# Topic 4 – Project Management & Planning

# PROJECT MANAGEMENT

# Topics covered

- ✧ Risk management
- ✧ Managing people
- ✧ Teamwork

What is the largest you worked on?

How much time did it took you?

What where its main parts?

Did you collaborate with someone?

What where the deliverables?

# Software project management

## ✧ What is it?

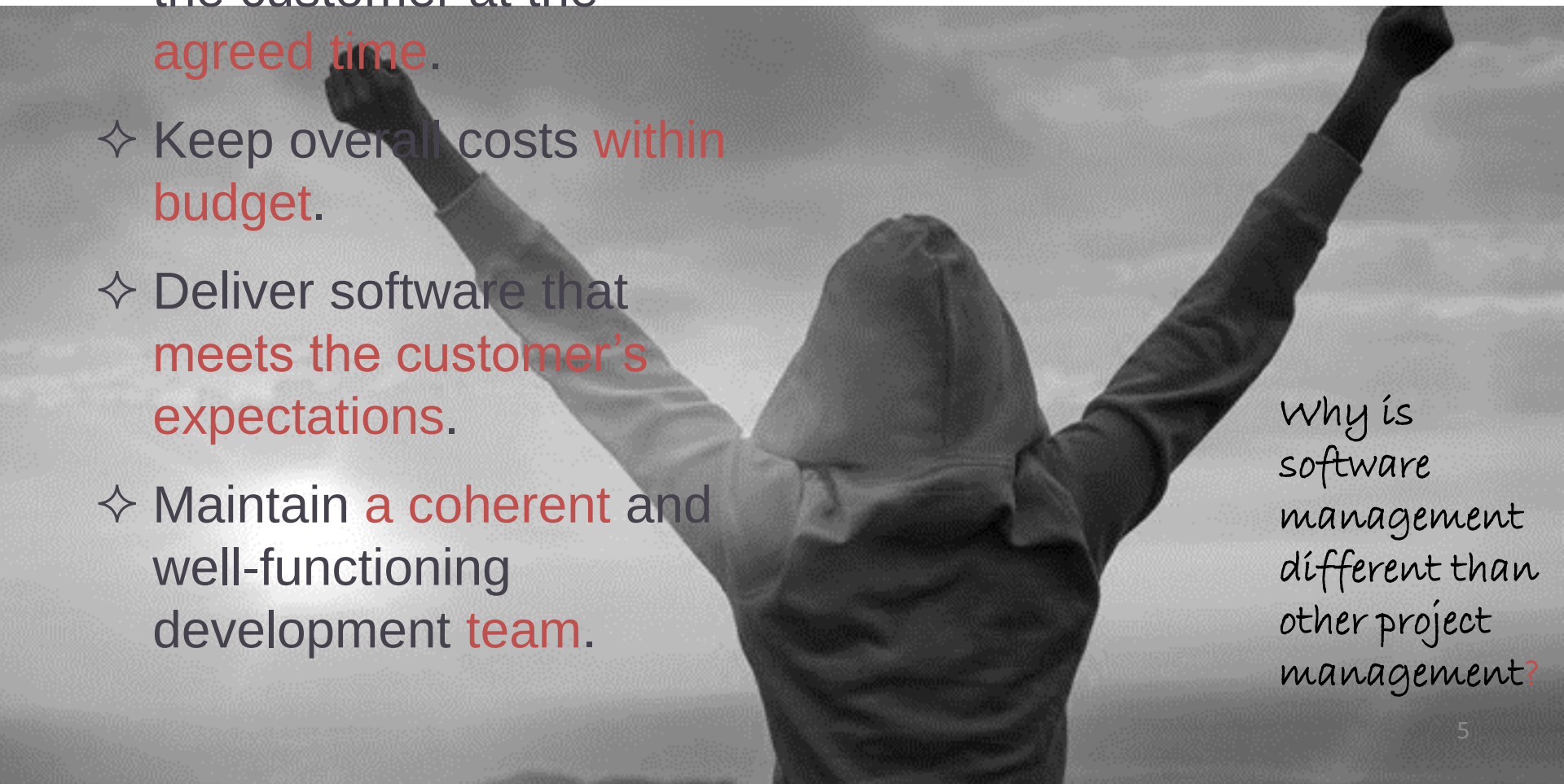
- Concerned with activities involved in ensuring that software is delivered on time and on schedule and in accordance with the requirements of the organisations developing and procuring the software.

## ✧ Why is it important?

- 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.



Why is software management different than other project management?

# Software management distinctions. Why is it different?

- ✧ The product is intangible.
  - Software cannot be seen or touched. Software project managers cannot see progress by simply looking at the artefact that is being constructed.
- ✧ Many software projects are 'one-off' projects.
  - Large software projects are usually different in some ways from previous projects. Even managers who have lots of previous experience may find it difficult to anticipate problems.
- ✧ Software processes are variable and organization specific.
  - We still cannot reliably predict when a particular software process is likely to lead to development problems.

## Factors influencing project management

- ✧ Company size
- ✧ Software customers
- ✧ Software size
- ✧ Software type
- ✧ Organizational culture
- ✧ Software development processes

These factors mean that project managers in different organizations may work in quite different ways.

# Universal management activities

## ✧ *Project planning*

- Project managers are responsible for planning, estimating and scheduling project development and assigning people to tasks.

## ✧ *Risk management*

- Project managers assess the risks that may affect a project, monitor these risks and take action when problems arise.

## ✧ *People management*

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

# Management activities

## ✧ *Reporting*

- Project managers are usually responsible for reporting on the progress of a project to customers and to the managers of the company developing the software.

## ✧ *Proposal writing*

- The first stage in a software project may involve writing a proposal to win a contract to carry out an item of work. The proposal describes the objectives of the project and how it will be carried out.

# Risk management

# Risk management

- Risk management is concerned with identifying risks and drawing up plans to minimise their effect on a project.
- Software risk management is important because of the inherent uncertainties in software development.
  - These uncertainties stem from loosely defined requirements, requirements changes due to changes in customer needs, difficulties in estimating the time and resources required for software development, and differences in individual skills.
- You have to anticipate risks, understand the impact of these risks on the project, the product and the business, and take steps to avoid these risks.

# Risk classification

- There are two dimensions of risk classification
  - The type of risk (technical, organizational, ..)
  - what is affected by the risk:
- *Project risks* affect schedule or resources;
- *Product risks* affect the quality or performance of the software being developed;
- *Business risks* affect the organisation developing or procuring the software.

