# Lecture 14: Software Project Management

EGCI341: WEEK10

# Outline

- Measures of Project Success
- Management Activities
- Project Plan Structure
- Activity Organization
- Project Scheduling
- PERT Chart
- Gantt Chart
- Activity Network
- Risk Management

# Software Project Management

- Concerned with activities involved in ensuring that:
  - Software is delivered *on time* and *on schedule*

- 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

- Product is intangible (abstract)
- Product is uniquely flexible
- Software engineering is not recognized as an engineering discipline
- Software development process is *not standardised*
- Many software projects are 'one-off' projects (made only once)

# Measures of Project Success

- The resulting information system is acceptable to the customer
- System was delivered "*on time*"
- System was delivered "*within budget*"
- System development process had a minimal impact on ongoing business operations

# Project Staffs

- May not be possible to appoint the ideal people to work on a project
  - Project budget may not allow for the use of highly-paid staff
  - Staff with the appropriate experience may not be available

- Managers have to work within these constraints especially when there are shortages of trained staffs

# Management Activities

- Proposal writing
- Project planning and scheduling
- Project costing
- Project monitoring and reviews
- Personnel selection and evaluation
- Report writing and presentations

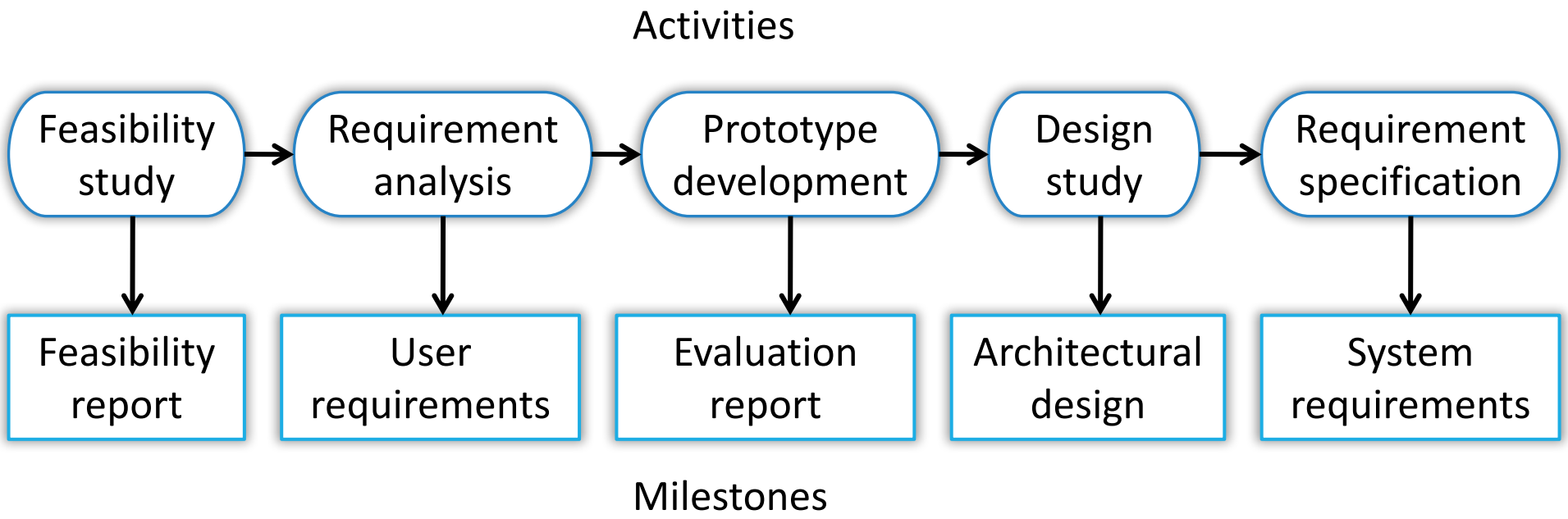
# Project Plan

- Resources available to the project
- Work breakdown
- Schedule for the work

