Introduction to Software Engineering

Project Management and Planning

Topics covered

- Risk management
- Managing people
- Teamwork

Project Management

- Software pricing
- Plan-driven development
- Project scheduling

Project Planning

Project Management: Risk Management

Project management goals

- 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**.

General 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.

General 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 classification

- **Project risks**
  - Risks that affect the project schedule or resources
  - Example: Main software architect quits

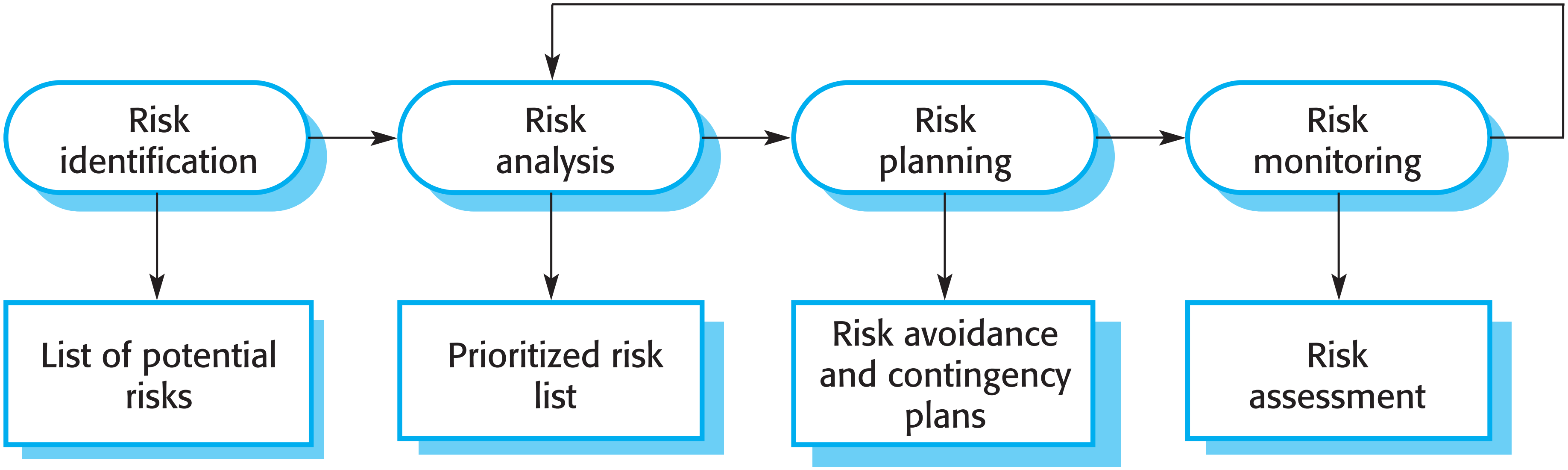
- **Product risks**
  - Affect the quality or performance of the software being developed
  - Example: Failure of a purchased component to perform as expected

- **Business risks**
  - Affect the organisation developing or procuring the software
  - Example: Competitor release their product faster.

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 activity or the individual project manager’s responsibility.

- A checklist of common risks may be used to identify risks in a project, e.g.:
  - Estimation risks
  - Organizational risks
  - People risks
  - Requirements risks
  - Technology risks
  - Tool risks

Examples of different risk types

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

Risk analysis

- Assess **probability** (i.e. likelihood) and **seriousness** (i.e. consequences) of each risk.

- Exemplary risk probability scale:
  - very low (<10%), low (10-25%), moderate (25-50%), high (50-75%) or very high (>75%).

- Exemplary risk consequence scale:
  - Catastrophic (threaten the survival of the project), serious (would cause major delays), tolerable (delays are within allowed contingency) or insignificant.

Risk types and examples

Risk	Probability	Effects
Organizational financial problems force reductions in the project budget.	Low	Catastrophic
It is impossible to recruit staff with the skills required for the project.	High	Catastrophic
Key staff are ill at critical times in the project.	Moderate	Serious
Faults in reusable software components have to be repaired before these components are reused.	Moderate	Serious
Changes to requirements that require major design rework are proposed.	Moderate	Serious
The organization is restructured so that different management are responsible for the project.	High	Serious
The database used in the system cannot process as many transactions per second as expected.	Moderate	Serious

Risk types and examples

Risk	Probability	Effects
The time required to develop the software is underestimated.	High	Serious
Software tools cannot be integrated.	High	Tolerable
Customers fail to understand the impact of requirements changes.	Moderate	Tolerable
Required training for staff is not available.	Moderate	Tolerable
The rate of defect repair is underestimated.	Moderate	Tolerable
The size of the software is underestimated.	High	Tolerable
Code generated by code generation tools is inefficient.	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 (as close to zero as possible)
- **Minimization** strategies
  - The impact of the risk on the project or product will be reduced
- **Contingency** strategies
  - 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
Staff illness	Reorganize team so that there is more overlap of work and people therefore understand each other's jobs.
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.
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.

