Software Project Management

King Saud University
College of Computer and Information Sciences
Department of Computer Science

Dr. S. HAMMAMI

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.

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.

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 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.

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.

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.

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.

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.

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.

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

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.

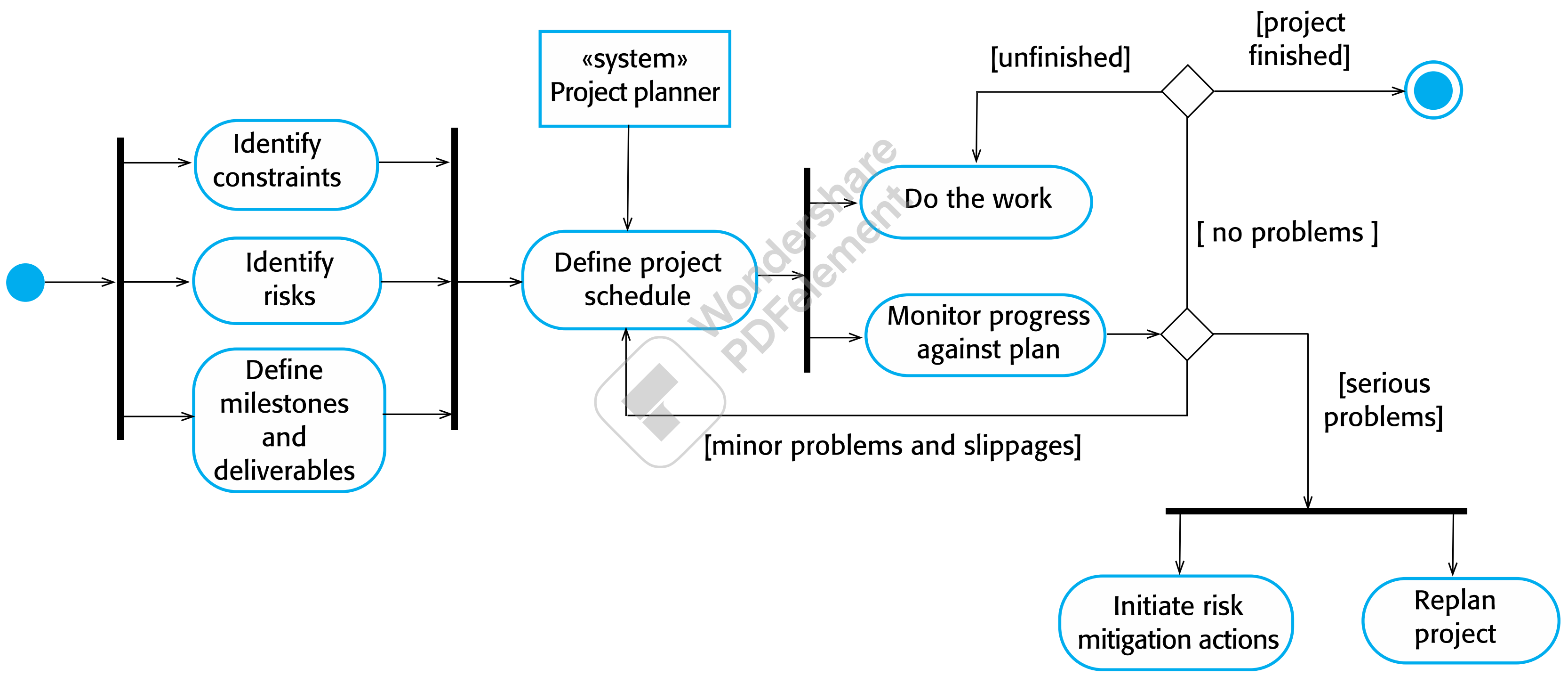
Project plans

- 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

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.

The project planning process



Risk mitigation

- If there are serious problems with the development work that are likely to lead to significant delays, you need to initiate risk mitigation actions to reduce the risks of project failure.
- In conjunction with these actions, you also have to re-plan the project.
- This may involve renegotiating the project constraints and deliverables with the customer. A new schedule of when work should be completed also has to be established and agreed with the customer.

