System Analysis and Design

Project Planning

Prof. Rania Elgohary
rania.elgohary@cis.asu.edu.eg
Dr. Yasmine Afify
yasmine.afify@cis.asu.edu.eg

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 into account factors such as staff costs, hardware costs, software costs, etc..

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 progres**ses and you know more about the software and its development
- The project schedule, cost-estimate and risks have to be regularly **revised**

Software Pricing

Software pricing

- Estimates are made to discover the cost, to the developer, of producing a software system.
  - You take into account, hardware, software, travel, training and effort costs.
- There is not a simple relationship between the development cost and the price charged to the customer.
- Broader organisational, economic, political and business considerations influence the price charged.

Factors affecting software pricing

Factor	Description
Contractual terms	A customer may be willing to allow the developer to retain ownership of the source code and reuse it in other projects. The price charged may then be less than if the software source code is handed over to the customer.
Cost estimate uncertainty	If an organization is unsure of its cost estimate, it may increase its price by a contingency over and above its normal profit.
Financial health	Developers in financial difficulty may lower their price to gain a contract. It is better to make a smaller than normal profit or break even than to go out of business. Cash flow is more important than profit in difficult economic times.

Factors affecting software pricing

Factor	Description
Market opportunity	A development organization may quote a low price because it wishes to move into a new segment of the software market. Accepting a low profit on one project may give the organization the opportunity to make a greater profit later. The experience gained may also help it develop new products.
Requirements volatility	If the requirements are likely to change, an organization may lower its price to win a contract. After the contract is awarded, high prices can be charged for changes to the requirements.

Pricing strategies

- Under pricing
  - A company may underprice a system in order **to gain a contract that allows them to retain staff for future opportunities**
  - A company may underprice a system **to gain access to a new market area**
- Increased pricing
  - The price may be increased when **a buyer wishes a fixed-price contract and so the seller increases the price to allow for unexpected risks**

Plan-driven Development

Plan-driven development

- Plan-driven development is an approach to software engineering where the development process is planned in detail.
- Plan-driven development 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

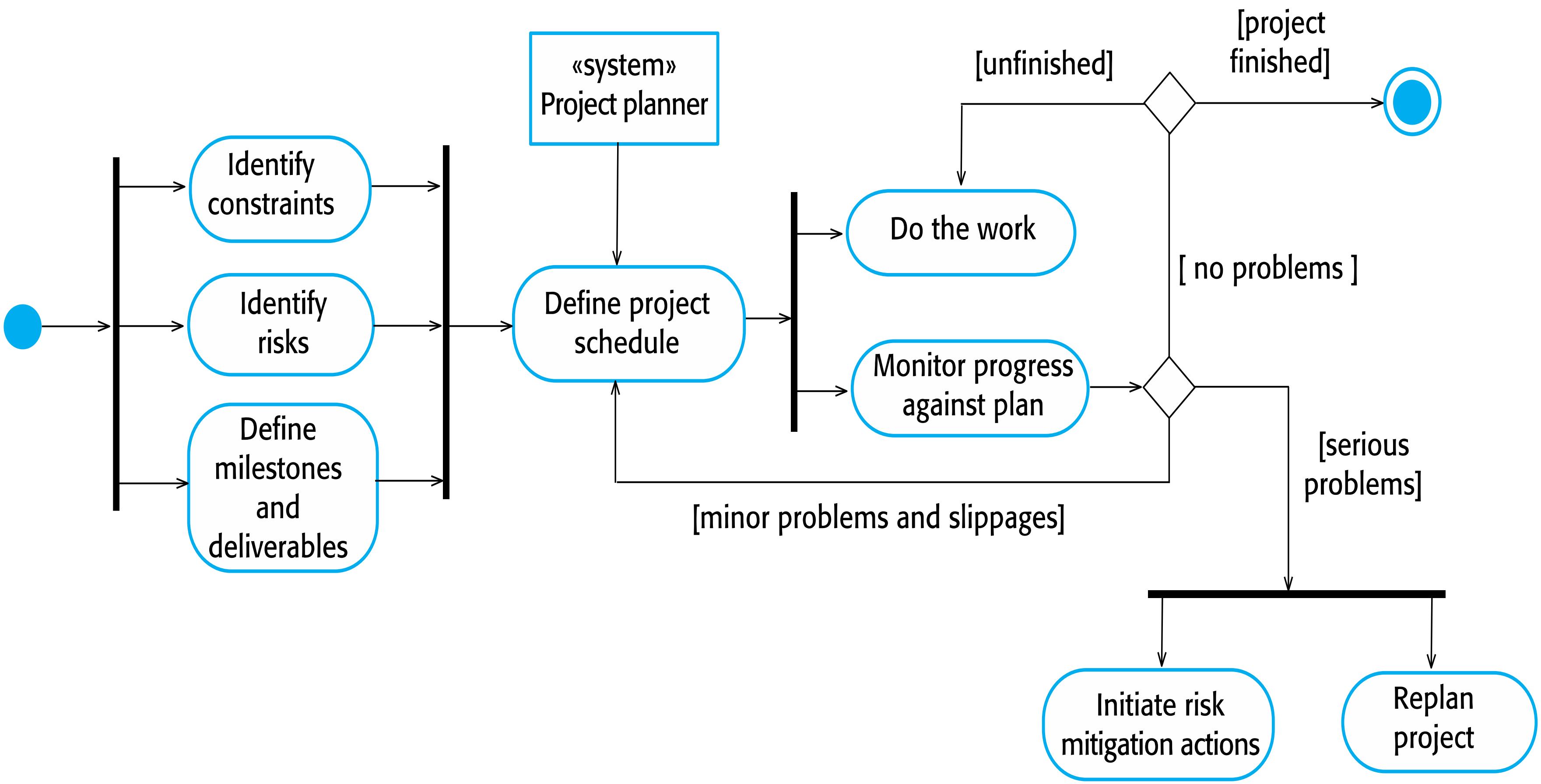
Project plan supplements

Plan	Description
Configuration management plan	Describes the configuration management procedures and structures to be used.
Deployment plan	Describes how the software and associated hardware (if any) will be deployed in the customer's environment. This should include a plan for migrating data from existing systems.
Maintenance plan	Predicts the maintenance requirements, costs, and effort.
Quality plan	Describes the quality procedures and standards that will be used in a project.
Validation plan	Describes the approach, resources, and schedule used for system validation.

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



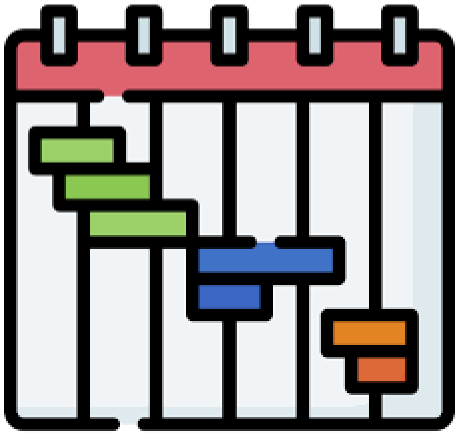
Planning assumptions

- You should make **realistic** rather than optimistic assumptions when you are defining a project plan.
- Problems always arise during a project, and these lead to project **delays**.
- Your initial assumptions and scheduling should therefore take unexpected problems into account.
- You should include **contingency** in your plan so that if things go wrong, then your delivery schedule is not seriously interrupted.

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

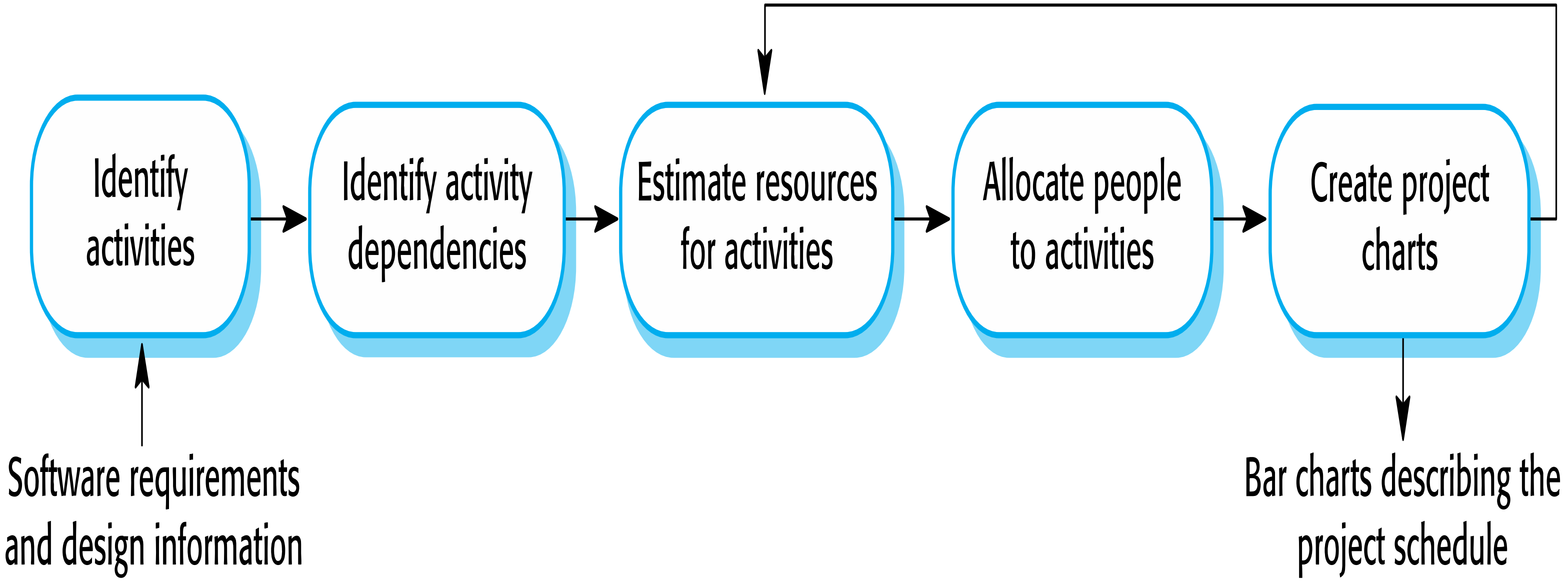
- Project scheduling is the process of deciding how the work in a project will be organized as **separate tasks**, and **when** and how these tasks will be executed.
- You **estimate** the **calendar time** needed to complete each task, the effort required, and **who** will work on the tasks that have been identified.
- You also 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.

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 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.

System project management

- ✧ **ensuring that system is delivered**

1. **on time and**
2. **on schedule and**
3. **in accordance with the Rs**.

- ✧ SD subject to **budget and schedule** constraints.

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 **the critical path**.
- **Bar charts show schedule against calendar time.**

Activity Definition

- An **activity** or **task** is an element of work normally found on the WBS that has an expected duration, a cost, and resource requirements.
- Project schedules grow out of the basic documents that initiate a project.
  - The project charter includes start and end dates and budget information.
  - The scope statement and WBS help define what will be done.
- Activity definition involves developing a more detailed WBS and supporting explanations to understand all the work to be done, so you can develop realistic cost and duration estimates.

Activity Lists and Attributes

- An **activity list** is a tabulation of activities to be included on a project schedule. The list should include:
  - The activity name
  - An activity identifier or number
  - A brief description of the activity
- **Activity attributes** provide more information about each activity, such as predecessors, successors, logical relationships, resource requirements, constraints, imposed dates, and assumptions related to the activity.

Milestones

- A **milestone** is a significant event that normally using for controlling, monitoring and evaluating if the work of progress done on time, schedule within budget or not.

- It often takes several activities and a lot of work to complete a milestone.

- Milestones are useful tools for setting schedule goals and monitoring progress.

- **Examples include completion and customer sign-off on key documents and completion of specific products.**

Activity Sequencing

- ✧ Involves reviewing activities and determining dependencies.

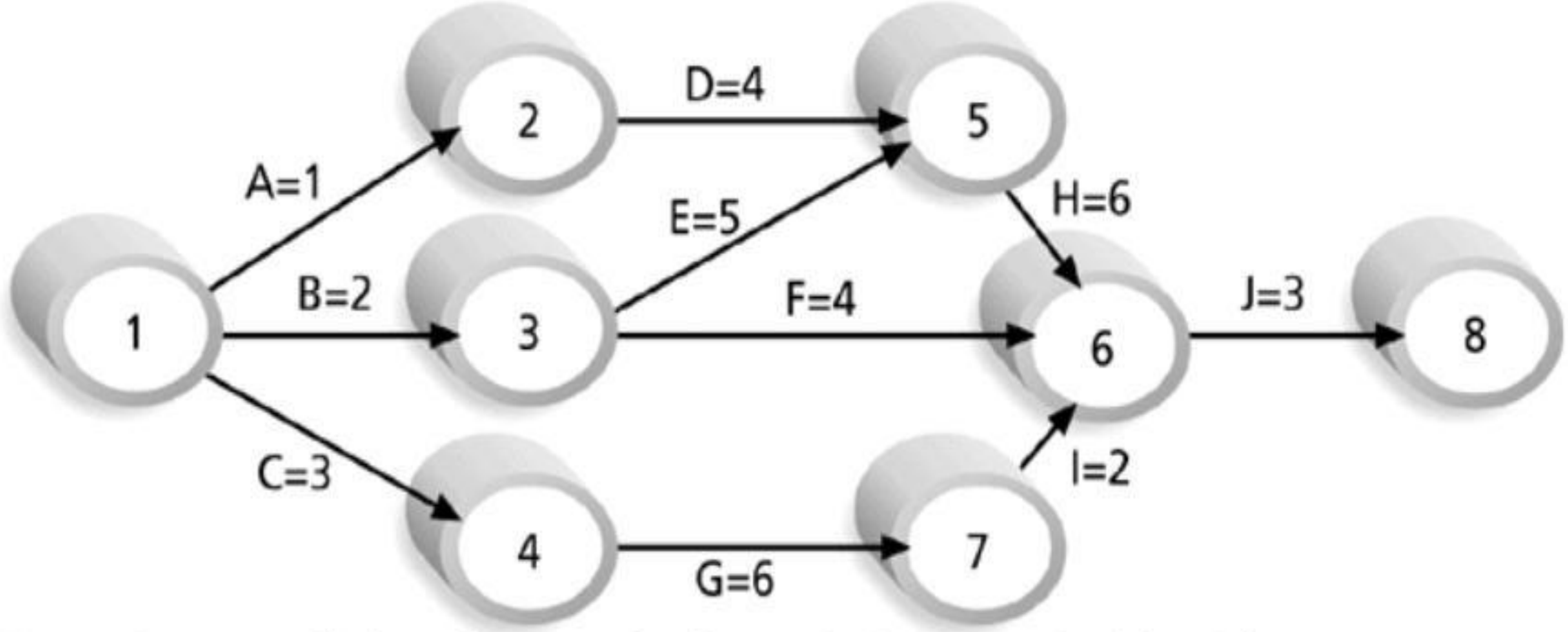
- ✧ A **dependency** or **relationship** relates to the sequencing of project activities or tasks.