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.

Project Management: 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.

Project Management: Teamwork

Teamwork

- Most software engineering tasks are a **group activities**.
- A good group 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.

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 communication between group members, and between the software engineering team and other project stakeholders, is essential.

Project Planning: Software Pricing

Software pricing

- Main parameters used to compute the cost of software development
  - Effort costs (paying for software engineers and managers)
  - Hardware and software costs, including maintenance
  - Travel and training cost

- There is not a simple relationship between the development cost and the price charged to the customer

- Broader organisational, economic, political and business considerations influence the price charged

Factors affecting software pricing

Factor	Description
Market opportunity	A development organization may quote a low price because it wishes to **move into a new segment of the software market**. Accepting a low profit on one project may give the organization the opportunity to make a greater profit later. The experience gained may also help it develop new products.
Requirements volatility	If the requirements are likely to change, an organization may lower its price to win a contract. After the contract is awarded, high prices can be charged for changes to the requirements.

Factors affecting software pricing

Factor	Description
Contractual terms	A customer may be willing to allow the developer to retain ownership of the source code and reuse it in other projects. The price charged may then be less than if the software source code is handed over to the customer.
Financial health	Developers in financial difficulty may lower their price to gain a contract. It is better to make a smaller than normal profit or break even than to go out of business. Cash flow is more important than profit in difficult economic times.

Pricing strategies

- Under pricing
  - A company may underprice a system in order to **gain a contract** that allows them to retain staff for future opportunities
  - A company may underprice a system to **gain access to a new market** area

- Increased pricing
  - The price may be increased when a buyer wishes a fixed-price contract and so the seller increases the price to allow for **unexpected risks**

Estimation techniques

- The two types of technique to estimate software effort and cost:
  - ***Experience-based techniques.*** The estimate of future effort requirements is based on the manager's experience of past projects and the application domain. Essentially, the manager makes an informed judgment of what the effort requirements are likely to be.
  - ***Algorithmic cost modeling*** In this approach, a formulaic approach is used to compute the project effort based on estimates of product attributes such as size, experience of staff involved etc.

Project Planning:
Plan-Driven Development

Plan-driven development

- Plan-driven or plan-based development is an approach to software engineering where the **development process is planned** in detail.
  - Plan-driven development is the 'traditional' way of managing large software development projects.

- A project plan is created that records the work to be done, who will do it, the development schedule and the work products.

- Managers use the plan to support project decision making and as a way of measuring progress.

Plan-driven development: Pros and Cons

- The arguments in favor of a plan-driven approach are:
  - Early planning allows organizational issues (availability of staff, other projects, etc.) to be closely taken into account
  - Potential problems and dependencies are discovered before the project starts, rather than once the project is underway.

- The principal argument against plan-driven development is:
  - Many early decisions have to be revised because of changes to the environment in which the software is to be developed and used.

Project plans

- In a plan-driven development project, a project plan sets out the
  - **Resources** available to the project
  - The **work breakdown**
  - **Schedule** for carrying out the work.

- Exemplary sections in a **principal project plan**:
  - Introduction
  - Project organization
  - Risk analysis
  - Hardware and software resource requirements
  - Work breakdown
  - Project schedule
  - Monitoring and reporting mechanisms

Agile planning

- Agile methods of software development are iterative approaches where the software is developed and delivered to customers in **increments**.

- Unlike plan-driven approaches, the functionality of these increments is decided during the development.
  - The decision on what to include in an increment depends on progress and on the customer's priorities.

- The customer's priorities and requirements change so it makes sense to have a **flexible plan** that can accommodate these changes.

Project Planning:
Project Scheduling

Project scheduling

- Project scheduling is the process of deciding how the work in a project will be organized as
  - Separate **tasks**
  - **When** these tasks will be executed
  - **How** these tasks will be executed

- You estimate the calendar time needed to complete each task, the effort required and **who** will work on the tasks that have been identified.

