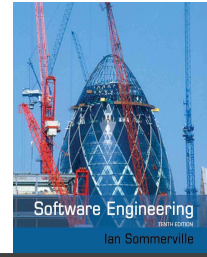


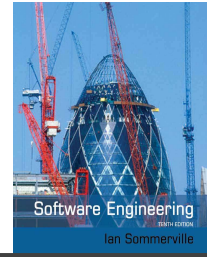
Project Management

Topics covered



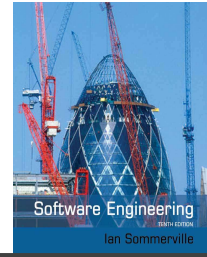
- ✧ Risk management
- ✧ Managing people
- ✧ Teamwork

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 organisations 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 organisation developing the software.

Success criteria



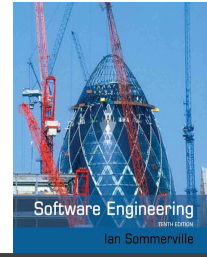
- ✧ Deliver the software to the customer at the agreed time.
- ✧ Keep overall costs within budget.
- ✧ Deliver software that meets the customer's expectations.
- ✧ Maintain a coherent and well-functioning development team.



Software management distinctions

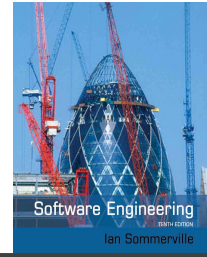
- ✧ The product is intangible.
 - Software cannot be seen or touched. Software project managers cannot see progress by simply looking at the artefact that is being constructed.
- ✧ Many software projects are 'one-off' projects.
 - Large software projects are usually different in some ways from previous 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.
- Covered in next lecture.

✧ *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



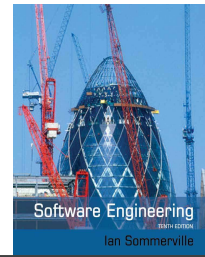
- ✧ Risk management is concerned with identifying risks and drawing up plans to minimise their effect on a project.
- ✧ Software risk management is important because of the inherent uncertainties in software development.
 - These uncertainties stem from loosely defined requirements, requirements changes 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



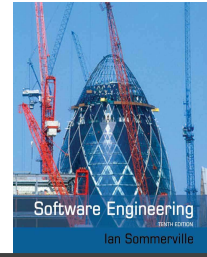
- ✧ There are two dimensions of risk classification
 - The type of risk (technical, organizational, ..)
 - what is affected by the risk:
- ✧ *Project risks* affect schedule or resources (e.g. experienced designer resigns);
- ✧ *Product risks* affect the quality or performance of the software being developed (e.g. purchased component fails);
- ✧ *Business risks* affect the organisation developing or procuring the software (e.g. competitor introducing a new product).

Examples of project, product, and business risks



| 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. |

The risk management process



✧ Risk identification

- Identify project, product and business risks;

✧ Risk analysis

- Assess the likelihood and consequences of these risks;

