

UNIT 1

Software Project Management Concepts

(8 Hours) — Software Project Management (CACS407)

NOTE: These notes may contain errors. Please verify important points, definitions and calculations with your textbook and teacher before relying on them.

1.1 Introduction

Software Project Management (SPM) is the discipline of planning, organizing, staffing, directing and controlling a software development effort so that the product is delivered on time, within budget and with the required quality. It is a sub-field of general project management, adapted to the special nature of software.

Software is **intangible**, **complex** and **easy to change**. Because progress cannot be seen or touched, managers need special techniques (planning, estimation, scheduling, risk and quality management) to know how much work has really been done.

1.2 Project and Software Project

1.2.1 What is a project?

A **project** is a temporary endeavour undertaken to create a unique product, service or result. It has a definite start and end, a specific objective, limited resources and a degree of uncertainty.

Characteristic	Meaning
Non-routine	Tasks are not repetitive; each project is different from the last.
Planned	Work is carried out according to a plan with defined stages.
Specific objectives	A clear goal (product, service or result) is to be achieved.
Defined life span	There is a clear start and finish date.
Multiple resources	People, money, tools and time are shared among activities.
Customer	A client or sponsor for whom the result is produced.

1.2.2 Software project

A **software project** is a project whose deliverable is a software product (and its documentation). It covers the complete set of activities from requirement gathering to delivery and maintenance, together with the management activities that surround them.

Example 1.1 — Software projects

- Developing a hospital management system for a client.
- Building a mobile banking application.
- Migrating an old accounting package to a web-based platform.

1.2.3 The triple constraint

Every project is bounded by **scope**, **time** and **cost**, and the quality of the result depends on how these are balanced. Changing one constraint normally affects the others.

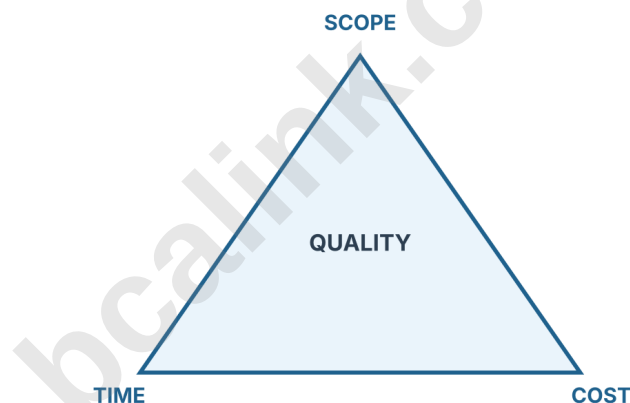


Figure 1.1 — Project constraints (scope, time, cost) and quality

1.3 Software Project vs Other Projects

Projects such as building a bridge or constructing a house are also managed projects, but software projects have features that make them harder to plan and control.

Aspect	Software project	Other (e.g. civil/engineering) project
Visibility	Product is invisible; progress hard to see.	Physical progress easy to see.
Complexity	Many interacting parts; complexity per cost is high.	Complexity usually lower per unit cost.
Conformity	Must conform to human clients and changing business rules.	Governed largely by physical laws and standards.

Aspect	Software project	Other (e.g. civil/engineering) project
Flexibility	Easy (and tempting) to change; frequent requirement changes.	Changes are costly after construction begins.
Estimation	Hard; every project is largely new.	Easier due to past similar projects and standard rates.
Maturity of methods	Techniques still evolving.	Well-established, standardized practice.
Product nature	Does not wear out, but deteriorates through changes.	Physical wear and tear.

Note: Brooks described these properties of software as complexity, conformity, changeability and invisibility.

1.4 Importance of Software Project Management

- Large numbers of software projects run late, over budget or fail to meet user needs; good management reduces this.
- Provides a framework for **planning** work, estimating effort and cost, and scheduling activities.
- Helps control **risk** early, before it turns into a problem.
- Ensures proper **use of resources** (people, tools, money) and avoids wastage.
- Improves **communication** between clients, developers and management.
- Improves **quality** through reviews, testing and standards.
- Gives the organization a basis for **measuring performance** and learning from past projects.

1.5 Problems in Software Project Management

Problem area	Description
Unclear requirements	Client does not know exactly what is needed; requirements change often (scope creep).
Poor estimation	Unrealistic effort, cost and time estimates cause overruns.
Inadequate planning	Missing tasks, unrealistic schedules, no risk plan.
Team issues	Lack of skills, high turnover, poor motivation or communication.
Intangible progress	Difficult to measure actual progress ("90% done" syndrome).
Technology change	New tools and platforms make earlier decisions obsolete.
Poor quality control	Late detection of defects increases rework.

Problem area	Description
Unrealistic expectations	Management or clients demand impossible delivery dates.

1.6 Process of Software Project Management

The management process is usually described through five process groups (as in the PMBOK model). They overlap and repeat throughout the project.

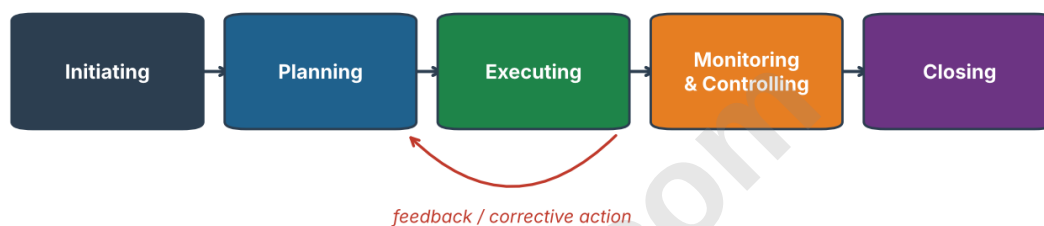


Figure 1.2 — Project management process groups

Process group	Main activities
Initiating	Define the project idea, objectives, feasibility, stakeholders; get approval (project charter).
Planning	Define scope, create work breakdown structure, estimate effort/cost, schedule, plan quality, risk, staffing and communication.
Executing	Carry out the plan: develop, test, manage team and resources.
Monitoring & Controlling	Track progress against plan, manage change, manage risk, take corrective action.
Closing	Deliver product, get acceptance, release resources, document lessons learned.