- You also have to estimate the **resources** needed to complete each task such as:
  - Hardware
  - Software
  - People
  - Travel budget

Schedule presentation

- **Graphical notations** are normally used to illustrate the project schedule.
- These show the project breakdown into tasks. Tasks should not be too small. They should take about a week or two.
- **Calendar-based (Gantt chart)**
  - Bar charts are the most commonly used representation for project schedules. They show the schedule as activities or resources against time.
- **Activity networks**
  - Show task dependencies

Project activities

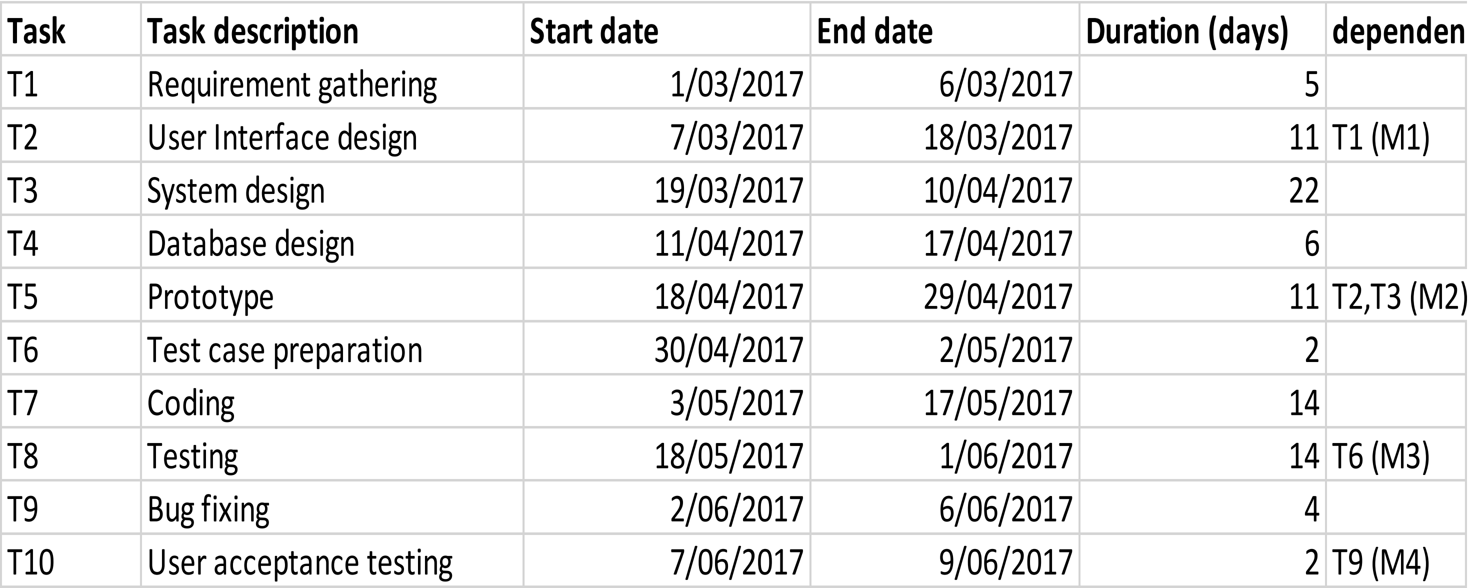
- Project activities (tasks) are the basic planning element. Each activity has:
  - A **duration** in calendar days or months
  - An **effort estimate**, which shows the number of person-days or person-months to complete the work
  - A **deadline** by which the activity should be complete
  - A **defined end-point**, which might be a document, the holding of a review meeting, the successful execution of all tests, etc.

Milestones and deliverables

- **Milestones** are points in the schedule against which you can assess progress
  - E.g., the handover of the system for testing.

