Software Engineering - Design

Summer Term 2025
11 Project Management

Prof. Dr. Robert Lokaiczyk
Frankfurt University of Applied Sciences
Faculty of Computer Science and Engineering
robert.lokaiczyk@fb2.fra-uas.de

Agenda

Managing people

Managing projects

Managing risk

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.

Source: Ian Sommerville - Software Engineering 2016

Success 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.

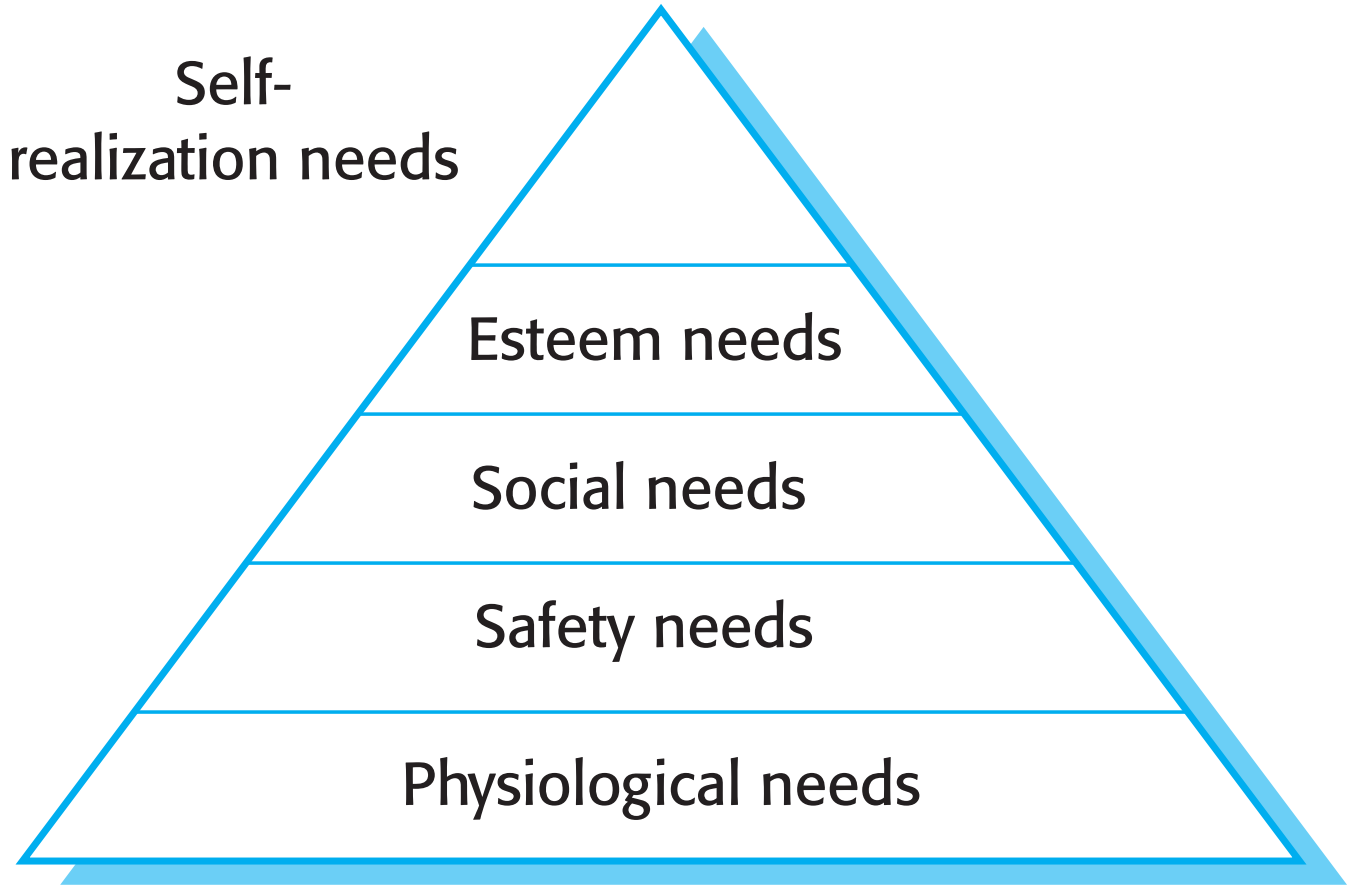
Source: Ian Sommerville - Software Engineering 2016

Motivating people

- An important role of a manager is to **motivate** the people working on a project.
- Motivation means organising 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 organisation.
- Motivation is a complex issue but it appears that their 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).

