MODULE 4

Software Project Management - Risk management, Managing people, Teamwork. Project Planning, Software pricing, Plan-driven development, Project scheduling, Agile planning. Estimation techniques, COCOMO cost modeling. Configuration management, Version management, System building, Change management, Release management, Agile software management - SCRUM framework. Kanban methodology and lean approaches.

Software project management

- 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.
- Good management cannot guarantee project success.
- However, bad management usually results in project failure: The software may be delivered late, cost more than originally estimated, or fail to meet the expectations of customers.
- The success criteria for project management obviously vary from project to project, but, for most projects, important goals are:
  - to deliver the software to the customer at the agreed time;
  - to keep overall costs within budget;
  - to deliver software that meets the customer's expectations;
  - to maintain a coherent and well-functioning development team.

  These goals are not unique to software engineering but are the goals of all engineering projects.

Software project management

- 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 : Software is intangible. It cannot be seen or touched. Software project managers cannot see progress by 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: Every large software development project is unique because every environment where software is developed is, in some ways, different from all others.
3. Software processes are variable and organization-specific: Different companies use quite different software development processes. We cannot reliably predict when a particular software process is likely to lead to development problems.

Software project management

It is impossible to write a standard job description for a software project manager. Some of the most **important factors that affect how software projects are managed are**:

1. **Company size**: Small companies can operate with informal management and team communications and do not need formal policies and management structures. In larger organizations, management hierarchies, formal reporting and budgeting, and approval processes must be followed.
2. **Software customers**: If the customer is an internal customer, then customer communications can be informal and there is no need to fit in with the customer's ways of working. If custom software is being developed for an external customer, agreement has to be reached on more formal communication channels. If the customer is a government agency, the software company must operate according to the agency's policies and procedures, which are likely to be bureaucratic.
3. **Software size**: Small systems can be developed by a small team, which can get together in the same room to discuss progress and other management issues. Large systems usually need multiple development teams that may be geographically distributed and in different companies. The project manager has to coordinate the activities of these teams and arrange for them to communicate with each other.
4. **Software type**: If the software being developed is a consumer product, formal records of project management decisions are unnecessary. On the other hand, if a safety-critical system is being developed, all project management decisions should be recorded and justified as these may affect the safety of the system.
5. **Organizational culture**: Some organizations have a culture that is based on supporting and encouraging individuals, while others are group focused. Large organizations are often bureaucratic. Some organizations have a culture of taking risks, whereas others are risk averse.
6. **Software development processes:** Agile processes typically try to operate with "lightweight" management. More formal processes require management monitoring to ensure that the development team is following the defined process.

Software project management

- A number of **fundamental project management activities** are **common to all organizations**:

1. **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 they monitor progress to check that the development is on time and within budget.
2. **Risk management**: Project managers have to assess the risks that may affect a project, monitor these risks, and take action when problems arise.
3. **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.
4. **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.
5. **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.

Risk management

Risk identification

Risk analysis

Risk planning

Risk monitoring

Risk management

- Risk management is one of the most important jobs for a project manager.
- You can think of a risk as something that you'd prefer not to have happen.
- Risks may threaten the project, the software that is being developed, or the organization.
- 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 can be categorized according to **type of risk** (technical, organizational, etc.).
- A complementary classification is to **classify risks according to what these risks affect**:
    1. Project risks: affect the project schedule or resources. An example of a project risk is the loss of an experienced system architect.
    2. Product risks: 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: affect the organization developing or procuring the software. For example, a competitor introducing a new product is a business risk
- These risk categories overlap.

Risk management

- For **large projects**, you should record the results of the risk analysis in a risk register along with a consequence analysis.
- This sets out the consequences of the risk for the project, product, and business.
- Effective risk management makes it easier to cope with problems and to ensure that these do not lead to unacceptable budget or schedule slippage.
- For **small projects**, formal risk recording may not be required, but the project manager should be aware of them.

Risk management

- The specific risks that may affect a project depend on the project and the organizational environment in which the software is being developed.
- However, there are also common risks that are independent of the type of software being developed.
- These can occur in any software development project. Some examples of these common risks are shown in Figure.

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 company 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.
Software tool underperformance	Product	Software tools that 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.

Risk management

- Software risk management is important because of the inherent uncertainties in software development.
- These uncertainties stem from loosely defined requirements, requirements changes due to changes in customer needs, difficulties in estimating the time and resources required for software development, and differences in individual skills.
- You have to anticipate risks, understand their impact on the project, the product, and the business, and take steps to avoid these risks.
- You may need to draw up contingency plans so that, if the risks do occur, you can take immediate recovery action.

