionality" would be frozen and any further requirements would be handled in subsequent software's releases.

5. Other Unavoidable Risks

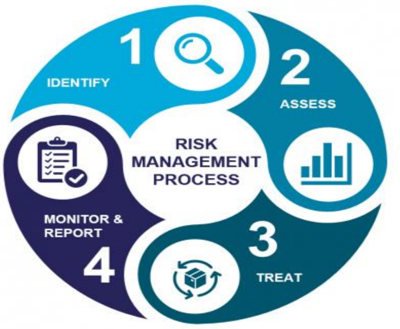
All the risks described above are those which can be anticipated to a certain extent and planned for in advance. However, there are certain risks which are unavoidable in nature. The reasons for such unavoidable risks are described below.

- Changes in government policy
- Obsolescence of software due to new technology from a rival company
- Loss of contracts due to changes at customers end

Although these risks are broadly unavoidable, an organization may anticipate and thereby reduce the impact of such risks by

- Keeping abreast with changes in government policy
- Monitoring the competition
- Catering to the needs of the customer and ensuring customer satisfaction

Risk Management involves following processes:



1. Software Risk Identification
2. Software Risk Analysis
3. Software Risk Planning
4. Software Risk Monitoring

1. Software Risk Identification

Risk identification is the first step in risk management. We need to identify both project and product risk by using certain techniques. Some of the most common techniques which can be applied to identify different risks are using risk templates, interviewing the stakeholders, project retrospectives etc.

To identify the risks that your project may be subjected to, it is importantto first study the problems faced byprevious projects. Study the project plan properly and check for all the possible areasthat are vulnerable to some or the other typeof risks. The best ways of analyzing a project plan are by converting it to a flowchart and examining all essential areas. It is important to conduct a fewbrainstorming sessions to identify the known unknowns that can affect the project. Any decision taken related to technical, operational, political, legal, social, internal, or external factors should be evaluated properly.

In this phase of Risk management, you must define processes that are important for risk identification. All the details of the risk such as unique Id, date on which it was identified, description and so on should be clearly mentioned.

2. Software Risk Analysis

Software risk analysis is the process of identifying and assessing potential risks and challenges associated with the development, deployment, and maintenance of software systems. This process involves identifying potential sources of risk, evaluating the likelihood and impact of those risks, and developing strategies to mitigate or manage those risks.

The simplest way to understand software risk analysis is to view it as a framework or technique employed to counteract issues during software development within an organization. It focuses on finding the problems that could cause a project to fail and categorizing them as risks.

Ultimately, the goal of software risk analysis is to ensure that software systems are delivered on time, within budget, and with the desired quality.

Software Risk analysis is a very important aspect of risk management. In this phase the risk is identified and then categorized. After the categorization of risk, the level, likelihood (percentage) and impact of the risk is analyzed. Likelihood is defined in percentage after examining what are the chances of risk to occur due to various technical conditions. These technical conditions can be:

- Complexity of the technology
- Technical knowledge possessed by the testing team.
- Conflicts within the team
- Teams being distributed over a large geographical area.
- Usage of poor-quality testing tools

Steps of software risk analysis

When we seek to combat the risks associated with this line of work, it is best to structure our approach and develop a process for teams to follow. This enables them to manage risks effectively while assessing issues as they arise, allowing them to complete projects successfully.

1. **Define your scope:** Identify the software project's scope, including its objectives, requirements, stakeholders, and constraints.
2. **Identify the risks:** Identify potential risks that could impact the success of your project. This can be done by reviewing past projects, consulting with stakeholders, and analyzing industry standards and best practices.
3. **Assess their impact**: Assessing the impact of potential risk is the next step. Once you know what problems may affect your project, you will need to figure out how these issues could affect you if they did indeed come to pass.
4. **Prioritize risks**: There is no way to cut out risks entirely, but with software risk analysis, you can prioritize issues ahead of time so that adequate measures are in place to deal with problems as soon as they surface.
5. **Implement justification strategies**: Risk mitigation is crucial for software risk analysis. To properly shield your project from disaster, you will need to create the necessary mitigation strategies to help you quickly bounce back in the event of a severe issue.
6. **Monitor and make revisions**: Once you have established a way to monitor the current state of risk in your software risk analysis, it is crucial to keep it updated. Revisions are a key aspect of maintaining an effective mitigation strategy, so updating your approach is essential as your project evolves.