Source: Ian Sommerville - Software Engineering 2016

The human needs hierarchy



Maslow: A Theory of Human Motivation, 1943

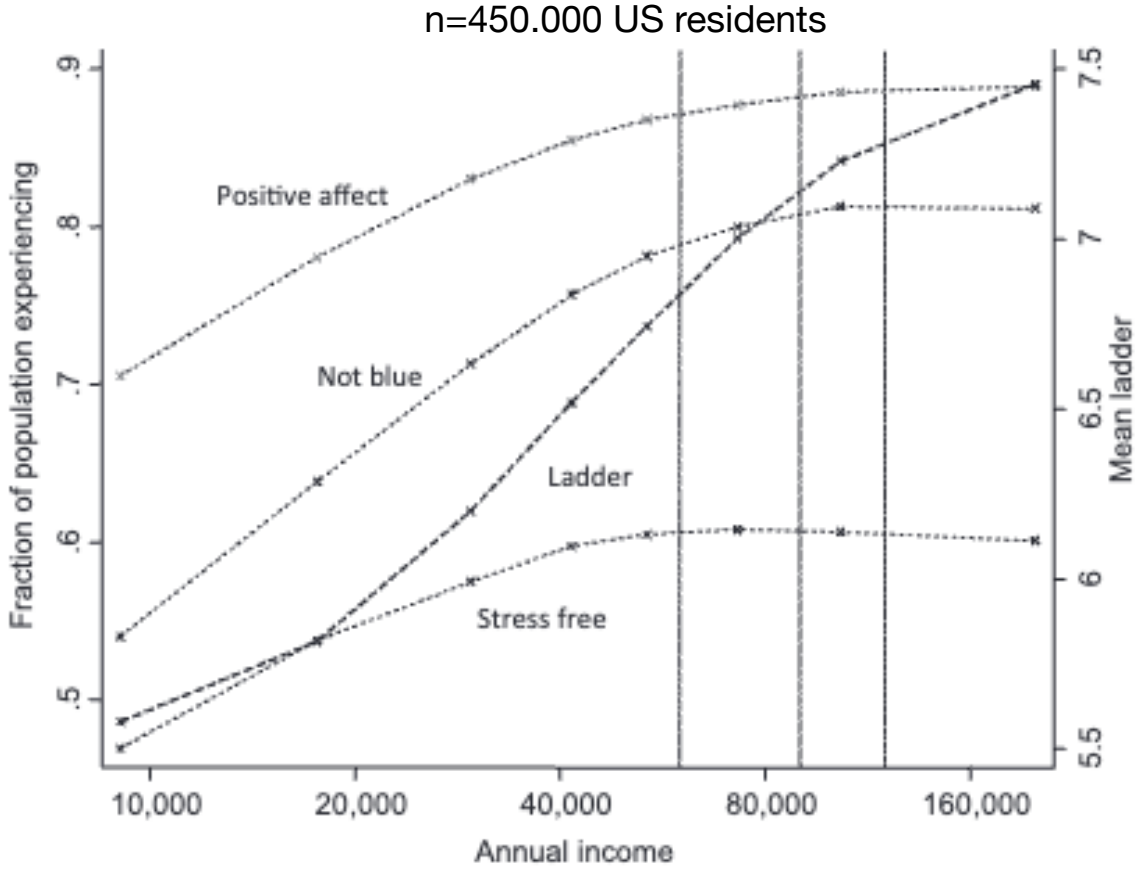
Does money really affect motivation?

Positive affect: reports of happiness, enjoyment, and frequent smiling and laughter

Blue affect: worry and sadness

Cantrill ladder: life evaluation

n=450.000 US residents



Kahnemann: High income improves evaluation of life but note motional well-being, 2010

Does money really affect motivation?