# 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

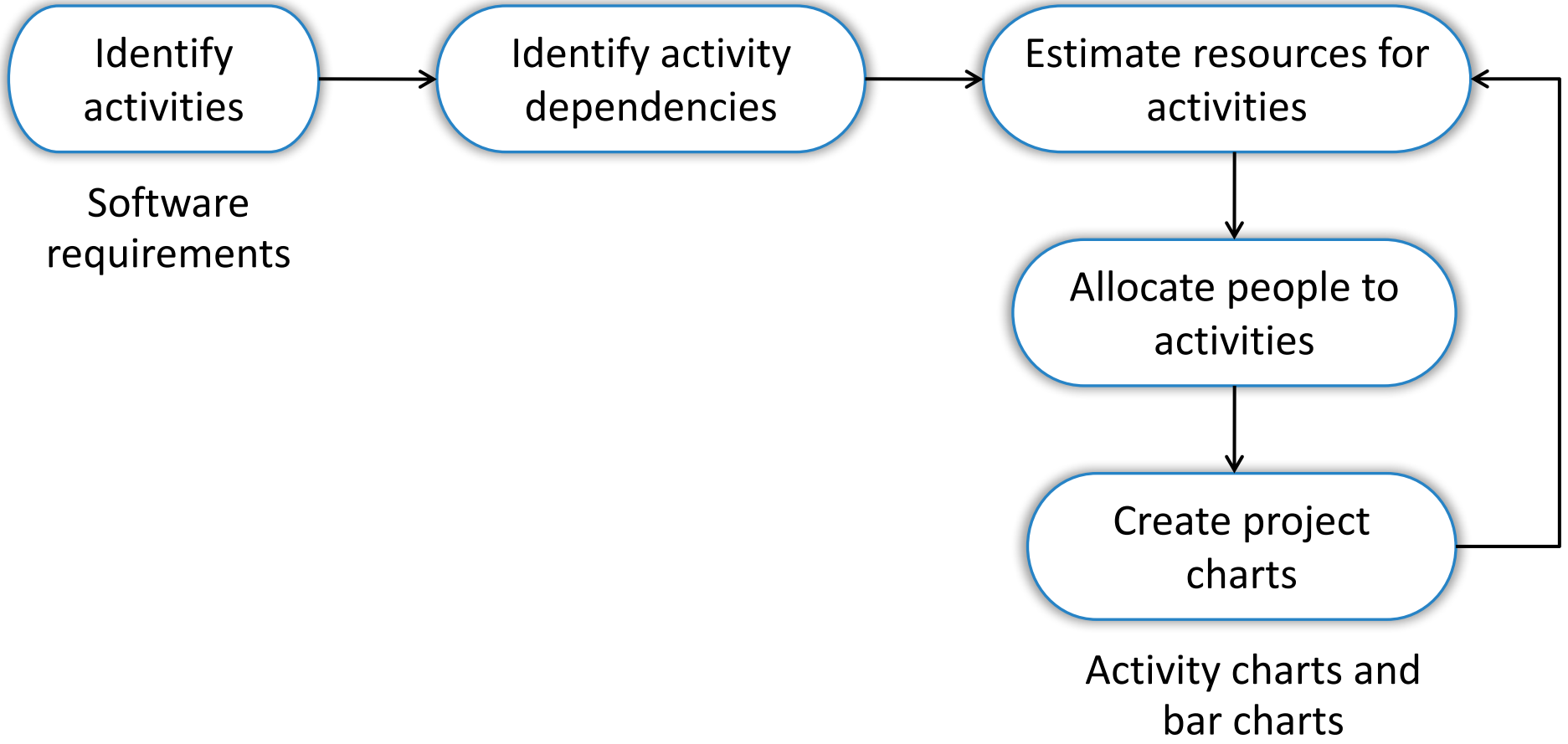
# Activities & Milestones



# 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 managers intuition and experience

# Project Scheduling Process



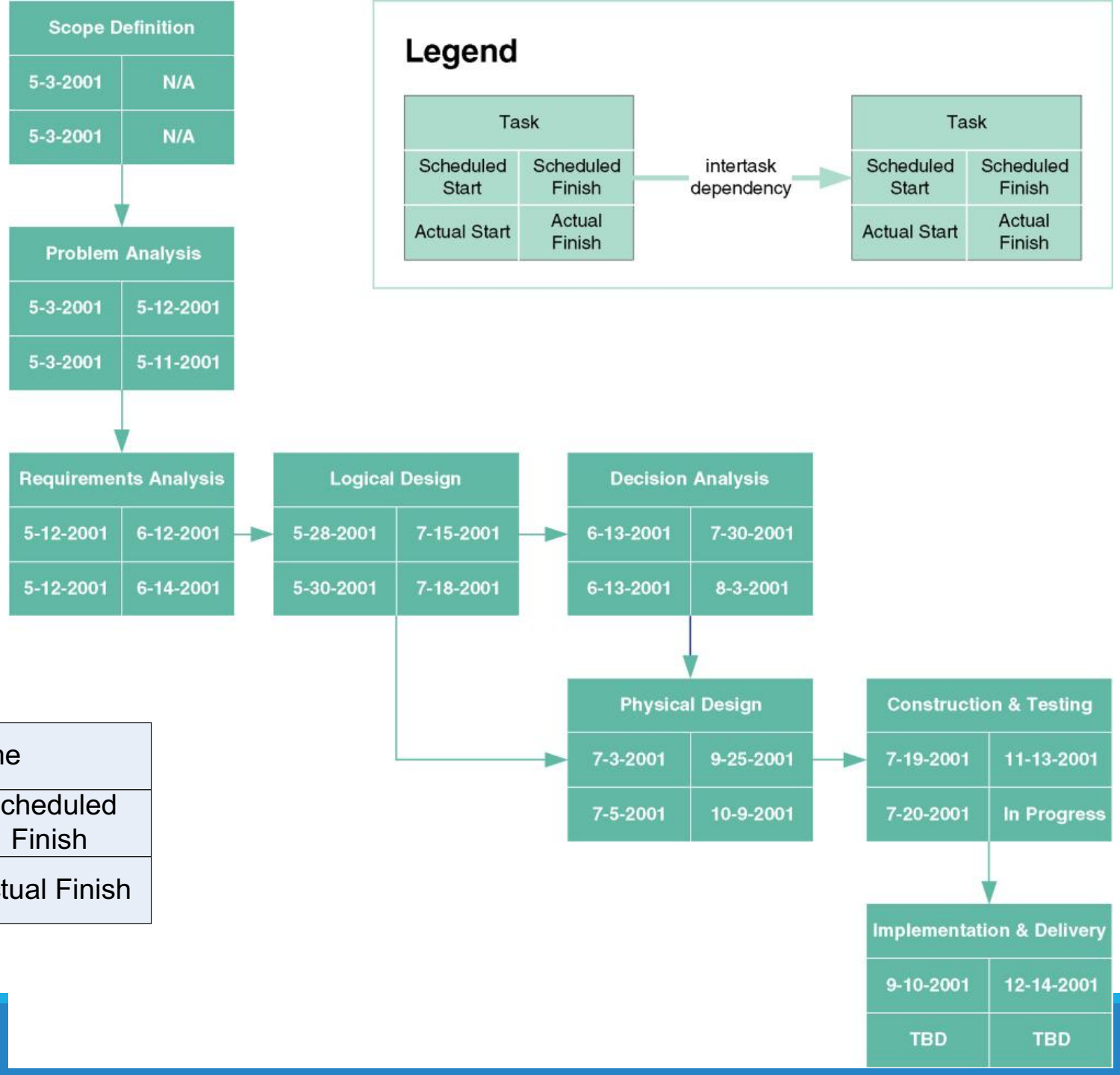
# Scheduling Problems

- 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
- Unexpected always happens
  - Always allow contingency in planning

# Bar Charts and Activity Networks

- Graphical notations used to illustrate the project schedule
- Show project breakdown into tasks
  - Tasks should not be too small
  - They should take about a week or two
- Activity charts show task dependencies and the critical path
- Bar charts show schedule against calendar time

# PERT Chart




| Task Name | |
|---|---|
| Scheduled Start | Scheduled Finish |
| Actual Start | Actual Finish |

