Thakur Educational Trust's (Regd.)

THAKUR RAMNARAYAN COLLEGE OF ARTS & COMMERCE

ISO 21001:2018 Certified

TRCAC

PROGRAMME: B.Sc (I.T)

CLASS: S.Y.B.Sc (I.T)

SUBJECT NAME: SOFTWARE ENGINEERING

SEMESTER: IV

FACULTY NAME: Ms. SMRITI DUBEY

UNIT III

Chapter 3 – Project Management

TRCAC

Concepts:

Introduction

Types of Project Plans

Management Activities

Risk Management

Project Scheduling

Managing People

Teamwork

Introduction

Software project management is an essential part of software engineering. Projects need to be managed because professional software engineering is always subject to organizational budget and schedule constraints. The project manager's job is to ensure that the software project meets and overcomes these constraints as well as delivering high-quality software.

The success criteria for project management obviously vary from project to project but, for most projects, important goals are:

1. Deliver the software to the customer at the agreed time.
2. Keep overall costs within budget.

3. Deliver software that meets the customer's expectations.
4. Maintain a happy and well-functioning development team

These goals are not unique to software engineering but are the goals of all engineering projects. However, software engineering is different from other types of engineering in a number of ways that make software management particularly challenging. Some of these differences are:

1. **The product is intangible:** A manager of a shipbuilding or a civil engineering project can see the product being developed. If a schedule slips, the effect on the product is visible—parts of the structure are obviously unfinished. Software is intangible. It cannot be seen or touched. Software project managers cannot see progress by simply looking at the artifact that is being constructed. Rather, they rely on others to produce evidence that they can use to review the progress of the work.

2. **Large software projects are often 'one-off' projects:** Large software projects are usually different in some ways from previous projects. Therefore, even managers who have a large body of previous experience may find it difficult to anticipate problems. Furthermore, rapid technological changes in computers and communications can make a manager's experience obsolete

3. **Software processes are variable and organization-specific :** The engineering process for some types of system, such as bridges and buildings, is well understood. However, software processes vary quite significantly from one organization to another. Although there has been significant progress in process standardization and improvement, we still cannot reliably predict when a particular software process is likely to lead to development problems. This is especially true when the software project is part of a wider systems engineering project.

Types of Project Plans

Effective management of a software project depends on thoroughly planning the progress of the project. Managers must anticipate problems that might arise and prepare tentative solutions to those problems.

A plan drawn up at the start of the project should be used as a driver of the project. This initial plan should be the best possible plan given the available information. It evolves as the project progresses and better information becomes available. Along with project plan, managers may also have to draw up other types of plans. They are:

1.Quality plan: Describes the quality procedures and standards that will be used in the project.

TRCAC

2.Validation plan: Describes the resources and schedule used for system validation.

3.Configuration management plan: Includes configuration management standards and procedure.

4. Maintenance plan: Predicts maintenance costs and efforts required.

5.Staff development plan: Describes how the skills and experience of the project team members will be developed.

Management Activities

Software systems are often new and technically innovative. Engineering projects (such as new transport systems) that are innovative often also have schedule problems. It is impossible to write a standard job description for a software project manager.

The job varies tremendously depending on the organization and the software product being developed. However, most managers take responsibility at some stage for some or all of the following activities:

1. Proposal writing: The first stage in a software project may involve writing a proposal to win a contract to carry out an item of work. The proposal describes the objectives of the project and how it will be carried out. It usually includes cost and schedule estimates and justifies why the project contract should be awarded to a particular organization or team. Proposal writing is a critical task as the survival of many software companies depends on having enough proposals accepted and contracts awarded. There can be no set guidelines for this task; proposal writing is a skill that you acquire through practice and experience.

2. Project planning: Project managers are responsible for planning, estimating and scheduling project development, and assigning people to tasks. They supervise the work to ensure that it is carried out to the required standards and monitor progress to check that the development is on time and within budget.

3. **Risk management**: Project managers have to assess the risks that may affect a project, monitor these risks, and take action when problems arise.

TPCAC

4. **People management**: Project managers are responsible for managing a team of people. They have to choose people for their team and establish ways of working that lead to effective team performance.

5.**Reporting:** Project managers are usually responsible for reporting on the progress of a project to customers and to the managers of the company developing the software. They have to be able to communicate at a range of levels, from detailed technical information to management summaries. They have to write concise, coherent documents that abstract critical information from detailed project reports. They must be able to present this information during progress reviews.