Level of risk is identified with the help of:

i. **Qualitative Risk Analysis:** Here you define risk as:
   - High
   - Low
   - Medium

ii. **Quantitative Risk Analysis:** can be used for software risk analysis but is considered inappropriate because risk level is defined in % which does not give a very clear picture.

3. Software Risk Planning

Risk planning is the process of identifying, prioritizing, and managing risk. Every project or initiative has objectives, that is, goals that it seeks to accomplish. These are often called Critical Success Factors (CSF). Risk events threaten the successful completion of these critical success factors. Thus, risk planning involves identifying the most important risk events in advance, prioritizing them, and developing the appropriate risk response plans. There are three steps to risk planning:

i. Identifying Risks
ii. Prioritizing Risks
iii. Determining Response Plans

i. Identifying Risks:

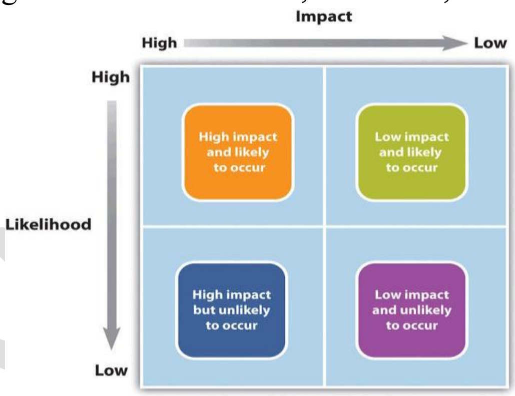
A strong risk identification process is important to the successful completion of the critical success factors. This is particularly true for large or inherently risky projects, like nuclear power plants. But if it's beneficial for large projects, an appropriately sized risk planning process will benefit small projects too. A Risk Management Plan is prepared which includes items such as:

- Risk Register
- Risk Breakdown Structure
- Risk Analysis

ii. Prioritizing Risks

Identifying risks to a project’s success is a great first step that would benefit most projects that I’ve seen. But to create a strong risk management plan, those risks must be analyzed and prioritized to determine which require the project manager’s time and attention, how often, and what resources are required.

Stakeholders can be sensitive to issues the project manager considers minor. Some stakeholders seem to demand excessive communication requirements. Prioritizing risks ensures that stakeholders recognize the importance placed on their areas of concern which goes a long way toward placating them. Each of these factors should be prioritized. The scale is not important, but it is often 1-10, low-medium-high, or a similar scale.


Impact
High
Low
High
Likelihood
Low
High impact and likely to occur
Low impact and likely to occur
High impact but unlikely to occur
Low impact and unlikely to occur

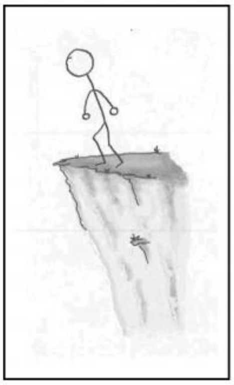
Since risk has two components ⇨ *Risk = Probability* × *Impact*

iii. Determine Risk Response Plans

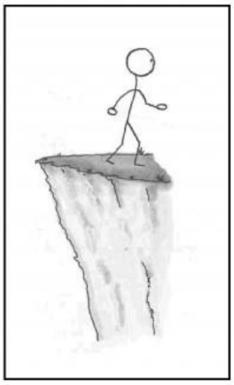
The final piece of information that completes the risk register is a risk response plan. Now that you’ve identified the triggers that allow you to quickly identify when a risk has occurred (or is occurring), the response plan gives you a head start in the response. Some responses occur at the beginning of the project (when the risk planning process is taking place) and others occur when the risk event occurs. Still others occur at any applicable time during the project.