# PERT Example

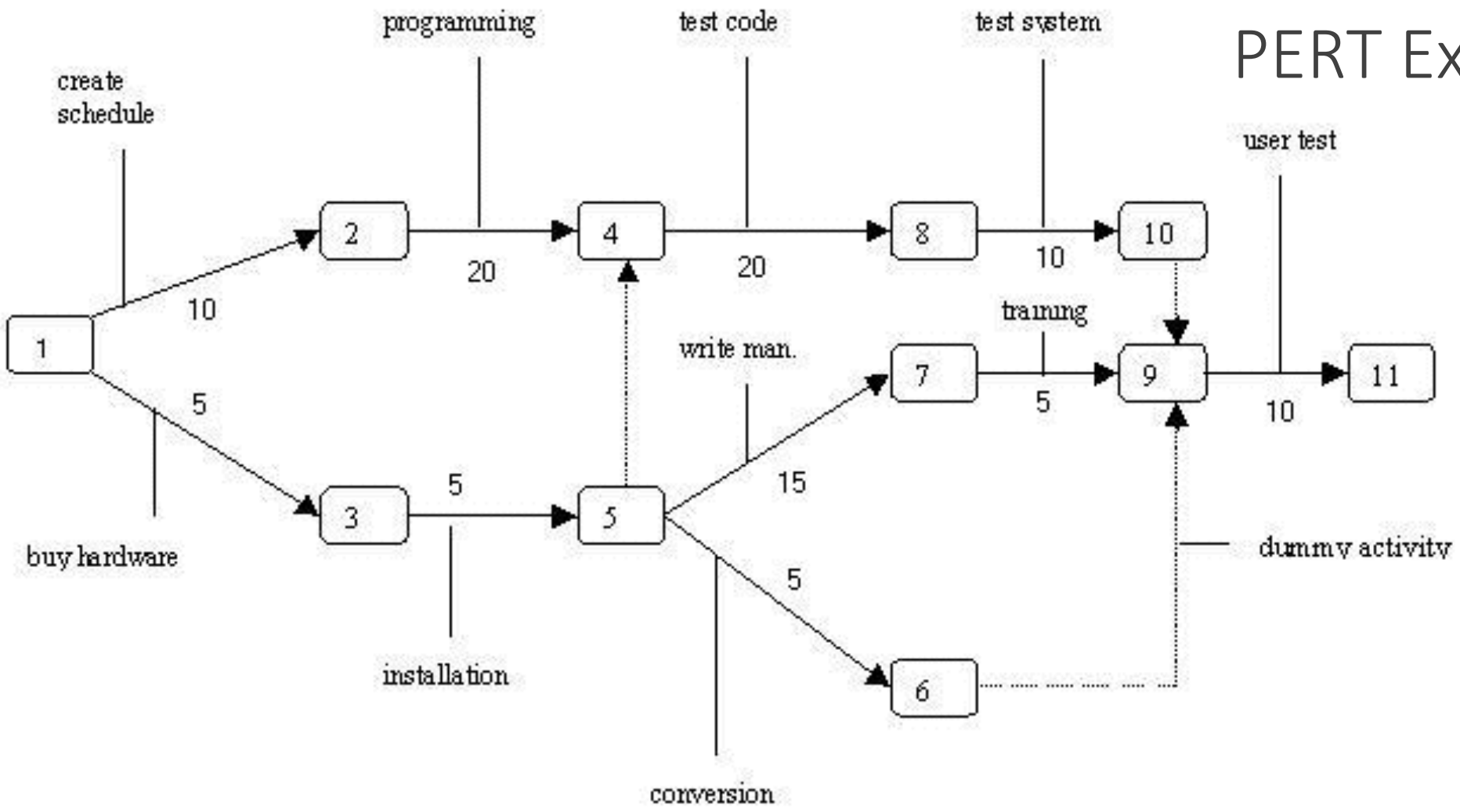