Risk management

Risk management is one of the most important jobs for a project manager. Risk management involves anticipating risks that might affect the project schedule or the quality of the software being developed, and then taking action to avoid these risks. Risks may threaten the project, the software that is being developed, or the organization. There are, therefore, three related categories of risk:

1. Project risks: Risks that affect the project schedule or resources. An example of a project risk is the loss of an experienced designer. Finding a replacement designer with appropriate skills and experience may take a long time and, consequently, the software design will take longer to complete.

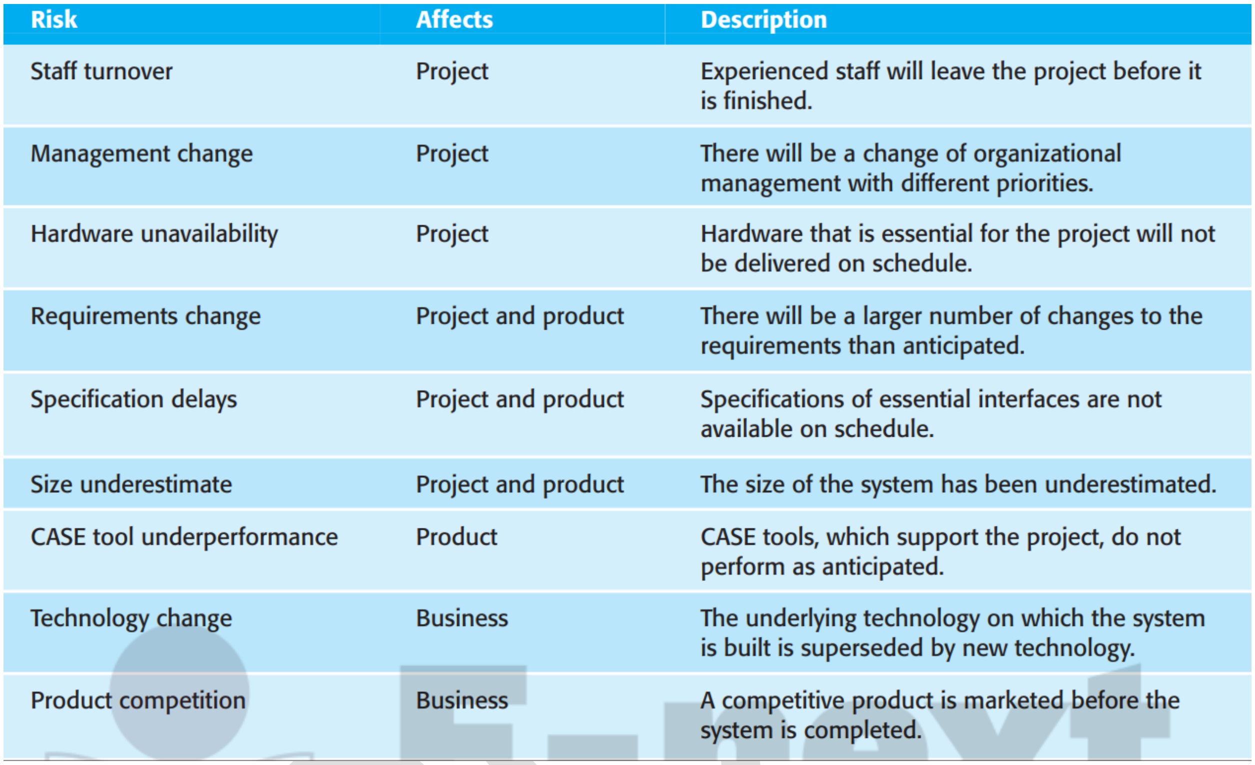
2. Product risks: Risks that affect the quality or performance of the software being developed. An example of a product risk is the failure of a purchased component to perform as expected. This may affect the overall performance of the system so that it is slower than expected.

3. Business risks: Risks that affect the organization developing or procuring the software. For example, a competitor introducing a new product is a business risk. The introduction of a competitive product may mean that the assumptions made about sales of existing software products may be unduly optimistic.

These risk types overlap. If an experienced programmer leaves a project this can be a project risk because, even if they are immediately replaced, the schedule will be affected. It inevitably takes time for a new project member to understand the work that has been done, so they cannot be immediately productive. Consequently, the delivery of the system may be delayed. The loss of a

TRCAC

team member can also be a product risk because a replacement may not be as experienced and so could make programming errors. Finally, it can be a business risk because that programmer's experience may be crucial in winning new contracts.