They must contain an appropriate level of detail. For major risks a good action plan is necessary in advance and could warrant its own write-up. For medium risks a small action plan could be placed within the risk register, and for small risks there could be no action plan at all. It isn’t a necessity for all risks, but it is important to have one for the most important ones.

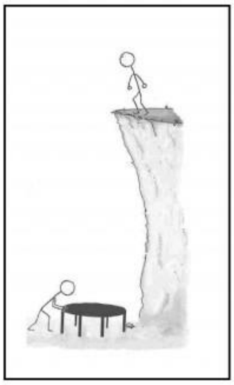
There are four possible responses to risk events:



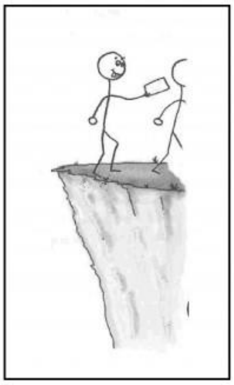
Your project



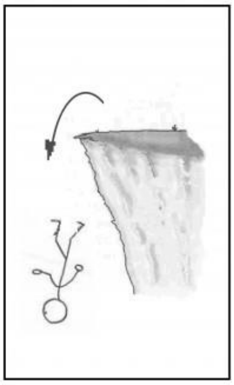
Avoid



Mitigate



Transfer



Accept

1. **Avoid**: Eliminate the threat. For example, change the scope of the project, spin off a certain business unit, or change the objectives that the risk event is threatening.

2. **Transfer:** Off-load the risk to a third party. For example, buy insurance, issue a performance bond, or change the contract from a lump sum to a unit price (or vice versa).

3. **Mitigate:** Reduce the probability or impact of the risk event. For example, cover the project area to prevent work stoppages due to inclement weather, or purchase materials in advance to ensure they can be returned without threatening the project completion date.

4. **Accept:** Sometimes there is no other alternative than to proceed with the project and accept the risk. But producing documentation, holding meetings, and communicating the risk with stakeholders can go a long way toward minimizing the damage.

4. Software Risk Monitoring

Software risk monitoring is integrated into project activities and regular checks are conducted on top risks. Software risk monitoring comprises of:

- Tracking of risk plans for any major changes in actual plan, attribute, etc.
- Preparation of status reports for project management.
- Review risks and risks whose impact or likelihood has reached the lowest possible level should be closed.
- Regularly search for new

Software Project Management Tools

1. **Planning Tools:** There are several very popular planning tools available for project management planning. Microsoft offers a popular product called Microsoft Project. There is a similar program available that is offered for free. It is Niku's Open Workbench. Primavera offers Primavera Project Planner and Inter Plan systems offers ATC Professional for use in project planning.
2. **Estimation Tools:** Estimation is an important aspect of project management. Like planning, there are numerous tools available to help with estimation. The most important tool for estimation is experience, which cannot be bought or installed.
3. **Risk Management Tools:** Risk management is extremely important in software project management. Dr. Dannelly stated in class that it is easier and cheaper to detect and fix the errors at the beginning of the project than to find them at the end of the project. Detecting risks and planning for them is extremely important in planning costs. Some of the tools to help with this include Risk Radar by ICE, Inc., Risk Trak, and CORA (Cost-Of-Risk-Analysis) by International Security Technology, Inc.
4. **Resource Management Tools:** Resource management is important to project management in many areas and tasks. Managing personnel between several projects and tasks is vital in having a productive and profitable project plan. Some of the tools available that help with resource management are Frog Point, Tracker Office, and Time Scope.
5. **Change Management Tools:** Change management tools are important in keeping track of versions and changes in projects relating to software development. Managers must keep track of components and milestone changes. Two change management tools that help keep track of these changes are Track-IT Pro Suite by DOVICO Inc. etc.

Risk Management Framework

The Risk Management Framework is a template and guideline used by companies to identify, eliminate, and minimize risks. It was originally developed by the National Institute of Standards and Technology to help protect the information systems of the United States government.