# Examples of project, product, and business risks

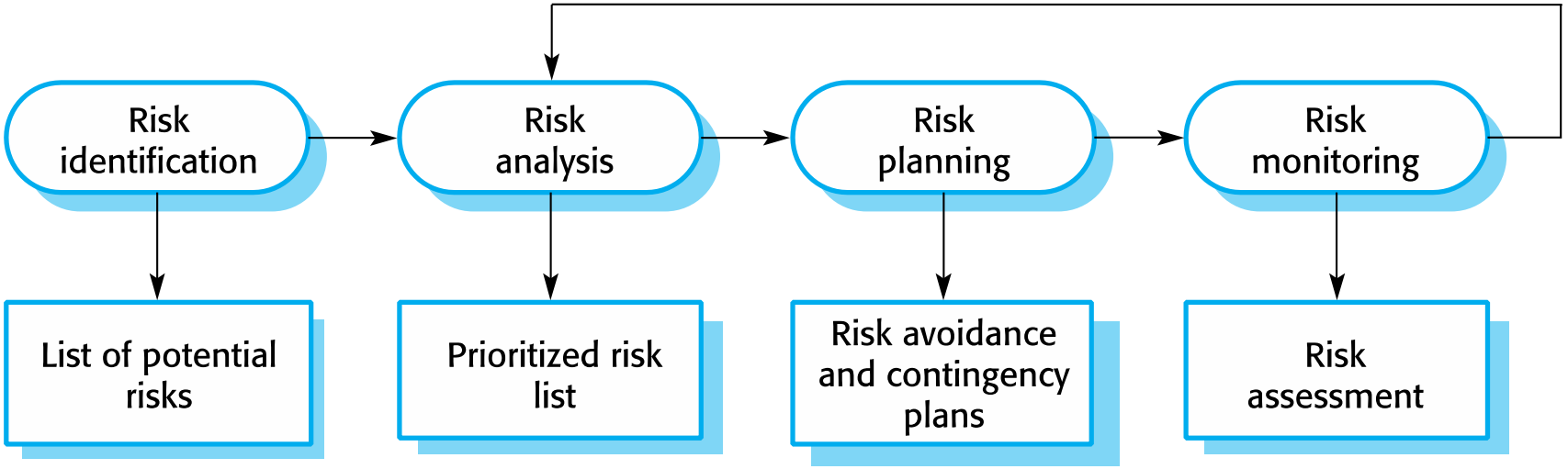
| Risk | Affects | Description |
|---|---|---|
| Staff turnover | Project | Experienced staff will leave the project before it is finished. |
| Management change | Project | There will be a change of organizational management with different priorities. |
| Hardware unavailability | Project | Hardware 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. |

*Given a specific project, can you identify the potential risks?*

# 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)<br>The rate of defect repair is underestimated. (13)<br>The size of the software is underestimated. (14) |
| Organizational | The organization is restructured so that different management are responsible for the project. (6)<br>Organizational financial problems force reductions in the project budget. (7) |
| People | It is impossible to recruit staff with the skills required. (3)<br>Key staff are ill and unavailable at critical times. (4)<br>Required training for staff is not available. (5) |
| Requirements | Changes to requirements that require major design rework are proposed. (10)<br>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)<br>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)<br>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. |
| 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. |

# Risk monitoring

- Assess each identified risk 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.

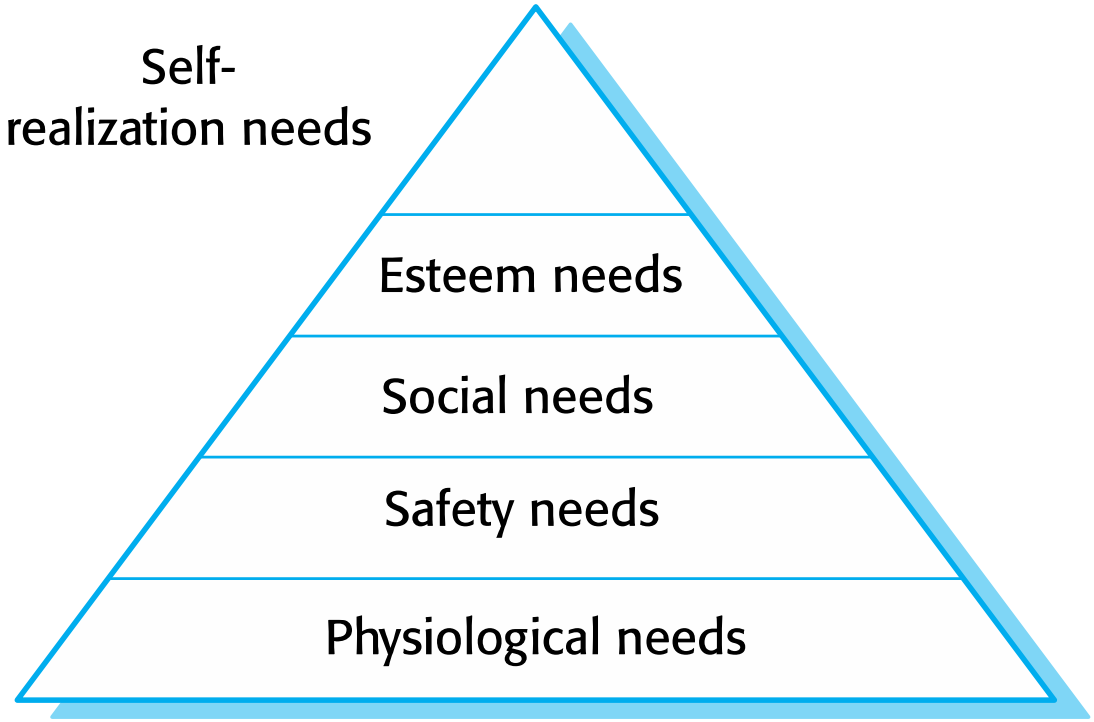
## ✧ Honesty

- 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 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).

# Human needs hierarchy



## Need satisfaction

- 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.