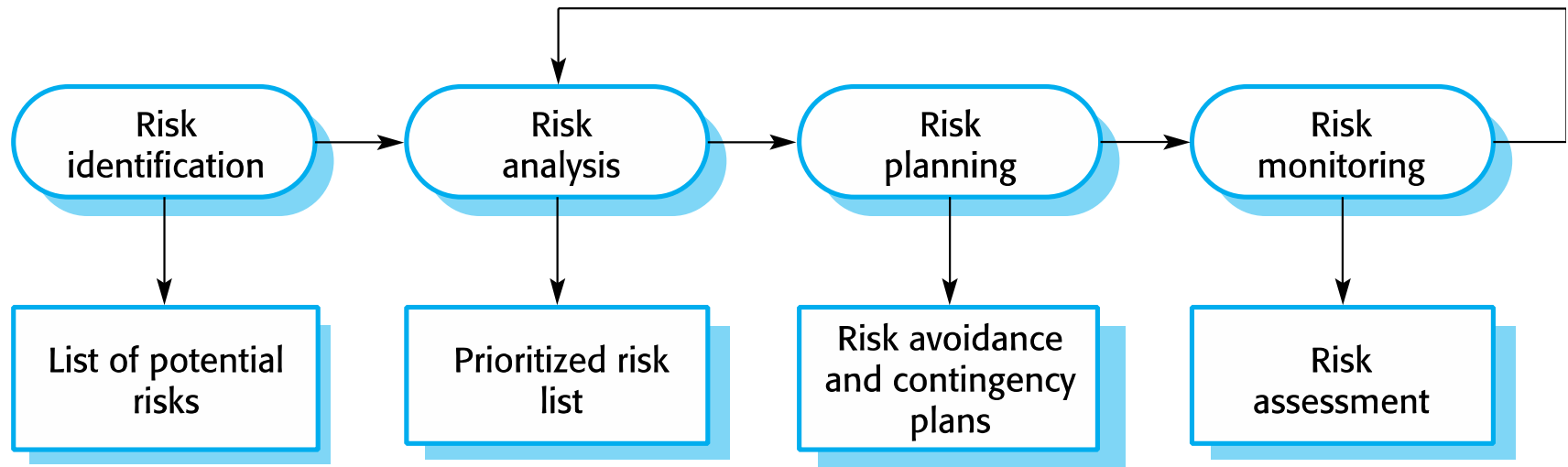
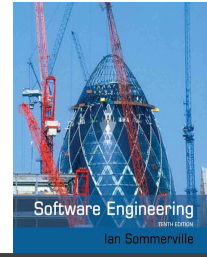
✧ Risk planning

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

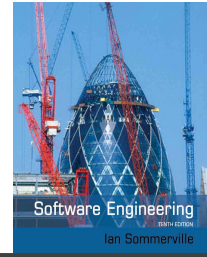
✧ Risk monitoring

- Monitor the risks throughout the project;

The risk management process

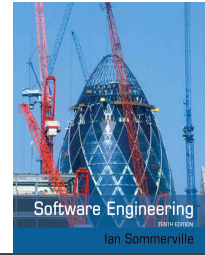


Risk identification



- ✧ 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.

Examples of different risk types



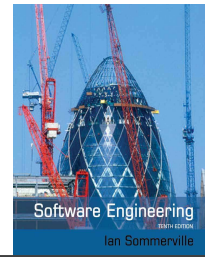
| Risk type | Possible risks |
|----------------|---|
| Estimation | The time required to develop the software is underestimated. (12) The rate of defect repair is underestimated. (13) The size of the software is underestimated. (14) |
| Organizational | The organization is restructured so that different management are responsible for the project. (6) Organizational financial problems force reductions in the project budget. (7) |
| 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) |
| Requirements | Changes to requirements that require major design rework are proposed. (10) Customers fail to understand the impact of requirements changes. (11) |
| 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) |
| Tools | The code generated by software code generation tools is inefficient. (8) Software tools cannot work together in an integrated way. (9) |

Risk analysis



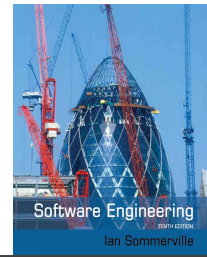
- ✧ Assess probability and seriousness of each risk.
- ✧ Probability may be very low, low, moderate, high or very high.
- ✧ Risk consequences might be catastrophic, serious, tolerable or insignificant.