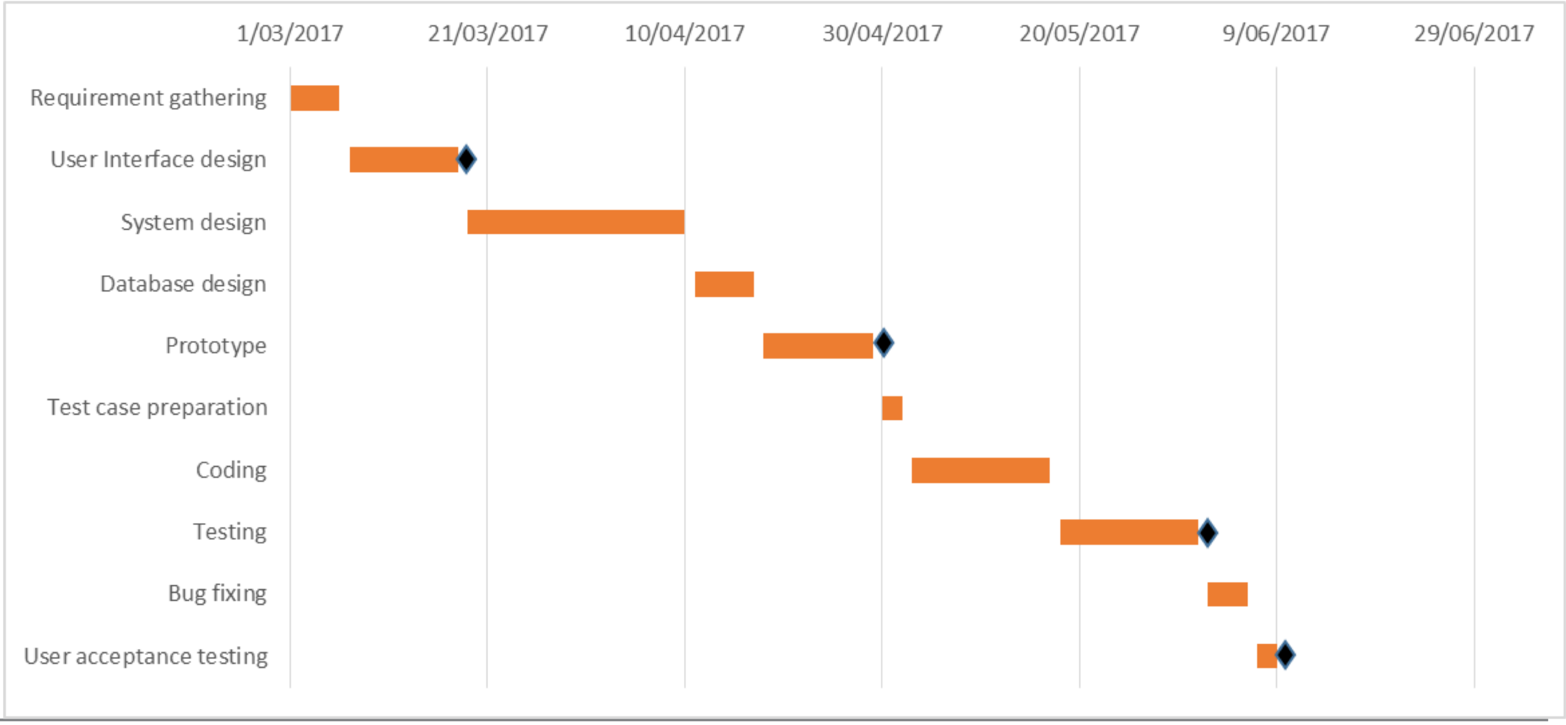
- **Deliverables** are work products that are delivered to the customer
  - E.g., a requirements document for the system.

