

# UNIT 8

## Managing Software Projects

(2 Hours) — Software Engineering (CACS253)

### 8.1 Need for Proper Management of Software Projects

**Software project management** ensures that software is delivered **on time, within budget and in accordance with requirements**. Software engineering is always subject to organisational budget and schedule constraints, so management is essential. Without it, projects suffer from cost overruns, missed deadlines, poor quality and unhappy customers.

| Reason for Management      | Explanation                                                                                    |
|----------------------------|------------------------------------------------------------------------------------------------|
| Intangible product         | Software cannot be seen; managers must rely on documents and deliverables to measure progress. |
| No standard process        | Development processes vary, so planning must be adapted to each project.                       |
| Many projects are one-off  | Each project is different; past experience may not transfer.                                   |
| Rapid technological change | Estimates become outdated as tools and platforms evolve.                                       |
| Scale and complexity       | Large projects need coordination across many people and tasks.                                 |
| Constraints                | Time, cost and quality must be balanced.                                                       |

**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.

### 8.2 Management Activities

Software managers are responsible for planning and scheduling development projects. They supervise the work to ensure that it is carried out to the required standards and monitor progress to check that development is on time and within budget. The main management activities are:

- **Project planning**
- **Estimating costs**
- **Project scheduling**
- **Risk management**
- **Managing people**

- Also: reporting, proposal writing, and monitoring and review.

## 8.3 Project Planning

Project planning involves breaking down the work into parts and assigning these to project team members, anticipating problems that might arise and preparing tentative solutions. The **project plan** is used to communicate how the work will be done to the project team and customer, and to help assess progress.

| Section of Project Plan                     | Content                                                                  |
|---------------------------------------------|--------------------------------------------------------------------------|
| Introduction                                | Objectives and constraints (budget, time)                                |
| Project organisation                        | Team composition, roles and responsibilities                             |
| Risk analysis                               | Likely risks, their probability and reduction strategies                 |
| Hardware and software resource requirements | Support software, hardware needed and their prices                       |
| Work breakdown                              | Activities, milestones and deliverables                                  |
| Project schedule                            | Dependencies between activities, time estimates and allocation of people |
| Monitoring and reporting mechanisms         | Management reports, how often and how progress is checked                |

**Milestones and deliverables:** a **milestone** is an end-point of a process activity (e.g., completion of the design review), while a **deliverable** is a project result delivered to the customer (e.g., SRS document). Milestones enable managers to assess progress.

## 8.4 Estimating Costs

Cost estimation predicts the effort, time and money needed to develop the software. The main cost components are: hardware and software costs; travel and training costs; and **effort costs** (salaries of engineers) – usually the dominant one. Effort is measured in **person-months**.

| Technique                       | Description                                                                     |
|---------------------------------|---------------------------------------------------------------------------------|
| Expert judgement                | Several experts estimate; results compared and discussed (e.g., Delphi method). |
| Estimation by analogy           | Compare with similar completed projects.                                        |
| Algorithmic (parametric) models | Formulae relating size and factors to effort – e.g., COCOMO.                    |

| Technique                        | Description                                                                                     |
|----------------------------------|-------------------------------------------------------------------------------------------------|
| Function point analysis          | Estimate size from functionality: inputs, outputs, inquiries, files and interfaces.             |
| Top-down / bottom-up             | Estimate whole then split; or estimate each task then sum.                                      |
| Parkinson's law / pricing to win | Work expands to fill time / the estimate equals the customer's budget – <i>not</i> recommended. |

## COCOMO (Constructive Cost Model)

Boehm's **Basic COCOMO** estimates effort in person-months from size in thousands of delivered source instructions (KDSI):

### Basic COCOMO Formulae

**Effort (E) = a × (KDSI)<sup>b</sup> person-months      Time (D) = c × (E)<sup>d</sup> months**

**Team size = E ÷ D**

| Project Class | a   | b    | c   | d    | Description                                          |
|---------------|-----|------|-----|------|------------------------------------------------------|
| Organic       | 2.4 | 1.05 | 2.5 | 0.38 | Small teams, familiar environment, simple projects   |
| Semi-detached | 3.0 | 1.12 | 2.5 | 0.35 | Medium size, mixed experience                        |
| Embedded      | 3.6 | 1.20 | 2.5 | 0.32 | Tight hardware, software and operational constraints |

### Example – Organic Project of 32 KDSI

**Effort E** =  $2.4 \times (32)^{1.05}$ . Since  $32^{1.05} \approx 38.0$ ,  $E \approx 2.4 \times 38.0 \approx \mathbf{91 \text{ person-months}}$ .