Risk	Affects	Description
Staff turnover	Project	Experienced staff will leave the project before it is finished.
Management change	Project	There will be a change of organizational management with different priorities.
Hardware unavailability	Project	Hardware that is essential for the project will not be delivered on schedule.
Requirements change	Project and product	There will be a larger number of changes to the requirements than anticipated.
Specification delays	Project and product	Specifications of essential interfaces are not available on schedule.
Size underestimate	Project and product	The size of the system has been underestimated.
CASE tool underperformance	Product	CASE tools, which support the project, do not perform as anticipated.
Technology change	Business	The underlying technology on which the system is built is superseded by new technology.
Product competition	Business	A competitive product is marketed before the system is completed.

An outline of the process of risk management. It involves several stages:

1. Risk identification: You should identify possible project, product, and business risks.

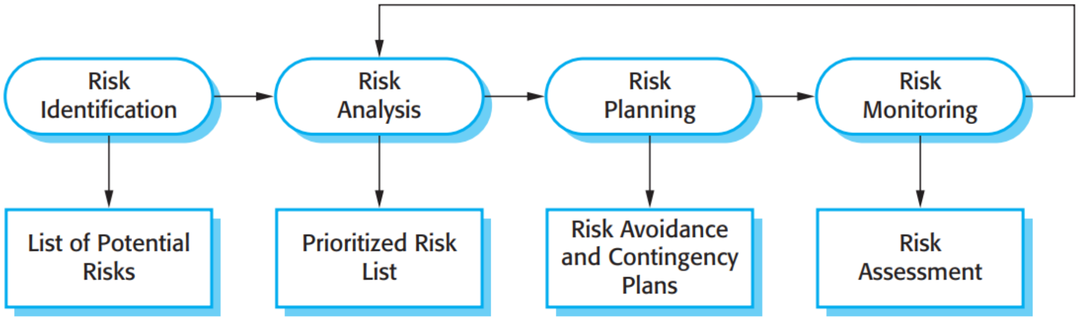
2. Risk analysis: You should assess the likelihood and consequences of these risks.

3. Risk planning: You should make plans to address the risk, either by avoiding it or minimizing its effects on the project.

4. Risk monitoring: You should regularly assess the risk and your plans for risk mitigation and revise these when you learn more about the risk.

You should document the outcomes of the risk management process in a risk management plan. This should include a discussion of the risks faced by the project, an analysis of these risks, and information on how you propose to manage the risk if it seems likely to be a problem.

The risk management process is an iterative process that continues throughout the project. Once you have drawn up an initial risk management plan, you monitor the situation to detect emerging risks. As more information about the risks becomes available, you have to reanalyze the risks and decide if the risk priority has changed. You may then have to change your plans for risk avoidance and contingency management.



TRCA

Risk identification

Risk identification is the first stage of the risk management process. It is concerned with identifying the risks that could pose a major threat to the software engineering process, the software being developed, or the development organization.

Risk identification may be a team process where a team get together to brainstorm possible risks. Alternatively, the project manager may simply use his or her experience to identify the most probable or critical risks.

As a starting point for risk identification, a checklist of different types of risk may be used. There are at least six types of risk that may be included in a risk checklist:

1. **Technology risks:** Risks that derive from the software or hardware technologies that are used to develop the system.

2. **People risks:** Risks that are associated with the people in the development team.

3. **Organizational risks:** Risks that derive from the organizational environment where the software is being developed.

4. **Tools risks:** Risks that derive from the software tools and other support software used to develop the system.

5. Requirements risks: Risks that derive from changes to the customer requirements and the process of managing the requirements change**.**