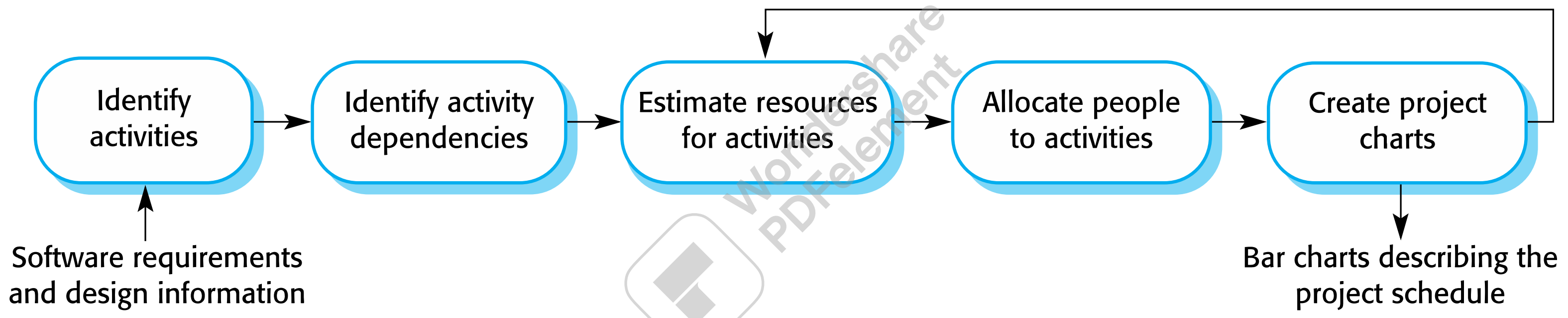
Project scheduling

- 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


Identify activities
Identify activity dependencies
Estimate resources for activities
Allocate people to activities
Create project charts
Software requirements and design information
Bar charts describing the project schedule

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

Milestones and deliverables

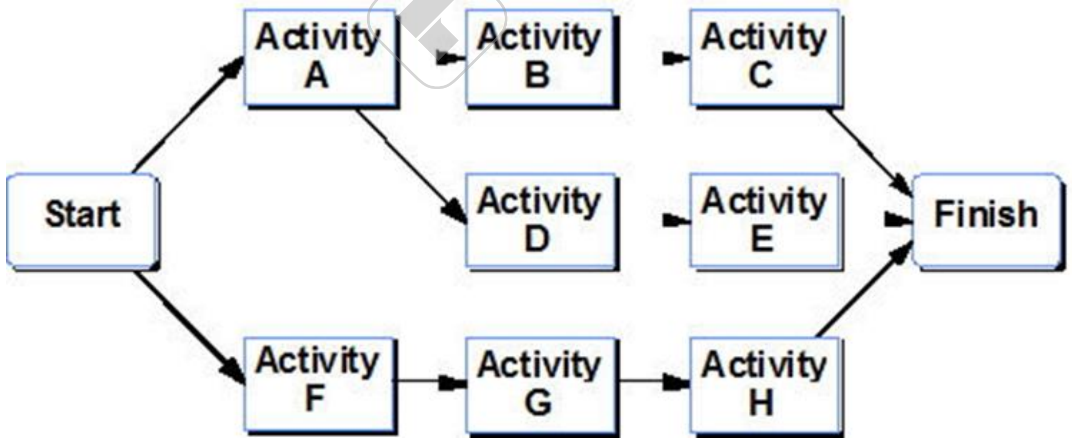
- Milestones are points in the schedule against which you can assess progress, for example, the handover of the system for testing.
- Deliverables are work products that are delivered to the customer, e.g. a requirements document for the system.

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.

Activity networks

- Graphical notations used to illustrate the project schedule.
- Show project breakdown into tasks.
- They show task dependencies and the critical path (the shortest time possible to complete the project, i.e. "critical" activities on the longest path).



What is Critical Path Method CPM?

- CPM calculates
  - the shortest time possible to complete the project.
  - The earliest and latest that each activity can start and finish without making the project longer
- Determines “critical” activities (on the longest path)
- Prioritize activities for the effective management and to shorten the planned critical path of a project by:
  - Pruning critical path activities (cutting some of them if possible)
  - “Fast tracking" (performing more activities in parallel)
  - “Crashing the critical path" (shortening the durations of critical path activities by adding resources)

Project Precedence Table

- The essential technique for using CPM is to construct a model of the project that includes the following:
  - A list of all activities required to complete the project (also known as Work Breakdown Structure)
  - The time (duration) that each activity will take to completion
  - The dependencies between the activities.

