Chapter 22

Project Management

“…a huge topic.” See Part 4, “Software Management”.
Topics covered

- Risk management
- Managing people
- Teamwork
Software project management

- Concerned with ensuring that software is delivered *on time, within budget* and *in accordance with the requirements* of both the *developing and procuring* organizations.

- Needed because *software development is always subject to budget and schedule constraints*. 
Success criteria

- Deliver the software to the customer *on time*.
- Keep overall costs *within budget*.
- Meet the customer’s needs/expectations.
- Maintain a happy 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". (one-of-a-kind)
  - 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.

(cont’d)
Software management distinctions (cont’d)

- Software processes are *variable* and *organization specific*.
  - We still cannot reliably predict when a particular software process is likely to lead to development problems.
Project Manager activities

- **Project planning**
  - Planning, estimating, and scheduling project development and assigning people to tasks.

- **Reporting**
  - Reporting on the progress of a project to customers and to the managers of the company developing the software.

(cont’d)
Project Manager activities (cont’d)

- **Risk management**
  - Assess and monitor risks; take action when problems arise.

- **People management**
  - Select team members and establish ways of working that lead to effective team performance

- **Proposal writing**
  - May be involved in writing a proposal describing the objectives of the project and how it will be carried out.
Topics covered

- Risk management
- Managing people
- Teamwork
Risk management

- Risk management is concerned with identifying risks and drawing up plans to minimize their effect.

(cont’d)
Risk management (cont’d)

- A risk exists when there is a probability that some adverse circumstance will occur.
  - *Project risks* affect schedule or resources.
  - *Product risks* affect software quality or performance.
  - *Business risks* affect the organization developing or procuring the software.

(Taxonomy based on *Effect*)
## Risk examples

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

- May be a team activity or based on the individual project manager’s experience.

- A checklist of common risks types may be used to identify risks in a project:
  - Technology risks
  - People risks
  - Organizational risks
  - Requirements risks
  - Estimation risks

Taxonomy based on Source
## Examples of risk types

<table>
<thead>
<tr>
<th>Risk type</th>
<th>Possible risks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technology</td>
<td>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)</td>
</tr>
<tr>
<td>People</td>
<td>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)</td>
</tr>
<tr>
<td>Organizational</td>
<td>The organization is restructured so that different management is responsible for the project. (6) Organizational financial problems force reductions in the project budget. (7)</td>
</tr>
<tr>
<td>Tools</td>
<td>The code generated by software code generation tools is inefficient. (8) Software tools cannot work together in an integrated way. (9)</td>
</tr>
<tr>
<td>Requirements</td>
<td>Changes to requirements that require major design rework are proposed. (10) Customers fail to understand the impact of requirements changes. (11)</td>
</tr>
<tr>
<td>Estimation</td>
<td>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)</td>
</tr>
</tbody>
</table>
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.
## Risk types and examples of analysis results

<table>
<thead>
<tr>
<th>Risk</th>
<th>Probability</th>
<th>Effects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organizational financial problems force reductions in the project budget (7).</td>
<td>Low</td>
<td>Catastrophic</td>
</tr>
<tr>
<td>It is impossible to recruit staff with the skills required for the project (3).</td>
<td>High</td>
<td>Catastrophic</td>
</tr>
<tr>
<td><strong>Key staff are ill at critical times in the project (4).</strong></td>
<td>Moderate</td>
<td>Serious</td>
</tr>
<tr>
<td>Faults in reusable software components have to be repaired before these components are reused. (2).</td>
<td>Moderate</td>
<td>Serious</td>
</tr>
<tr>
<td>Changes to requirements that require major design rework are proposed (10).</td>
<td>Moderate</td>
<td>Serious</td>
</tr>
<tr>
<td><strong>The organization is restructured so that different management is responsible for the project (6).</strong></td>
<td>High</td>
<td>Serious</td>
</tr>
<tr>
<td><strong>The database used in the system cannot process as many transactions per second as expected (1).</strong></td>
<td>Moderate</td>
<td>Serious</td>
</tr>
</tbody>
</table>
Risk planning

- Develop a strategy to manage each risk...
  - **Avoidance strategies** – reduce the *probability*.
  - **Minimization strategies** – reduce the *impact* by developing *contingency plans*. 
## Risk management strategies