Risk types and examples



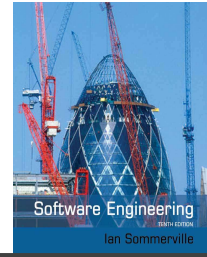
| Risk | Probability | Effects |
|---|-------------|--------------|
| Organizational financial problems force reductions in the project budget (7). | Low | Catastrophic |
| It is impossible to recruit staff with the skills required for the project (3). | High | Catastrophic |
| Key staff are ill at critical 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 |

Risk types and examples



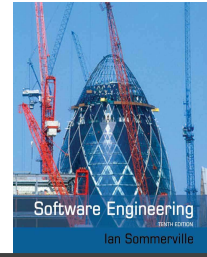
| Risk | Probability | Effects |
|---|-------------|---------------|
| The time required to develop the software is underestimated (12). | High | Serious |
| Software tools cannot be integrated (9). | High | Tolerable |
| Customers fail to understand the impact of requirements changes (11). | Moderate | 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 |

Risk planning



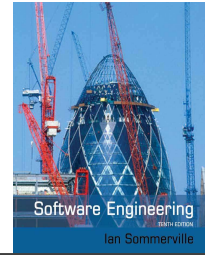
- ✧ Consider each risk and develop a strategy to manage that risk.
- ✧ Avoidance strategies
 - The probability that the risk will arise is reduced;
- ✧ Minimization strategies
 - The impact of the risk on the project or product will be reduced;
- ✧ Contingency plans
 - If the risk arises, contingency plans are plans to deal with that risk;

What-if questions



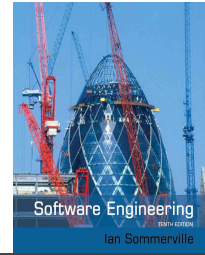
- ✧ What if several engineers are ill at the same time?
- ✧ What if an economic downturn leads to budget cuts of 20% for the project?
- ✧ What if the performance of open-source software is inadequate and the only expert on that open source software leaves?
- ✧ What if the company that supplies and maintains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

Strategies to help manage risk



| 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. |

Strategies to help manage risk



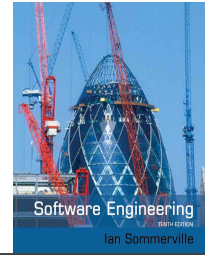
| Risk | Strategy |
|---------------------------------|---|
| Organizational restructuring | Prepare a briefing document for senior management showing how the project is making a very important contribution to the goals of the business. |
| Database performance | Investigate the possibility of buying a higher-performance database. |
| Underestimated development time | Investigate buying-in components; investigate use of a program generator. |

Risk monitoring

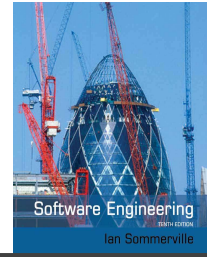


- ✧ Assess each identified risks regularly to decide whether or not it is becoming less or more probable.
- ✧ Also assess whether the effects of the risk have changed.
- ✧ Each key risk should be discussed at management progress meetings.

Risk indicators

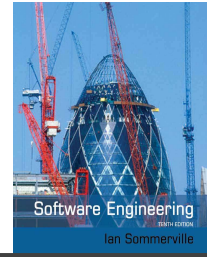


| Risk type | Potential indicators |
|----------------|--|
| Estimation | Failure to meet agreed schedule; failure to clear reported defects. |
| Organizational | Organizational gossip; lack of action by senior management. |
| People | Poor staff morale; poor relationships amongst team members; high 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 CASE tools; demands for higher-powered workstations. |



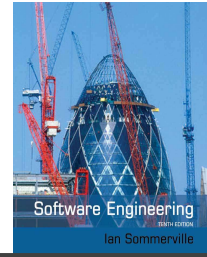
Managing people

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.

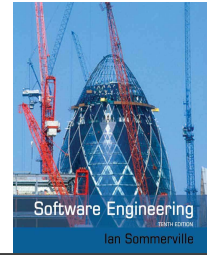
✧ Inclusion

- Involve all team members and make sure that people's views are considered.