Risk management

An **outline of the process of risk management** is presented in Figure. 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 by minimizing its effects on the project.
4. **Risk monitoring**: You should regularly assess the risk and your plans for risk mitigation and revise these plans when you learn more about the risk

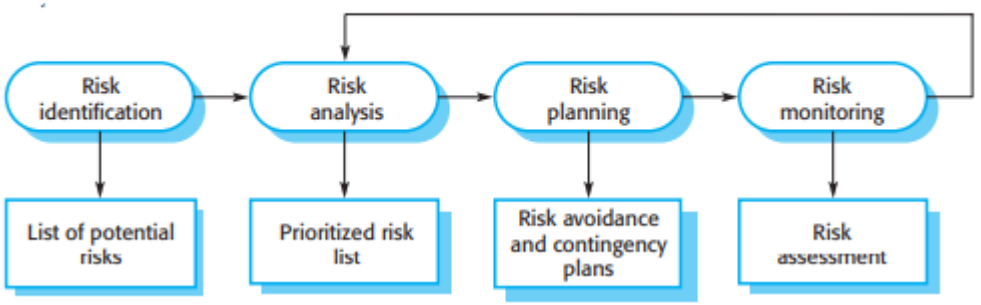

Figure: The risk management process

Risk management

- For large projects, you should document the outcomes of the risk management process in a risk management plan.
- The risk management process is an iterative process that continues throughout a 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 vailable, you have to re-analyze the risks and decide if the risk priority has changed. You may then have to change your plans for risk avoidance and contingency management.
- Risk management in agile development is less formal.

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 in which a team gets together to brainstorm possible risks.
- Alternatively, project managers may identify risks based on their experience of what went wrong on previous projects.
- As a starting point for risk identification, a checklist of different types of risk may be used. Six types of risk may be included in a risk checklist:

1. Estimation risks: arise from the management estimates of the resources required to build the system.
2. Organizational risks: arise from the organizational environment where the software is being developed.
3. People risks: are associated with the people in the development team.
4. Requirements risks: come from changes to the customer requirements and the process of managing the requirements change.
5. Technology risks: come from the software or hardware technologies that are used to develop the system.
6. Tools risks: come from the software tools and other support software used to develop the system

Risk identification

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

Figure 1: Examples of different types of risks

Risk identification

- When you have finished the risk identification process, you should have a long list of risks that could occur and that could affect the product, the process, and the business.
- You then need to prune this list to a manageable size. If you have too many risks, it is practically impossible to keep track of all of them

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.
- You have to rely on your 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 insignificant, low, moderate, high, or very high.

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. Y

You may then tabulate the results of this analysis process using a table ordered according to the seriousness of the risk. (table in the next slide)

Risk analysis

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

Figure 2: Risk types and examples

Risk analysis

- Both the probability and the assessment of the effects of a risk may change as more information about the risk becomes available and as risk management plans are implemented.
- Update the risk table during each iteration of the risk management process.
- Once the risks have been analyzed and ranked, assess which of these risks are most significant. The 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 planning

- The risk planning process develops strategies to manage the key risks that threaten the project.
- For each risk, 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 should also think about the information that you need to collect while monitoring the project so that emerging problems can be detected before they become serious.
- In risk planning, you have to ask “what-if” questions that consider both individual risks, combinations of risks, and external factors that affect these risks.
- For example, questions that you might ask are:

1. What if several engineers are ill at the same time?
2. What if an economic downturn leads to budget cuts of 20% for the project?
3. What if the performance of open-source software is inadequate and the only expert on that open-source software leaves?
4. What if the company that supplies and maintains software components goes out of business?
5. What if the customer fails to deliver the revised requirements as predicted?

Risk planning

Based on the answers to these “what-if” questions, you may devise strategies for managing the risks.

These strategies fall into three categories:

1. Avoidance strategies: Following these strategies means that the probability that the risk will arise is reduced. An example of a risk avoidance strategy is the strategy for dealing with defective components shown in figure 3.
2. Minimization strategies :Following these strategies means that the impact of the risk is reduced. An example of a risk minimization strategy is the strategy for staff illness shown in Figure 3
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 shown in Figure 3.

Figure 3 shows possible risk management strategies that have been identified for the key risks (i.e., those that are serious or intolerable) durig the previous phase(risk analysis)(figure 2).

Risk planning

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 automated code generation.

Figure 3: Strategies to help manage risks

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.
- Figure 4 gives some examples of factors that may be helpful in assessing these risk types.

Risk monitoring

Risk type	Potential indicators
Estimation	Failure to meet agreed schedule; failure to clear reported defects.
Organizational	Organizational gossip; lack of action by senior management.
People	Poor staff morale; poor relationships among team members; high staff turnover.
Requirements	Many requirements change requests; customer complaints.
Technology	Late delivery of hardware or support software; many reported technology problems.
Tools	Reluctance by team members to use tools; complaints about software tools; requests for faster computers/more memory, and so on.