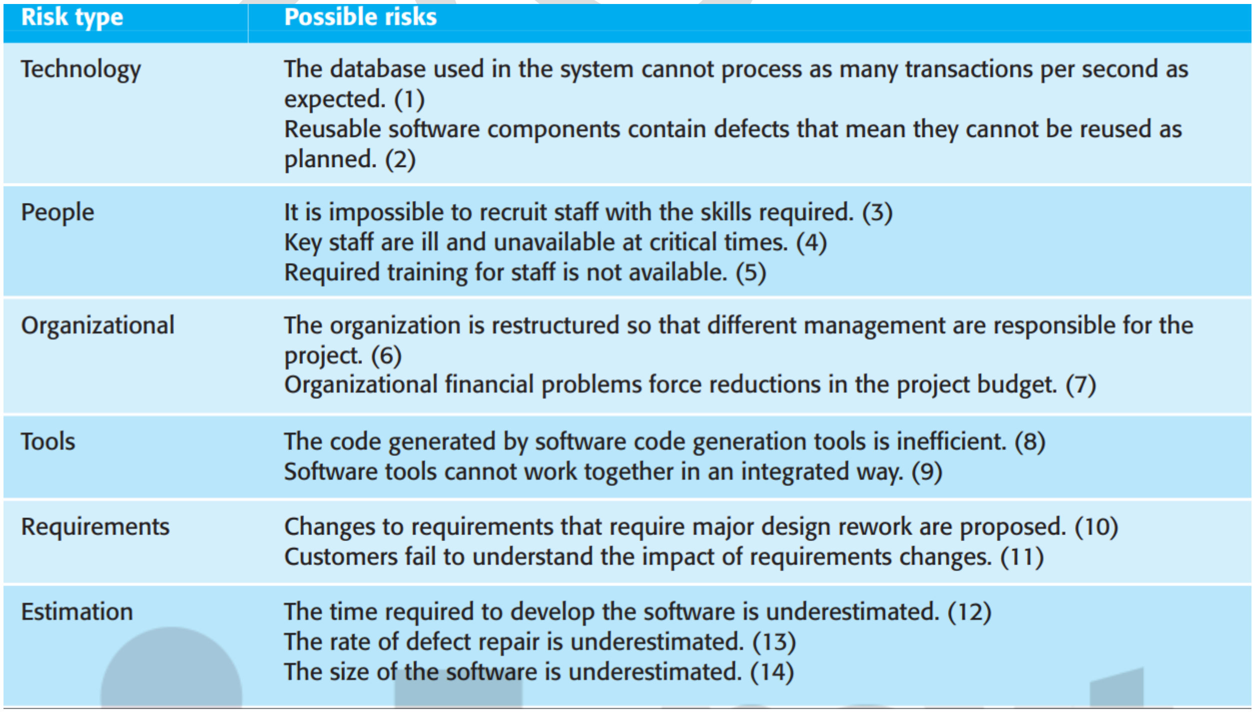
6. Estimation risks: Risks that derive from the management estimates of the resources required to build the system.

Risk analysis

During the risk analysis process, you have to consider each identified risk and make a judgment about the probability and seriousness of that risk. There is no easy way to do this. You have to rely on your own judgment and experience of previous projects and the problems that arose in them. It is not possible to make precise, numeric assessment of the probability and seriousness of each risk. Rather, you should assign the risk to one of a number of bands:

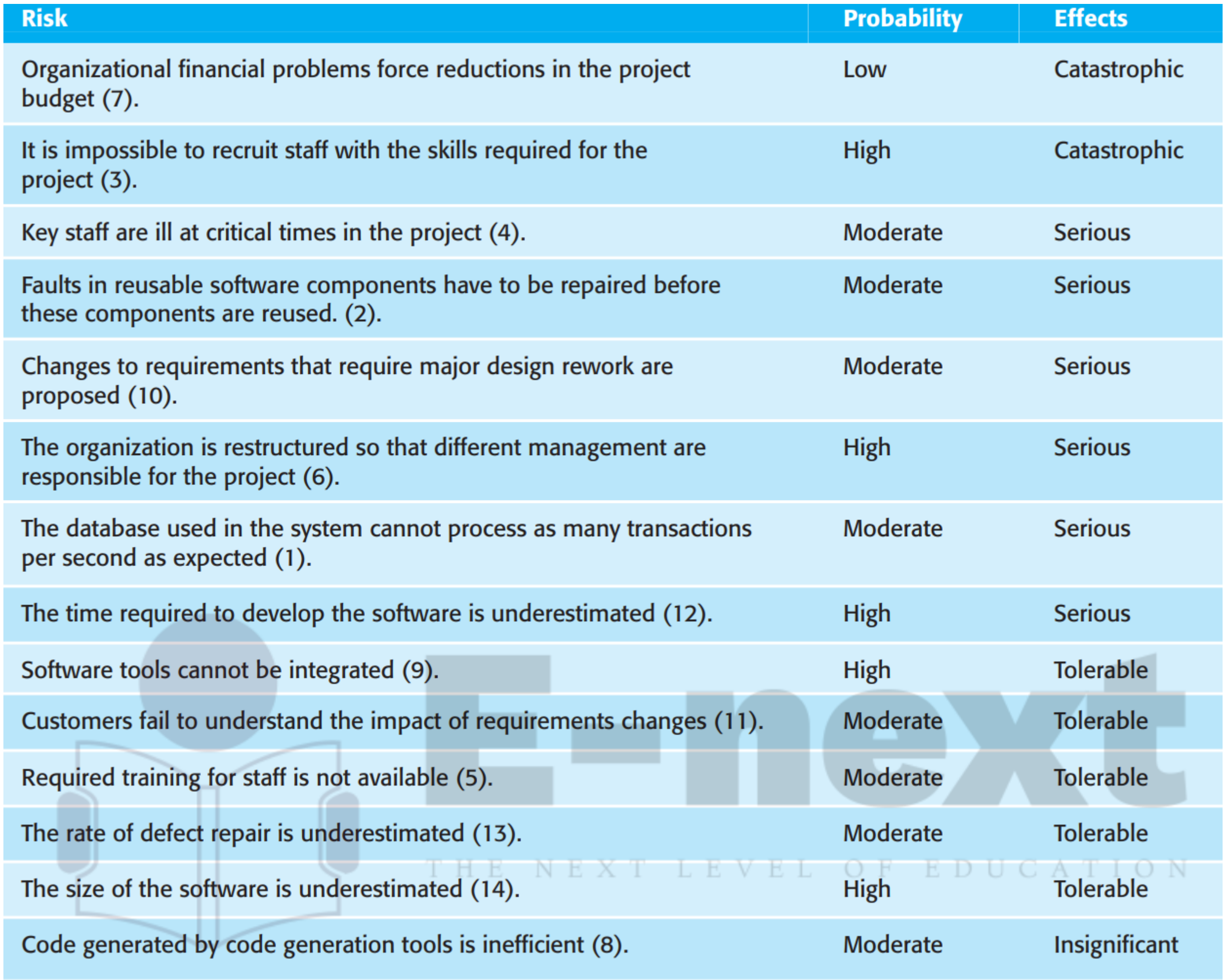
1. The probability of the risk might be assessed as very low (<10%), low (10–25%), moderate (25–50%), high (50–75%), or very high (> 75%).