## Personality types

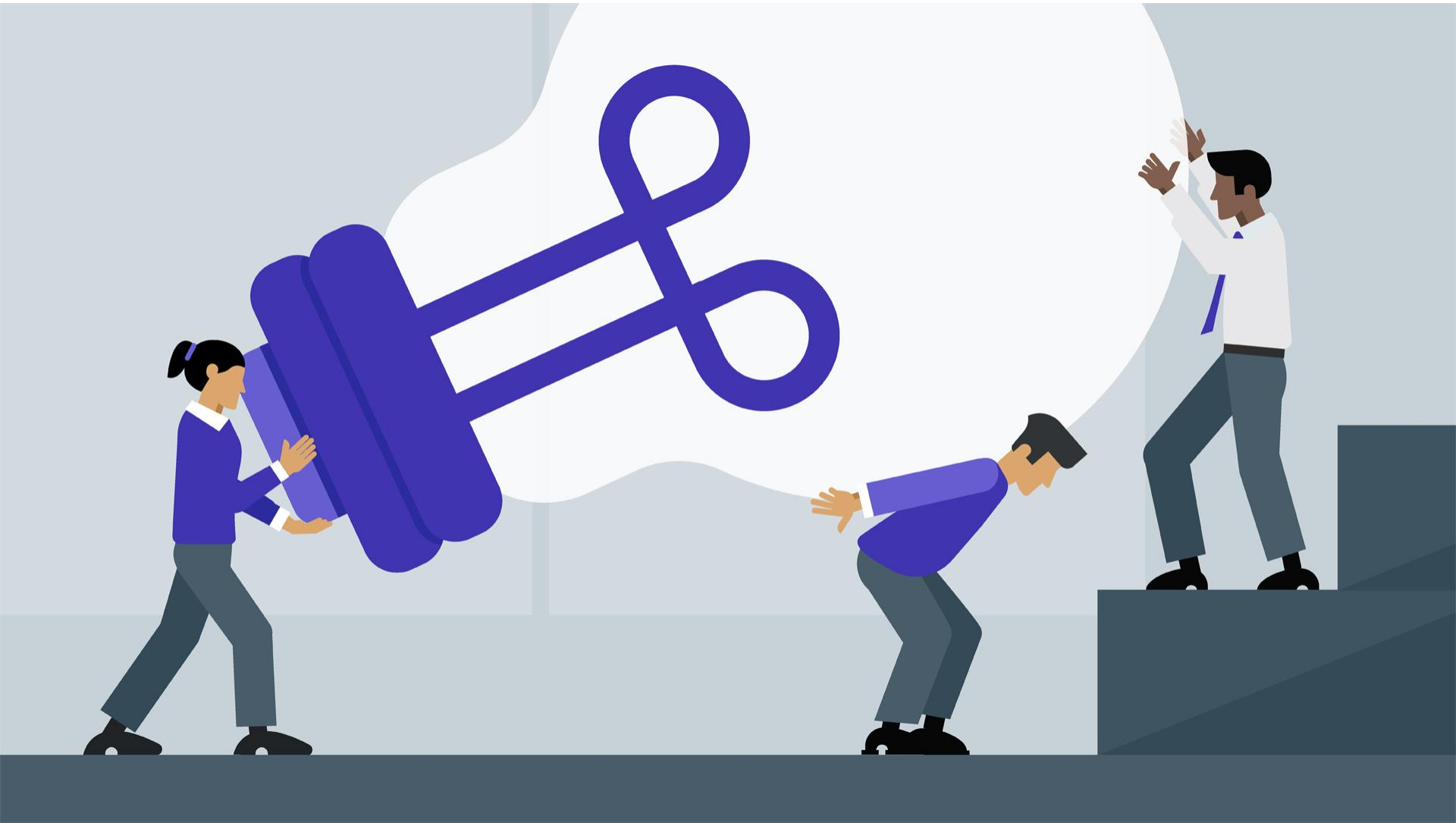
- The needs hierarchy is almost certainly an over-simplification of motivation in practice.
- Motivation should also take into account different personality types:
  - Task-oriented people, who are motivated by the work they do
  - Interaction-oriented people, who are motivated by the presence and actions of co-workers.
  - Self-oriented people, who are principally motivated by personal success and recognition.

# 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.

# What is your type?

# 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.

# Group cohesiveness

- In a cohesive group, members consider the group to be more important than any individual in it.
- The advantages of a cohesive group are:
  - Group quality standards can be developed by the group members.
  - Team members learn from each other and get to know each other's work; Inhibitions caused by ignorance are reduced.
  - Knowledge is shared. Continuity can be maintained if a group member leaves.
  - Refactoring and continual improvement is encouraged. Group members work collectively to deliver high quality results and fix problems, irrespective of the individuals who originally created the design or program.

# The effectiveness of a team

- The people in the group
  - You need a mix of people in a project group as software development involves diverse activities such as negotiating with clients, programming, testing and documentation.
- The group organization
  - A group should be organized so that individuals can contribute to the best of their abilities and tasks can be completed as expected.
- Technical and managerial communications
  - Good communications between group members, and between the software engineering team and other project stakeholders, is essential.

## Selecting group members

- A manager or team leader's job is to create a cohesive group and organize their group so that they can work together effectively.
- This involves creating a group with the right balance of technical skills and personalities, and organizing that group so that the members work together effectively.

# Assembling a team

- May not be possible to appoint the ideal people to work on a project
  - Project budget may not allow for the use of highly-paid staff;
  - Staff with the appropriate experience may not be available;
  - An organisation may wish to develop employee skills on a software project.
- Managers have to work within these constraints especially when there are shortages of trained staff.

## Group composition

- Group composed of members who share the same motivation can be problematic
  - Task-oriented - everyone wants to do their own thing;
  - Self-oriented - everyone wants to be the boss;
  - Interaction-oriented - too much chatting, not enough work.
- An effective group has a balance of all types.
- This can be difficult to achieve software engineers are often task-oriented.
- Interaction-oriented people are very important as they can detect and defuse tensions that arise.

# Group organization

- The way that a group is organized affects the decisions that are made by that group, the ways that information is exchanged and the interactions between the development group and external project stakeholders.
  - Key questions include:
    - Should the project manager be the technical leader of the group?
    - Who will be involved in making critical technical decisions, and how will these be made?
    - How will interactions with external stakeholders and senior company management be handled?
    - How can groups integrate people who are not co-located?
    - How can knowledge be shared across the group?

## 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.

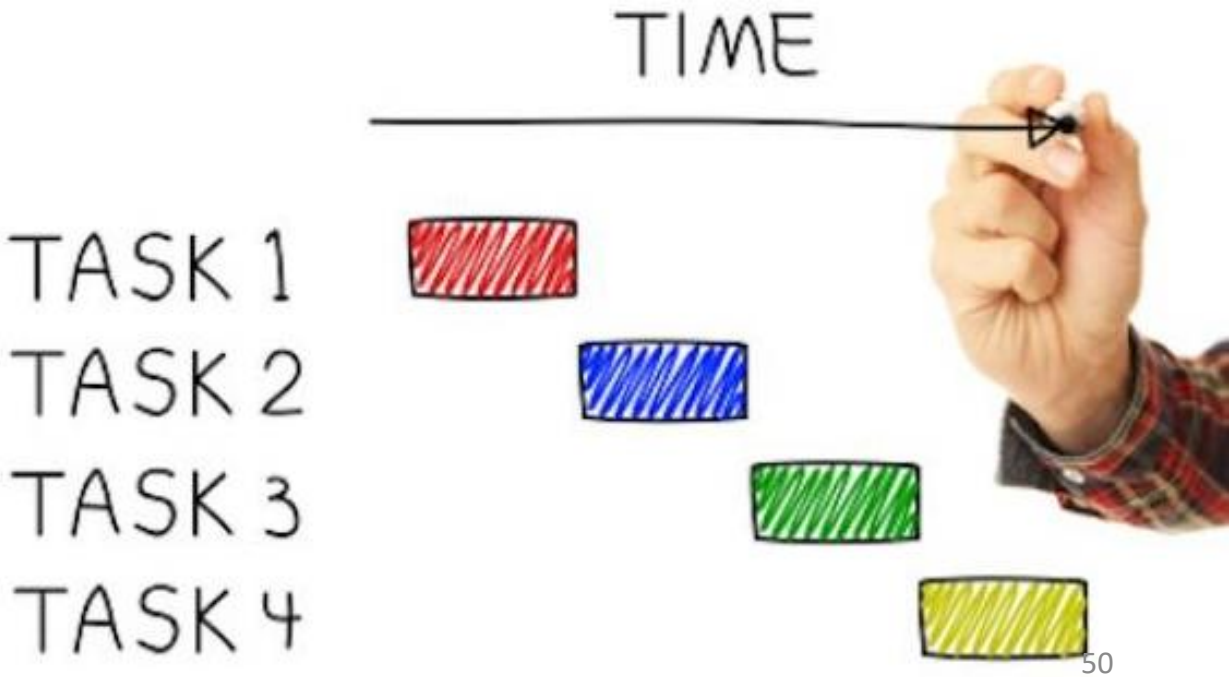
# Key points

- 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 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.
- 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. You should make plans to avoid, manage or deal with likely risks if or when they arise.