Task	Duration (Weeks)	Precedence
A	8	-
B	15	-
C	3	A
D	10	A, B
E	5	B
F	2	C, D
G	7	E, F

Phases of CPM Approach

- Phase I
  - Break project into operations necessary for completion
  - Determine sequential relationship of operations
- Phase II
  - Create time estimates for each operation (duration)
  - Determine earliest possible start date, earliest possible finish date , latest start & finish
    - Determine "free float" and "total float"
  - Revised after completion of Phase III
- Phase III
  - Establish time-cost relationship
  - Establish scheduling variations
    - Determine most favorable balance between time-cost

Definitions

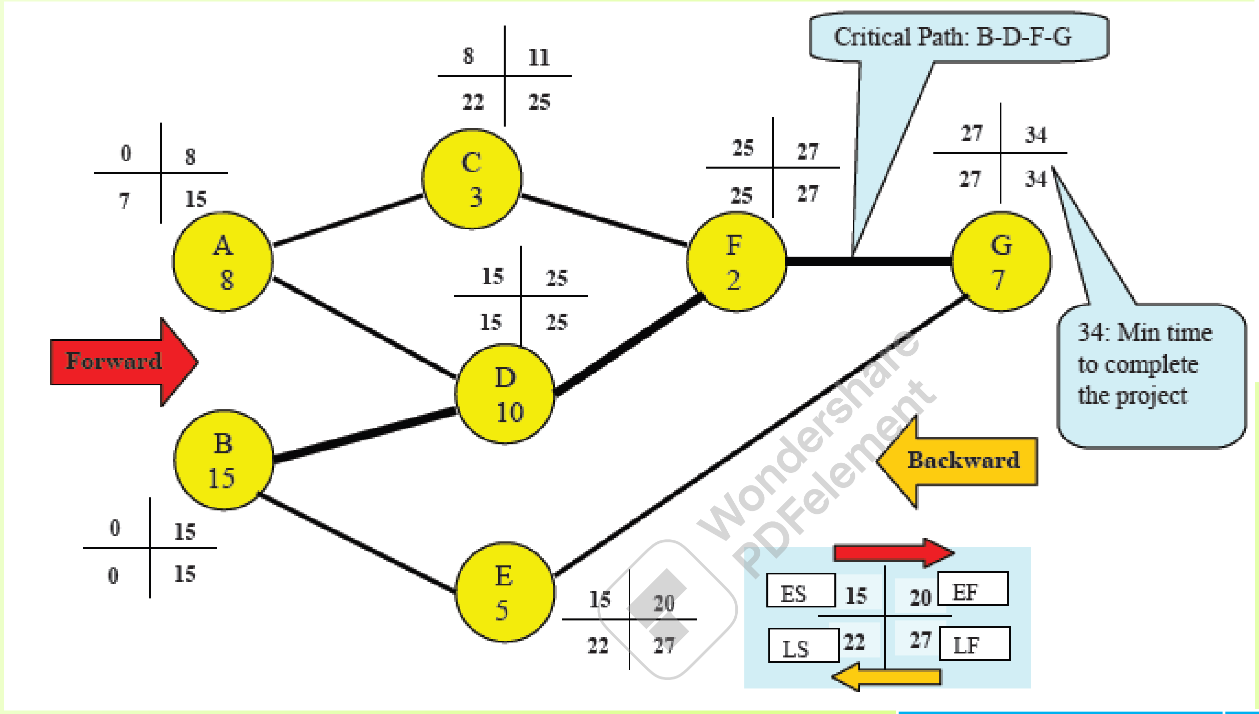
- **Float (slack) -** amount of time that a task can be delayed without causing a delay to:
  - subsequent tasks (free float): Free float is how long an activity can be delayed without delaying the Early Start of its successor.
    - free float = min(suc. ES) – ES – Duration
  - project completion date (total float)
    - LF - EF = total float = LS - ES
- **Critical path** is the sequence of activities which add up to the longest overall duration. It is the shortest time possible to complete the project. Any delay of an activity on the critical path directly impacts the planned project completion date (there is no float on the critical path).
- **Critical activity** – activity with zero total float