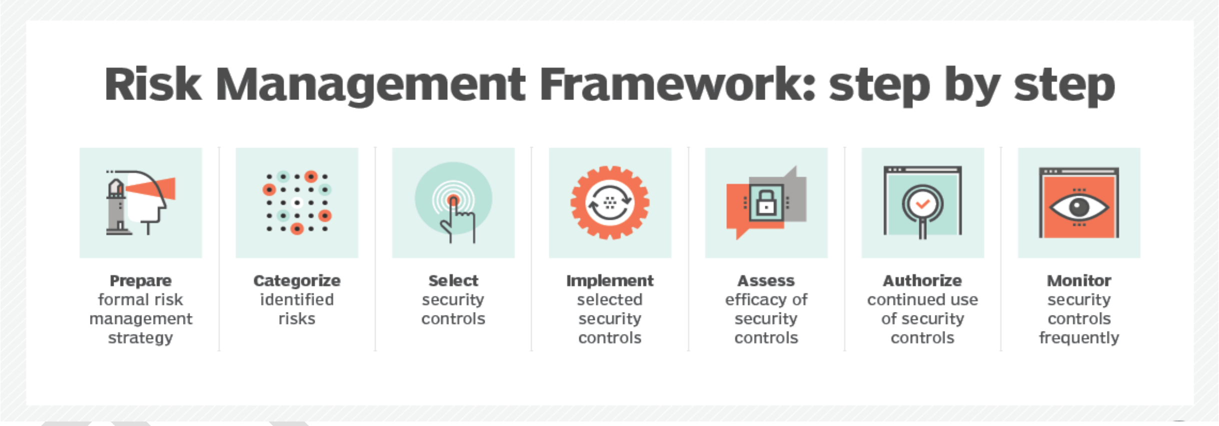
The RMF was initially designed for use by federal agencies but can be easily adopted by organizations operating in the private sector. Businesses cannot exist without exposing themselves to risks such as IT problems, litigation, and loss of capital. While it is impossible to eliminate all risks involved in running a business, they can be minimized.

A Risk Management Framework (RMF) establishes principles and guidelines to which an organization must follow to effectively manage risks. (Without one, you're not managing risks efficiently, if at all.)

Key features & components in RMFs:

- ➢ Risk identification
- ➢ Risk identification
- ➢ Risk mitigation
- ➢ Risk monitoring & reporting

Steps of the Risk Management Framework



Risk Management Framework: step by step

Prepare formal risk management strategy

Categorize identified risks

Select security controls

Implement selected security controls

Assess efficacy of security controls

Authorize continued use of security controls

Monitor security controls frequently

Prepare: The preparation stage of the RMF focuses on getting the organization ready to adopt a formalized risk management strategy. This might include identifying organizational risks and determining key risk-management roles.

Categorize: The categorized stage is where organizations begin assessing the risks that have been identified. This may mean assessing the impact of the various risks and prioritizing the risks that need to be addressed.

Select: The select stage involves choosing the controls that will be used to protect affected systems to minimize or mitigate the risks that have been identified. These controls will vary widely from one system to the next. They may include anything from adopting monitoring solutions to shaping policies that will help to alleviate concerns.

Implement: Once an organization has selected the solutions it will be adopting as part of its risk mitigation strategy, the next stage is implementation. This is where the selected controls are put into place to head off risks that might exist.

Assess: The assessment stage comes after implementation of any selected solutions. It seeks to determine whether the selected controls were implemented correctly and if those controls are delivering the desired result. This means making sure any mechanisms that have been implemented are reducing risks in a quantifiable way without accidentally introducing new risks in the process.

Authorize: In some instances, the authorized stage is tied to executive approval of the risk mitigation mechanisms that have been put into place. More often, however, the authorize phase is more of an overview by senior members of the organization who are looking to make sure that risk mitigation strategies are working and that those strategies adhere to any applicable laws and policies that may exist within the organization.

Monitor: The monitor phase is designed to provide situational awareness on an ongoing basis. Organizations should continuously evaluate their risk mitigation strategies to ensure they continue to work as intended.