# Key points

- People management involves choosing the right people to work on a project and organizing the team and its working environment.
- People are motivated by interaction with other people, the recognition of management and their peers, and by being given opportunities for personal development.
- Software development groups should be fairly small and cohesive. The key factors that influence the effectiveness of a group are the people in that group, the way that it is organized and the communication between group members.
- 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.

## Part B – Project planning



## Topics covered

- ✧ Software pricing
- ✧ Plan-driven development
- ✧ Project scheduling
- ✧ Agile planning

## Project planning

- Project planning involves breaking down the work into parts and assign these to project team members, anticipate problems that might arise and prepare tentative solutions to those problems.
- The project plan, which is created at the start of a project, is used to communicate how the work will be done to the project team and customers, and to help assess progress on the project.

## Planning stages

- At the proposal stage, when you are bidding for a contract to develop or provide a software system.
- During the project startup phase, when you have to plan who will work on the project, how the project will be broken down into increments, how resources will be allocated across your company, etc.
- Periodically throughout the project, when you modify your plan in the light of experience gained and information from monitoring the progress of the work.

## Proposal planning

- Planning may be necessary with only outline software requirements.
- The aim of planning at this stage is to provide information that will be used in setting a price for the system to customers.
- Project pricing involves estimating how much the software will cost to develop, taking factors such as staff costs, hardware costs, software costs, etc. into account

## Project startup planning

- At this stage, you know more about the system requirements but do not have design or implementation information
- Create a plan with enough detail to make decisions about the project budget and staffing.
  - This plan is the basis for project resource allocation
- The startup plan should also define project monitoring mechanisms
- A startup plan is still needed for agile development to allow resources to be allocated to the project

# Development planning

- The project plan should be regularly amended as the project progresses and you know more about the software and its development
- The project schedule, cost-estimate and risks have to be regularly revised

## Software pricing

## Software pricing

- Estimates are made to discover the cost, to the developer, of producing a software system.
  - You take into account, hardware, software, travel, training and effort costs.
- 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 |
| --- | --- |
| 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. |
| Cost estimate uncertainty | If an organization is unsure of its cost estimate, it may increase its price by a contingency over and above its normal profit. |
| 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. |

# 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. |

# 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

# Pricing to win

- The software is priced according to what the software developer believes the buyer is willing to pay
- If this is less that the development costs, the software functionality may be reduced accordingly with a view to extra functionality being added in a later release
- Additional costs may be added as the requirements change and these may be priced at a higher level to make up the shortfall in the original price

# 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 based on engineering project management techniques and 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 that early planning allows organizational issues (availability of staff, other projects, etc.) to be closely taken into account, and that 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 that 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 and a schedule for carrying out the work.
- Plan sections
  - Introduction
  - Project organization
  - Risk analysis
  - Hardware and software resource requirements
  - Work breakdown
  - Project schedule
  - Monitoring and reporting mechanisms

FAILING TO PLAN

=

PLANNING TO FAIL

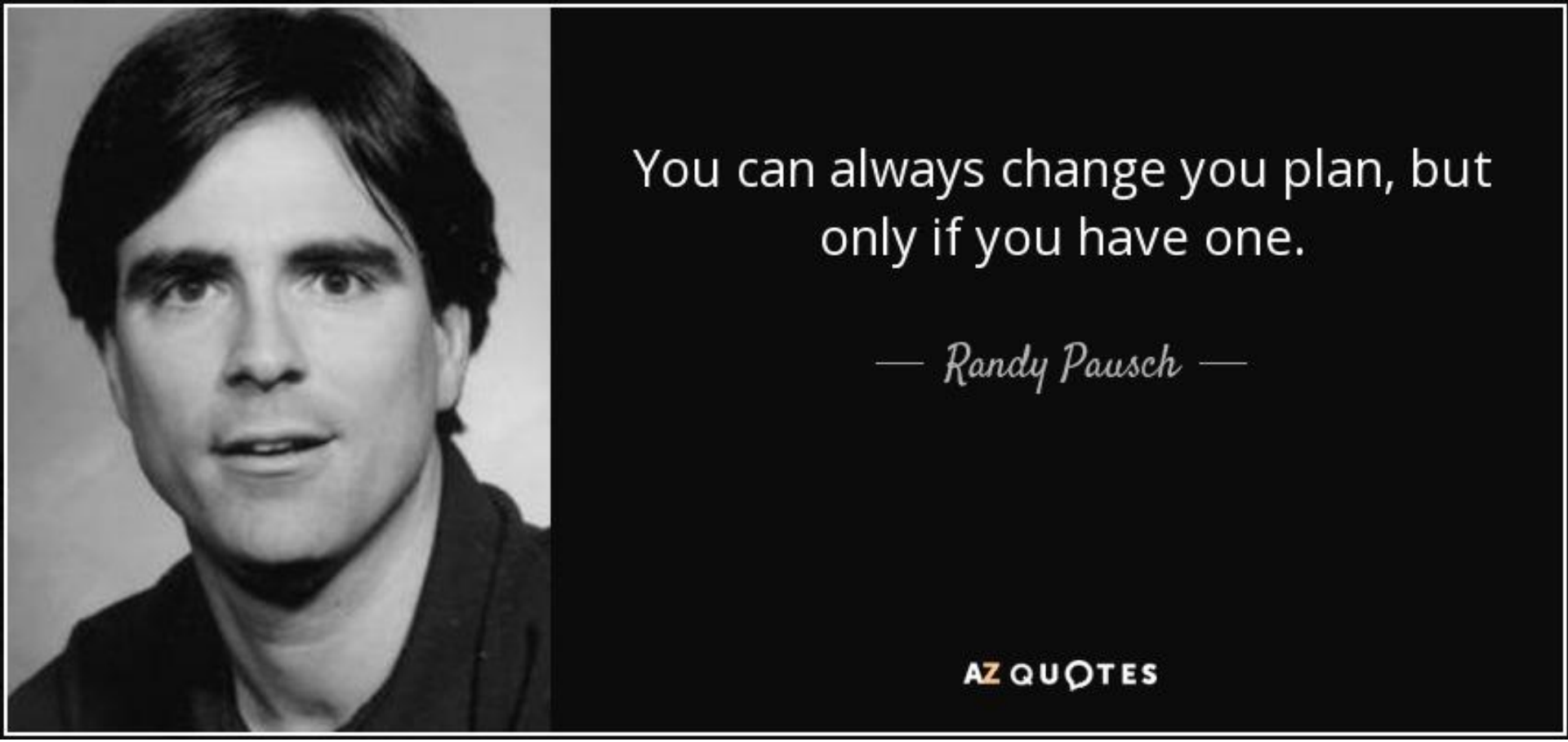
# Project plan supplements

| Plan | Description |
|---|---|
| Configuration management plan | Describes the configuration management procedures and structures to be used. |
| Deployment plan | Describes how the software and associated hardware (if required) will be deployed in the customer's environment. This should include a plan for migrating data from existing systems. |
| Maintenance plan | Predicts the maintenance requirements, costs, and effort. |
| Quality plan | Describes the quality procedures and standards that will be used in a project. |
| Validation plan | Describes the approach, resources, and schedule used for system validation. |