2. The effects of the risk might be assessed as catastrophic (threaten the survival of the project), serious (would cause major delays), tolerable (delays are within allowed contingency), or insignificant.



Risk type	Possible risks
Technology	The database used in the system cannot process as many transactions per second as expected. (1) Reusable software components contain defects that mean they cannot be reused as planned. (2)
People	It is impossible to recruit staff with the skills required. (3) Key staff are ill and unavailable at critical times. (4) Required training for staff is not available. (5)
Organizational	The organization is restructured so that different management are responsible for the project. (6) Organizational financial problems force reductions in the project budget. (7)
Tools	The code generated by software code generation tools is inefficient. (8) Software tools cannot work together in an integrated way. (9)
Requirements	Changes to requirements that require major design rework are proposed. (10) Customers fail to understand the impact of requirements changes. (11)
Estimation	The time required to develop the software is underestimated. (12) The rate of defect repair is underestimated. (13) The size of the software is underestimated. (14)

Once the risks have been analyzed and ranked, you should assess which of these risks are most significant. Your judgment must depend on a combination of the probability of the risk arising and the effects of that risk. In general, catastrophic risks should always be considered, as should all serious risks that have more than a moderate probability of occurrence.



Risk	Probability	Effects
Organizational financial problems force reductions in the project budget (7).	Low	Catastrophic
It is impossible to recruit staff with the skills required for the project (3).	High	Catastrophic
Key staff are ill at critical times in the project (4).	Moderate	Serious
Faults in reusable software components have to be repaired before these components are reused. (2).	Moderate	Serious
Changes to requirements that require major design rework are proposed (10).	Moderate	Serious
The organization is restructured so that different management are responsible for the project (6).	High	Serious
The database used in the system cannot process as many transactions per second as expected (1).	Moderate	Serious
The time required to develop the software is underestimated (12).	High	Serious
Software tools cannot be integrated (9).	High	Tolerable
Customers fail to understand the impact of requirements changes (11).	Moderate	Tolerable
Required training for staff is not available (5).	Moderate	Tolerable
The rate of defect repair is underestimated (13).	Moderate	Tolerable
The size of the software is underestimated (14).	High	Tolerable
Code generated by code generation tools is inefficient (8).	Moderate	Insignificant

E-next THE NEXT LEVEL OF EDUCATION

Risk planning

The risk planning process considers each of the key risks that have been identified, and develops strategies to manage these risks. For each of the risks, you have to think of actions that you might take to minimize the disruption to the project if the problem identified in the risk occurs. You also should think about information that you might need to collect while monitoring the project so that problems can be anticipated.