- ✧ You *must* determine **dependencies** in order to use critical path analysis.

Activity Network Diagram

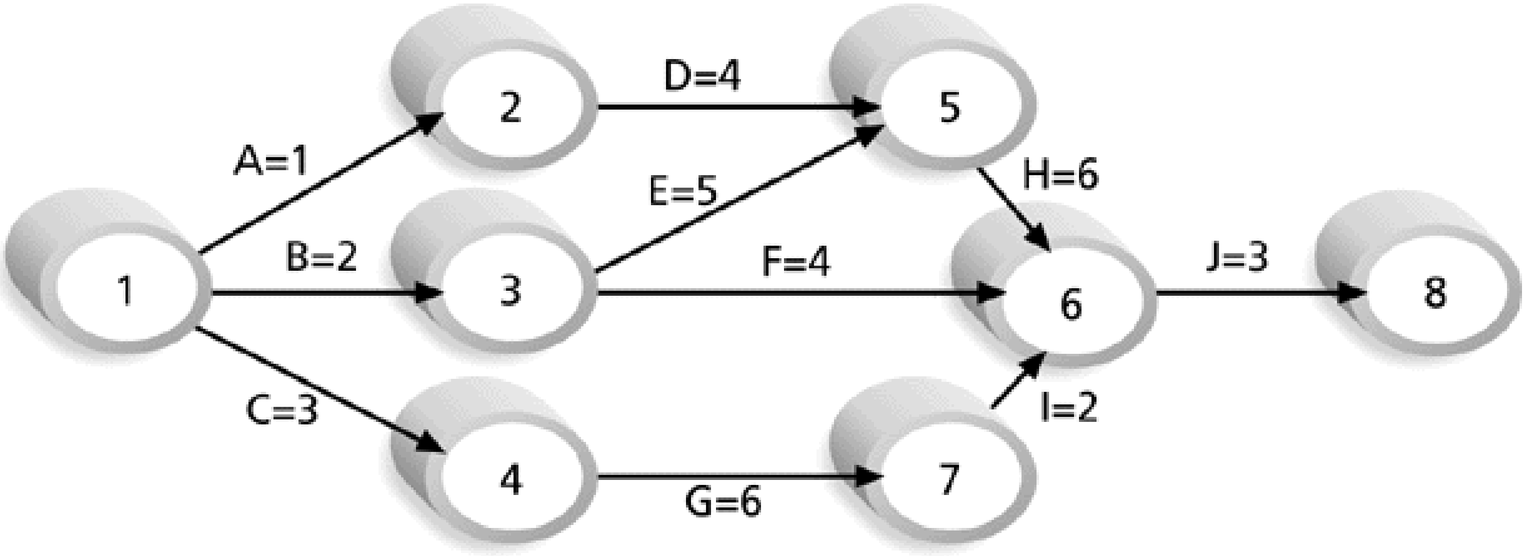
- It is a diagram of project activities that shows the sequential **relationships of activities** using arrows and nodes.
- An activity network diagram tool is necessary for the identification of a **project's critical path** (which is used to determine the expected completion time of the project).

Figure -1. Sample Activity-on-Arrow (AOA) Network Diagram for Project X



Note: Assume all durations are in days; A=1 means Activity A has a duration of 1 day.

Determining the Critical Path for Project X

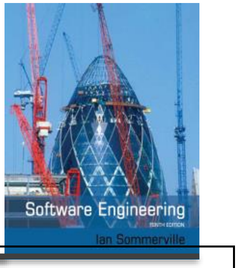


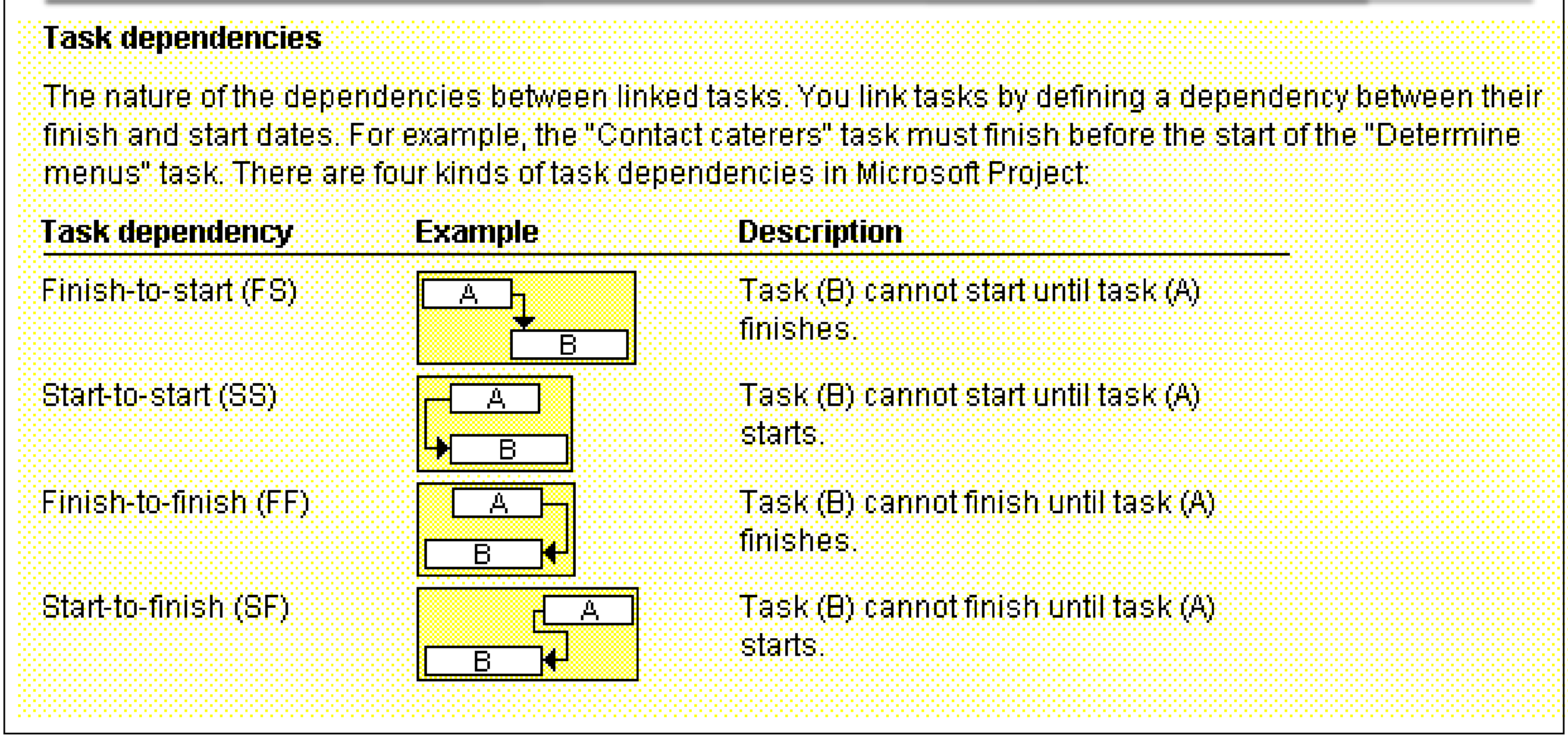
Note: Assume all durations are in days.

Path 1:	A-D-H-J	Length = 1+4+6+3 = 14 days
Path 2:	**B-E-H-J**	**Length = 2+5+6+3 = 16 days**
Path 3:	B-F-J	Length = 2+4+3 = 9 days
Path 4:	C-G-I-J	Length = 3+6+2+3 = 14 days

Since the critical path is the longest path through the network diagram, Path 2, B-E-H-J, is the critical path for Project X.

Figure 2. Task Dependency Types





Task dependencies

The nature of the dependencies between linked tasks. You link tasks by defining a dependency between their finish and start dates. For example, the "Contact caterers" task must finish before the start of the "Determine menus" task. There are four kinds of task dependencies in Microsoft Project:

Task dependency	Example	Description
Finish-to-start (FS)	A B	Task (B) cannot start until task (A) finishes.
Start-to-start (SS)	A B	Task (B) cannot start until task (A) starts.
Finish-to-finish (FF)	A B	Task (B) cannot finish until task (A) finishes.
Start-to-finish (SF)	A B	Task (B) cannot finish until task (A) starts.

Activity Duration Estimating

- **Duration** includes the actual amount of time worked on an activity *plus* the elapsed time.

- **Effort** is the number of workdays or work hours required to complete a task.

- Effort does not normally equal duration.

- People doing the work should help create estimates, and an expert should review them.

Three-Point Estimates

- Instead of providing activity estimates as a discrete number, such as four weeks, it’s often helpful to create a **three-point estimate**:
  - An estimate that includes an optimistic, most likely, and pessimistic estimate, such as three weeks for the optimistic, four weeks for the most likely, and five weeks for the pessimistic estimate.
- **Three-point estimates are needed for PERT estimates and Monte Carlo simulations.**

Schedule Development

- Uses results of the other time management processes to determine the start and end dates of the project.

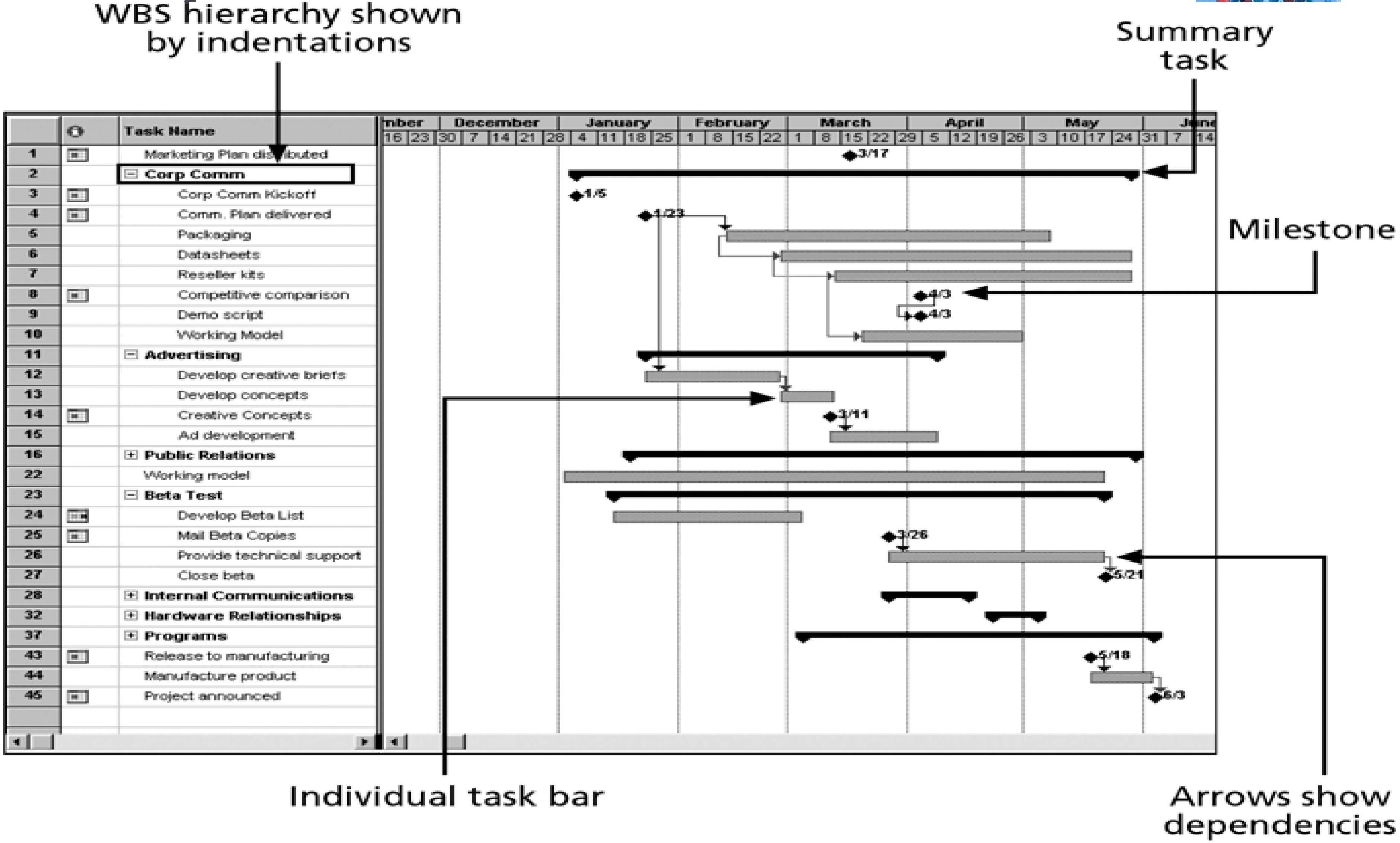
- Ultimate goal is to create a realistic project schedule that provides a basis for monitoring project progress for the time dimension of the project.