Activity Identity box

Early Start (ES) = max (pre. EF)	ID	Early Finish (EF) = Duration+ ES
Late Start (LS) = LF- Duration	Duration	Late Finish (LF) = min (suc. LS)

Activity network – Critical Pat

Task	Duration (Weeks)	Precedence
A	8	-
B	15	-
C	3	A
D	10	A, B
E	5	B
F	2	C, D
G	7	E, F

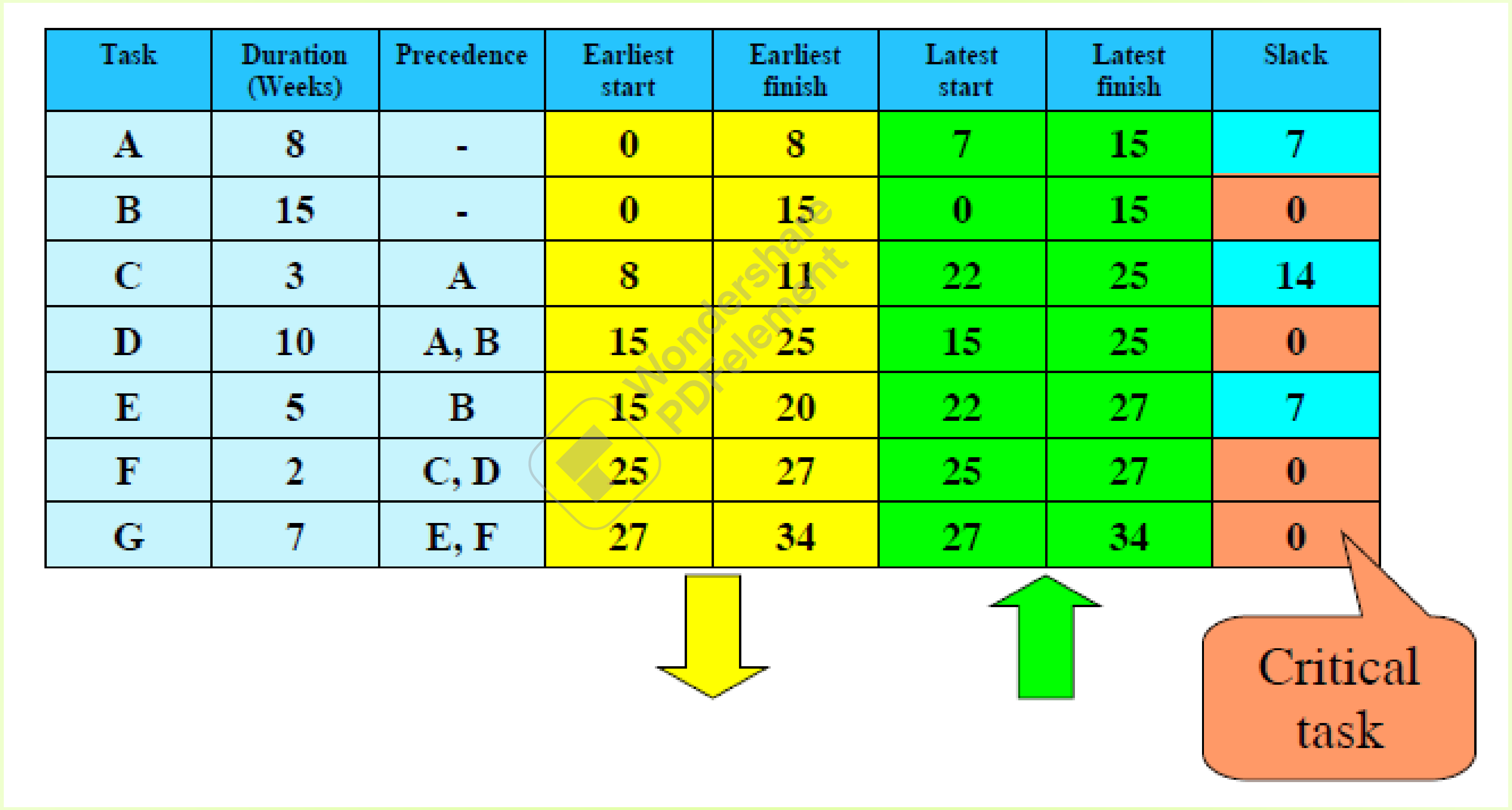


Early Start (ES) = max (pre. EF)	ID	Early Finish (EF) = Duration+ ES
Late Start (LS) = LF- Duration	Duration	Late Finish (LF) = min (suc. LS)

