

Project Management Activities

Project management

- Organising, planning and scheduling software projects.
- Good project management cannot guarantee success, but poor management on significant projects always leads to failure.

Software project management

- 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.
- Project management is needed because software development is always subject to budget and schedule constraints that are set by the organisation developing the software.

Software management distinctions

- The product is intangible.
- The product is uniquely flexible.
- Software engineering is not recognized as an engineering discipline with the same status as mechanical, electrical engineering, etc.
- The software development process is not standardised.
- Many software projects are 'one-off' projects.

Management activities

- Project planning.
- Project scheduling.
- Risk management.
- Managing people.

Management commonalities

- These activities are not peculiar to software management
- Many techniques of engineering project management are equally applicable to software project management
- Technically complex engineering systems tend to suffer from the same problems as software systems

The 4 P's

- People — the most important element of a successful project
- Product — the software to be built
- Process — the set of framework activities and software engineering tasks to get the job done
- Project — all work required to make the product a reality

Project Planning

- The most time-consuming project management activity.
- Continuous activity from initial concept through to system delivery.
- Plans must be regularly revised as new information becomes available.
- Various different types of plan may be developed to support the main software project plan that is concerned with schedule and budget.

Project Planning

- The biggest single problem that affects software developing is that of under estimating resources required for a project.
- Developing a realistic project plan is essential to gain an understanding of the resources required, and how these should be applied.

Types of Project Plan

- *Software development plan.*
 - The central plan, which describes how the system will be developed.
- *Quality assurance plan.*
 - Specifies the quality procedures & standards to be used.
- *Validation plan.*
 - Defines how a client will validate the system that has been developed.
- *Configuration management plan.*
 - Defines how the system will be configured and installed.
- *Maintenance plan.*
 - Defines how the system will be maintained.
- *Staff development plan.*
 - Describes how the skills of the participants will be developed.

Project planning process

Establish the project constraints

Make initial assessments of the project parameters

Define project milestones and deliverables

while project has not been completed or cancelled **loop**

 Draw up project schedule

 Initiate activities according to schedule

 Wait (for a while)

 Review project progress

 Revise estimates of project parameters

 Update the project schedule

 Re-negotiate project constraints and deliverables

if (problems arise) **then**

 Initiate technical review and possible revision

end if

end loop

The Software Development Plan

- Specifies the order of work to be carried out.
- Varies from small and relatively informal to large and very formal.
- Developing a project plan is as important as properly designing code:
- On the basis of a project plan, contracts will be signed and careers made or broken.
- Important not to:
 - Overestimate your team's ability;
 - Be pressured by developers;

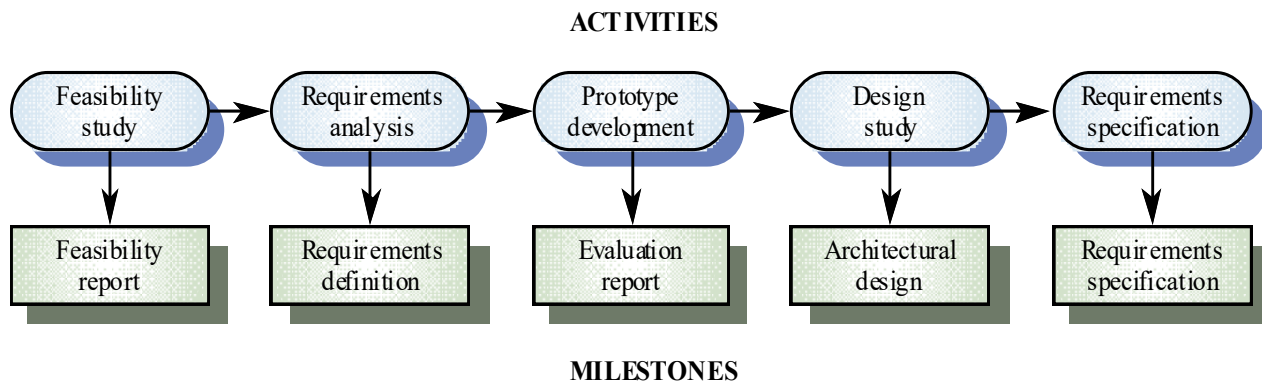
Project plan structure

- Introduction
 - Brief intro to project— references to requirements spec
- Project organization
 - Introduction to organizations, people, and their roles
- Risk Analysis
 - what are the key risks to the project?
- Hardware and software resources
 - what h/w and s/w resources will be required for the project and when?
- Work breakdown
 - The project divided into activities, milestones, deliverables; dependencies between tasks etc
- Project schedule
 - Actual time required— allocation of dates
- Reporting and progress measurement
 - Mechanisms to monitor progress.

Activity organization

- Activities in a project should be organised to produce tangible outputs for management to judge progress
- *Milestones* are the end-point of a process activity
- *Deliverables* are project results delivered to customers
- The waterfall process allows for the straightforward definition of progress milestones

Milestones in the RE process



Why Are Projects Late?

- An unrealistic deadline established by someone outside the software development group.
- Changing customer requirements that are not reflected in schedule changes;
- An honest underestimate of the amount of effort and/or the number of resources that will be required to do the job;
- Predictable and/or unpredictable risks that were not considered when the project commenced;
- Technical difficulties that could not have been foreseen in advance;

Why Are Projects Late?