- Important tools and techniques include Gantt charts, critical path analysis, critical chain scheduling, and PERT analysis.

Gantt Charts

- **Gantt charts** provide a standard format for displaying project schedule information by listing project activities and their corresponding start and finish dates in a calendar format.
- Symbols include:
  - **Black diamonds**: Milestones
  - **Thick black bars**: Summary tasks
  - **Lighter horizontal bars**: Durations of tasks
  - **Arrows**: Dependencies between tasks

Gantt Chart for Software Launch Project



WBS hierarchy shown by indentations

Summary task

Milestone

Individual task bar

Arrows show dependencies

	Task Name
1	Marketing Plan distributed
2	⊟ Corp Comm
3	Corp Comm Kickoff
4	Comm. Plan delivered
5	Packaging
6	Datasheets
7	Reseller kits
8	Competitive comparison
9	Demo script
10	Working Model
11	⊟ Advertising
12	Develop creative briefs
13	Develop concepts
14	Creative Concepts
15	Ad development
16	⊞ Public Relations
22	Working model
23	⊟ Beta Test
24	Develop Beta List
25	Mail Beta Copies
26	Provide technical support
27	Close beta
28	⊞ Internal Communications
32	⊞ Hardware Relationships
37	⊞ Programs
43	Release to manufacturing
44	Manufacture product
45	Project announced

Critical Path Method (CPM)

✧ **CPM is a network diagramming technique used to predict total project duration.**

✧ A **critical path** for a project is the series of activities that determines **the *earliest time*** by which the project can be completed.

✧ The critical path is **the *longest path*** through the network diagram and **has the least amount of slack or float**.

✧ **Slack** or **float** is **the amount of time an activity can be delayed without delaying a succeeding activity or the project finish date.**

Program Evaluation and Review Technique (PERT)

- ✧ PERT is a network analysis technique used to estimate project duration when there is a high degree of uncertainty about the individual activity duration estimates.

- ✧ PERT uses **probabilistic time estimates**:

  - ▪ Duration estimates based on using optimistic, most likely, and pessimistic estimates of activity durations, or a three-point estimate.

PERT Formula and Example

- **PERT weighted average =**

$$\frac{\textbf{optimistic time + 4 X most likely time + pessimistic time}}{\textbf{6}}$$

- Example:

PERT weighted average =

$$\frac{\text{8 workdays + 4 X 10 workdays + 24 workdays}}{6} \quad = \textbf{12 days}$$

where:

optimistic time= 8 days

most likely time = **10 days**

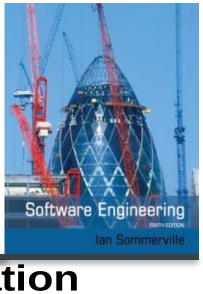
pessimistic time = 24 days

Therefore, you’d use **12 days** on the network diagram instead of 10 when using PERT for the above example.

Schedule Control

- Perform reality checks on schedules.
- Allow for contingencies.
- Don’t plan for everyone to work at 100 percent capacity all the time.
- Hold progress meetings with stakeholders and be clear and honest in communicating schedule issues.

Critical Path Method (CPM) Problem – *Question*



Activity	Dependencies	Duration
A	-	2
B	-	2
C	-	4
D	-	8
E	A, F	3
F	B	4
G	C, D, E	3
H	D, G	2
I	E	7
J	G	6

Critical Path Method (CPM) Problem – *Question*

Using the table of data provided, answer the following questions:

✧ What is the *minimum time* required to complete the project?

✧ What is the *Critical Path* for this project?

✧ What is the *Early Start* (ES) date for activity E?

✧ What is the *Late Start* (LS) date for activity A?

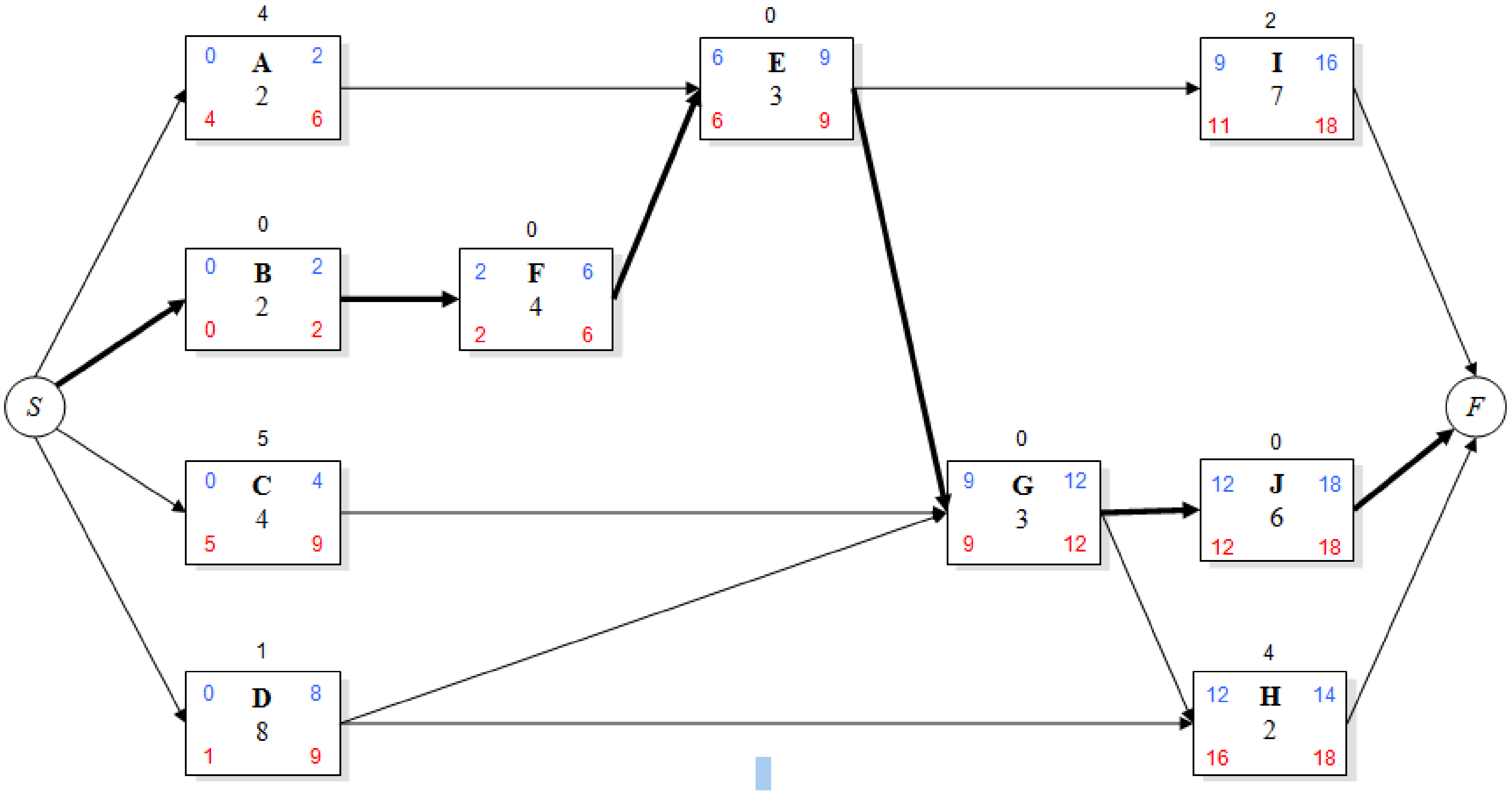
✧ What is the *Late Finish* (LF) date for activity F?

✧ What is the *Total Float / Slack* for activity H?

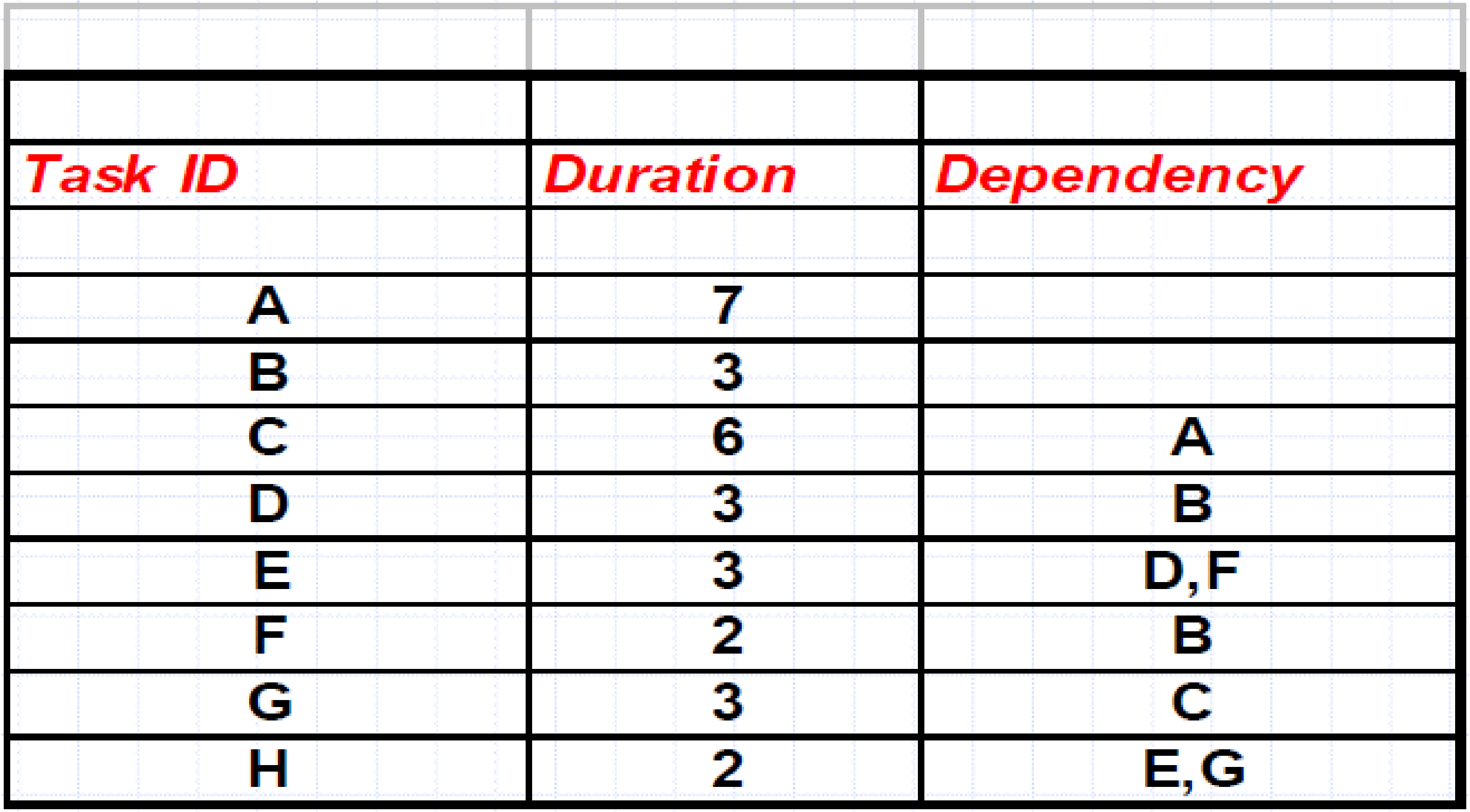
✧ What is the *Free Float / Slack* between activities D & G?

Solution

Software Engineering
Ian Sommerville

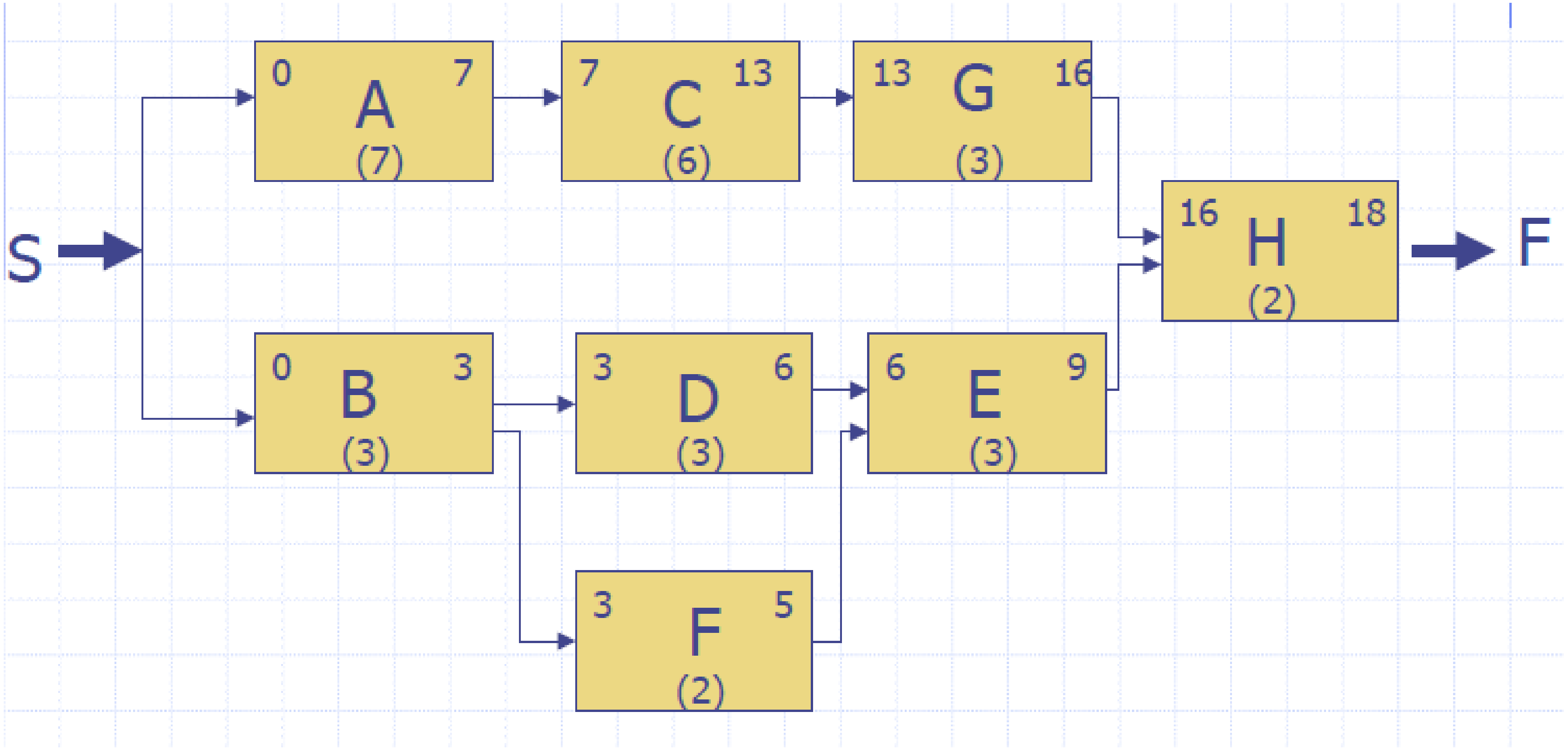

4
0 A 2
2
4 6
0
6 E 9
3
6 9
2
9 I 16
7
11 18
0
0 B 2
2
0 2
0
2 F 6
4
2 6
S
F
5
0 C 4
4
5 9
0
9 G 12
3
9 12
0
12 J 18
6
12 18
1
0 D 8
8
1 9
4
12 H 14
2
16 18