<table>
<thead>
<tr>
<th>Risk</th>
<th>Strategy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organizational financial problems</td>
<td>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.</td>
</tr>
<tr>
<td>Recruitment problems</td>
<td>Alert customer to potential difficulties and the possibility of delays; investigate buying-in components.</td>
</tr>
<tr>
<td>Staff illness</td>
<td>Reorganize team so that there is more overlap of work and people therefore understand each other’s jobs.</td>
</tr>
<tr>
<td>Defective components</td>
<td>Replace potentially defective components with bought-in components of known reliability.</td>
</tr>
<tr>
<td>Requirements changes</td>
<td>Derive traceability information to assess requirements change impact; maximize information hiding in the design.</td>
</tr>
</tbody>
</table>
Risk monitoring

- Regularly assess each identified risk to determine if it is becoming less or more probable.
- Also assess whether the effects of the risk have changed.
- Significant risks should be discussed at management progress meetings.
## Risk indicators

*(≈ warning signs)*

<table>
<thead>
<tr>
<th>Risk type</th>
<th>Potential indicators</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Technology</strong></td>
<td>Late delivery of hardware or support software; many reported technology problems.</td>
</tr>
<tr>
<td><strong>People</strong></td>
<td>Poor staff morale; poor relationships amongst team members; high staff turnover.</td>
</tr>
<tr>
<td><strong>Organizational</strong></td>
<td>Organizational gossip; lack of action by senior management.</td>
</tr>
<tr>
<td><strong>Tools</strong></td>
<td>Reluctance by team members to use tools; complaints about CASE tools; demands for higher-powered workstations.</td>
</tr>
<tr>
<td><strong>Requirements</strong></td>
<td>Many requirements change requests; customer complaints.</td>
</tr>
<tr>
<td><strong>Estimation</strong></td>
<td>Failure to meet agreed schedule; failure to clear reported defects.</td>
</tr>
</tbody>
</table>
Topics covered

- Risk management
- Managing people
- Teamwork
Managing people

- People are an organization’s most important assets.
- The tasks of a manager are essentially people-oriented.
- Successful management requires a good understanding of people.
- Poor people-management is an important contributor to project failure.
People management factors

- **Consistency**: treating all team members in an equitable way without favourites or discrimination.

- **Respect**: recognizing that team members have different skills and respecting (indeed, *capitalizing on*) these differences.

- **Inclusion**: developing an environment in which all views, even those of the most junior staff, are considered.

- **Honesty**: being honest about what is going well and what is going badly in a project.
Motivating people

- Involves organizing project tasks and the working environment to encourage people to work effectively.
- Maslow (‘54) suggests that people are motivated by satisfying their needs...
Needs satisfaction

- In software development groups, basic **physiological** and **safety** needs are not *(usually!)* an issue.
- **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
Case study: Individual motivation

Alice is a software project manager working in a company that develops alarm systems. This company wishes to enter the growing market of assistive technology to help elderly and disabled people live independently. Alice has been asked to lead a team of 6 developers than can develop new products based around the company’s alarm technology.

Alice’s assistive technology project starts well. Good working relationships develop within the team and creative new ideas are developed. The team decides to develop a peer-to-peer messaging system using digital televisions linked to the alarm network for communications. However, some months into the project, Alice notices that Dorothy, a hardware design expert, starts coming into work late, the quality of her work deteriorates and, increasingly, that she does not appear to be communicating with other members of the team.

Alice talks about the problem informally with other team members to try to find out if Dorothy’s personal circumstances have changed, and if this might be affecting her work. They don’t know of anything, so Alice decides to talk with Dorothy to try to understand the problem.

(cont’d)
After some initial denials that there is a problem, Dorothy admits that she has lost interest in the job. **She expected that she would be able to develop and use her hardware interfacing skills.** However, because of the product direction that has been chosen, she has little opportunity for this. **Basically, she is working as a C programmer with other team members.**

Although she admits that the work is challenging, **she is concerned that she is not developing her interfacing skills. She is worried that finding a job that involves hardware interfacing will be difficult after this project.** Because she does not want to upset the team by revealing that she is thinking about the next project, she has decided that it is best to minimize conversation with them.
Personality types

- Maslow’s needs hierarchy is almost certainly an over-simplification of what drives motivation...
- One should also take into account different personality types:
  - Task-oriented
  - Self-oriented
  - Interaction-oriented

(cont’d)
Personality types (cont’d)

- **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.