✧ Honesty

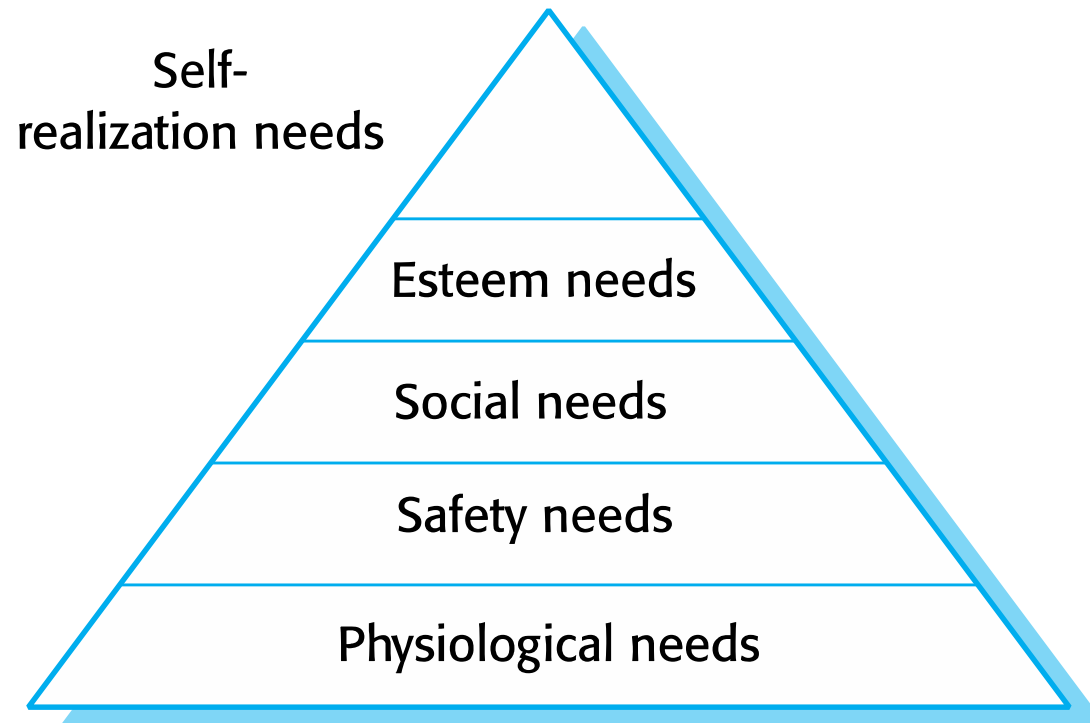
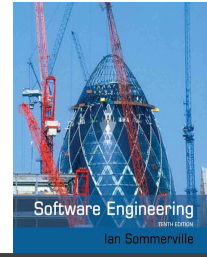
- You should always be honest about what is going well and what is going badly in a project.

Motivating people

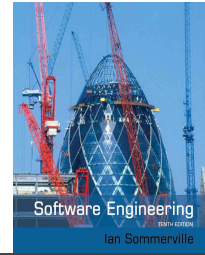


- ✧ An important role of a manager is to motivate the people working on a project.
- ✧ Motivation means organizing the work and the working environment to encourage people to work effectively.
 - If people are not motivated, they will not be interested in the work they are doing. They will work slowly, 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).

Human needs hierarchy



Case study: Individual motivation

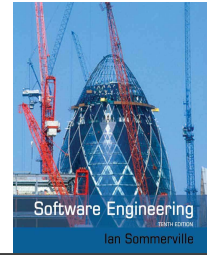


Alice is a software project manager working in a company that develops alarm systems. This company wishes to enter the growing market of assistive technology to help elderly and disabled people live independently. Alice has been asked to lead a team of 6 developers than can develop new products based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

Case study: Individual motivation



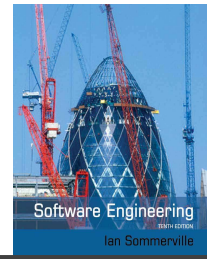
After some initial denials that there is a problem, Dorothy admits that she has lost interest in the job. She expected that she would be able to develop and use her hardware interfacing skills. However, because of the product direction that has been chosen, she has little opportunity for this. Basically, 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.

