

# Project Planning

Fattane Zarrinkalam

1

# 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

# 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

A grayscale photograph of a large crowd of people, likely at a stadium or arena, with the text "Software pricing" overlaid in the center. The crowd is dense and fills the frame, with many people standing and some sitting. The text is in a large, black, sans-serif font.

# 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.
Requirement's 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 strategies

---

- The software is priced according to what the **software developer believes the buyer is willing to pay**
  - If this is less than 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



# 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.

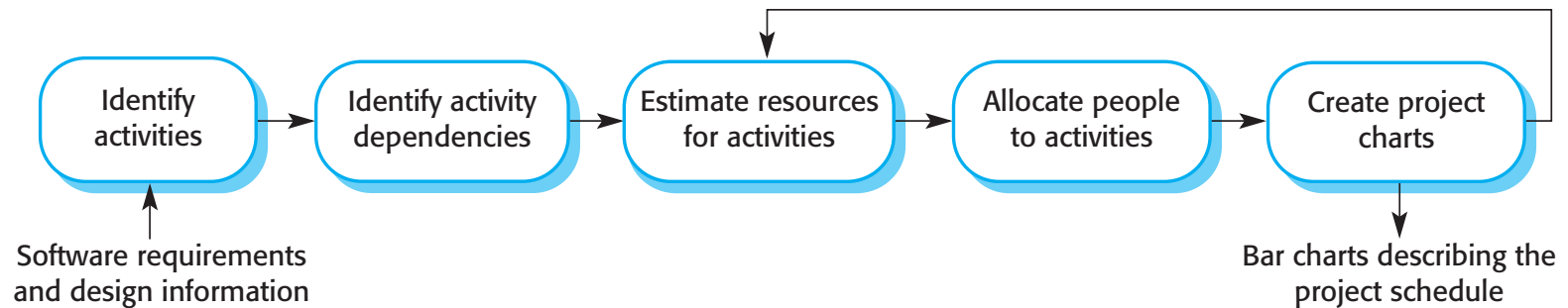
# Project scheduling activities

---

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

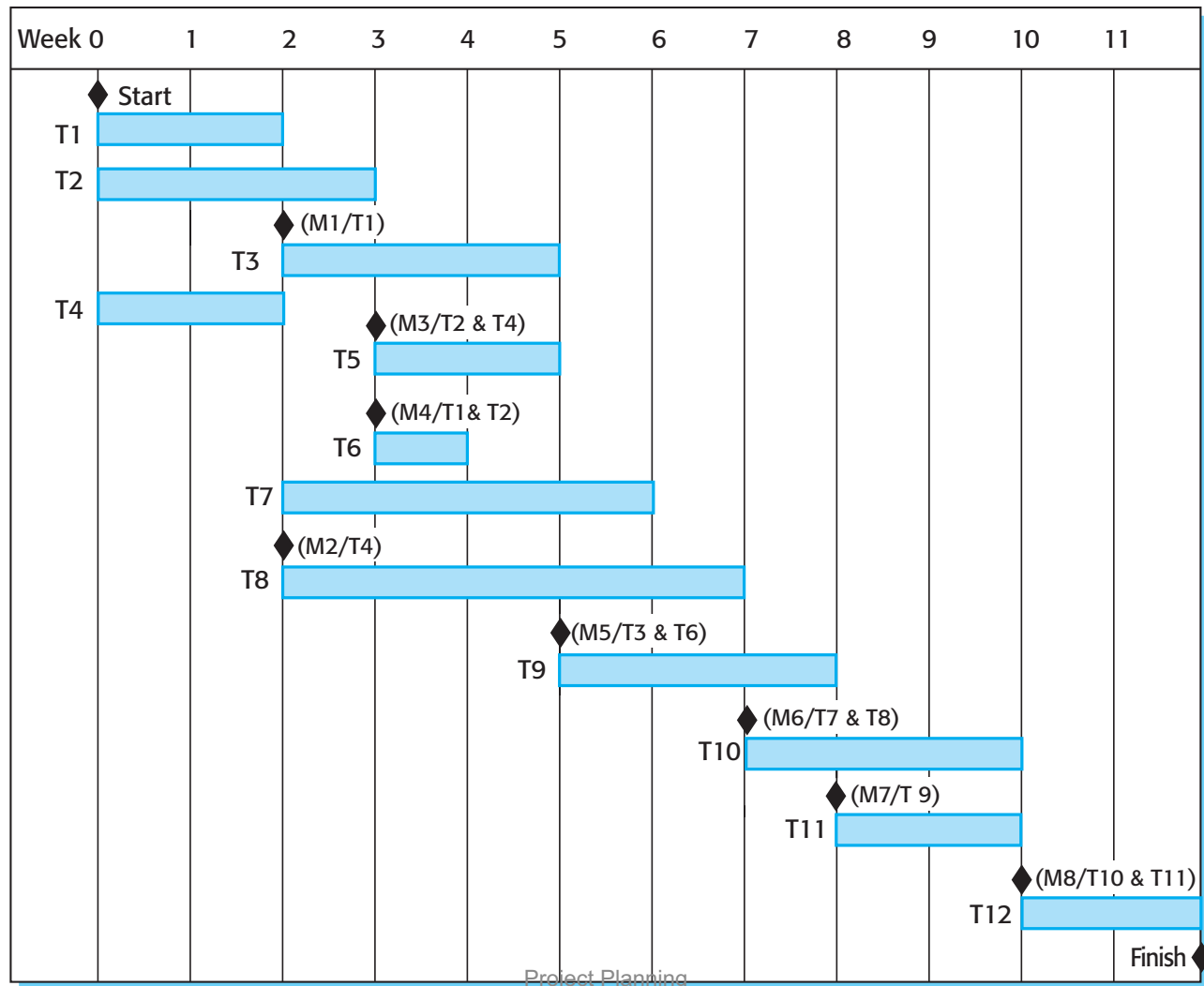
---

- 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.