Example

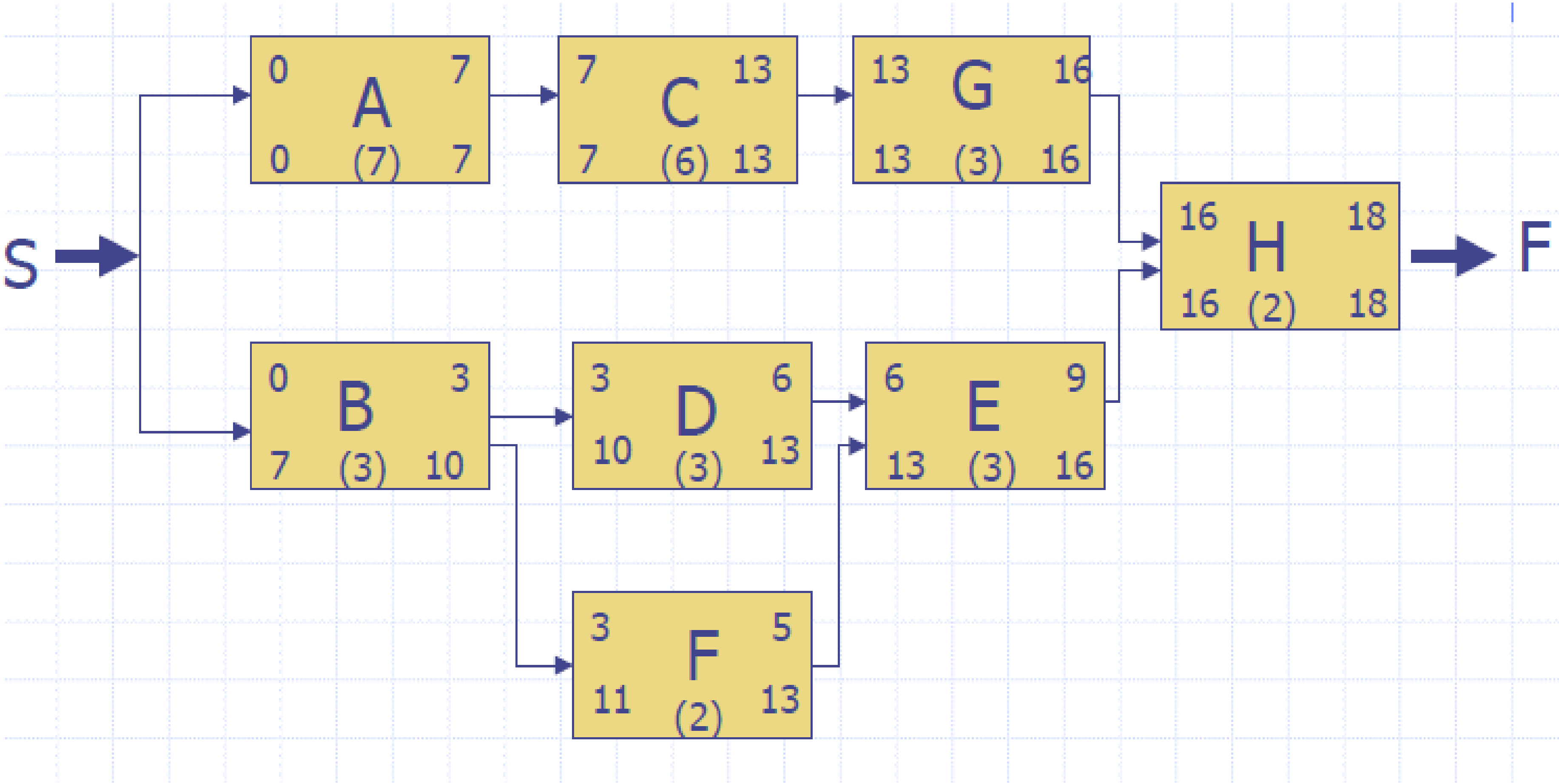


Task ID	Duration	Dependency
A	7	
B	3	
C	6	A
D	3	B
E	3	D,F
F	2	B
G	3	C
H	2	E,G

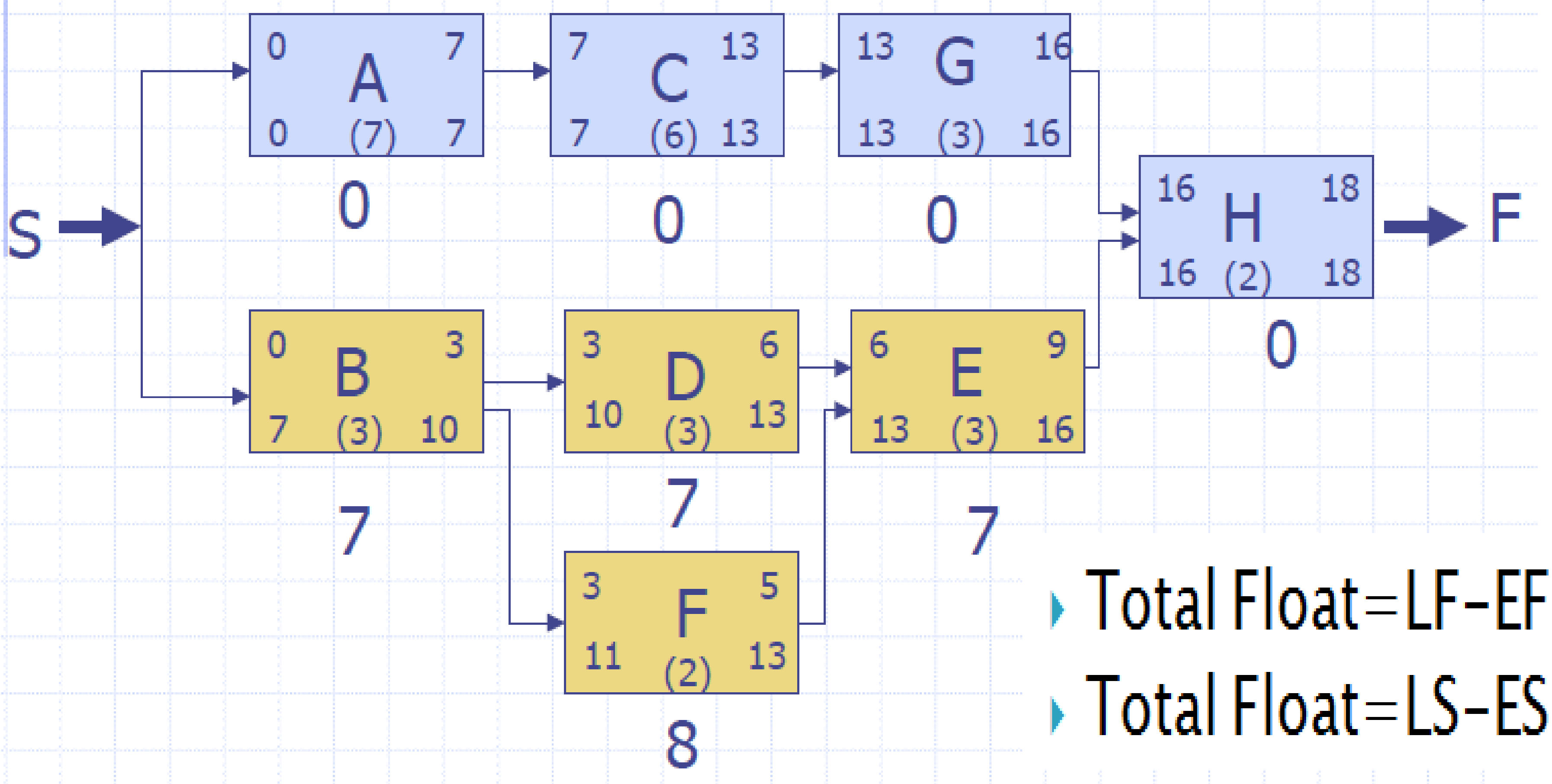
Calculating the forward pass



Calculating the backward pass



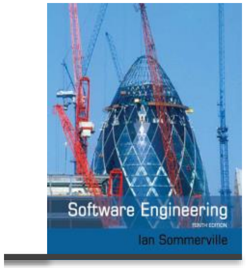
Calculating the float/critical path



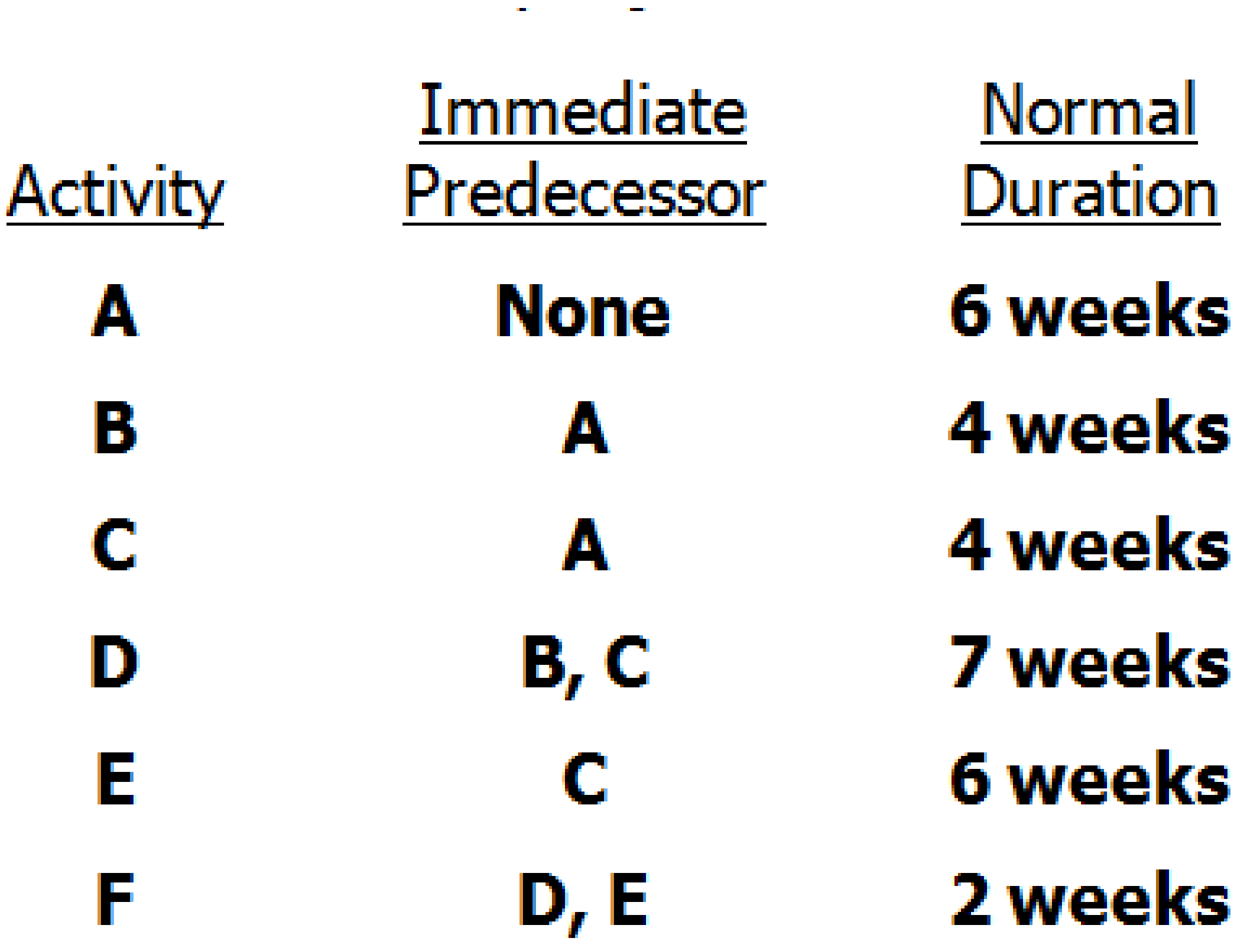
- Total Float=LF-EF
- Total Float=LS-ES

What’s the critical path?