Comments on case study



- ✧ If you don't sort out the problem of unacceptable work, the other group members will become dissatisfied and feel that they are doing an unfair share of the work.
- ✧ Personal difficulties affect motivation because people can't concentrate on their work. They need time and support to resolve these issues, although you have to make clear that they still have a responsibility to their employer.
- ✧ Alice gives Dorothy more design autonomy and organizes training courses in software engineering that will give her more opportunities after her current project has finished.



Teamwork

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.

Team spirit

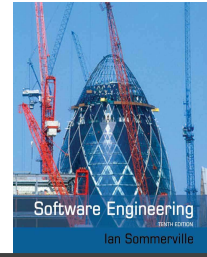


Alice, an experienced project manager, understands the importance of creating a cohesive group. As they are developing a new product, she takes the opportunity of involving all group members in the product specification and design by getting them to discuss possible technology with elderly members of 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.

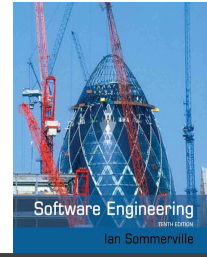
Every few months, Alice organizes an 'away day' for the group where the team spends two days on 'technology updating'. Each team member prepares an update on a relevant technology and presents it to the group. This is an off-site meeting in a good hotel and plenty of time is scheduled for discussion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

Assembling a team



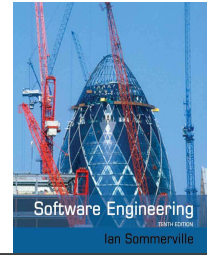
- ✧ May not be possible to appoint the ideal people to work on a project
 - Project budget may not allow for the use of highly-paid staff;
 - Staff with the appropriate experience may not be available;
 - An organisation may wish to develop employee skills on a software project.
- ✧ Managers have to 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 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:

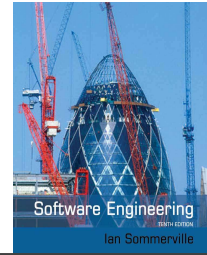
- Alice—self-oriented
- Brian—task-oriented
- Bob—task-oriented
- Carol—interaction-oriented
- Dorothy—self-oriented
- Ed—interaction-oriented
- Fred—task-oriented

Group organization



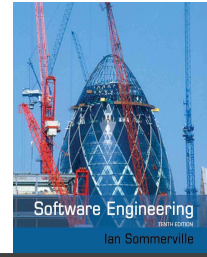
- ✧ Small software engineering groups are usually organised informally without a rigid structure.
- ✧ For large projects, there may be a hierarchical structure where different groups are responsible for different sub-projects.
- ✧ Agile development is always based around an informal group on the principle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

Informal groups

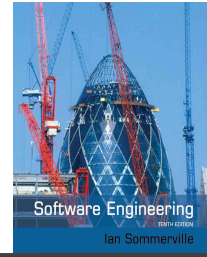


- ✧ The group acts as a whole and comes to a consensus on decisions affecting the system.
- ✧ The group leader serves as the external interface of the group but does not allocate specific work items.
- ✧ Rather, work is discussed by the group as a whole and tasks are allocated according to ability and 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 communications also strengthens group cohesion as it promotes understanding.



Group communications

✧ 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

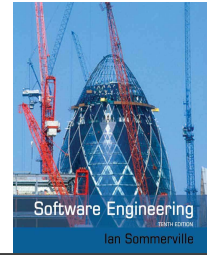
- Good workplace organisation can help encourage communications.

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 processes are not as mature as traditional engineering processes.
- ✧ Risk management involves identifying and assessing project risks to establish the probability that they will occur and the consequences for the project if that risk does arise.

Key points



- ✧ People management involves choosing the right people to work on a project and organizing the team and its working environment.
- ✧ Software development groups should be fairly small and cohesive.
- ✧ Communications within a group are influenced by factors such as the status of group members, the size of the group, the gender composition of the group, personalities and available communication channels.