Table

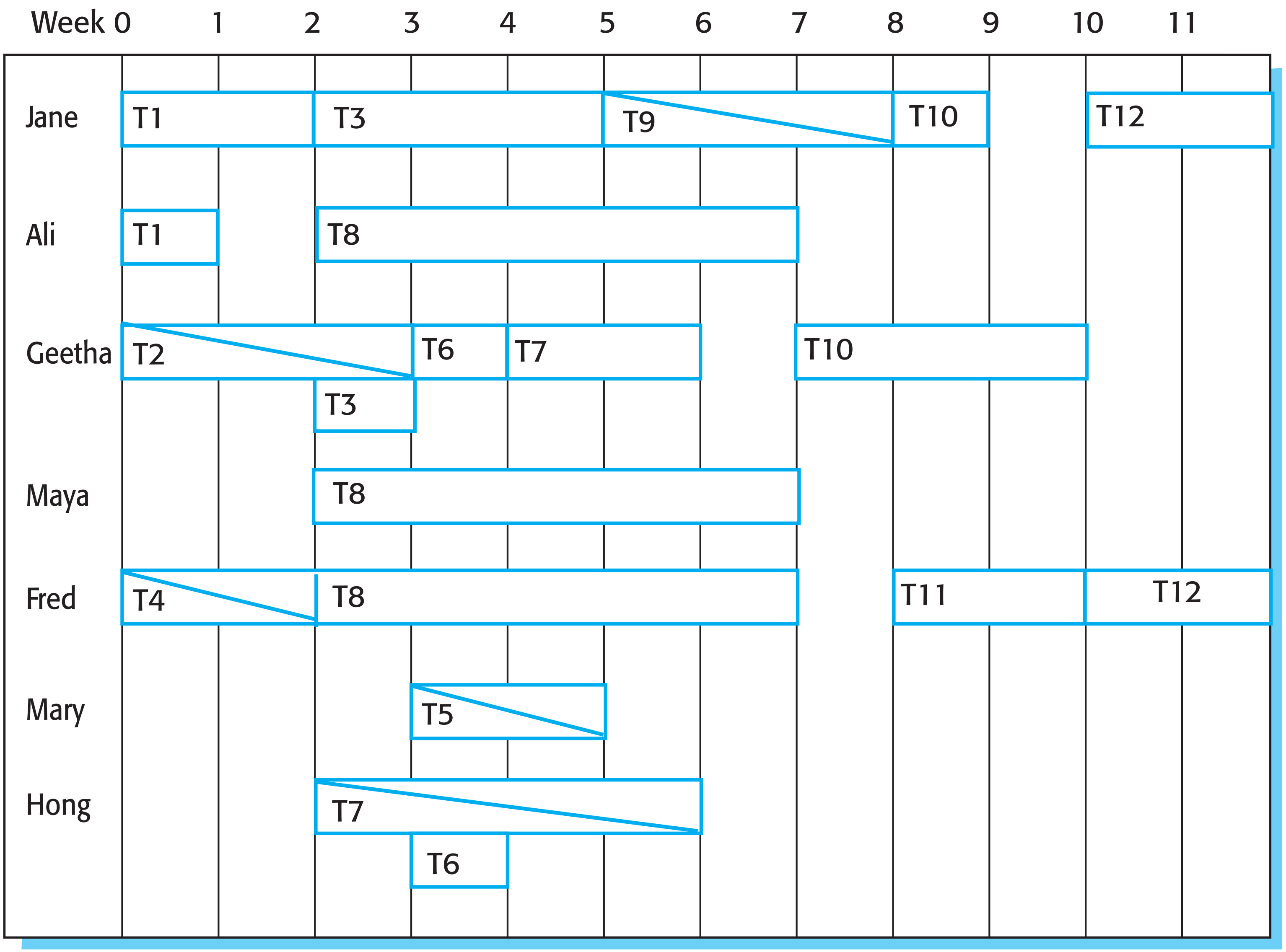


Task	Task description	Start date	End date	Duration (days)	dependen
T1	Requirement gathering	1/03/2017	6/03/2017	5	
T2	User Interface design	7/03/2017	18/03/2017	11	T1 (M1)
T3	System design	19/03/2017	10/04/2017	22	
T4	Database design	11/04/2017	17/04/2017	6	
T5	Prototype	18/04/2017	29/04/2017	11	T2,T3 (M2)
T6	Test case preparation	30/04/2017	2/05/2017	2	
T7	Coding	3/05/2017	17/05/2017	14	
T8	Testing	18/05/2017	1/06/2017	14	T6 (M3)
T9	Bug fixing	2/06/2017	6/06/2017	4	
T10	User acceptance testing	7/06/2017	9/06/2017	2	T9 (M4)