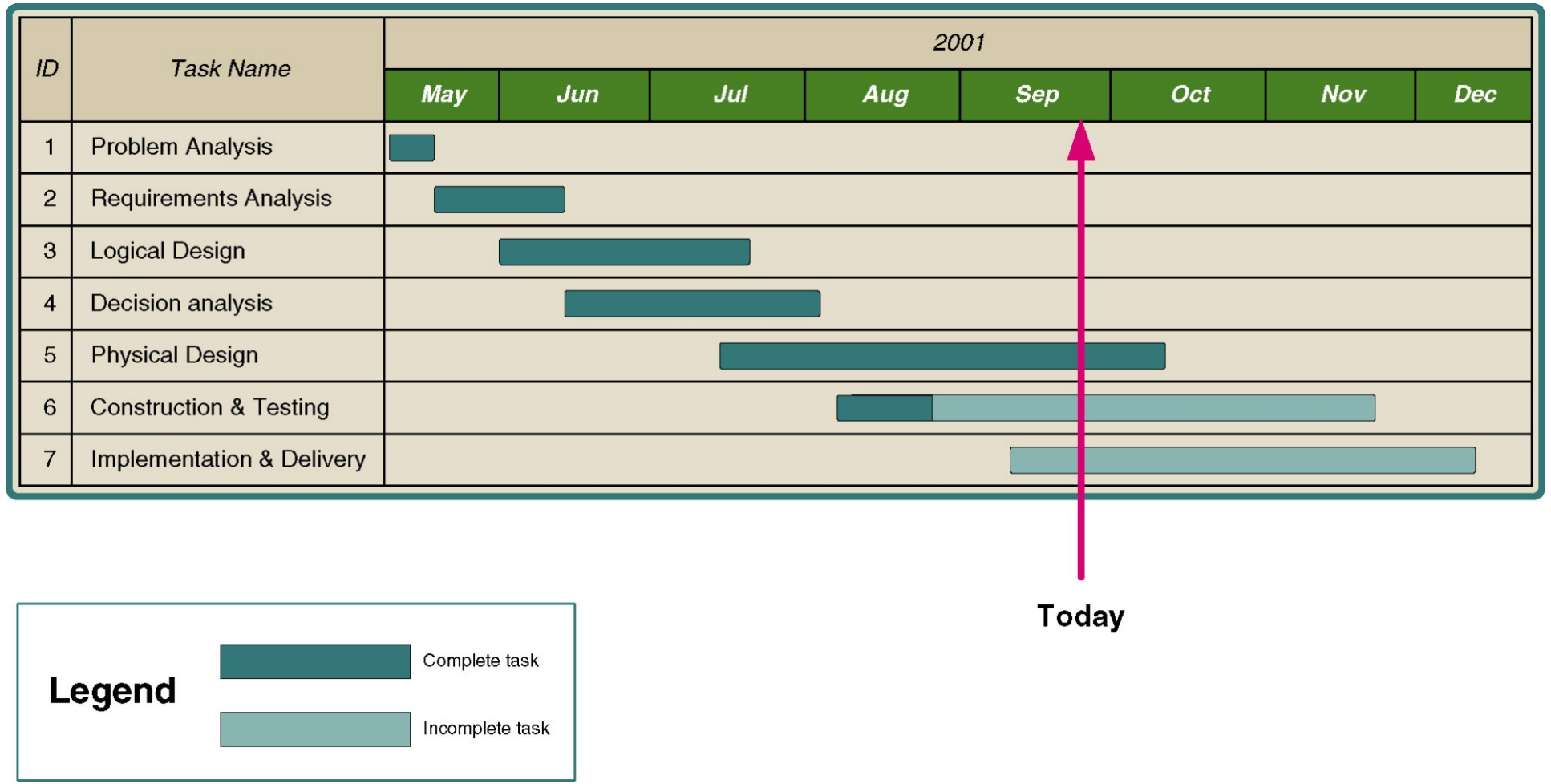