**Duration D** =  $2.5 \times (91)^{0.38}$ . Since  $91^{0.38} \approx 5.5$ ,  $D \approx \mathbf{13.9 \text{ months}}$ .

**Average staff** =  $91 \div 13.9 \approx \mathbf{6.5 \approx 7 \text{ people}}$ .

*Note: Estimates are only as good as the size estimate. Estimate uncertainty is greatest early in the project and narrows as requirements are refined ("cone of uncertainty").*

## 8.5 Project Scheduling

Project scheduling is the process of deciding how the work in a project will be organised as separate tasks, and when and how these tasks will be executed. Estimate the calendar time needed, effort required and who will work on the tasks; identify task dependencies.

- Split the project into tasks (work breakdown structure – WBS) of about 1–2 weeks each.
- Minimise task dependencies to avoid delays caused by one task waiting for another.

- Rules of thumb: allocate contingency (e.g., estimate then add 30–50%), and remember that adding people to a late project makes it later (Brooks' law).

## Bar Charts (Gantt Charts)

Show project activities against a calendar; each bar's length represents the task duration. Easy to read and useful for showing who does what and when.

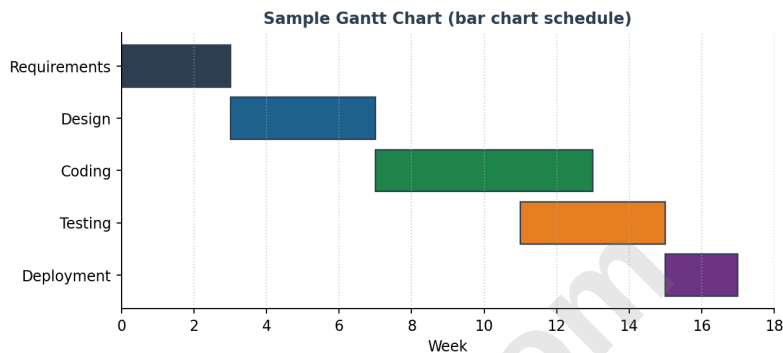


Figure 8.1: Sample Gantt chart

## Activity Networks and Critical Path

Activity networks (PERT/CPM) show dependencies between tasks. The **critical path** is the longest path through the network; it determines the minimum project duration. Any delay in a critical task delays the whole project. Non-critical tasks have **slack (float)**.

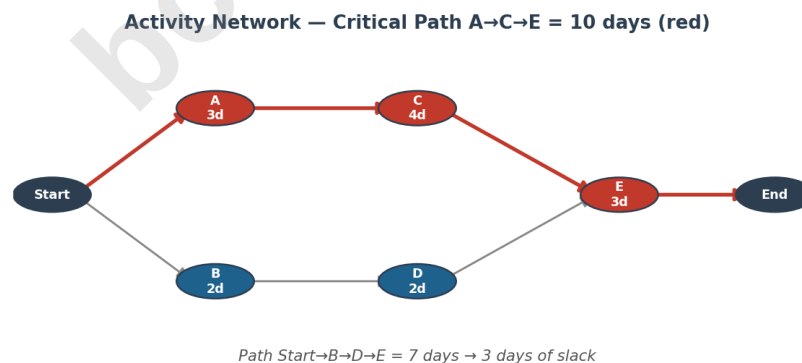


Figure 8.2: Activity network and critical path

| Task | Depends on | Duration (days) |
|------|------------|-----------------|
| A    | –          | 3               |
| B    | –          | 2               |
| C    | A          | 4               |
| D    | B          | 2               |

| Task | Depends on | Duration (days) |
|------|------------|-----------------|
| E    | C, D       | 3               |

Path A→C→E = 3 + 4 + 3 = **10 days (critical)**; path B→D→E = 2 + 2 + 3 = 7 days, so slack of 3 days for B and D.