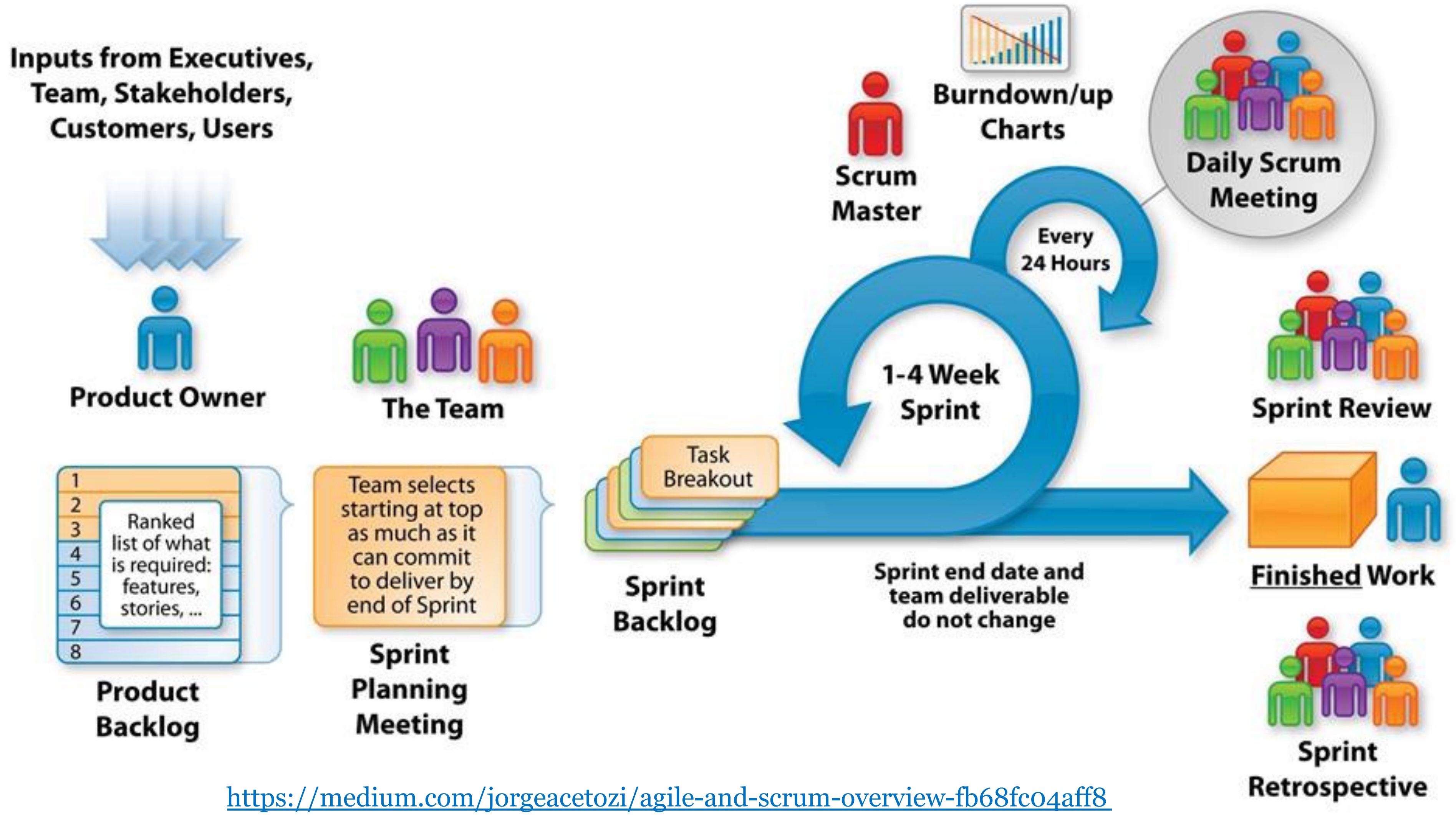
Gantt chart



Staff allocation chart



The Agile - Scrum Framework



Inputs from Executives, Team, Stakeholders, Customers, Users

Product Owner

The Team

Scrum Master

Burndown/up Charts

Daily Scrum Meeting

Every 24 Hours

1-4 Week Sprint

Sprint Review

1
2
3
4
5
6
7
8

Ranked list of what is required: features, stories, ...

Product Backlog

Team selects starting at top as much as it can commit to deliver by end of Sprint

Sprint Planning Meeting

Task Breakout

Sprint Backlog

Sprint end date and team deliverable do not change

Finished Work

Sprint Retrospective

https://medium.com/jorgeacetozi/agile-and-scrum-overview-fb68fc04aff8

Summary

Summary

- Good project management is important to software engineering projects.
- **Risk management** involves identifying and assessing project risks. Plan to avoid, manage or deal with likely risks.
- Key factors are **people**, **organisation,** and **communication**.
- The system **price** includes estimated development costs, the company's profits, and **organizational factors**.

Key Points: Project Planning

- **Plan-driven development** focuses on the project plan, defining activities, effort, the schedule, and human resources.

- **Agile planning** involves the whole team in project planning.

- **Project scheduling** involves the creation of various graphical representations of part of the project plan.

- A **project milestone** is a predictable outcome of an activity or set of activities. A **deliverable** is a work product that is delivered to the project customer.