✧ Path: A-C-G-H

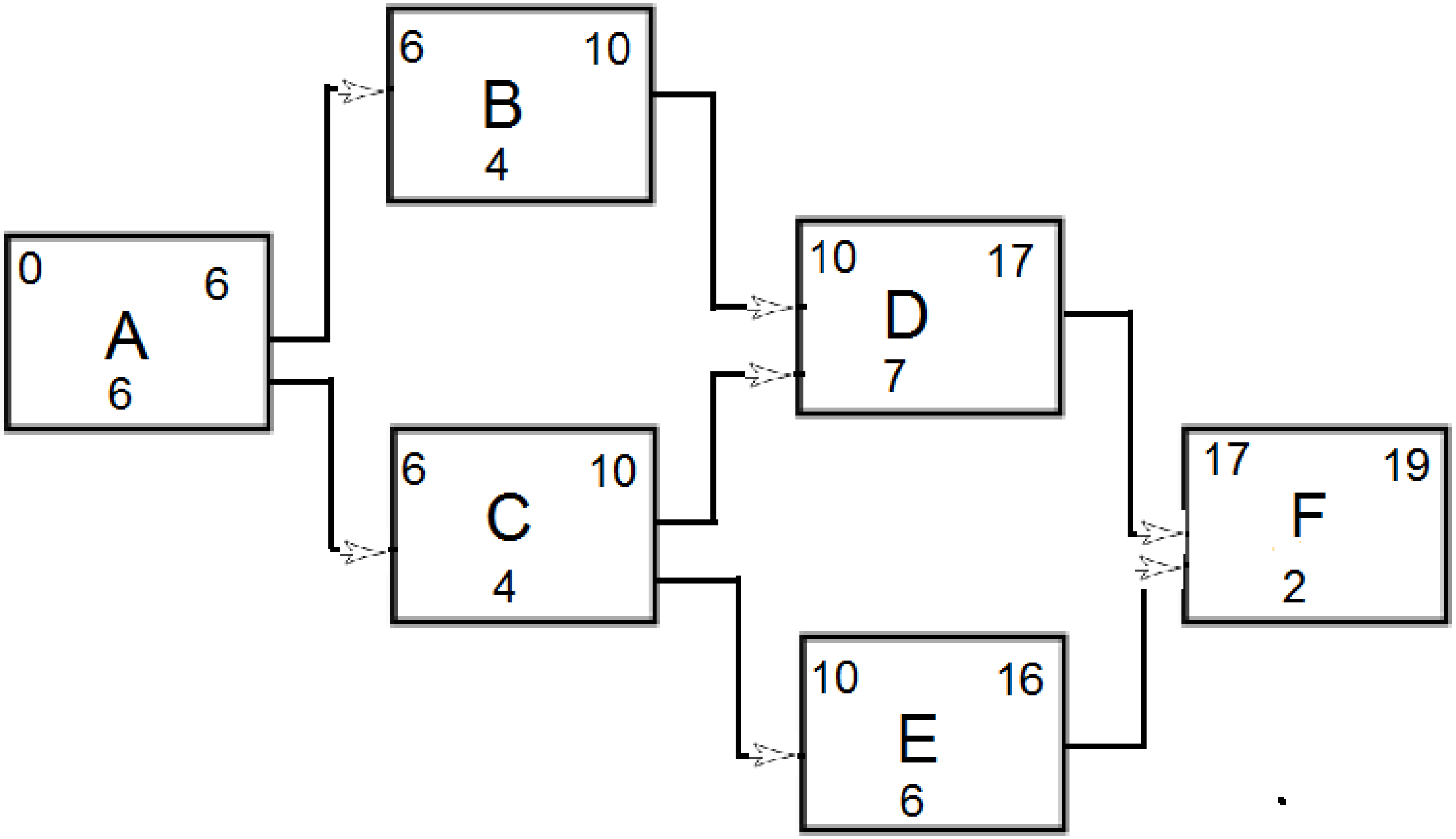


Example

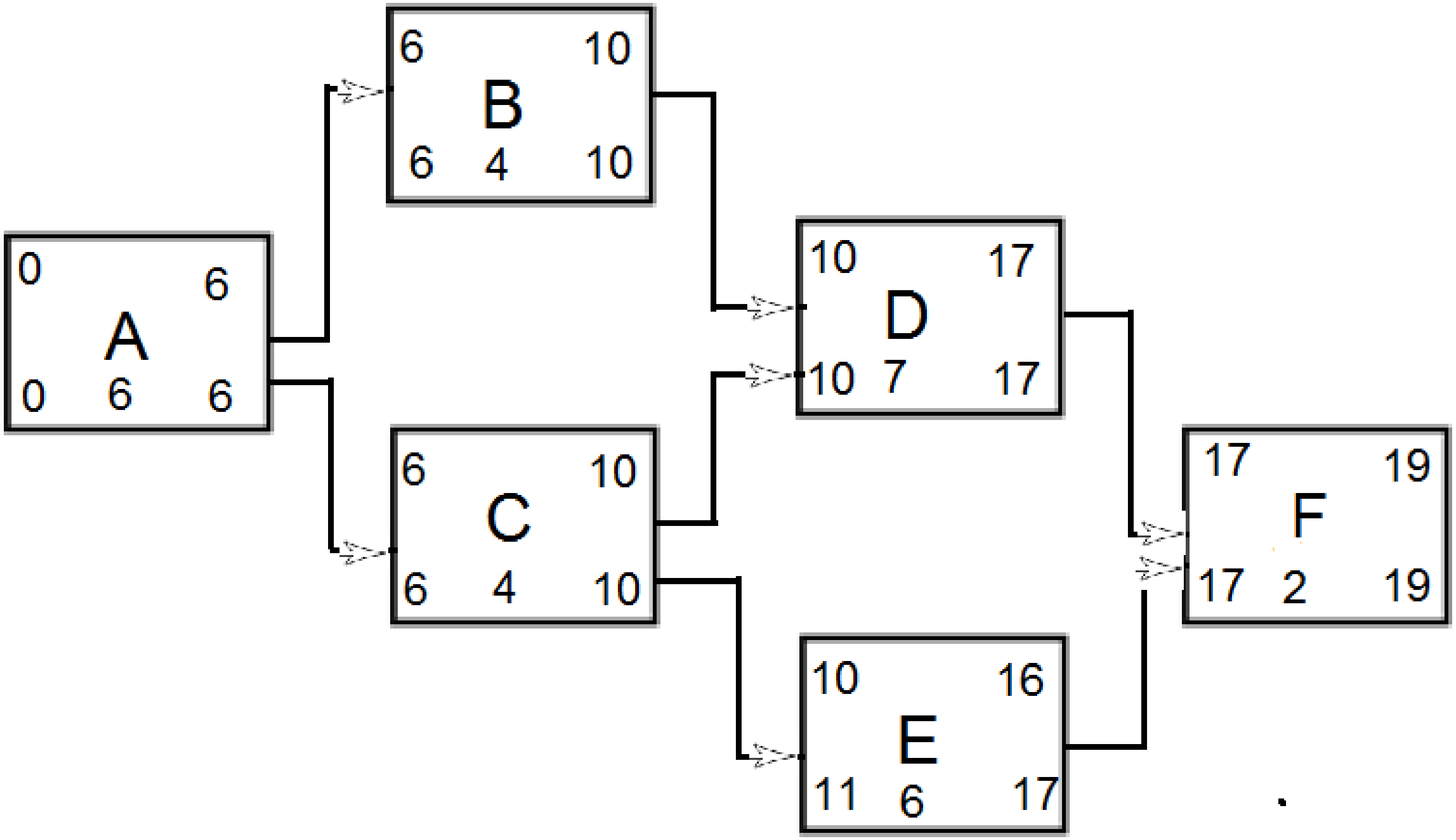


Activity	Immediate Predecessor	Normal Duration
A	None	6 weeks
B	A	4 weeks
C	A	4 weeks
D	B, C	7 weeks
E	C	6 weeks
F	D, E	2 weeks

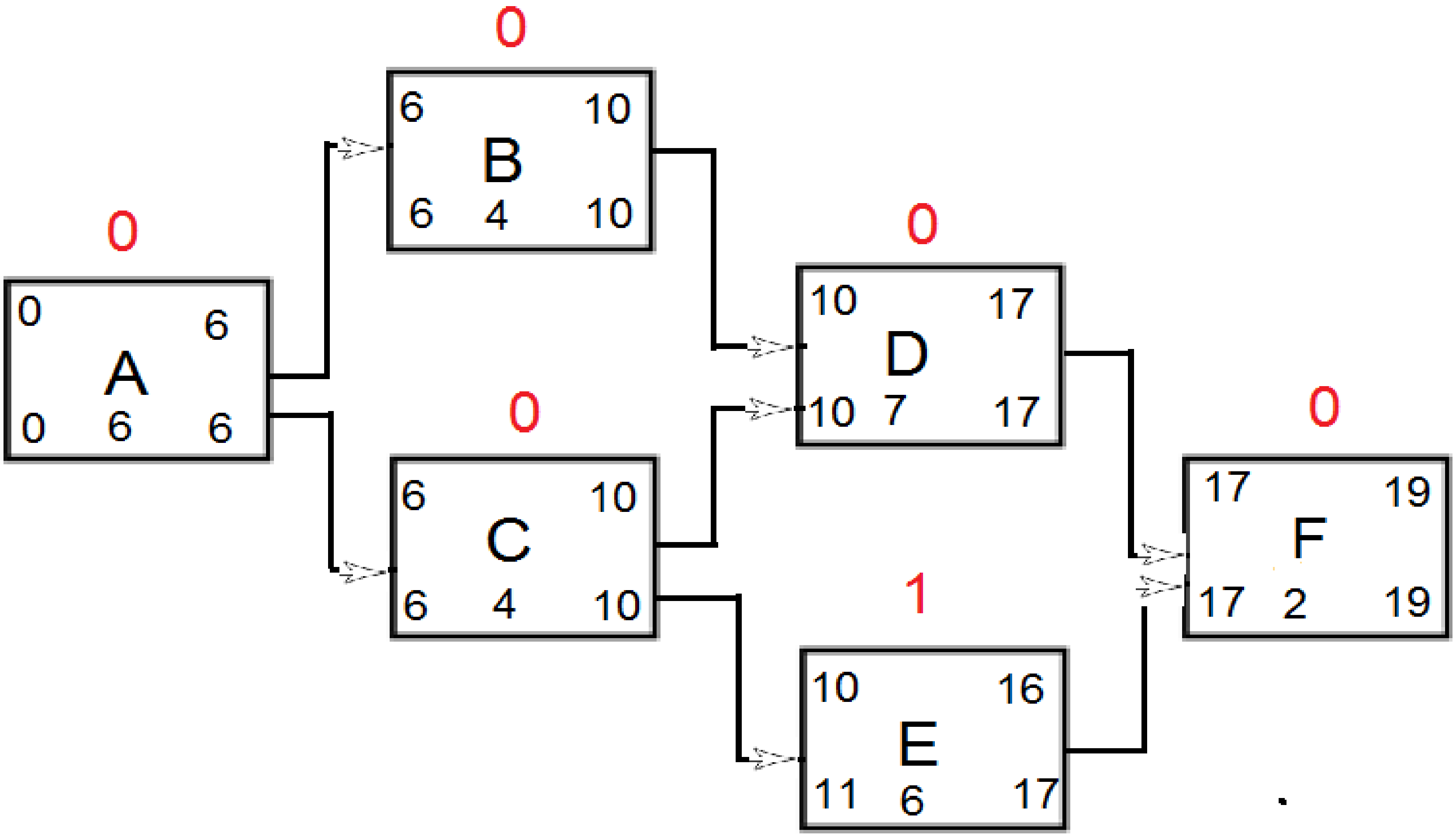
Calculating forward pass



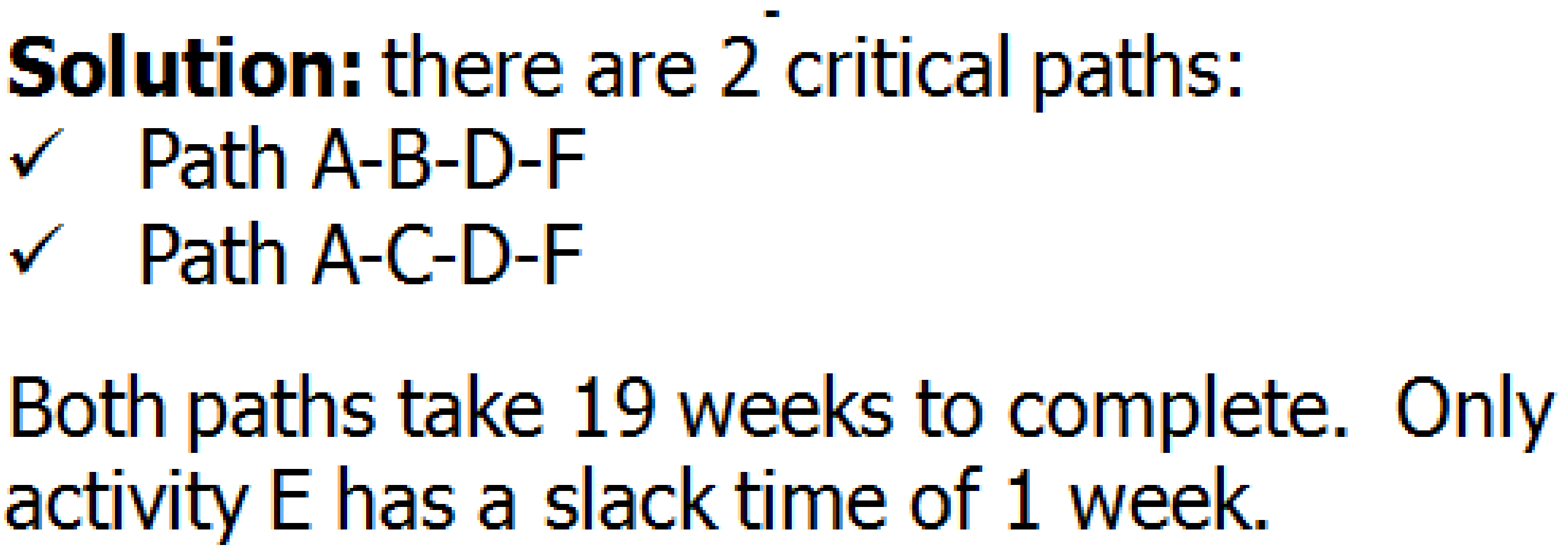
Calculating backward pass



Calculating the float



What’s the critical path?