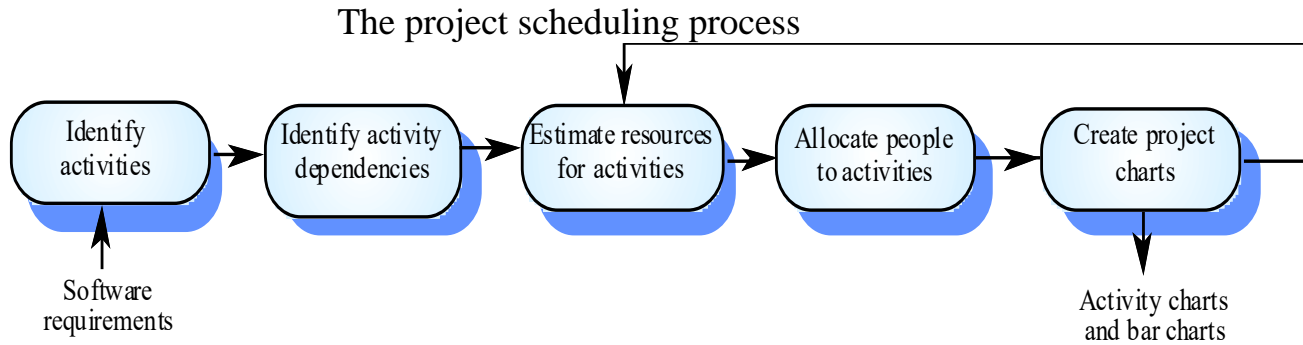
- Human difficulties that could not have been foreseen in advance;
- Miscommunication among project staff that results in delays;
- A failure by project management to recognize that the project is falling behind schedule and a lack of action to correct the problem

Project Scheduling

- 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 manager's intuition and experience.

Scheduling Principles

- *Compartmentalization*—define distinct tasks
- *Interdependency*—indicate task interrelationship
- *Effort validation* —be sure resources are available
- *Defined responsibilities* —people must be assigned
- *Defined outcomes* —each task must have an output
- *Defined milestones* —review for quality



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 Tracking

- Conduct periodic project status meetings in which each team member reports progress and problems.
- Evaluate the results of all reviews conducted throughout the software engineering process.
- Determine whether formal project milestones have been accomplished by the scheduled date.
- Compare actual start-date to planned start-date for each project task listed in the resource table.
- Meet informally with practitioners to obtain their subjective assessment of progress to date and problems on the horizon.
- Use earned value analysis to assess progress quantitatively.

Example: Work Breakdown Structure (WBS)

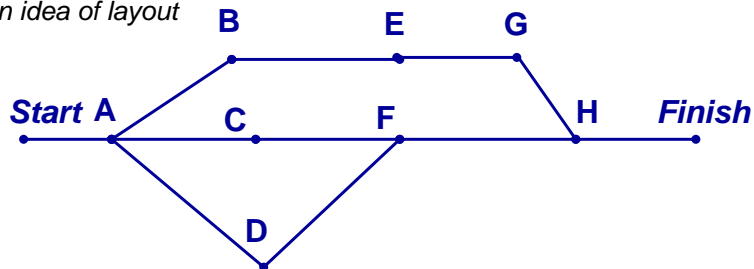
•Consider:

Task	Label	Predecessor	Staff reqd	Estimation duration (days)
Obtain materials	A	-	2	30
Produce component 1	B	A	3	40
Produce component 2	C	A	3	30
Produce component 3	D	A	3	35
Finish component 1	E	B	2	25
Finish component 2	F	B, C	3	30
Assemble	G	E	2	15
Test and deliver	H	G,F	3	20

Graph: Activity on Node

Graph: Activity on Node

- Gain idea of layout



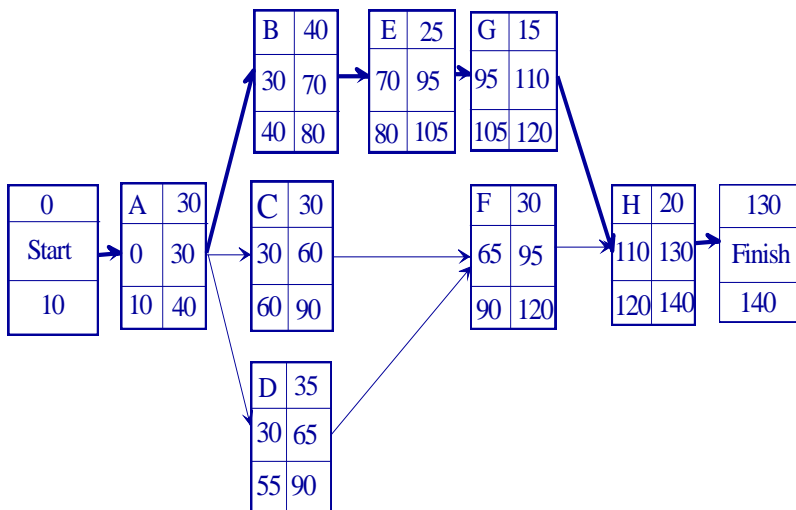
- Re-draw graph introducing boxes at nodes:

Task Label	Duration
Early start time	Early finish time
Late start time	Late finish time

Labels

- Early start time (EST) = Previous early finish time
 - Early finish time (EFT) = EST + Duration
 - Late finish time (LFT) = Following late start time
 - Late start time (LST) = LFT - Duration
- Critical path is such that it links tasks where difference between finish times differ by the difference between target time and finish time

Network Diagram



Gantt Chart