free float = min (suc. ES) – ES - Duration

total float = LF - EF = LS - ES

Project Precedence Table

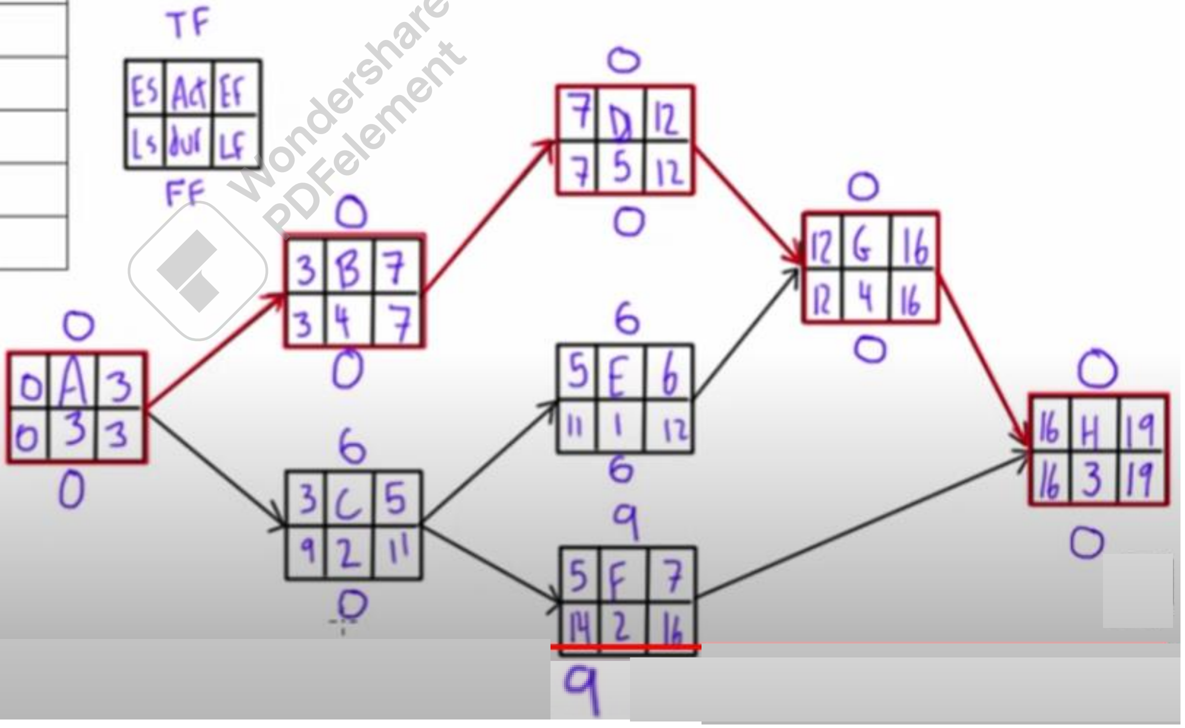


Task	Duration (Weeks)	Precedence	Earliest start	Earliest finish	Latest start	Latest finish	Slack
A	8	-	0	8	7	15	7
B	15	-	0	15	0	15	0
C	3	A	8	11	22	25	14
D	10	A, B	15	25	15	25	0
E	5	B	15	20	22	27	7
F	2	C, D	25	27	25	27	0
G	7	E, F	27	34	27	34	0

Critical task

Example of Free Float

Activity	Predecessor	Duration (days)
A	-	3
B	A	4
C	A	2
D	B	5
E	C	1
F	C	2
G	D,E	4
H	F,G	3