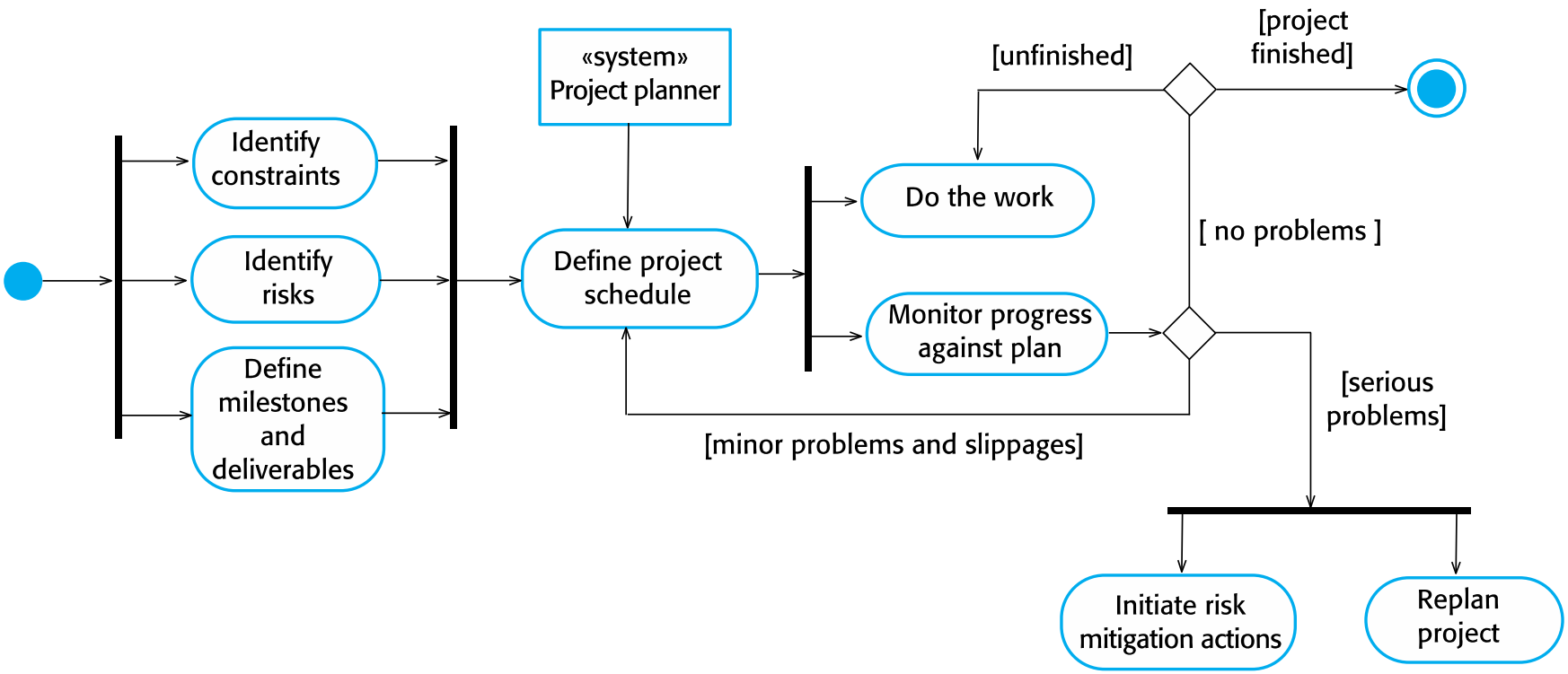
## The planning process

- Project planning is an iterative process that starts when you create an initial project plan during the project startup phase.

- Plan changes are inevitable.
  - As more information about the system and the project team becomes available during the project, you should regularly revise the plan to reflect requirements, schedule and risk changes.
  - Changing business goals also leads to changes in project plans. As business goals change, this could affect all projects, which may then have to be re-planned.


You can always change you plan, but only if you have one.
— Randy Pausch —
AZ QUOTES

# The project planning process



## Planning assumptions

- You should make realistic rather than optimistic assumptions when you are defining a project plan.
- Problems of some description always arise during a project, and these lead to project delays.
- Your initial assumptions and scheduling should therefore take unexpected problems into account.
- You should include contingency in your plan so that if things go wrong, then your delivery schedule is not seriously disrupted.

# Project scheduling

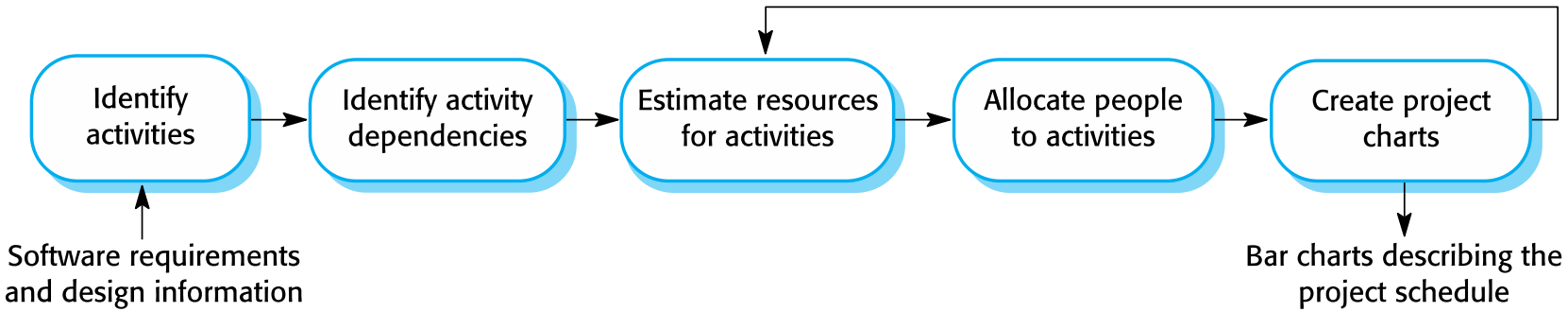
## Project scheduling

- Project scheduling is the process of deciding how the work in a project will be organized as separate tasks, and when and 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 the disk space required on a server, the time required on specialized hardware, such as a simulator, and what the travel budget will be.