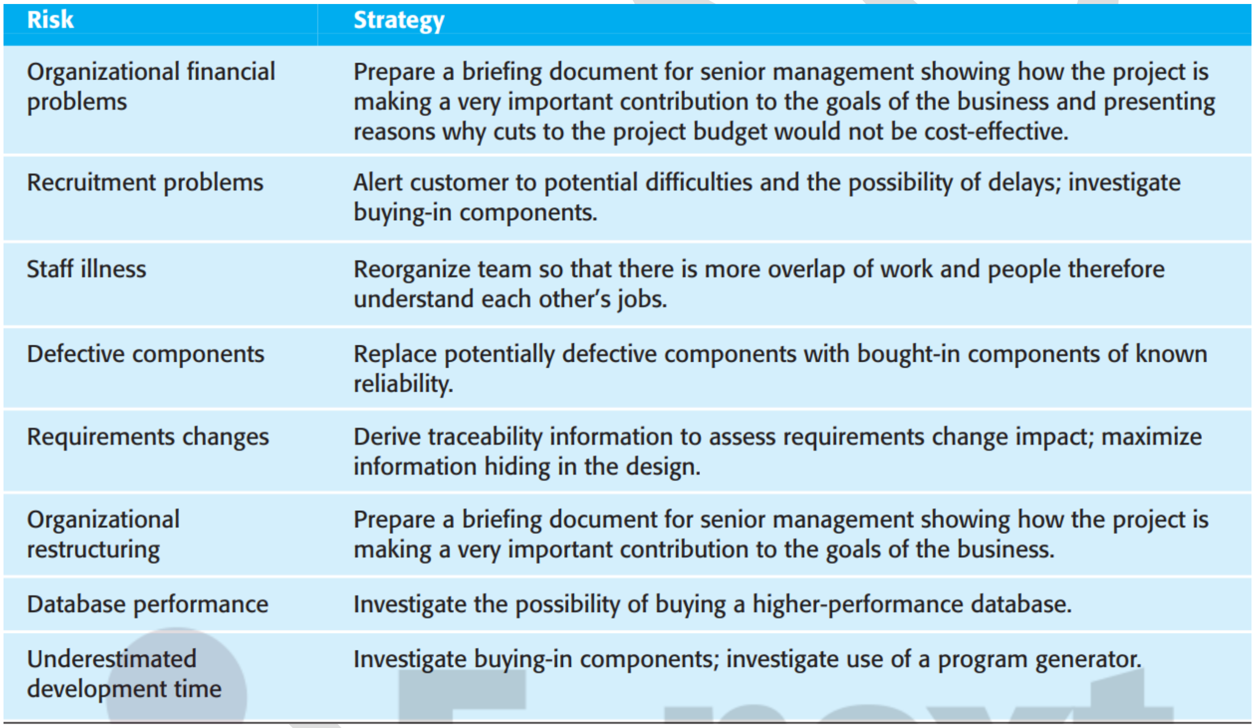
Again, there is no simple process that can be followed for contingency planning. It relies on the judgment and experience of the project manager.

Figure shows possible risk management strategies that have been identified for the key risks (i.e., those that are serious or intolerable) .These strategies fall into three categories:

1. Avoidance strategies: Following these strategies means that the probability that the risk will arise will be reduced. An example of a risk avoidance strategy is the strategy for dealing with defective components

2. Minimization strategies: Following these strategies means that the impact of the risk will be reduced. An example of a risk minimization strategy is the strategy for staff illness

3. Contingency plans: Following these strategies means that you are prepared for the worst and have a strategy in place to deal with it. An example of a contingency strategy is the strategy for organizational financial problems.



Risk	Strategy
Organizational financial problems	Prepare a briefing document for senior management showing how the project is making a very important contribution to the goals of the business and presenting reasons why cuts to the project budget would not be cost-effective.
Recruitment problems	Alert customer to potential difficulties and the possibility of delays; investigate buying-in components.
Staff illness	Reorganize team so that there is more overlap of work and people therefore understand each other's jobs.
Defective components	Replace potentially defective components with bought-in components of known reliability.
Requirements changes	Derive traceability information to assess requirements change impact; maximize information hiding in the design.
Organizational restructuring	Prepare a briefing document for senior management showing how the project is making a very important contribution to the goals of the business.
Database performance	Investigate the possibility of buying a higher-performance database.
Underestimated development time	Investigate buying-in components; investigate use of a program generator.

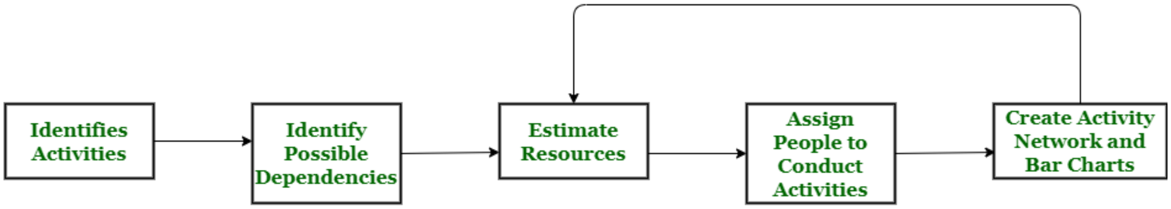
Risk monitoring

Risk monitoring is the process of checking that your assumptions about the product, process, and business risks have not changed. You should regularly assess each of the identified risks to decide whether or not that risk is becoming more or less probable. You should also think about whether or not the effects of the risk have changed. To do this, you have to look at other factors, such as the number of requirements change requests, which give you clues about the risk probability and its effects.