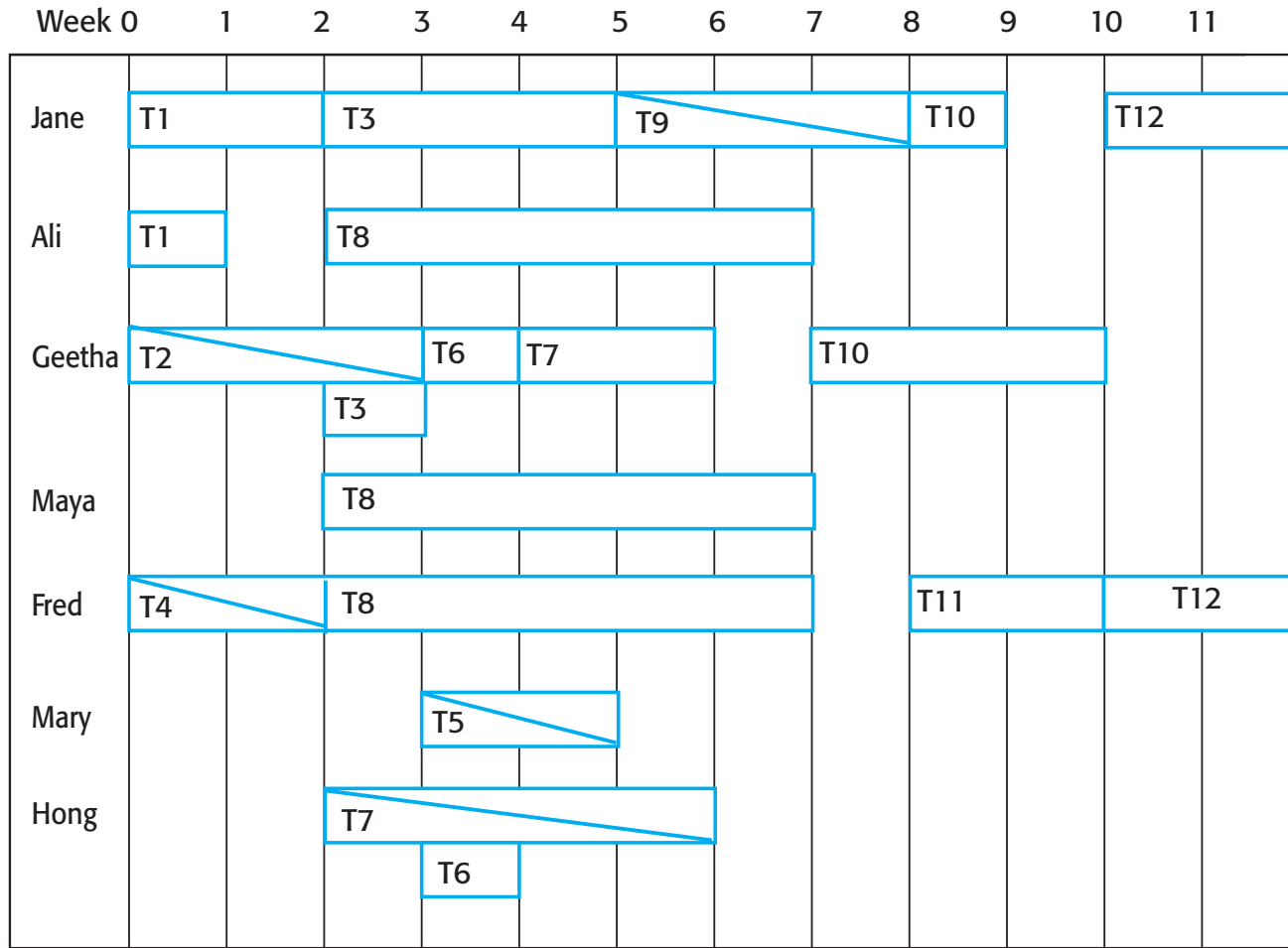
# Tasks, durations, and dependencies

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



# Staff allocation chart





# Estimation techniques



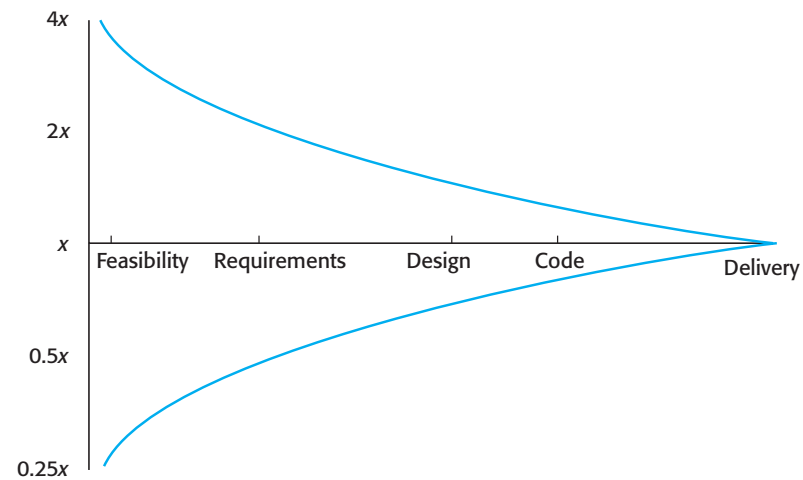
# Estimation techniques

---

- Organizations need to make software effort and cost estimates. There are two types of technique that can be used to do this:
  - *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, and process characteristics, such as experience of staff involved.

# Estimate uncertainty

---



# Experience-based approaches

---

- Experience-based techniques rely on judgments based on experience of past projects and the effort expended in these projects on software development activities.
- Typically, you identify the deliverables to be produced in a project and the different software components or systems that are to be developed.
- You document these in a spreadsheet, estimate them individually and compute the total effort required.
- It usually helps to get a group of people involved in the effort estimation and to ask each member of the group to explain their estimate.

# Problem with experience-based approaches

---

- The difficulty with experience-based techniques is that a new software project may not have much in common with previous projects.
- Software development changes very quickly and a project will often use unfamiliar techniques such as web services, application system configuration or HTML5.
- If you have not worked with these techniques, your previous experience may not help you to estimate the effort required, making it more difficult to produce accurate costs and schedule estimates.

# Algorithmic cost modelling

---

- Cost is estimated as a mathematical function of product, project and process attributes whose values are estimated by project managers:
  - $\text{Effort} = A \times \text{Size}^B \times M$ 
    - A is an organisation-dependent constant,