## Project scheduling activities

- Split project into tasks and estimate time and resources required to complete each task.
- Organize tasks concurrently to make optimal use of workforce.
- Minimize task dependencies to avoid delays caused by one task waiting for another to complete.
- Dependent on project managers intuition and experience.

# The project scheduling process



## Scheduling problems

- Estimating the difficulty of problems and hence the cost of developing a solution is hard.
- Productivity is not proportional to the number of people working on a task.
- Adding people to a late project makes it later because of communication overheads.
- The unexpected always happens. Always allow contingency in planning.

## 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
  - 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

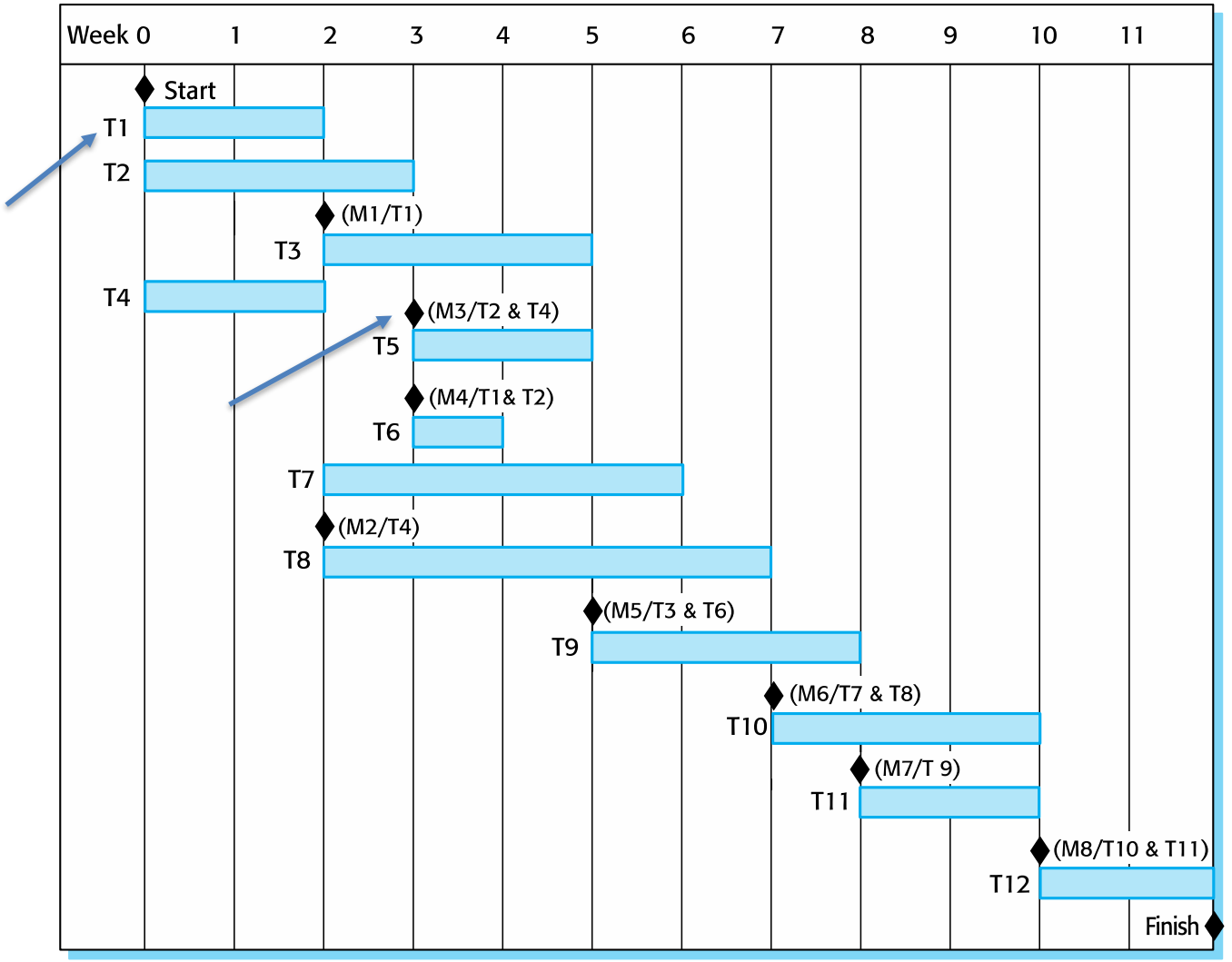
- Milestones are points in the schedule against which you can assess progress, for example, the handover of the system for testing.
- Deliverables are work products that are delivered to the customer, e.g. a requirements document for the system.

What is the difference between milestones and deliverables?