1.6.1 Project life cycle effort

The amount of effort is low at the start, peaks during execution and falls near closure.

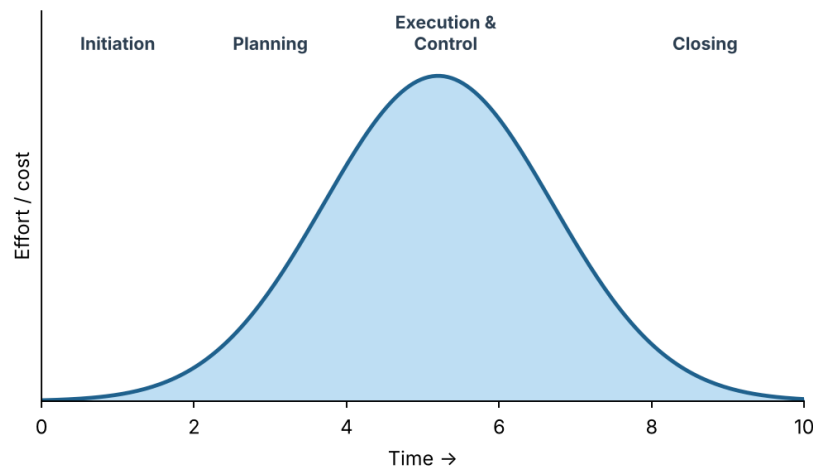


Figure 1.3 — Typical effort distribution over the project life cycle

1.6.2 Management functions

- **Planning** — deciding what is to be done, how, when and by whom.
- **Organizing** — assigning roles and responsibilities and structuring the team.
- **Staffing** — selecting, training and motivating people.
- **Directing/Leading** — guiding and supervising team members.
- **Monitoring and Controlling** — comparing actual progress with the plan and correcting deviations.

1.7 Characteristics of a Good Project Manager

The project manager is the person accountable for the success of the project. Important qualities are:

- **Leadership** — inspires and guides the team toward the goal.
- **Communication skills** — listens, explains clearly and reports honestly to the team and stakeholders.
- **Technical knowledge** — understands the software process well enough to judge estimates and risks.
- **Planning and organizing ability** — breaks work into manageable pieces and tracks them.
- **Decision making** — takes timely decisions even with incomplete information.
- **Problem solving and negotiation** — resolves conflicts and negotiates with clients and management.
- **Team building and motivation** — builds trust and a positive working atmosphere.
- **Adaptability** — responds to change in requirements and technology.

- **Integrity and accountability** — takes responsibility for outcomes.

1.8 Successful Software Project Manager

A successful project manager balances the technical, people and business sides of the project. Typical roles include:

Role	Description
Planner	Prepares plans, schedules and estimates.
Leader	Motivates and directs the team.
Communicator	Links developers, client, sponsor and top management.
Risk manager	Identifies and reduces threats to the project.
Quality champion	Ensures standards and reviews are followed.
Controller	Monitors cost, schedule and scope and acts on deviations.

Signs of success: the product is delivered on time, within budget, meets requirements and quality targets, the client is satisfied and the team remains motivated.

1.9 Overview of Software Project Planning

Project planning is the process of deciding in advance what will be done, how, by whom, when and at what cost. A plan is a living document: it is revised as the project progresses.

1.9.1 Steps in project planning

- **Select the project** — identify the need and decide whether it is worth doing.
- **Identify project scope and objectives** — stakeholders, goals and measures of success.
- **Identify project infrastructure** — standards, tools, organization.
- **Analyze project characteristics** — nature of the product and risks.
- **Identify products and activities** — work breakdown structure and deliverables.
- **Estimate effort for each activity** — size, effort, cost.
- **Identify activity risks** — and plan how to handle them.
- **Allocate resources** — people, equipment and money.
- **Review and publicize the plan** — get approval and share with the team.
- **Execute the plan and lower-level planning** — detail the plan as work proceeds.

Note: These steps follow Hughes & Cotterell's stepwise approach to project planning.

1.9.2 Contents of a project plan

- Introduction and objectives, scope and deliverables.
- Work breakdown structure and schedule.
- Effort and cost estimates, budget.
- Resource and staffing plan, project organization.
- Risk management plan, quality plan.
- Configuration management and communication plan.

Example 1.2 — Planning a small project

A college wants an online admission system. The project manager (i) defines the scope: online form, fee payment, merit list; (ii) breaks it into modules; (iii) estimates effort for each; (iv) assigns developers; (v) lists risks such as payment gateway delay; (vi) prepares the schedule and shares it with the client.

Key Takeaways

- ✓ A project is temporary, unique and goal-oriented; a software project delivers software.
- ✓ Software projects are harder to manage because the product is invisible, complex and changeable.
- ✓ SPM follows five process groups: initiating, planning, executing, monitoring/controlling, closing.
- ✓ Common problems: unclear requirements, poor estimates, weak planning, team and quality issues.
- ✓ A good manager combines leadership, communication, technical and planning skills.
- ✓ Planning (scope, WBS, estimates, schedule, risk, resources) is the foundation of the whole project.

— End of Unit 1 —