Fig. 1:
PERT Chart

* Numbered rectangles are nodes and represent events or milestones.
* Directional arrows represent dependent tasks that must be completed sequentially.
* Diverging arrow directions (e.g. 1-2 & 1-3) indicate possibly concurrent tasks
* Dotted lines indicate dependent tasks that do not require resources.

http://www.mckinnonsc.vic.edu.au/la/it/ipmnotes/ganttpert/pert_chart.gif

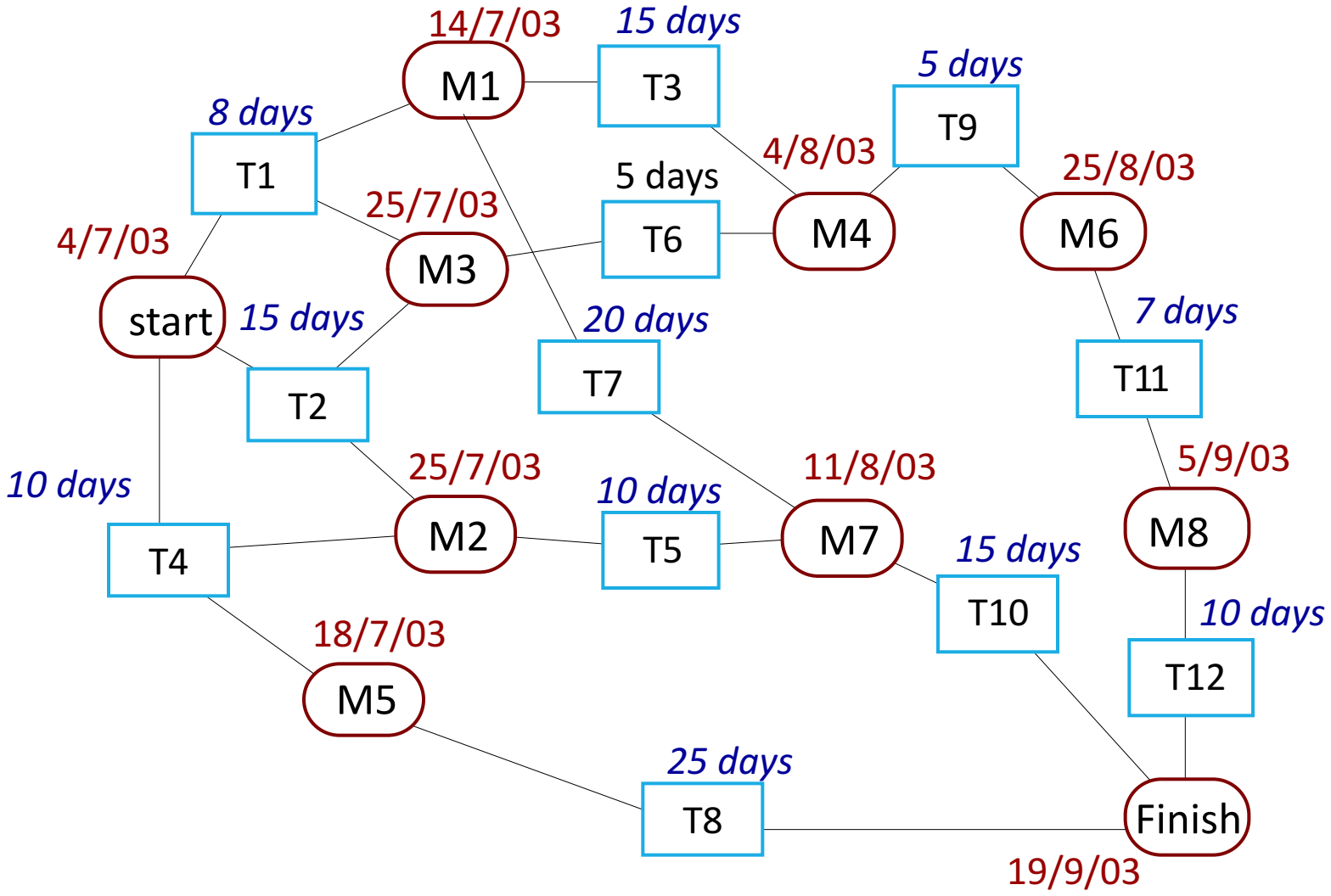
# Gantt Chart


ID
Task Name
2001
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
1 Problem Analysis
2 Requirements Analysis
3 Logical Design
4 Decision analysis
5 Physical Design
6 Construction & Testing
7 Implementation & Delivery
Today
Legend
Complete task
Incomplete task