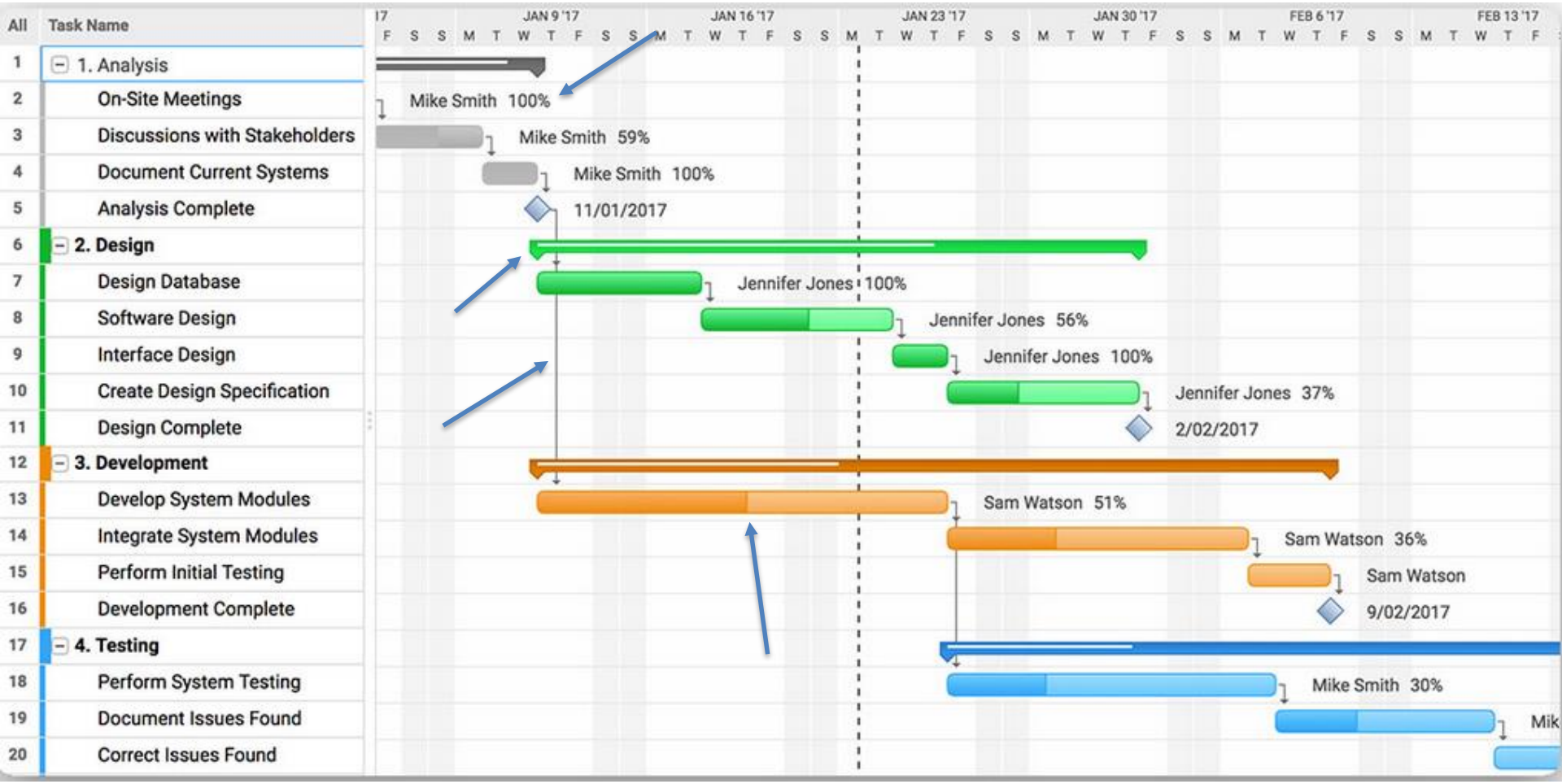
# Tasks, durations, and dependencies

| Task | Effort (person-days) | Duration (days) | Dependencies |
|---|---|---|---|
| T1 | 15 | 10 | |
| T2 | 8 | 15 | |
| T3 | 20 | 15 | T1 (M1) |
| T4 | 5 | 10 | |
| T5 | 5 | 10 | T2, T4 (M3) |
| T6 | 10 | 5 | T1, T2 (M4) |
| T7 | 25 | 20 | T1 (M1) |
| T8 | 75 | 25 | T4 (M2) |
| T9 | 10 | 15 | T3, T6 (M5) |
| T10 | 20 | 15 | T7, T8 (M6) |
| T11 | 10 | 10 | T9 (M7) |
| T12 | 20 | 10 | T10, T11 (M8) |

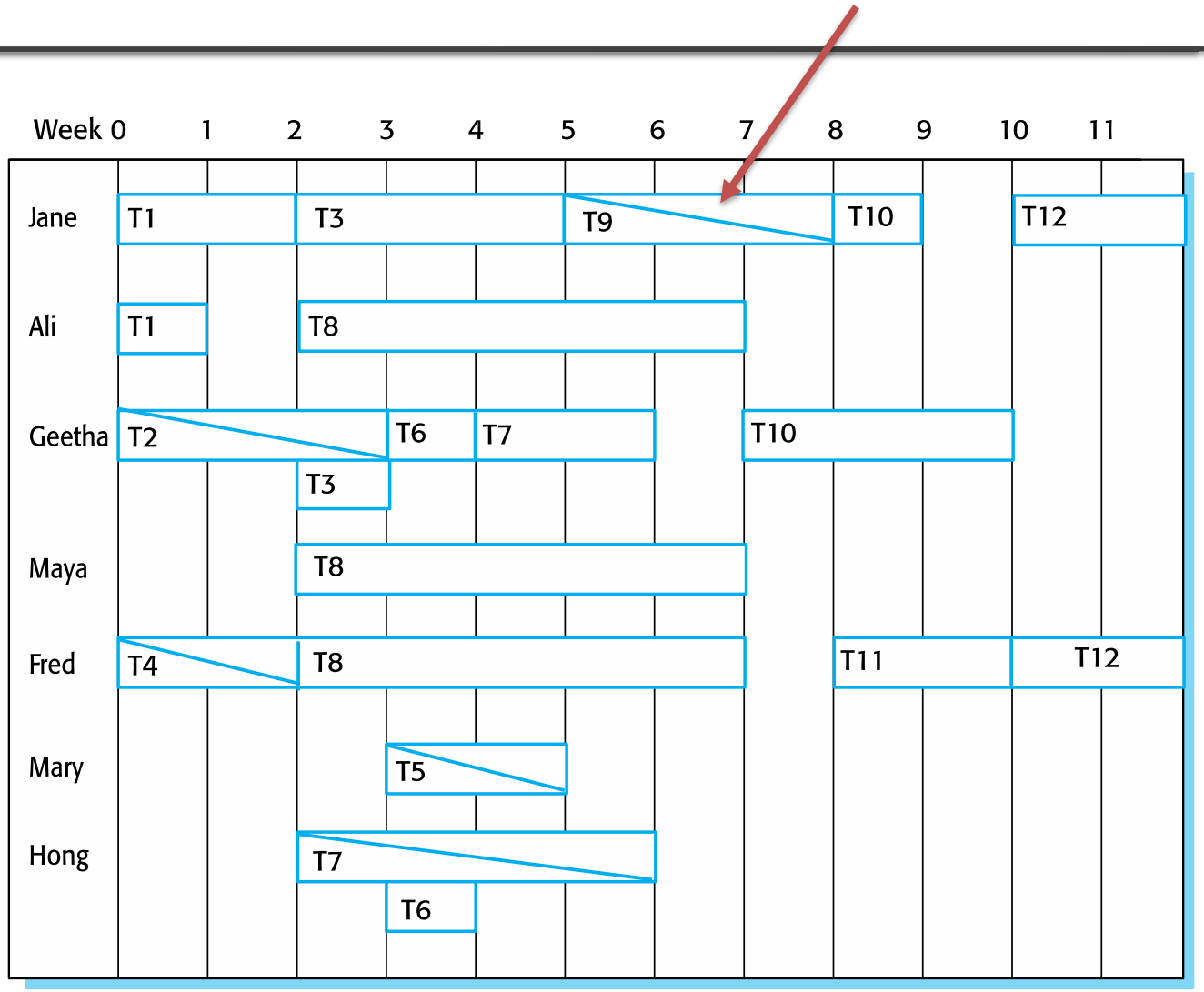
# Activity bar chart (Gantt Chart)