Solution: there are 2 critical paths:

- ✓ Path A-B-D-F
- ✓ Path A-C-D-F

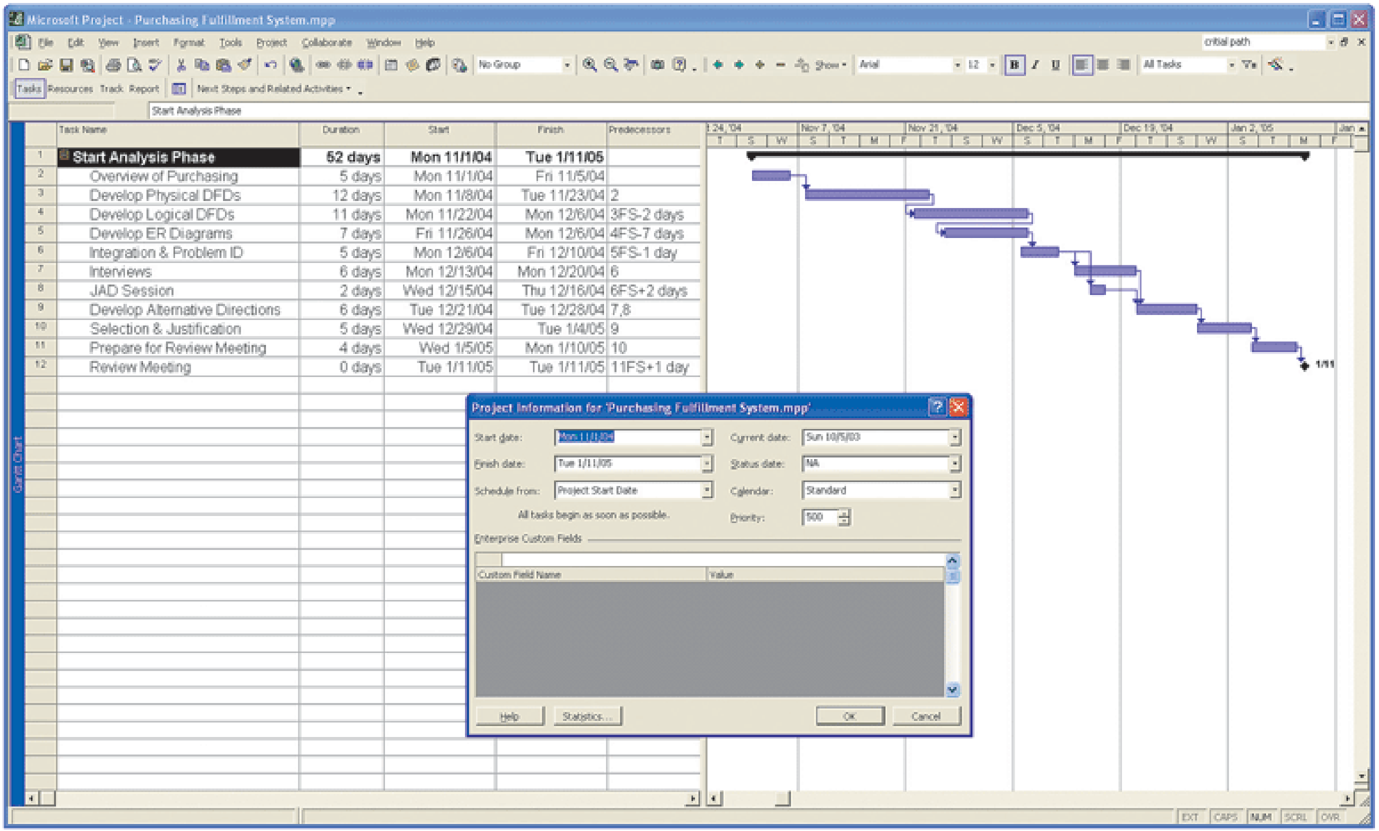
Both paths take 19 weeks to complete. Only activity E has a slack time of 1 week.

Using Project Management Software

- ✧ Many powerful software tools exist for assisting with project management.
- ✧ Example: Microsoft Project can help with
  - Entering project start date.
  - Establishing tasks and task dependencies.
  - Viewing project information as Gantt or Network diagrams.

Project Start Date

Figure 3-25 Establishing a project starting date in Microsoft Project for Windows



Microsoft Project - Purchasing Fulfillment System.mpp
File Edit View Insert Format Tools Project Collaborate Window Help
critial path
No Group
Arial
12
All Tasks
Tasks Resources Track Report
Next Steps and Related Activities
Start Analysis Phase
Gantt Chart
1/11
EXT CAPS NUM SCRL OVR

	Task Name	Duration	Start	Finish	Predecessors
1	**Start Analysis Phase**	**52 days**	**Mon 11/1/04**	**Tue 1/11/05**	
2	Overview of Purchasing	5 days	Mon 11/1/04	Fri 11/5/04	
3	Develop Physical DFDs	12 days	Mon 11/8/04	Tue 11/23/04	2
4	Develop Logical DFDs	11 days	Mon 11/22/04	Mon 12/6/04	3FS-2 days
5	Develop ER Diagrams	7 days	Fri 11/26/04	Mon 12/6/04	4FS-7 days
6	Integration & Problem ID	5 days	Mon 12/6/04	Fri 12/10/04	5FS-1 day
7	Interviews	6 days	Mon 12/13/04	Mon 12/20/04	6
8	JAD Session	2 days	Wed 12/15/04	Thu 12/16/04	6FS+2 days
9	Develop Alternative Directions	6 days	Tue 12/21/04	Tue 12/28/04	7,8
10	Selection & Justification	5 days	Wed 12/29/04	Tue 1/4/05	9
11	Prepare for Review Meeting	4 days	Wed 1/5/05	Mon 1/10/05	10
12	Review Meeting	0 days	Tue 1/11/05	Tue 1/11/05	11FS+1 day

Project Information for 'Purchasing Fulfillment System.mpp'

Start date: Mon 11/1/04

Finish date: Tue 1/11/05

Schedule from: Project Start Date

All tasks begin as soon as possible.

Current date: Sun 10/5/03

Status date: NA

Calendar: Standard

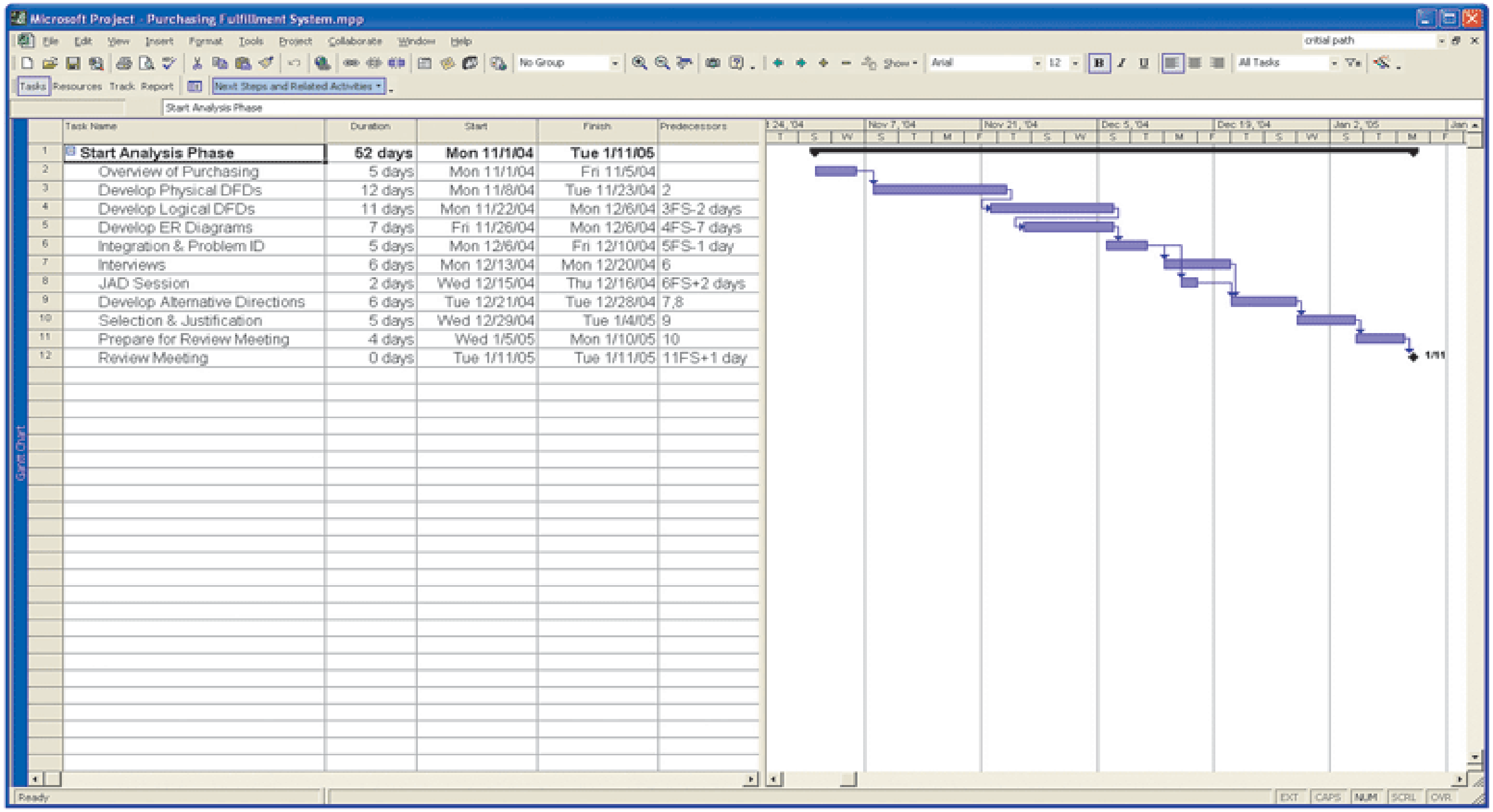
Priority: 500

Enterprise Custom Fields

Custom Field Name	Value

Entering Tasks

Figure 3-26 Entering tasks and assigning task relationships in Microsoft Project for Windows

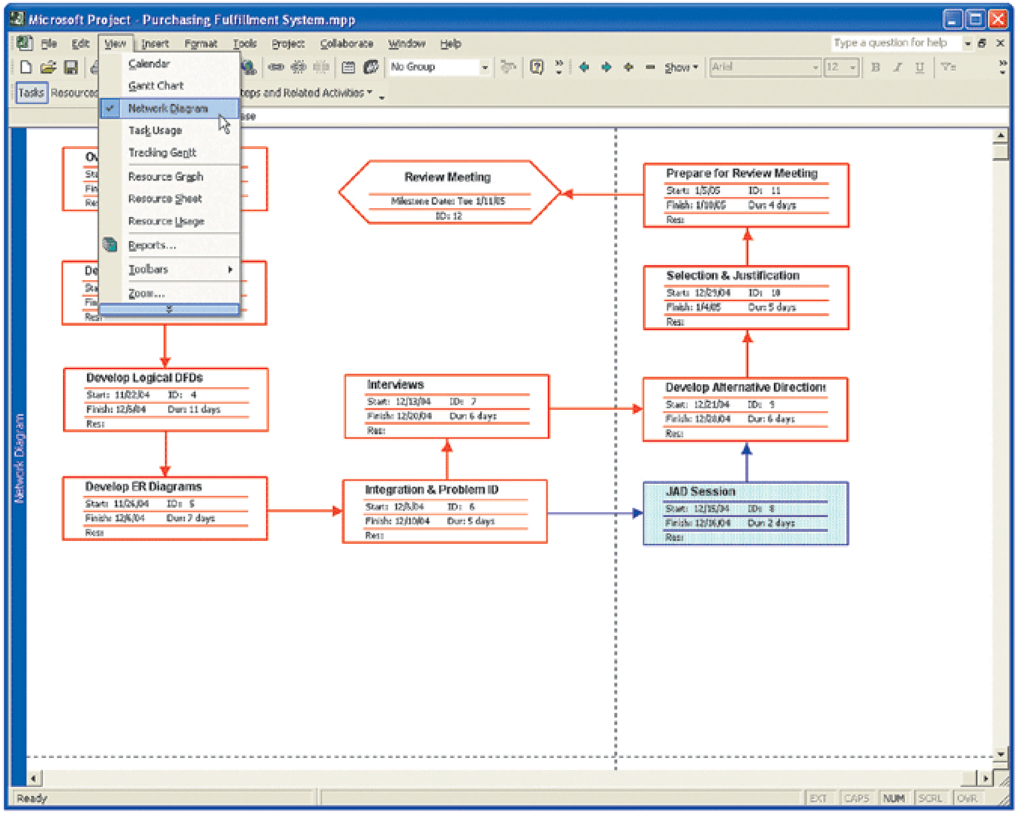


Start Analysis Phase

	Task Name	Duration	Start	Finish	Predecessors
1	**Start Analysis Phase**	**52 days**	**Mon 11/1/04**	**Tue 1/11/05**	
2	Overview of Purchasing	5 days	Mon 11/1/04	Fri 11/5/04	
3	Develop Physical DFDs	12 days	Mon 11/8/04	Tue 11/23/04	2
4	Develop Logical DFDs	11 days	Mon 11/22/04	Mon 12/6/04	3FS-2 days
5	Develop ER Diagrams	7 days	Fri 11/26/04	Mon 12/6/04	4FS-7 days
6	Integration & Problem ID	5 days	Mon 12/6/04	Fri 12/10/04	5FS-1 day
7	Interviews	6 days	Mon 12/13/04	Mon 12/20/04	6
8	JAD Session	2 days	Wed 12/15/04	Thu 12/16/04	6FS+2 days
9	Develop Alternative Directions	6 days	Tue 12/21/04	Tue 12/28/04	7,8
10	Selection & Justification	5 days	Wed 12/29/04	Tue 1/4/05	9
11	Prepare for Review Meeting	4 days	Wed 1/5/05	Mon 1/10/05	10
12	Review Meeting	0 days	Tue 1/11/05	Tue 1/11/05	11FS+1 day