These factors are obviously dependent on the types of risk. You should monitor risks regularly at all stages in a project. At every management review, you should consider and discuss each of the key risks separately. You should decide if the risk is more or less likely to arise and if the seriousness and consequences of the risk have changed.

Project Scheduling

Project-task scheduling is a significant project planning activity. It comprises deciding which functions would be taken up when. **Project schedule** simply means a mechanism that is used to communicate and know about that tasks are needed and has to be done or performed and which organizational resources will be given or allocated to these tasks and in what time duration or time frame work is needed to be performed. Effective project scheduling leads to success of project, reduced cost, and increased customer satisfaction.


Identifies Activities
Identify Possible Dependencies
Estimate Resources
Assign People to Conduct Activities
Create Activity Network and Bar Charts

Project Scheduling Process

Process

The first method in scheduling a software plan involves identifying all the functions required to complete the project. A good judgment of the intricacies of the project and the development process helps the supervisor to identify the critical role of the project effectively.

Next, the large functions are broken down into a valid set of small activities which would be assigned to various engineers. The work breakdown structure formalism supports the manager to breakdown the function systematically after the project manager has broken down the purpose and constructs the work breakdown structure; he has to find the dependency among the activities.

Dependency among the various activities determines the order in which the various events would be carried out. If an activity A necessary the results of another activity B, then activity A must be scheduled after activity B. In general, the function dependencies describe a partial ordering among functions, i.e., each service may precede a subset of other functions, but some functions might not have any precedence ordering describe between them (called concurrent function). The dependency among the activities is defined in the pattern of an activity network.

Once the activity network representation has been processed out, resources are allocated to every activity. Resource allocation is usually done using a Gantt chart. After resource allocation is completed, a PERT chart representation is developed.

The PERT chart representation is useful for program monitoring and control. For task scheduling, the project plan needs to decompose the project functions into a set of activities. The time frame when every activity is to be performed is to be determined.

The end of every action is called a milestone. The project manager tracks the function of a project by audit the timely completion of the milestones. If he examines that the milestones start getting delayed, then he has to handle the activities carefully so that the complete deadline can still be met.

Project Planning

Project planning is the process of defining the project scope, objectives, and steps needed to get the work done. It's one of the most important processes in project management. The output of the project planning process is a project management plan.

A project management plan—**also known as a project plan**—is a document that outlines the process your team will use to manage the project according to scope to meet its stated objectives. The purpose of a project plan is to map out the steps and resources it will take to complete a project on time and budget.

A project plan communicates vital information—such as deadlines, assignments, and key milestones—to all project stakeholders and is integral to project success. It is most commonly represented in the form of a Gantt chart to make it easy to ensure work stays on track.

Structure of the project plan includes:

1.Introduction: This briefly describes the objective of the project and sets out the constraints (e.g., budget, time, etc.) that affects the project management.

IRCAC

2. **Project organization**: This describes the way in which the development team is organized, the people involved and their roles in the team.

3.Risk analysis: This describes the possible project risks, the likelihood of these risks arising and the risk reduction strategies that are proposed.

4.Hardware and Software resource requirements: This specifies the hardware and the support software required to carry out the development. If hardware has to be bought, estimates of the prices and delivery schedule may be included.

5.Work breakdown: This sets out the breakdown of the project into activities and identifies the milestones and deliverables associated with each activity.

6. Project schedule: This shows the dependencies between activities, the estimated time required to reach out each milestone and the allocation of people to activities.

7.Monitoring and reporting mechanisms: This defines the management reports that should be produced, when these should be produced and the project monitoring mechanisms used.

TRCAC

Managing people

People are an organization's most important assets. The tasks of a manager are essentially people-oriented. Unless there is some understanding of people, management will be unsuccessful. Poor people management is an important contributor to project failure.

People management factors:

- **Consistency**: team members should all be treated in a comparable way without favourites or discrimination.
- **Respect**: different team members have different skills and these differences should be respected.
- **Inclusion**: involve all team members and make sure that people's views are considered.
- **Honesty**: you should always be honest about what is going well and what is going badly in a project.