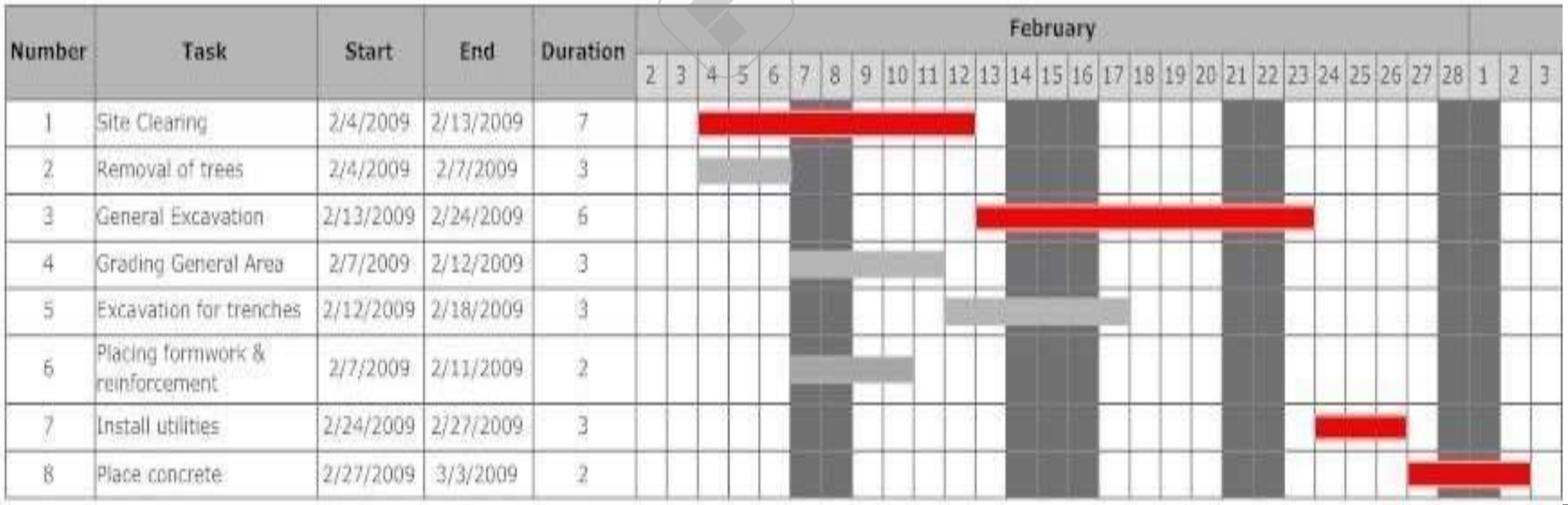


TF
ES Act EF
LS dur LF
FF
0 A 3 / 0 3 3 (TF 0, FF 0)
3 B 7 / 3 4 7 (TF 0, FF 0)
3 C 5 / 9 2 11 (TF 6, FF 0)
7 D 12 / 7 5 12 (TF 0, FF 0)
5 E 6 / 11 1 12 (TF 6, FF 6)
5 F 7 / 14 2 16 (TF 9, FF 9)
12 G 16 / 12 4 16 (TF 0, FF 0)
16 H 19 / 16 3 19 (TF 0, FF 0)

[redacted]

Bar or Gantt Chart

- One of the most popular and useful ways of showing activities (tasks or events) displayed against time.
- On the left of the chart is a list of the activities and along the top is a suitable time scale.
- Each activity is represented by a bar; the position and length of the bar reflects the start date, duration and end date of the activity.
- This allows you to see at a glance:
  - •What the various activities are
  - •When each activity begins and ends
  - •How long each activity is scheduled to last
  - •Where activities overlap with other activities, and by how much
  - •The start and end date of the whole project