Figure 4: Risk indicators

Managing people

Motivating people

Managing people

- The people working in a software organization are its greatest assets.
- It is expensive to recruit and retain good people, and it is up to software managers to ensure that the engineers working on a project are as productive as possible.
- It is important that software project managers understand the technical issues that influence the work of software development.
- As a project manager, you should be aware of the potential problems of people management and should try to develop people management skills.

Managing people

- There are four critical factors that influence the relationship between a manager and the people that he or she manages:

1. Consistency: All the people in a project team should be treated in a comparable way.
2. Respect: Different people have different skills, and managers should respect these differences. All members of the team should be given an opportunity to make a contribution.
3. Inclusion: It is important to develop a working environment where all views, even those of the least experienced staff, are considered.
4. Honesty: As a manager, you should always be honest about what is going well and what is going badly in the team. You should also be honest about your level of technical knowledge and be willing to defer to staff with more knowledge when necessary.

Motivating people

- As a project manager, you need to motivate the people who work with you so that they will contribute to the best of their abilities.
- In practice, motivation means organizing work and its environment to encourage people to work as effectively as possible.
- If people are not motivated, they will be less interested in the work they are doing.
- people are motivated by satisfying their needs. These needs are arranged in a series of levels(figure).
- The lower levels of this hierarchy represent fundamental needs for food, sleep, and so on.
- Social need is concerned with the need to feel part of a social grouping.
- Esteem need represents the need to feel respected by others, and self-realization need is concerned with personal development.
- People need to satisfy lower-level needs such as hunger before the more abstract, higher-level needs.

Human needs hierarchy

Motivating people

- Making sure that peoples' social, esteem, and self-realization needs are satisfied is most important from a management point of view.

1. To satisfy social needs, you need to give people time to meet their co-workers and provide places for them to meet.
2. To satisfy esteem needs, you need to show people that they are valued by the organization.
3. Finally, to satisfy self-realization needs, you need to give people responsibility for their work, assign them demanding (but not impossible) tasks, and provide opportunities for training and development where people can enhance their skills. Training is an important motivating influence as people like to gain new knowledge and learn new skills.

Motivating people

- Psychological personality type also influences motivation. Three classifications for professional workers are identified:

1. Task-oriented people, who are motivated by the work they do. In software engineering, these are people who are motivated by the intellectual challenge of software development.
2. Self-oriented people, who are principally motivated by personal success and recognition. They are interested in software development as a means of achieving their own goals. They often have longer-term goals, such as career progression, that motivate them, and they wish to be successful in their work to help realize these goals.
3. Interaction-oriented people, who are motivated by the presence and actions of co-workers.

Research has shown that interaction-oriented personalities usually like to work as part of a group, whereas task-oriented and self-oriented people usually prefer to act as individuals. Women are more likely to be interaction-oriented than men are. They are often more effective communicators.

Teamwork

Selecting group members

Group organization

Group communications

Teamwork

- Most professional software is developed by project teams that range in size from two to several hundred people.
- However, as it is impossible for everyone in a large group to work together on a single problem, large teams are usually split into a number of smaller groups.
- Each group is responsible for developing part of the overall system. The best size for a software engineering group is 4 to 6 members, and they should never have more than 12 members.
- When groups are small, communication problems are reduced.
- Putting together a group that has the right balance of technical skills, experience, and personalities is a critical management task.

Teamwork

- A good group is cohesive and thinks of itself as a strong, single unit. The people involved are motivated by the success of the group as well as by their own personal goals.
- In a cohesive group, members think of the group as more important than the individuals who are group members.
- Members of a well-led, cohesive group are loyal to the group. They identify with group goals and other group members.
- They attempt to protect the group, as an entity, from outside interference.
- The benefits of creating a cohesive group are:
    1. The group can establish its own quality standards
    2. Individuals learn from and support each other
    3. Knowledge is shared: Continuity can be maintained if a group member leaves.
    4. Refactoring and continual improvement is encouraged

Teamwork

- Good project managers should always try to encourage group cohesiveness.
- They may try to establish a sense of group identity by naming the group and establishing a group identity and territory.
- Social events for group members and their families are a good way to bring people together.
- One of the most effective ways of promoting cohesion is to be inclusive. That is, you should treat group members as responsible and trustworthy, and make information freely available.
- An effective way of making people feel valued and part of a group is to make sure that they know what is going on.
- Given a stable organizational and project environment, the three factors that have the biggest effect on team working are:

1. 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.
2. The way the group is organized :A group should be organized so that individuals can contribute to the best of their abilities and tasks can be completed as expected.
3. Technical and managerial communications: Good communication between group members, and between the software engineering team and other project stakeholders, is essential.