- In software development groups, basic physiological and safety needs are not 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.

Source: Ian Sommerville - Software Engineering 2016

Scenario: 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.

Source: Ian Sommerville - Software Engineering 2016

Scenario: 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.

Source: Ian Sommerville - Software Engineering 2016

Personality types

- The needs hierarchy is almost certainly an over-simplification of motivation in practice.
- Motivation should also take into account different personality types:
  - Task-oriented
  - Self-oriented
  - Interaction-oriented.

Source: Ian Sommerville - Software Engineering 2016

Personality types

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

Source: Ian Sommerville - Software Engineering 2016

Generational "gaps“

Characteristics	Baby Boomers	Generation X	Millenials	Generation Z
Born*	1945 - 1964	1961 - 1981	1975 - 1995	1995 - 2015
Avg. US National Population*	71.2 Million	63.7 Million	83.1 Million	86.6. Million
Family Values & Situations	Strong family values Multi-child families Strong respect towards elders	First latchkey children Increased divorce rates Redefinition of family values	Single-parented children Divorce becomes common Loss of traditions	Single or same sex parents Little attachment to traditional values Perceives family as support
Values & Attributes	Redefinition of traditional values Independent, liberal, experimental, free-spirited	Ethical, independent, adaptable, open to diversity	Optimistic, independent, adventurous, conservative, open-minded, entrepreneurial	Internet savvy, conservative, impatient, creative, innovative, social, visual, adventurous, hyper-aware
Workforce Values	Strong work ethic Loyal to their job Strong group workforce	Loyal to their job Increasing female workforce More individually focused	Seek good work/life balance Strong sense of entitlement	Multitaskers, adventurous Seek flexibility, reality & openness Creative, innovative, entrepreneurial
Global Events	Post-WWII Peace & Love Movement Rock & Roll Assassinations of JFK, RFK & MLK	Berlin Wall Watergate scandal Moon landing Vietnam War Civil Rights Movement	Rise of technology Environmental issues Financial crisis Terrorism	Rise of social media Same sex marriage Climate change Terrorism
Icons	JFK Mickey Mantle	Madonna Michael Jackson	Steve Jobs Mark Zuckerberg	Taylor Swift Roger Federer
Communication Mediums	Face-to-face Radio TV Print & direct mail	Email Mobile Face-to-face TV	Mobile / Smartphones Email Digital Instant messaging	Mobile / Smartphones Digital Facetime, Skype, Hangouts Instant messaging

https://www.bigarrowgroup.com/

Motivational balance

- Individual motivations are made up of elements of **each class**.
- The balance can change depending on personal circumstances and external events.
- However, people are not just motivated by personal factors but also by being **part of a group and culture**.
- People go to work because they are motivated by the people that they work with.

Source: Ian Sommerville - Software Engineering 2016

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.

Source: Ian Sommerville - Software Engineering 2016

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.

Source: Ian Sommerville - Software Engineering 2016

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.

Source: Ian Sommerville - Software Engineering 2016

The effectiveness of a team

- A) 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.
- B) 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.
- C) Technical and managerial communications
  - **Good communications** between group members, and between the software engineering team and other project stakeholders, is essential.

Source: Ian Sommerville - Software Engineering 2016

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.

Source: Ian Sommerville - Software Engineering 2016

Group organization

- 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

Source: Ian Sommerville - Software Engineering 2016

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.

Source: Ian Sommerville - Software Engineering 2016

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.
- Tool support:
  - Chat: Slack, Discord, Mattermost, Rocketchat
  - Video conferencing: Zoom, MS Teams, WebEx, BigBlueButton, Jitsi, GoWebinar
  - Collaboration: Asana, Trello, Jira, ClickUp, Notion, BaseCamp

Source: Ian Sommerville - Software Engineering 2016

Agenda

Managing people

Managing projects

Managing risk

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.

Source: Ian Sommerville - Software Engineering 2016

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 **happy** and well-functioning development **team**.

Source: Ian Sommerville - Software Engineering 2016

Why is a software project different?

