**Software project management**

• Concerned with activities involved in ensuring that software is

delivered on time and on schedule and in accordance with the

requirements of the organizations developing and procuring the

software.

• Project management is needed because software development is

always subject to budget and schedule constraints that are set by

the organization developing the software.

**Success criteria**

• Deliver the software to the customer at the agreed time.

• Keep overall costs within budget.

• Deliver software that meets the customer’s expectations.

• Maintain a coherent and well-functioning development team.

**Software management distinctions**

• The product is intangible.

• Software cannot be seen or touched. Software project managers cannot see

progress by simply looking at the artefact that is being constructed.

• Many software projects are 'one-off' projects.

• Large software projects are usually different in some ways from previous

projects. Even managers who have lots of previous experience may find it

difficult to anticipate problems.

• Software processes are variable and organization specific.

• We still cannot reliably predict when a particular software process is likely

to lead to development problems.

**Factors influencing project management**

• Company size

• Software customers

• Software size

• Software type

• Organizational culture

• Software development processes

• These factors mean that project managers in different

organizations may work in quite different ways.

**Universal management activities**

• Project planning

• Project managers are responsible for planning. estimating and scheduling

project development and assigning people to tasks.

• Risk management

• Project managers assess the risks that may affect a project, monitor these

risks and take action when problems arise.

• People management

• Project managers have to choose people for their team and establish ways

of working that leads to effective team performance.

**Management activities**

• 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.

• 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.

**Risk management**

• Risk management is concerned with identifying risks and drawing

up plans to minimise their effect on a project.

• 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 the impact of these risks on

the project, the product and the business, and take steps to avoid

these risks.

**Risk classification**

• There are two dimensions of risk classification

• The type of risk (technical, organizational, ..)

• what is affected by the risk:

• Project risks affect schedule or resources;

• Product risks affect the quality or performance of the software

being developed;

• Business risks affect the organisation developing or procuring the

software.

Table

Description automatically generated

**The risk management process**

• Risk identification

• Identify project, product and business risks;

• Risk analysis

• Assess the likelihood and consequences of these risks;

• Risk planning

• Draw up plans to avoid or minimise the effects of the risk;

• Risk monitoring

• Monitor the risks throughout the project;

Diagram

Description automatically generated

**Risk identification**

• May be a team activities or based on the individual project

manager’s experience.

• A checklist of common risks may be used to identify risks in a

project

• Technology risks.

• Organizational risks.

• People risks.

• Requirements risks.

• Estimation risks.

Text

Description automatically generated

**Risk analysis**

• Assess probability and seriousness of each risk.

• Probability may be very low, low, moderate, high or very high.

• Risk consequences might be catastrophic, serious, tolerable or

insignificant.

Table

Description automatically generated

Table

Description automatically generated

**Risk planning**

• Consider each risk and develop a strategy to manage that risk.

• Avoidance strategies

• The probability that the risk will arise is reduced;

• Minimization strategies

• The impact of the risk on the project or product will be reduced;

• Contingency plans

• If the risk arises, contingency plans are planned to deal with that risk;

Text

Description automatically generated with medium confidence

Timeline

Description automatically generated with low confidence

**Risk monitoring**

• Assess each identified risks regularly to decide whether or not it is

becoming less or more probable.

• Also assess whether the effects of the risk have changed.

• Each key risk should be discussed at management progress

meetings.

Table

Description automatically generated with medium confidence

**Managing people**

• People are an organisation’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.

**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.

**Group cohesiveness**

• 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.

**The effectiveness of a team**

• 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.

**Selecting group members**

• 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.

**Assembling a team**

• 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 organisation 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 composition**

• 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.

**Group organization**

• The way that a group is organized 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?

• Small software engineering groups are usually organised 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

33

**Informal groups**

• The 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.

**Group communications**

• 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 communication also strengthens group cohesion as it

promotes understanding.

• 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-gender.

• The physical work environment

• Good workplace organization can help encourage communications.