    - B reflects the disproportionate effort for large projects and
    - M is a multiplier reflecting product, process and people attributes.
- The most commonly used product attribute for cost estimation is code size.
- Most models are similar but they use different values for A, B and M.

# Estimation accuracy

---

- The size of a software system can only be known accurately when it is finished.
- Several factors influence the final size
  - Use of reused systems and components;
  - Programming language;
  - Distribution of system.
- As the development process progresses then the size estimate becomes more accurate.
- The estimates of the factors contributing to B and M are subjective and vary according to the judgment of the estimator.

# Effectiveness of algorithmic models

---

- Algorithmic cost models are a systematic way to estimate the effort required to develop a system. However, these models are complex and difficult to use.
- There are many attributes and considerable scope for uncertainty in estimating their values.
- This complexity means that the practical application of algorithmic cost modeling has been limited to a relatively small number of large companies, mostly working in defense and aerospace systems engineering.

A grayscale photograph of a large crowd of people, likely at a sporting event, with the word 'Summary' overlaid in the center.

# Summary



# Summary

---

- **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.
- Software is often **priced** to gain a contract, and the functionality of the system is then adjusted to meet the estimated price.
- **Project scheduling** involves the creation of various graphical representations of part of the project plan.
- **Estimation techniques** for software may be experience-based, where managers judge the effort required, or algorithmic, where the effort required is computed from other estimated project parameters.