- The product is **intangible**.
  - Software cannot be seen or touched. Software project managers cannot see progress by simply looking at the artifact 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 **organisation specific**.
  - We still cannot reliably predict when a particular software process is likely to lead to development problems.

Source: Ian Sommerville - Software Engineering 2016

Management activities

- **Project planning**
  - Project managers are responsible for planning, estimating and scheduling project development and assigning people to tasks.
- **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.
- **Risk management**
  - Project managers assess the risks that may affect a project, monitor these risks and take action when problems arise.

Source: Ian Sommerville - Software Engineering 2016

Management activities

- **People management**
  - Project managers have to choose people for their team and establish ways of working that leads to effective team performance
- **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.

Source: Ian Sommerville - Software Engineering 2016

Gantt chart

Example

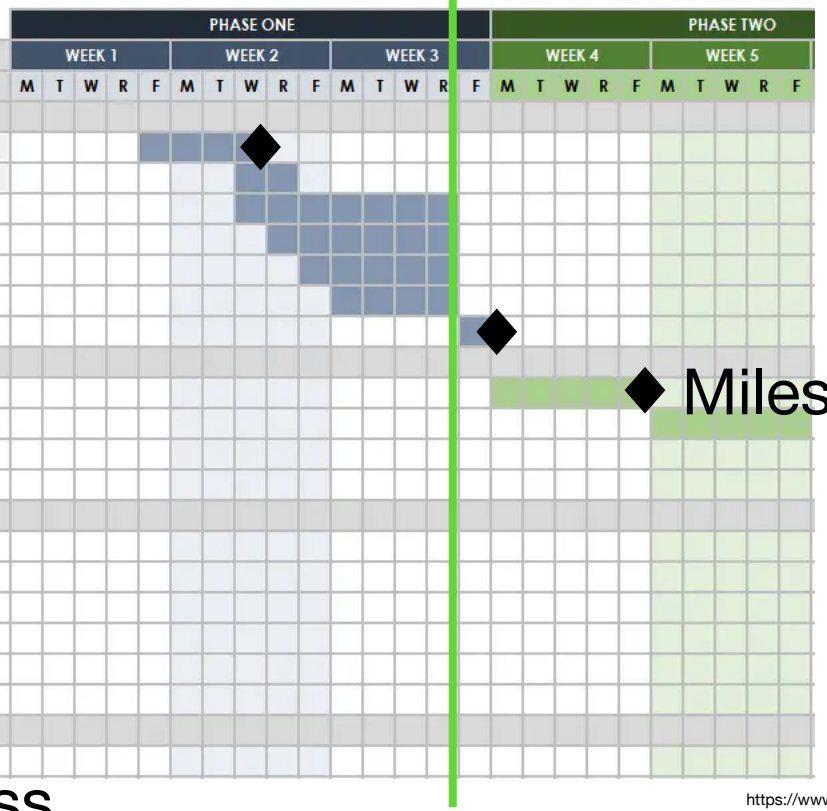
MARKETING CAMPAIGN GANTT CHART EXAMPLE

PROJECT TITLE

Project Alpha

PROJECT MANAGER	COMPANY NAME	DATE
John K.	Weller Corp.	MM/DD/YY

TASK ID	TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION IN DAYS	PCT OF TASK COMPLETE
1	Project Conception and Initiation					
1.1	Establish Goals	Leon W	03/12/26	03/15/26	4	100%
1.1.1	Determine Target Audience	Kylie R	03/15/26	03/16/26	2	100%
1.2	Develop Creative Concept	Pete S	03/15/26	03/21/26	7	90%
1.3	Identify Marketing Channels	Steve L	03/16/26	03/22/26	7	40%
1.4	Budget and Financial Projection	Allen W	03/17/26	03/22/26	6	70%
1.5	Set up Campaign Calendar	Malik M	03/18/26	03/22/26	5	60%
1.6	Creative Brief	Malik M	03/23/26	03/23/26	1	50%
2	Creative Development					
2.1	Message Planning	Steve L	03/24/26	03/28/26	5	5%
2.2	Create Marketing Assets	Allen W	03/29/26	04/02/26	5	30%
2.3	Gather Customer Testimonials	Malik M				0%
2.4	Other	Malik M				0%
3	Promotion Plan					
3.1	Email Campaign	Pete S				0%
3.2	Social Media	Leon W				0%
3.2.1	Sales Outreach	Kylie R				0%
3.2.2	Press Release	Kylie R				0%
3.3	Print Advertising	Pete S				0%
3.3.1	Other	Malik M				0%
4	Testing					
4.1	Split Testing	Steve L				0%