Viewing Network Diagram

Figure 3-27 Viewing project information as a network diagram in Microsoft Project for Windows



Review Meeting
Milestone Date: Tue 1/11/05
ID: 12

Prepare for Review Meeting
Start: 1/5/05 ID: 11
Finish: 1/10/05 Dur: 4 days
Res:

Selection & Justification
Start: 12/29/04 ID: 10
Finish: 1/4/05 Dur: 5 days
Res:

Develop Logical DFDs
Start: 11/22/04 ID: 4
Finish: 12/6/04 Dur: 11 days
Res:

Interviews
Start: 12/13/04 ID: 7
Finish: 12/20/04 Dur: 6 days
Res:

Develop Alternative Directions
Start: 12/21/04 ID: 9
Finish: 12/28/04 Dur: 6 days
Res:

Develop ER Diagrams
Start: 11/26/04 ID: 5
Finish: 12/6/04 Dur: 7 days
Res:

Integration & Problem ID
Start: 12/6/04 ID: 6
Finish: 12/10/04 Dur: 5 days
Res:

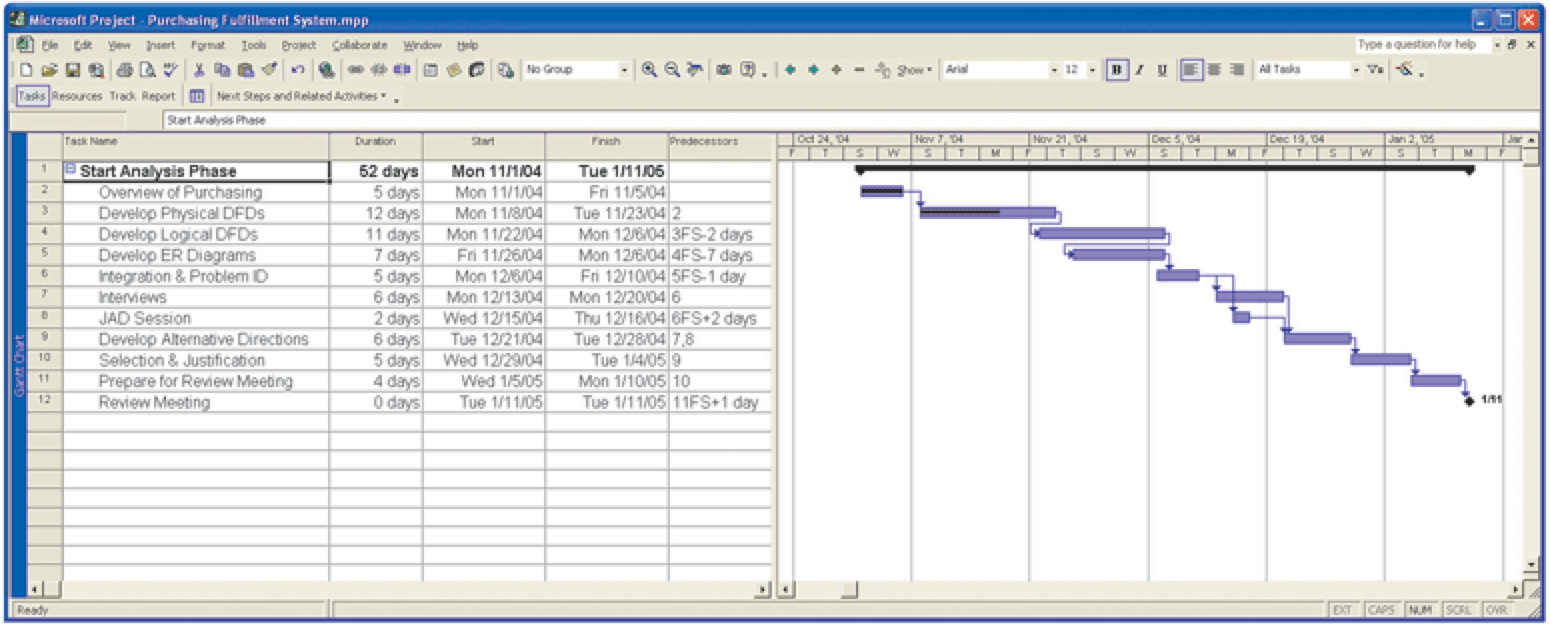
JAD Session
Start: 12/15/04 ID: 8
Finish: 12/16/04 Dur: 2 days
Res:

Hexagon shape indicates a milestone.

Red boxes and arrows indicate critical path (no slack).

Viewing Gantt Chart

Figure 3-28 Gantt chart showing progress of activities (right frame) versus planned activities (left frame)


Microsoft Project - Purchasing Fulfillment System.mpp
File Edit View Insert Format Tools Project Collaborate Window Help
Type a question for help
No Group
Show
Arial
12
B I U
All Tasks
Tasks Resources Track Report
Next Steps and Related Activities
Start Analysis Phase
Gantt Chart
Oct 24, '04
Nov 7, '04
Nov 21, '04
Dec 5, '04
Dec 19, '04
Jan 2, '05
Jar
F T S W S T M F T S W S T M F T S W S T M F
1/11
Ready
EXT CAPS NUM SCRL OVR

	Task Name	Duration	Start	Finish	Predecessors
1	**Start Analysis Phase**	**52 days**	**Mon 11/1/04**	**Tue 1/11/05**	
2	Overview of Purchasing	5 days	Mon 11/1/04	Fri 11/5/04	
3	Develop Physical DFDs	12 days	Mon 11/8/04	Tue 11/23/04	2
4	Develop Logical DFDs	11 days	Mon 11/22/04	Mon 12/6/04	3FS-2 days
5	Develop ER Diagrams	7 days	Fri 11/26/04	Mon 12/6/04	4FS-7 days
6	Integration & Problem ID	5 days	Mon 12/6/04	Fri 12/10/04	5FS-1 day
7	Interviews	6 days	Mon 12/13/04	Mon 12/20/04	6
8	JAD Session	2 days	Wed 12/15/04	Thu 12/16/04	6FS+2 days
9	Develop Alternative Directions	6 days	Tue 12/21/04	Tue 12/28/04	7,8
10	Selection & Justification	5 days	Wed 12/29/04	Tue 1/4/05	9
11	Prepare for Review Meeting	4 days	Wed 1/5/05	Mon 1/10/05	10
12	Review Meeting	0 days	Tue 1/11/05	Tue 1/11/05	11FS+1 day

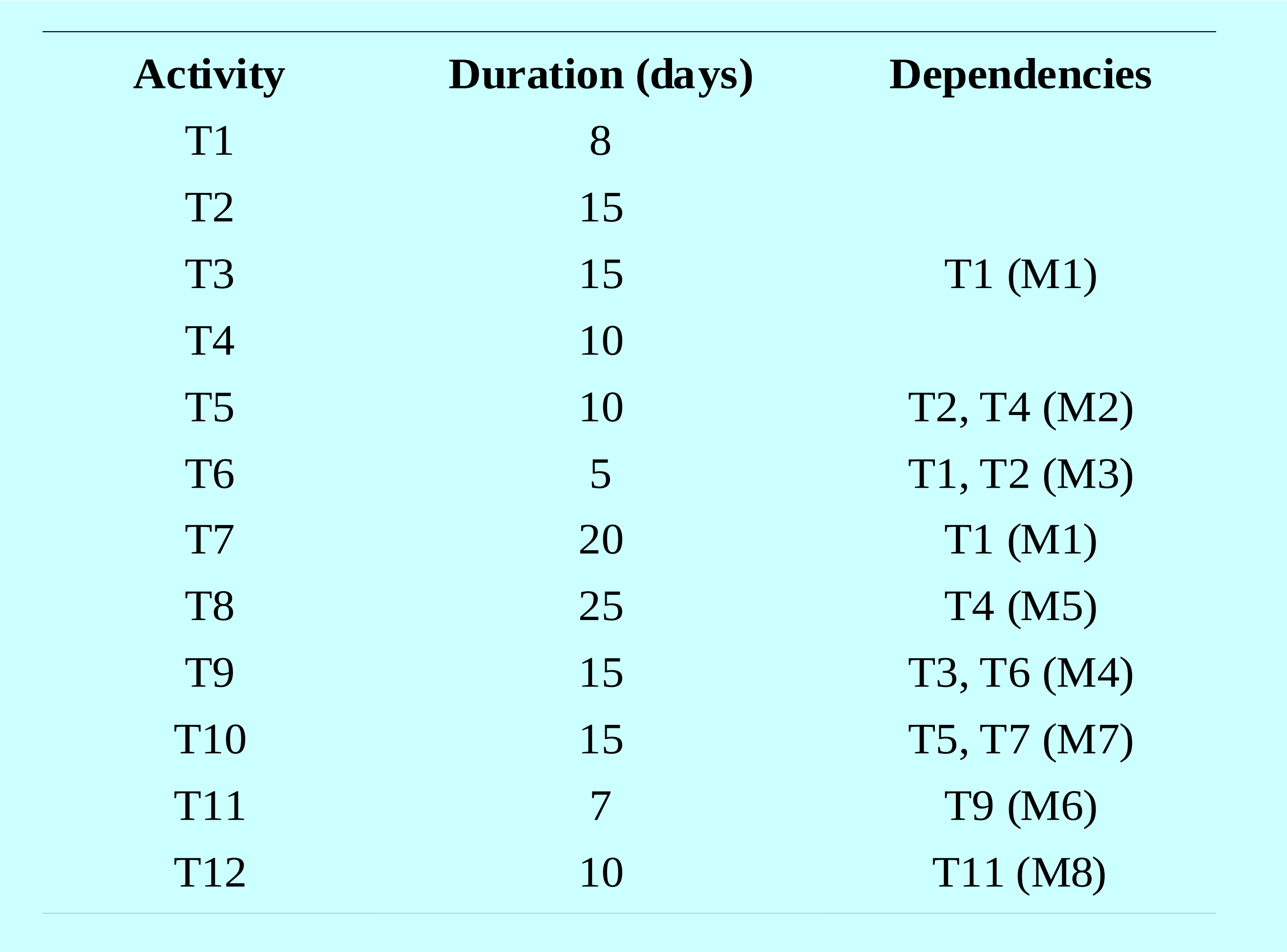
Black line at top indicates a summary activity (composed of subtasks).

Diamond shape indicates a milestone.

Milestones and deliverables

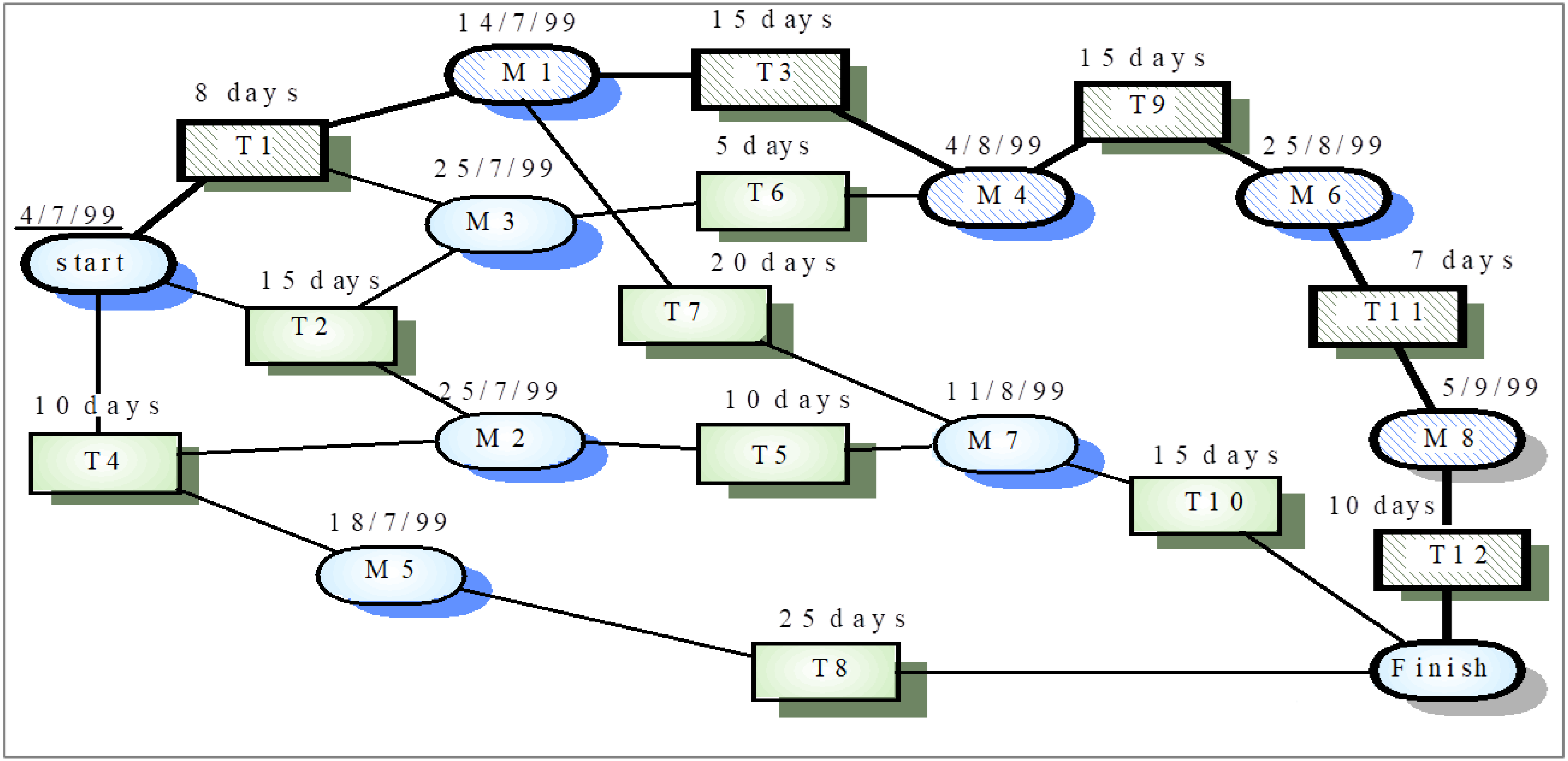
- **Milestones** are points in the schedule against which you can assess progress, for example, the handover of the system for testing.
- Deliverables are work products that are delivered to the customer, e.g. a requirements document for the system.
- Normally, you should use a project planning tool, such as the Basecamp or Microsoft project, to create, update, and analyze project schedule information.
- Project management tools usually expect you to input project information into a table, and they create a database of project information. Bar charts and activity charts can then be generated automatically from this database.