- **Interaction-oriented**
  - The principal motivation is the presence and actions of co-workers.
Motivation balance

- Most people have personalities that are a blend of these types, with one usually being dominant at any given time.
- The balance can change depending on personal circumstances and external events.
- Just being part of a cohesive group or team can be highly motivating to people...
Topics covered

- Risk management
- Managing people
- Teamwork
Teamwork

- Most software engineering is a **group activity**.
- A good group is **cohesive** and has team spirit. The people involved are motivated by the **success of the group** as well as by their own **personal goals**.
- Effective and efficient **group communication** is a key determinant of group performance.
- **Flexibility in composing groups is often limited** – managers must do the best they can with the people available.
Advantages of group cohesiveness

- Quality standards can be developed by the group members (and are therefore likely to be observed).
- Team members learn from each other and get to know each other’s work.
- Knowledge is shared, so continuity can be maintained if a group member leaves.
- Refactoring and continual improvement is encouraged.
- Members work collectively to deliver high quality results, regardless of who originally created the design or program.
Case study: Team spirit

Alice, an experienced project manager, understands the importance of creating a cohesive group. As they are developing a new product, she takes the opportunity of involving all group members in the product specification and design by getting them to discuss possible technology with elderly members of their families. She also encourages them to bring these family members to meet other members of the development group.

Alice also arranges monthly lunches for everyone in the group. These lunches are an opportunity for all team members to meet informally, talk around issues of concern, and get to know each other. At the lunch, Alice tells the group what she knows about organizational news, policies, strategies, and so forth. Each team member then briefly summarizes what they have been doing and the group discusses a general topic, such as new product ideas from elderly relatives.

Every few months, Alice organizes an ‘away day’ for the group where the team spends two days on ‘technology updating’. Each team member prepares an update on a relevant technology and presents it to the group. This is an off-site meeting in a good hotel and plenty of time is scheduled for discussion and social interaction.
Team effectiveness factors

- **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, documentation, etc.

- **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.
Assembling a team

- 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 (= train those who lack skills) on a software project.

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

- Groups composed of members who all 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 as software engineers are often task-oriented.
- Interaction-oriented people are very important as they can detect and defuse tensions that arise.
Case study: Group composition

In creating a group for assistive technology development, Alice is aware of the importance of selecting members with complementary personalities. When interviewing potential group members, she tried to assess whether they were task-oriented, self-oriented, or interaction-oriented. She felt that she was primarily a self-oriented type because she considered the project to be a way of getting noticed by senior management and possibly promoted. She therefore looked for one or perhaps two interaction-oriented personalities, with task-oriented individuals to complete the team. The final assessment that she arrived at was:

Alice—self-oriented
Brian—task-oriented
Bob—task-oriented
Carol—interaction-oriented
Dorothy—self-oriented
Ed—interaction-oriented
Fred—task-oriented
Group organization

- The way that a group is organized affects:
  - the decisions that are made,
  - the ways that information is exchanged, and
  - the interactions between the development group and stakeholders.

- Key organizational questions to consider 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?

(cont’d)
Group organization (cont’d)

- Key questions (cont’d):
  - 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?
Group organization (cont’d)

- **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.
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** (by mutual consent) 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 communications also strengthens group cohesion as it promotes understanding.

(cont’d)
Group communications are 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. (But size matters here...)
- **Group composition**: Communication is better when there are different personality types in a group and when groups are mixed rather than single sex.
- **Physical work environment**: The organization of the workplace can help encourage communications.
- **Available communication channels**: Different forms of communication are supported by a range of technologies.
Communication/coordination overhead: The “Mythical Man-Month”

- Unfortunately, the time required to complete a SE project is generally not inversely proportional to the number of people working on the project.

- Adding people to a late project can make it later (due to communication/coordination overhead).

(F. Brooks, The Mythical Man-Month)
The problem with "Man-Months"

![Graph showing the relationship between time and people. The x-axis represents few to many people, and the y-axis represents less to more time.](image)
Key points

- Good project management is **essential** if software engineering projects are to be developed on schedule and within budget.

- Software management is distinct from other engineering management.
  - Software is intangible.
  - Projects may be novel or innovative with no body of experience to guide their management.
  - Software processes are not as mature as traditional engineering processes.

(cont’d)
Key points (cont’d)

- Risk management is now recognized as one of the most important project management tasks.
- It involves identifying and assessing project risks to determine their probability and severity.
- One should develop strategies to both avoid and reduce the impact of risks.

(cont’d)
Key points (cont’d)

- People are **motivated** by interaction with other people, the recognition of management and their peers, and by being given opportunities for personal development.

- Ideally, software development groups should be small and cohesive.

- Factors that influence **group effectiveness** are its mix of people, its organization and the quality of communication among its members.

  (cont’d)
Key points (cont’d)

- **Communication** within a group is influenced by factors such as:
  - the status of group members,
  - the size of the group,
  - the gender composition of the group,
  - the personalities within the group, and
  - the available communication channels.
Chapter 22

Project Management