Responsible

Progress

https://www.smartsheet.com/content/gantt-chart-examples

Agenda

Managing people

Managing projects

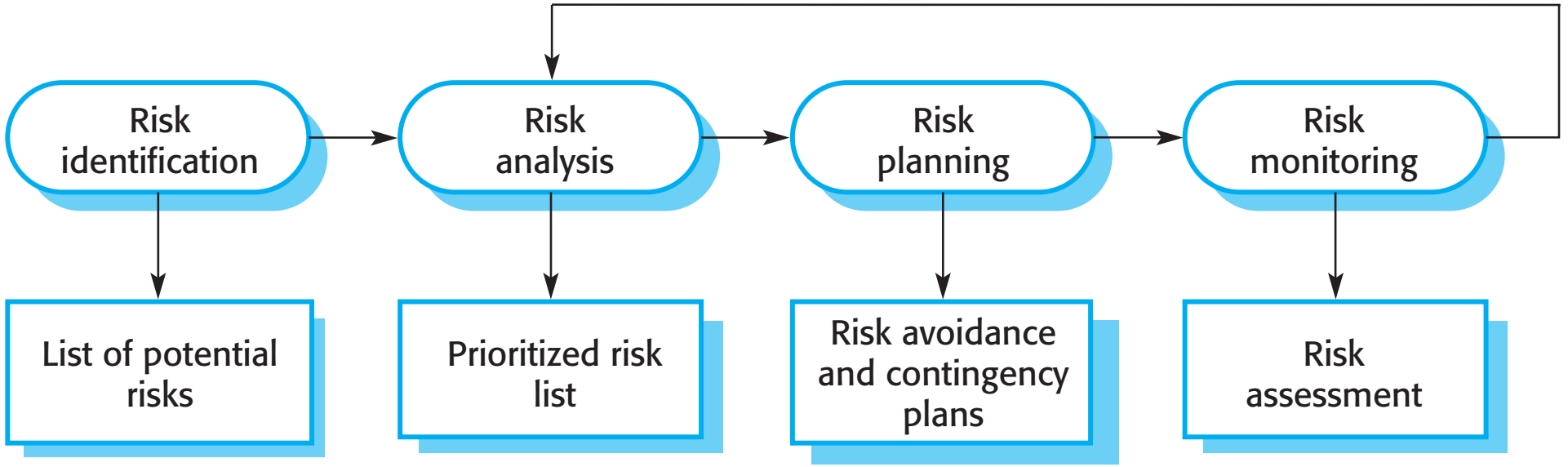
Managing risk

Risk management

- Risk management is concerned with **identifying risks and drawing up plans to minimise their effect** on a project.
- A risk **is a probability that some adverse circumstance will occur**
  - 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.

Source: Ian Sommerville - Software Engineering 2016

A risk management process



Source: Ian Sommerville - Software Engineering 2016

Risk list with risk types

Example

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

Source: Ian Sommerville - Software Engineering 2016

Add probability and severity

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

Source: Ian Sommerville - Software Engineering 2016

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;

Source: Ian Sommerville - Software Engineering 2016

Managing 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.
Requirements changes	Derive traceability information to assess requirements change impact; maximize information hiding in the design.

Source: Ian Sommerville - Software Engineering 2016

Risk planning

- **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.

Source: Ian Sommerville - Software Engineering 2016

Final remarks

- **What did you learn today?**
  - How to manage and motivate different types of people.
  - How to manage a project
  - Read and write Gantt charts
  - How to manage risk

- **Next week**: Preparation and questions for final exam
- **Please read**: Slides for all lectures again. Find open questions.