An important role of a manager is to **motivate the people** working on a project. Motivation means organizing the work and the working environment to encourage people to work effectively. If people are not motivated, they will not be interested in the work they are doing. They will work slowly, be more likely to make mistakes and will not contribute to the broader goals of the team or the organization. **Motivation is a complex issue** but it appears that there are different types of motivation based on:

- Basic needs (e.g., food, sleep, etc.);
- Personal needs (e.g., respect, self-esteem);
- Social needs (e.g., to be accepted as part of a group).

In software development groups, basic physiological and safety needs are not an issue. Here's how to satisfy other types of needs:

- **Social**: provide communal facilities; allow informal communications e.g., via social networking
- **Esteem**: recognition of achievements; appropriate rewards
- **Self-realization**: training - people want to learn more; responsibility

Motivation should also take into account different **personality types**:

- **Task-oriented**: the motivation for doing the work is the work itself
- **Self-oriented**: the work is a means to an end which is the achievement of individual goals - e.g. to get rich, to play tennis, to travel etc.;
- **Interaction-oriented**: the principal motivation is the presence and actions of co-workers. People go to work because they like to go to work.

Teamwork

Most **software engineering is a group activity.** The development schedule for most non-trivial software projects is such that they cannot be completed by one person working alone. A good group is cohesive and has a team spirit. The people involved are motivated by the success of the group as well as by their own personal goals. Group interaction is a key determinant of group performance. Flexibility in group composition is limited: managers must do the best they can with available people.

In a **cohesive group**, members consider the group to be more important than any individual in it. The **advantages** of a cohesive group are:

- Group **quality standards** can be developed by the group members.
- Team members **learn from each other** and get to know each other's work; inhibitions caused by ignorance are reduced.
- **Knowledge is shared**. Continuity can be maintained if a group member leaves.
- Refactoring and **continual improvement is encouraged**; group members work collectively to deliver high quality results and fix problems, irrespective of the individuals who originally created the design or program.

TRCAC

Three generic **factors that affect team effectiveness**:

The people in the group

You need a mix of people in a project group as software development involves diverse activities such as negotiating with clients, programming, testing and documentation.

The group organization

A group should be organized so that individuals can contribute to the best of their abilities and tasks can be completed as expected.

Technical and managerial communications

Good communications between group members, and between the software engineering team and other project stakeholders, is essential.

A manager or team leader's job is to create a **cohesive group** and organize their group so that they can work together effectively. This involves creating a group with the **right balance of technical skills and personalities**, and organizing that group so that the members work together effectively. It may not be possible to appoint the ideal people to work on a project:

- Project budget may not allow for the use of highly-paid staff;
- Staff with the appropriate experience may not be available;
- An organization may wish to develop employee skills on a software project.

Managers have to work within these constraints especially when there are shortages of trained staff.

Group composed of members who share the same motivation can be problematic:

- **Task-oriented** everyone wants to do their own thing;
- **Self-oriented** everyone wants to be the boss;
- **Interaction-oriented** too much chatting, not enough work.

An effective group has a balance of all types. This can be difficult to achieve software engineers are often task-oriented. Interaction-oriented people are very important as they can detect and defuse tensions that arise.

The way how **group organization** affects the decisions that are made by that group, the ways that information is exchanged and the interactions between the development group and external project stakeholders. Key questions include:

- Should the project manager be the technical leader of the group?
- Who will be involved in making critical technical decisions, and how will these be made?
- How will interactions with external stakeholders and senior company management be handled?
- How can groups integrate people who are not co-located?
- How can knowledge be shared across the group?

TRCAC

Small software engineering groups are usually organized **informally** without a rigid structure. For **large** projects, there may be a **hierarchical structure** where different groups are responsible for different sub-projects. **Agile** development is always based around an **informal group** on the principle that formal structure inhibits information exchange.

An **informal group** acts as a whole and comes to a consensus on decisions affecting the system. The group leader serves as the external interface of the group but does not allocate specific work items. Rather, work is discussed by the group as a whole and tasks are allocated according to ability and experience. This approach is successful for groups where all members are experienced and competent.

Good **communications** are essential for effective group working. Information must be exchanged on the status of work, design decisions and changes to previous decisions. Good communications also strengthen group cohesion as it promotes understanding. The effectiveness and efficiency of communications is influenced by:

- **Group size**: the larger the group, the harder it is for people to communicate with other group members.
- **Group structure**: communication is better in informally structured groups than in hierarchically structured groups.
- **Group composition**: communication is better when there are different personality types in a group and when groups are mixed rather than single sex.
- **The physical work environment**: good workplace organization can help encourage communications.

TRCAC