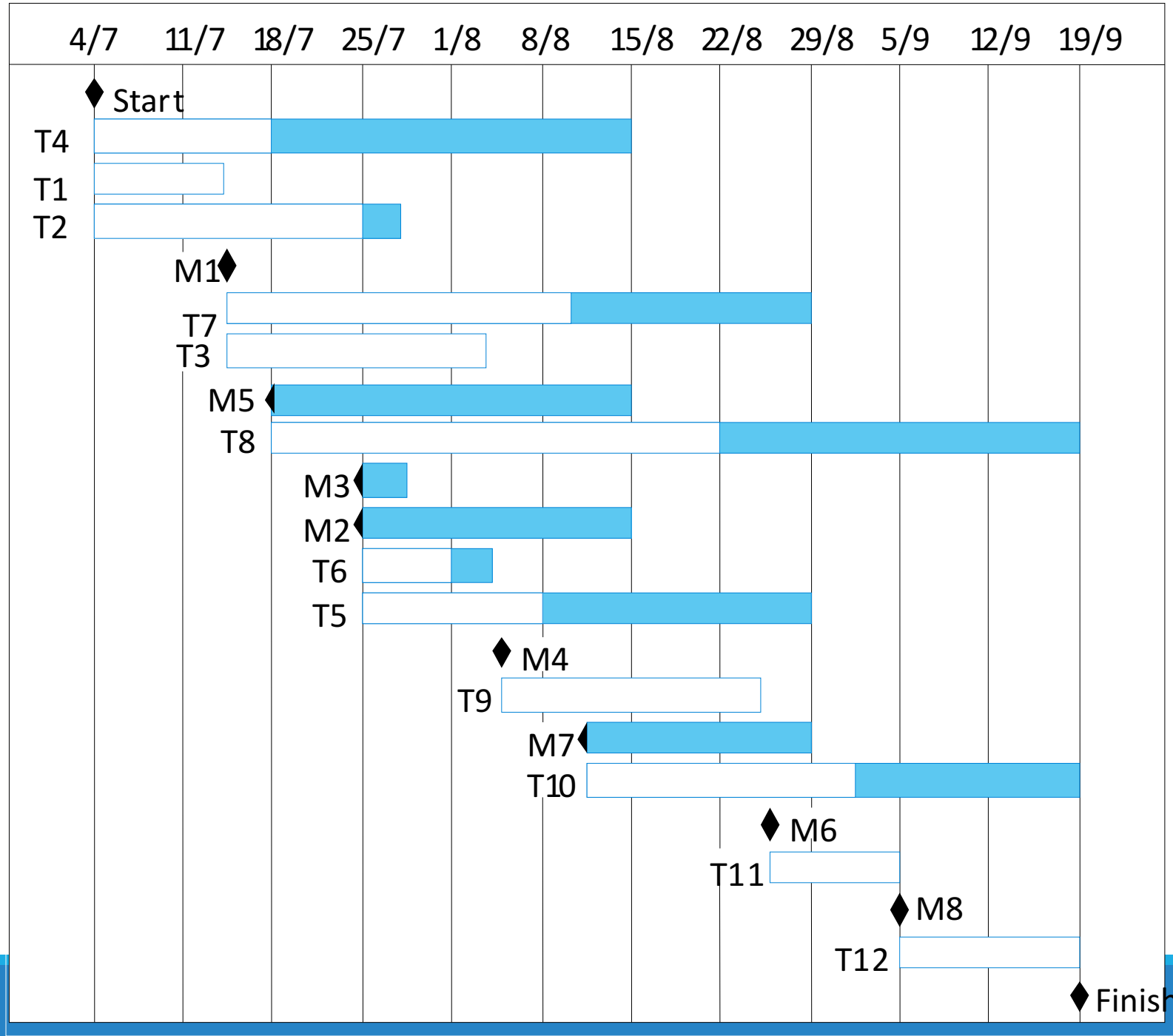
# Task Durations and Dependencies

| Activity | Duration (days) | Dependencies |
|---|---|---|
| T1 | 8 | |
| T2 | 15 | |
| T3 | 15 | T1 (M1) |
| T4 | 10 | |
| T5 | 10 | T2, T4 (M2) |
| T6 | 5 | T1, T2 (M3) |
| T7 | 20 | T1 (M1) |
| T8 | 25 | T4 (M5) |
| T9 | 15 | T3, T6 (M4) |
| T10 | 15 | T5, T7 (M7) |
| T11 | 7 | T9 (M6) |
| T12 | 10 | T11 (M8) |

# Activity Network



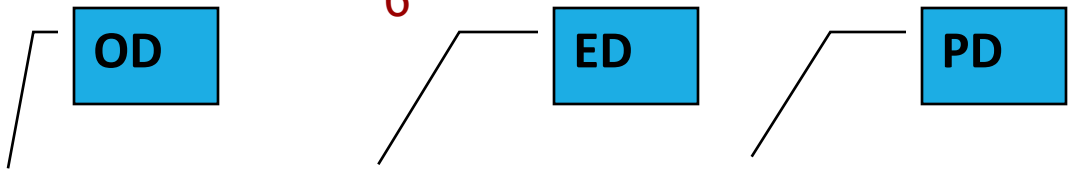
# Activity Timeline



# Estimate Task Durations

1. Estimate the minimum amount of time it would take to perform the task – the optimistic duration (OD)
2. Estimate the maximum amount of time it would take to perform the task – the pessimistic duration (PD)
3. Estimate the expected duration (ED) that will be needed to perform the task
4. Calculate a weighted average of the most likely duration (D) as follows:

$$D = \frac{(1 \times OD) + (4 \times ED) + (1 \times PD)}{6}$$

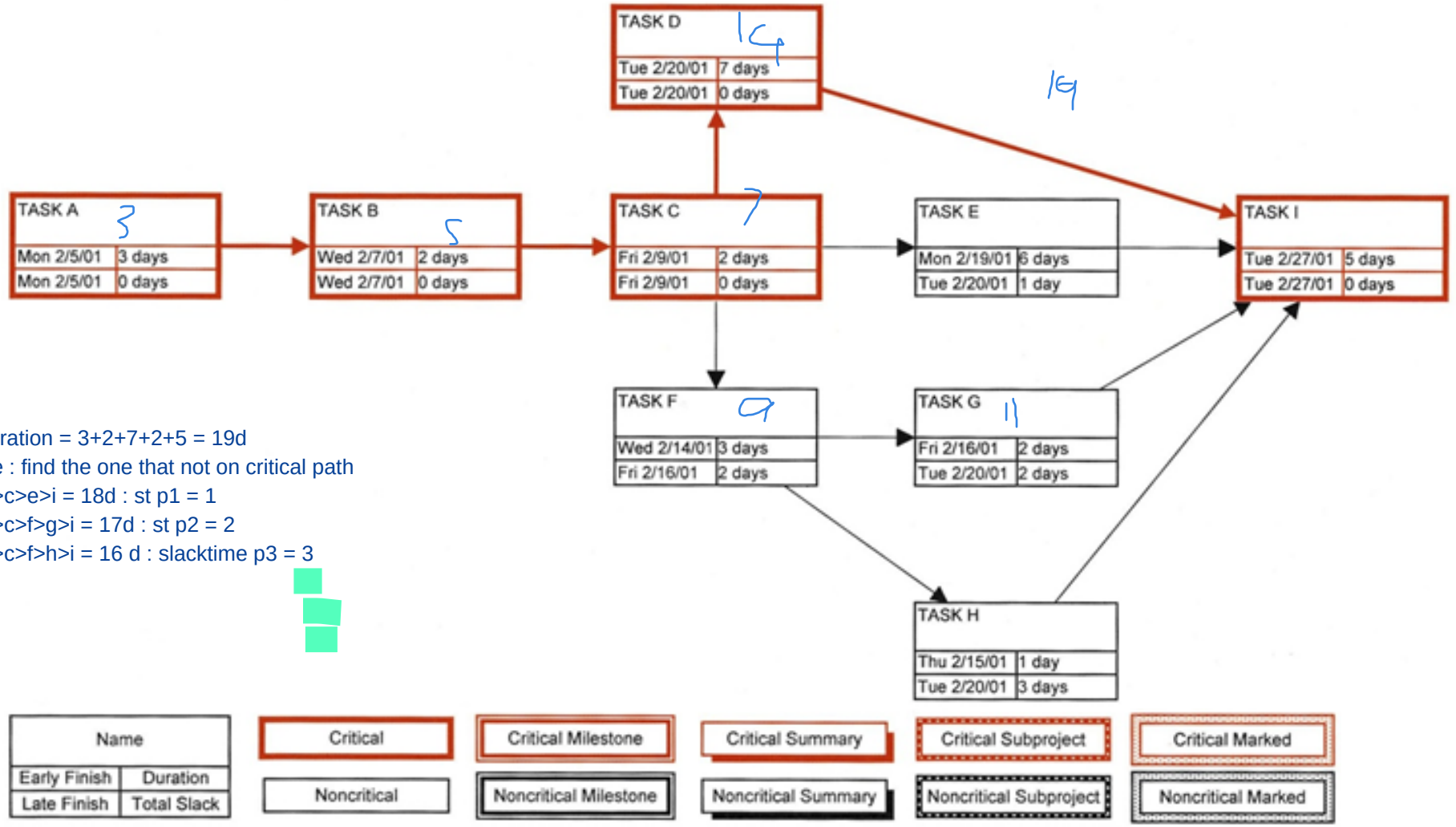



$$3.33 \text{ days} = \frac{(1 \times 2 \text{ days}) + (4 \times 3 \text{ days}) + (1 \times 6 \text{ days})}{6}$$

# Schedule Adjustments

- Using intertask dependencies, determine every possible path through the project
- Sum the durations of all tasks in each path
- **Path with the longest total duration** is the ***critical path***
  - **Critical Path** for a project is that sequence of dependent tasks that have *the largest sum of most likely durations*
    - Critical path determines the earliest completion date of the project
  - **Slack Time** available for any noncritical task is *the amount of delay that can be tolerated between the starting time and completion time of a task* without causing a delay in the completion date of the entire project

# Critical Path Analysis




Critical duration = 3+2+7+2+5 = 19d
Slack time : find the one that not on critical path
P1 = a>b>c>e>i = 18d : st p1 = 1
P2 = a>b>c>f>g>i = 17d : st p2 = 2
P3 = a>b>c>f>h>i = 16 d : slacktime p3 = 3

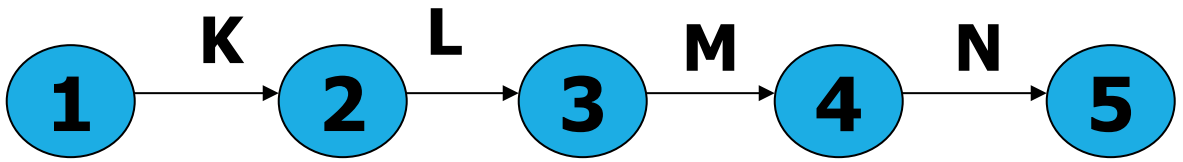
# Example I

K – integration testing

L – Install Software

M – Write Manual

N – Train Users



# Example II

| Task | Days | Dependencies/ Predecessor |
|---|---|---|
| A | 5 | |
| B | 3 | |
| C | 8 | A, B |
| D | 4 | B |
| E | 4 | A |
| F | 6 | |
| G | 4 | C, D, E, F |

# Example III

- Draw Pert diagram for the scheduling table below
- Show the critical path and calculate duration of critical path
- Draw Gantt chart

| Task | Description | Duration (Working Days) | Predecessor/s |
|---|---|---|---|
| A | Requirement Analysis | 5 | |
| B | Systems Design | 15 | A |
| C | Programming | 25 | B |
| D | telecoms | 15 | B |
| E | Hardware Installation | 30 | B |
| F | Integration | 10 | C, D |
| G | System Testing | 10 | E, F |
| H | Training/Support | 5 | G |
| I | Handover and Go-Live | 5 | H |

# Reference

- This set of slides and examples are modified  from Ian Sommerville, Software Engineering 8th Edition, Addison-Wesley; 2007

Any Questions?

:O)

Thank you