## 8.6 Risk Management

**Risk management** is concerned with identifying risks and drawing up plans to minimise their effect on the project. A risk is a probability that some adverse circumstance will occur. Risk types: **project risks** (affect schedule or resources), **product risks** (affect quality or performance) and **business risks** (affect the organisation).

| Risk                        | Affects             | Description                                                           |
|-----------------------------|---------------------|-----------------------------------------------------------------------|
| Staff turnover              | Project             | Experienced staff will leave the project before it is finished.       |
| Management change           | Project             | A new management with different priorities.                           |
| Hardware unavailability     | Project             | Hardware essential for the project will not be delivered on schedule. |
| Requirements change         | Project and product | Larger number of changes than anticipated.                            |
| Specification delays        | Project and product | Specifications of essential interfaces not available on schedule.     |
| Size underestimate          | Project and product | Size of the system has been underestimated.                           |
| CASE tool under-performance | Product             | Tools do not perform as expected.                                     |
| Technology change           | Business            | Underlying technology is superseded by newer technology.              |
| Product competition         | Business            | A competitive product is marketed before the system is complete.      |

### Risk Management Process

| Step                   | Description                                                                                                   |
|------------------------|---------------------------------------------------------------------------------------------------------------|
| 1. Risk identification | Identify project, product and business risks (checklists, brainstorming).                                     |
| 2. Risk analysis       | Assess probability (very low → very high) and consequences (catastrophic, serious, tolerable, insignificant). |

| Step               | Description                                                                                                                        |
|--------------------|------------------------------------------------------------------------------------------------------------------------------------|
| 3. Risk planning   | Draw up plans: <b>avoidance</b> (reduce probability), <b>minimisation</b> (reduce impact), <b>contingency</b> (plan if it occurs). |
| 4. Risk monitoring | Assess each identified risk regularly; check whether its probability or effects have changed; discuss in progress meetings.        |

### Risk Exposure – Example

**Risk exposure = probability × impact.** Key developer leaves: probability 0.3, impact 4 months of delay → exposure =  $0.3 \times 4 = 1.2$  months.

**Strategy:** cross-train team members, document code, keep pair programming so knowledge is shared.

## 8.7 Managing People

People are an organisation's most important assets. Poor people management is an important contributor to project failure. Four critical factors in people management: **consistency** (treat team members equally), **respect** (different skills are valued), **inclusion** (involve all team members and listen to views) and **honesty** (be honest about what is going well and badly).

### Motivating People

Motivation means organising work and its environment to encourage people to work effectively. Maslow's hierarchy is often used: **basic needs** (pay, working conditions), **social needs** (team belonging), **esteem needs** (recognition), and **self-realisation needs** (challenging work, growth).

| Personality Type     | Motivated By                                                                 |
|----------------------|------------------------------------------------------------------------------|
| Task-oriented        | The intellectual challenge of software development                           |
| Self-oriented        | Individual success; the goals of career advancement and personal recognition |
| Interaction-oriented | The presence and actions of co-workers; prefer to work in teams              |

### Teamwork and Team Organisation

- **Team size:** ideally 5–8 members; larger teams require sub-teams; communication paths grow as  $n(n-1)/2$  (10 people → 45 paths).
- **Choose group members** for a balance of technical skills and personalities, and ensure a mix of experience.

- **Group organisation:** informal democratic groups (Agile), chief-programmer teams (a lead with specialist support), hierarchical groups.
- **Group communication:** depends on group size, structure, composition and the physical and work environment; regular meetings and shared workspace help.
- **Roles:** project manager, business analyst, designer, developers, tester, QA, configuration manager.

*Note: Brooks' Law: "Adding manpower to a late software project makes it later." Training, communication overhead and task partitioning consume time.*

### Key Takeaways

- ✓ Project management delivers software on time, within budget and meeting requirements; software is intangible and one-off, so it is hard.
- ✓ Main activities: planning, cost estimating, scheduling, risk management, people management.
- ✓ The project plan covers objectives, organisation, risks, resources, work breakdown, schedule and monitoring.
- ✓ Estimation techniques: expert judgement, analogy, COCOMO, function points. Basic COCOMO:  $E = a \cdot (KDSI)^b$ ;  $D = c \cdot E^d$ .
- ✓ Schedule with Gantt charts and activity networks; the **critical path** decides minimum duration.
- ✓ Risk process: identify → analyse → plan → monitor; risk exposure = probability × impact.
- ✓ People management: consistency, respect, inclusion, honesty; keep teams small; avoid adding people to late projects.

— End of Unit 8 —