# Gantt Chart (Examples)



# Staff allocation chart

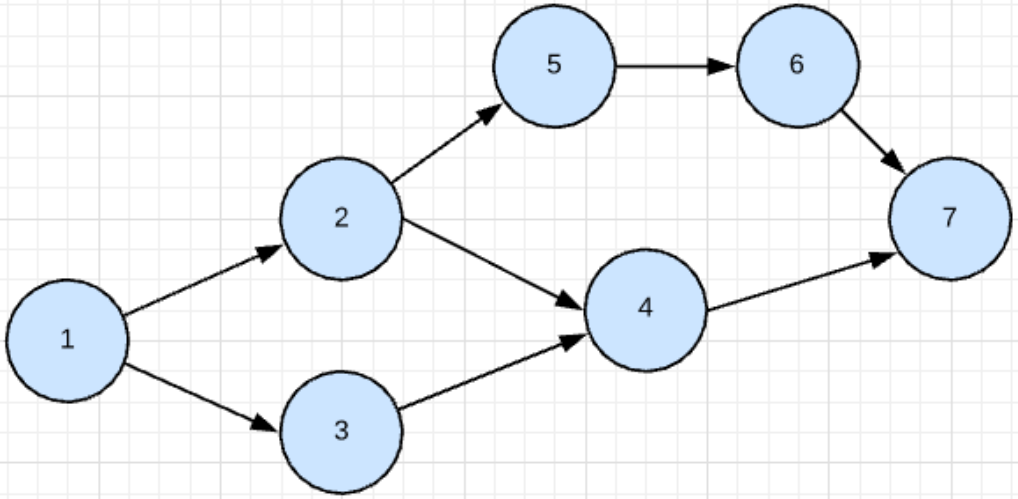


# PERT Chart (Program Evaluation and Review Technique)

- PERT charts are generally used before a project begins to plan and determine the duration of each task—so they don't have to show the actual dates of your project.
- They also do a better job of showing whether certain tasks need to be completed in order or whether they can be completed simultaneously.
- Use a PERT chart if you need to:
  - Show the interdependency of certain tasks.
  - Anticipate the amount of time it'll take to complete a project.
  - Determine the critical path to meet your deadlines.
  - Plan for large or more complex projects.

# How to

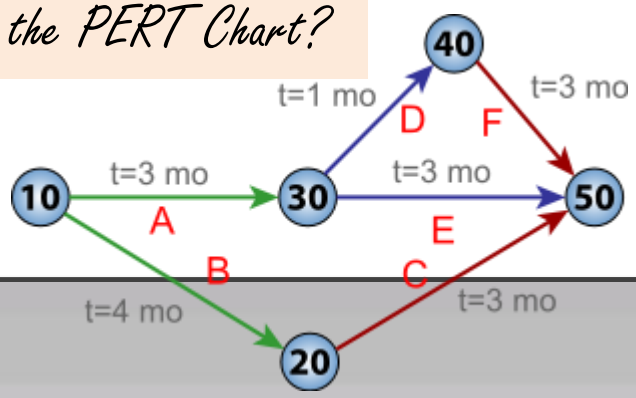
- Nodes represent events or milestones in your project. You can use either numbered circles or numbered boxes.
- Arrows represent tasks. The direction of the arrows shows the sequence of tasks. Diverging arrows indicate that you can complete those tasks concurrently. In the example below, tasks 1, 2, 4, and 6 have to be completed in order.



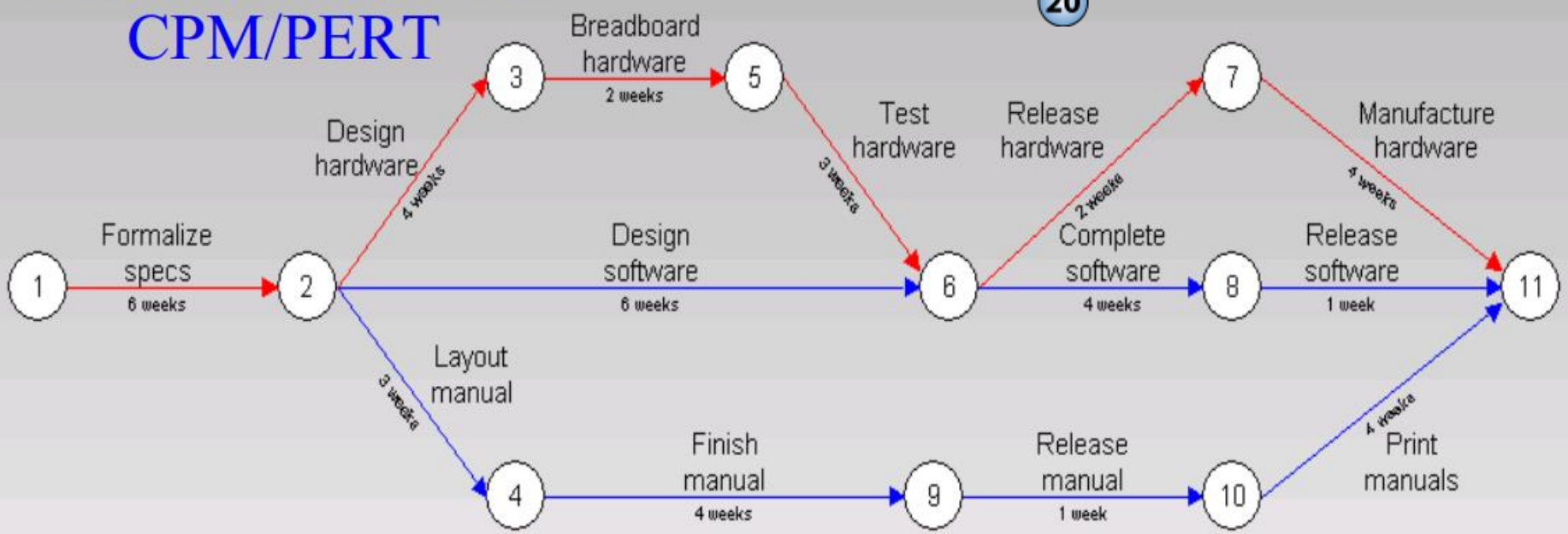

| Activity | Duration (weeks) | Immediate Predecessor Activities |
|---|---|---|
| 1 | 2.5 | - |
| 2 | 3 | 1 |
| 3 | 1 | 1 |
| 4 | 2 | 2, 3 |
| 5 | .5 | 2 |
| 6 | 1.5 | 5 |
| 7 | 4 | 4, 6 |

*What is the critical path and how do we calculate it on the PERT Chart?*

# PERT Chart




## Project Management
## CPM/PERT




| | | | | | |
|---|---|---|---|---|---|
| 1 | Start project | 5 | Breadboard running | 9 | Manual finalized |
| 2 | All specs finalized | 6 | Hardware fully functional | 10 | Manual ready for printer |
| 3 | Hardware design completed | 7 | PC Board released | 11 | Project complete |
| 4 | Manual layout completed | 8 | Software finished | | Critical path |

# Key points

- Project scheduling involves the creation of various graphical representations of part of the project plan. Bar charts, which show the activity duration and staffing timelines, are the most commonly used schedule representations.
- A project milestone is a predictable outcome of an activity or set of activities. At each milestone, a formal report of progress should be presented to management. A deliverable is a work product that is delivered to the project customer.

# Tools

- ✧ Project Management
  - Trello – trello.com
  - Asana – asana.com
- ✧ Team Communication
  - Slack – slack.com
- ✧ Gantt Chart
  - Even in Microsoft Excel….
- ✧ PERT Chart
  - www.draw.io
  - Even
    - Google Docs – Drawing
    - Libre Office - Draw