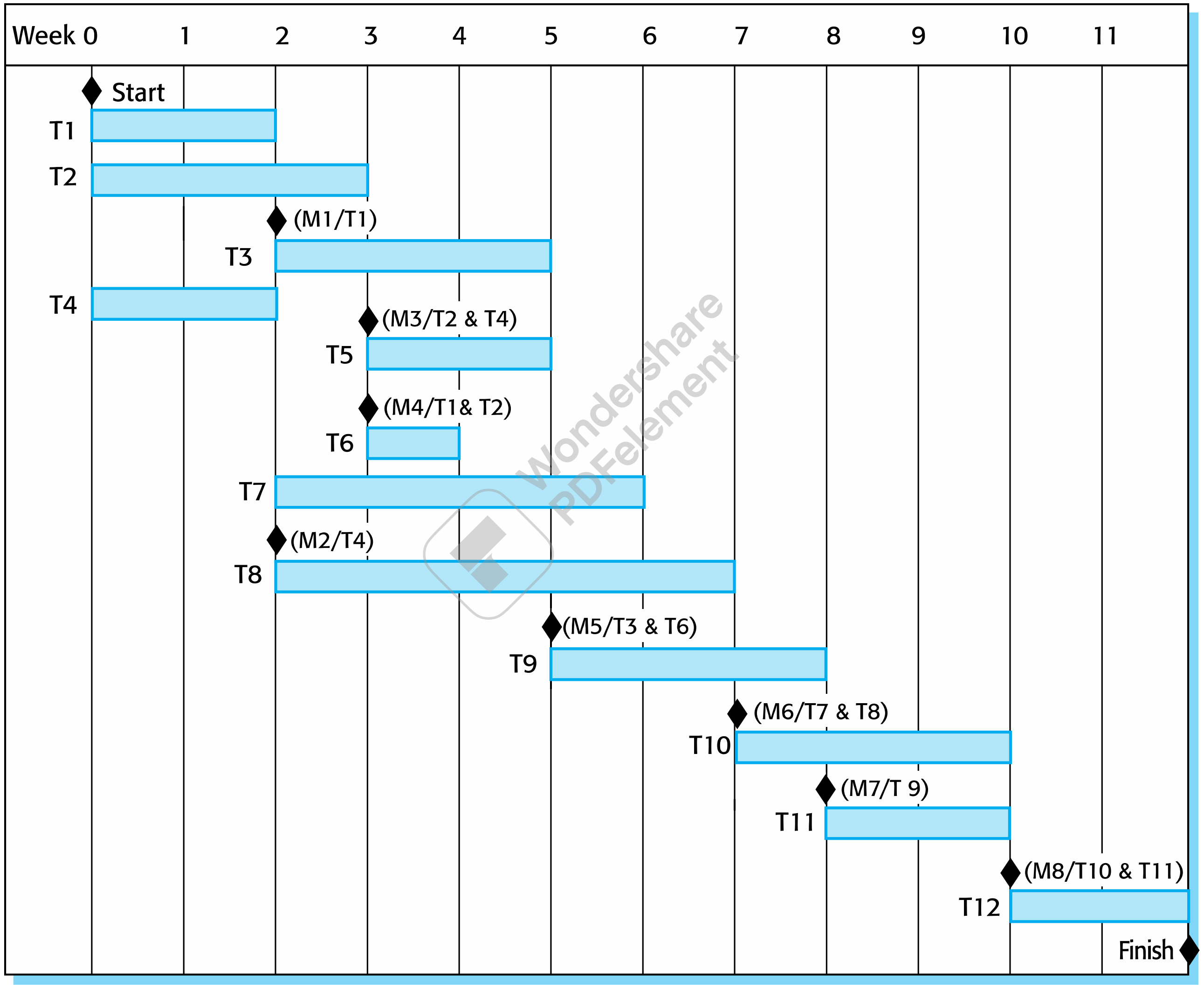


Number	Task	Start	End	Duration	February																													
					2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	1	2	3
1	Site Clearing	2/4/2009	2/13/2009	7			🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥																			
2	Removal of trees	2/4/2009	2/7/2009	3			▨	▨	▨																									
3	General Excavation	2/13/2009	2/24/2009	6												🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥								
4	Grading General Area	2/7/2009	2/12/2009	3						▨	▨	▨	▨	▨																				
5	Excavation for trenches	2/12/2009	2/18/2009	3											▨	▨	▨	▨	▨	▨														
6	Placing formwork & reinforcement	2/7/2009	2/11/2009	2						▨	▨	▨	▨																					
7	Install utilities	2/24/2009	2/27/2009	3																							🟥	🟥	🟥					
8	Place concrete	2/27/2009	3/3/2009	2																										🟥	🟥	🟥	🟥	