Selecting group members

- A manager or team leader's job is to create a cohesive group and organize that group so that they work together effectively.
- This task involves selecting a group with the right balance of technical skills and personalities.
- Sometimes people are hired from outside the organization; more often, software engineering groups are put together from current employees who have experience on other projects.
- Managers rarely have a completely free hand in team selection.
- They often have to use the people who are available in the company, even if they are not the ideal people for the job.
- Technical knowledge and ability should not be the only factor used to select group members.
- The "competing engineers" problem can be reduced if the people in the group have complementary motivations.
- People who are motivated by the work are likely to be the strongest technically.
- People who are self-oriented will probably be best at pushing the work forward to finish the job.
- People who are interaction-oriented help facilitate communications within the group.
- Interaction oriented people like to talk to people and can detect tensions and disagreements at an early stage, before these problems have a serious impact on the group.
- It is sometimes impossible to choose a group with complementary personalities. If this is the case, the project manager has to control the group so that individual goals do not take precedence over organizational and group objectives. This control is easier to achieve if all group members participate in each stage of the project

Group organization

- The way a group is organized affects the group's decisions, the ways information is exchanged, and the interactions between the development group and external project stakeholders.
- Important organizational questions for project managers include the following:

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

Group organization

- Small programming groups are usually organized in an informal way. The group leader gets involved in the software development with the other group members.
- In an informal group, the group as a whole discusses the work to be carried out, and tasks are allocated according to ability and experience. More senior group members may be responsible for the architectural design. However, detailed design and implementation is the responsibility of the team member who is allocated to a particular task.
- Agile development teams are always informal groups.
- If a group is composed mostly of inexperienced or incompetent members, informality can be a hindrance.
- In hierarchical groups the group leader is at the top of the hierarchy. He or she has more formal authority than the group members and so can direct their work. There is a clear organizational structure, and decisions are made toward the top of the hierarchy and implemented by people lower down. Communications are primarily instructions from senior staff; the people at lower levels of the hierarchy have relatively little communication with the managers at the upper levels.

Group organization

- Hierarchial model rarely works well for complex software engineering.
- In software development, effective team communications at all levels is essential:

1. Changes to the software often require changes to several parts of the system, and this requires discussion and negotiation at all levels in the hierarchy.

2. Software technologies change so fast that more junior staff may know more about new technologies than experienced staff. Top-down communications may mean that the project manager does not find out about the opportunities of using these new technologies. More junior staff may become frustrated because of what they see as old-fashioned technologies being used for development.

A major challenge facing project managers is the difference in technical ability between group members.

Group communications

- It is absolutely essential that group members communicate effectively and efficiently with each other and with other project stakeholders.
- Group members must exchange information on the status of their work, the design decisions that have been made, and changes to previous design decisions.
- They have to resolve problems that arise with other stakeholders and inform these stakeholders of changes to the system, the group, and delivery plans.
- Good communication also helps strengthen group cohesiveness.

Group communications

- The effectiveness and efficiency of communications are influenced by:

1. Group size: As a group gets bigger, it gets harder for members to communicate effectively. The number of one-way communication links is n * (n − 1), where n is the group size, so, with a group of eight members, there are 56 possible communication pathways. This means that it is quite possible that some people will rarely communicate with each other.
2. Group structure: People in informally structured groups communicate more effectively than people in groups with a formal, hierarchical structure.
3. Group composition: People with the same personality types may clash, and, as a result, communications can be inhibited. Communication is also usually better in mixed-sex groups than in single-sex groups. Women are often more interaction-oriented than men and may act as interaction controllers and facilitators for the group.
4. The physical work environment: The organization of the workplace is a major factor in facilitating or inhibiting communications.
5. The available communication channels: There are many different forms of communication—face to face, email messages, formal documents, telephone, and technologies such as social networking and wikis. As project teams become increasingly distributed, with team members working remotely, you need to make use of interaction technologies, such as conferencing systems, to facilitate group communications.

Group communications

- Effective communication is achieved when communications are two-way and the people involved can discuss issues and information and establish a common understanding of proposals and problems.
- All this can be done through meetings, although these meetings are often dominated by powerful personalities.
- Informal discussions when a manager meets with the team for coffee are sometimes more effective.
- More and more project teams include remote members, which also makes meetings more difficult.
- To involve them in communications, you may make use of wikis and blogs to support information exchange.
- Wikis and blogs allow project members and external stakeholders to exchange information, irrespective of their location.