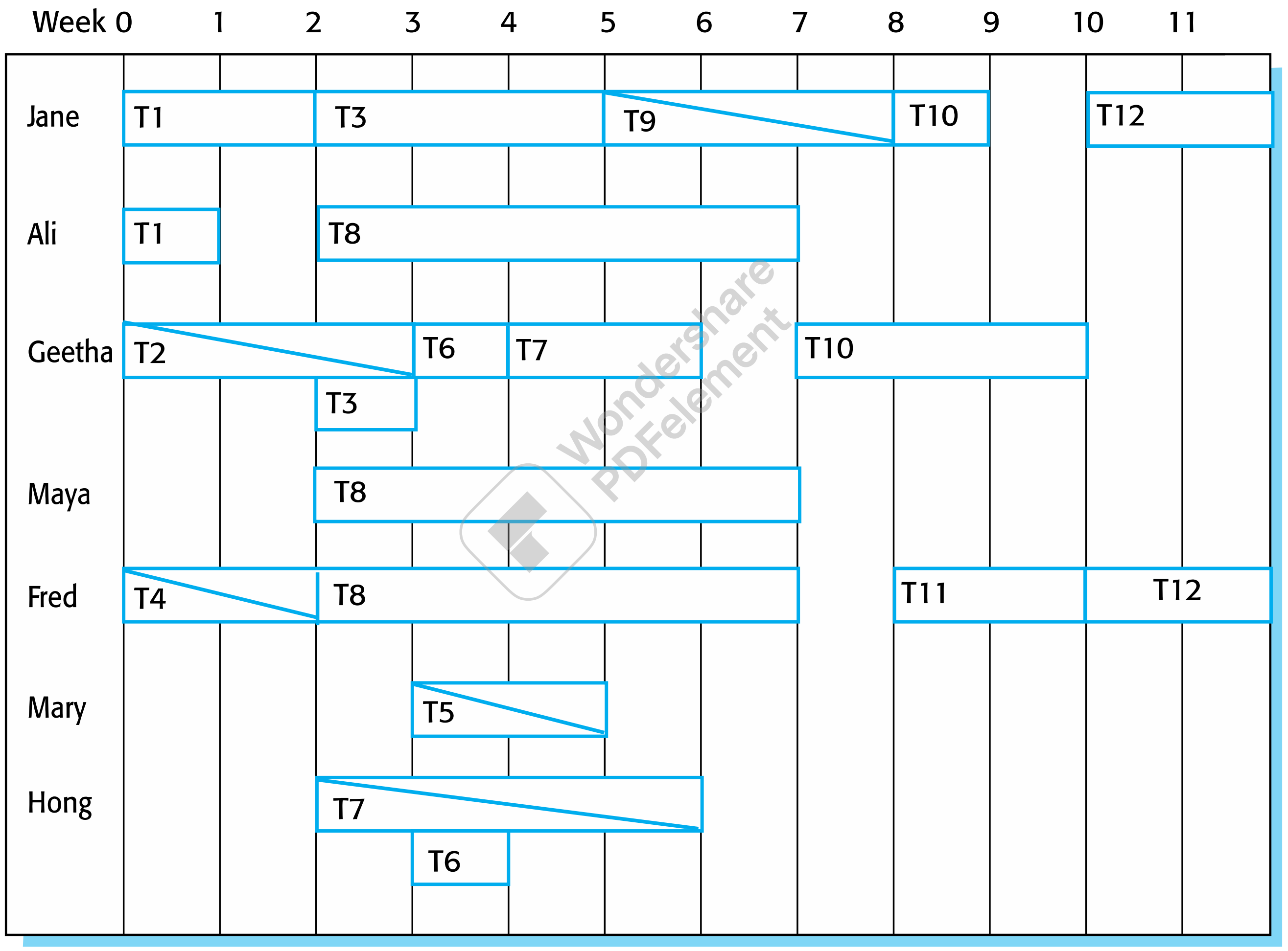
Tasks, durations, and dependencies

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)

Bar Chart


Week 0
1
2
3
4
5
6
7
8
9
10
11
Start
T1
T2
(M1/T1)
T3
T4
(M3/T2 & T4)
T5
(M4/T1& T2)
T6
T7
(M2/T4)
T8
(M5/T3 & T6)
T9
(M6/T7 & T8)
T10
(M7/T 9)
T11
(M8/T10 & T11)
T12
Finish

Staff allocation chart


Week 0
Jane
T1
T3
T9
T10
T12
Ali
T1
T8
Geetha
T2
T6
T7
T10
T3
Maya
T8
Fred
T4
T8
T11
T12
Mary
T5
Hong
T7
T6