Task durations and dependencies

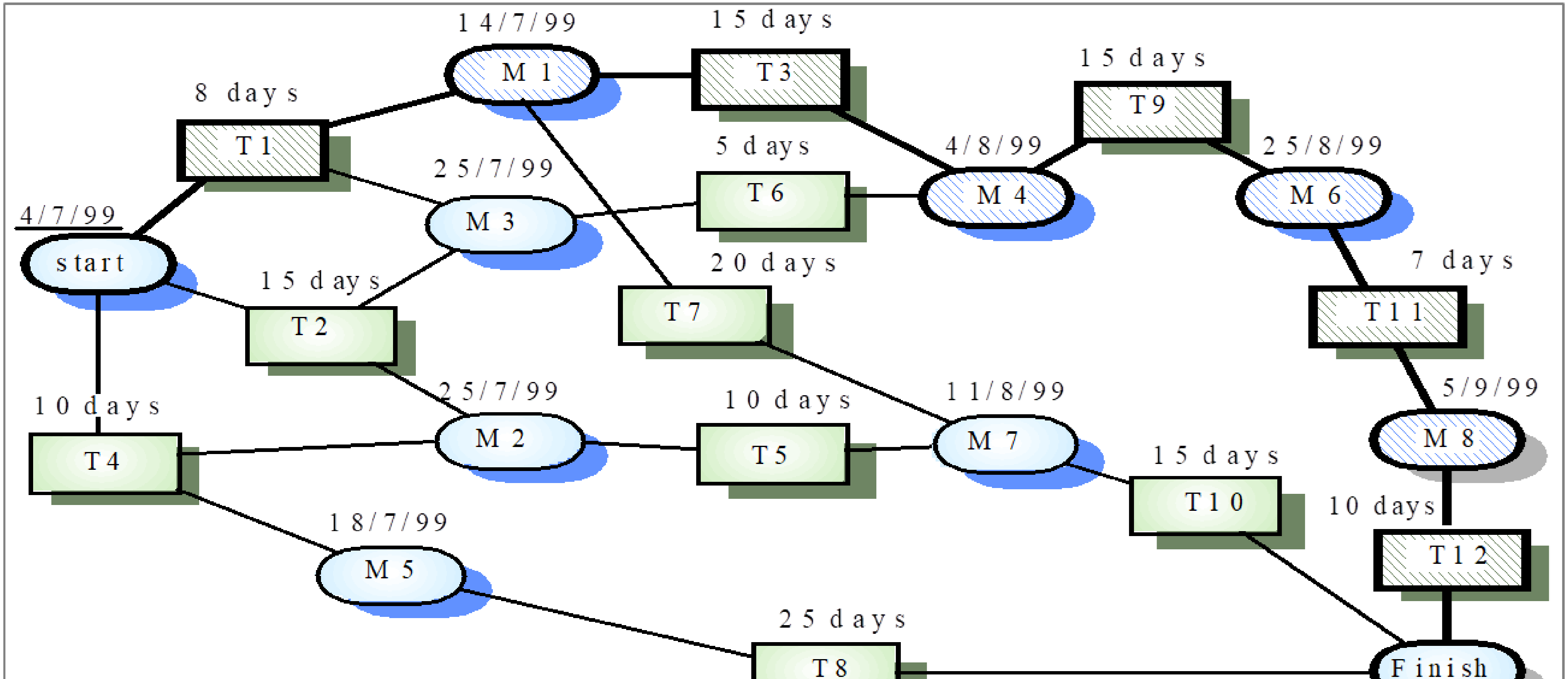


Activity	Duration (days)	Dependencies
T1	8	
T2	15	
T3	15	T1 (M1)
T4	10	
T5	10	T2, T4 (M2)
T6	5	T1, T2 (M3)
T7	20	T1 (M1)
T8	25	T4 (M5)
T9	15	T3, T6 (M4)
T10	15	T5, T7 (M7)
T11	7	T9 (M6)
T12	10	T11 (M8)

Activity Network



Activity Network



The critical path is 55 days

Attention

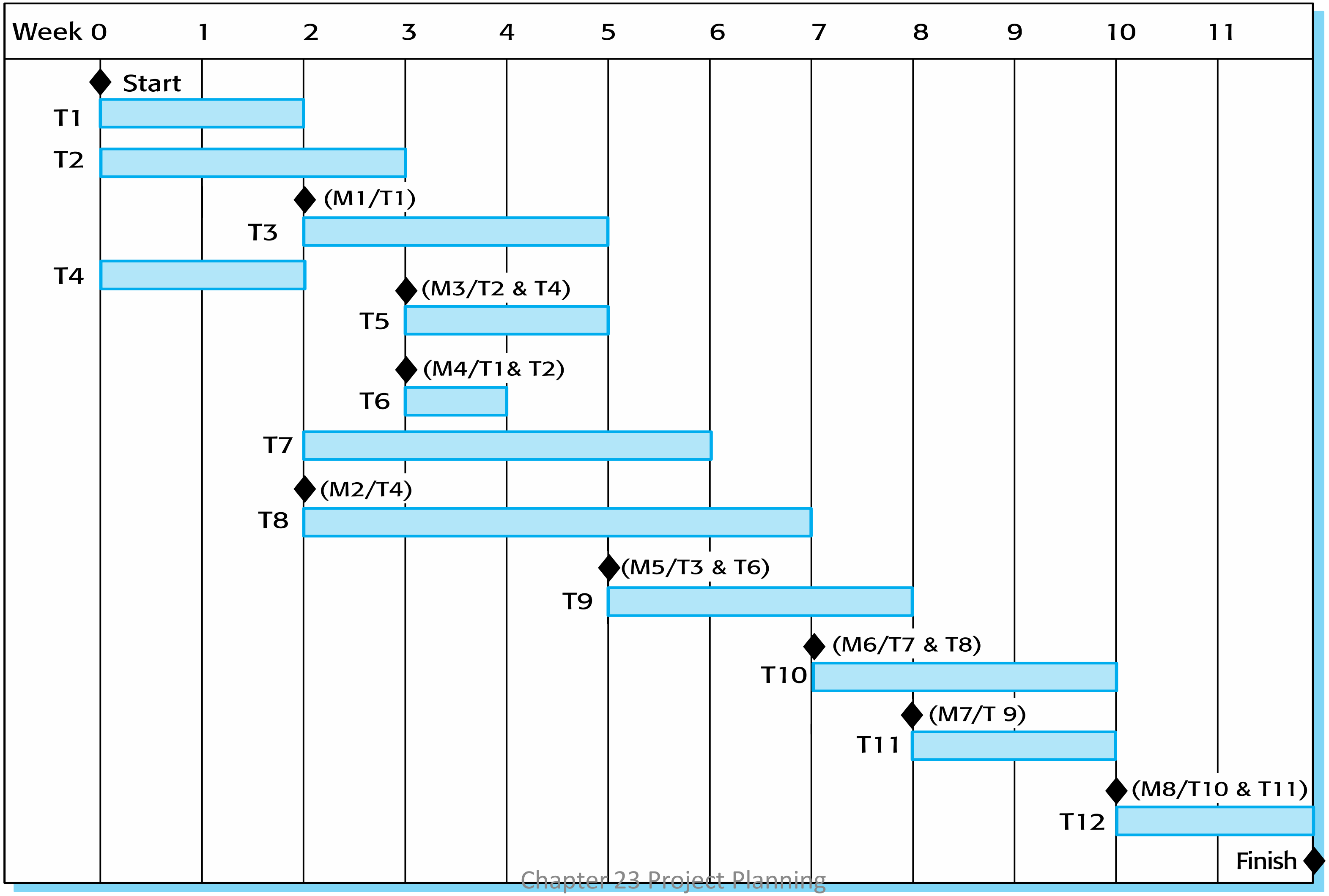
Q:What is the critical distinction between a milestone and a deliverable?

Task	Duration (days)	Dependencies
T1	15	
T2	20	T1
T3	14	T1, T2 (M1)
T4	15	T3(M2)
T5	25	T4(M3)

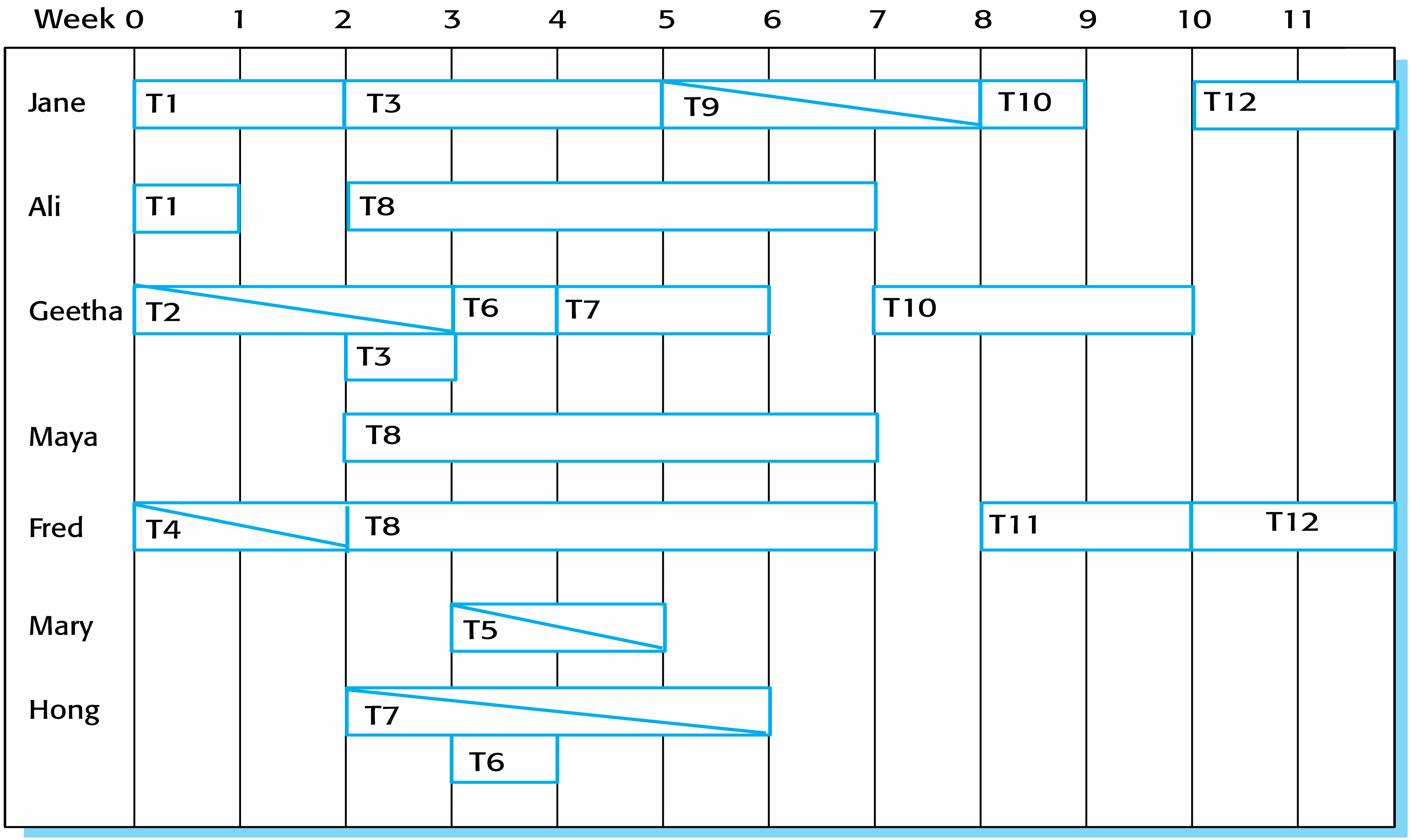
Tasks, durations, and dependencies

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)

Activity bar chart



Staff allocation chart



Week 0 1 2 3 4 5 6 7 8 9 10 11

Jane T1 T3 T9 T10 T12

Ali T1 T8

Geetha T2 T6 T7 T10
T3

Maya T8

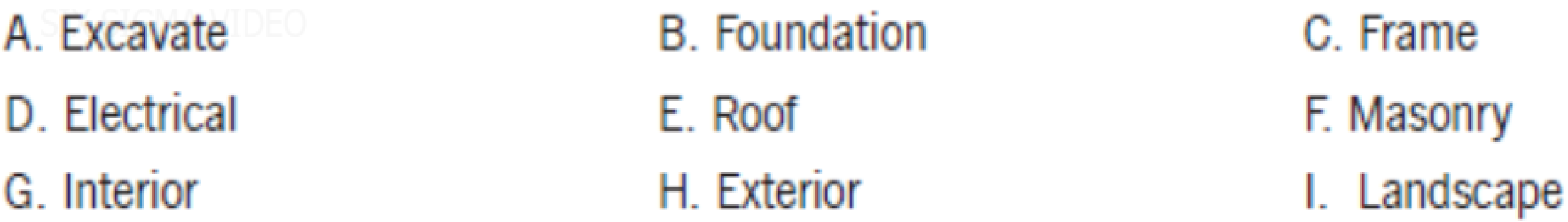
Fred T4 T8 T11 T12

Mary T5

Hong T7
T6

Example

Suppose the team is tasked with improving the process of building a house. The team lists the major steps: from the Excavate (digging) step through the landscaping step.



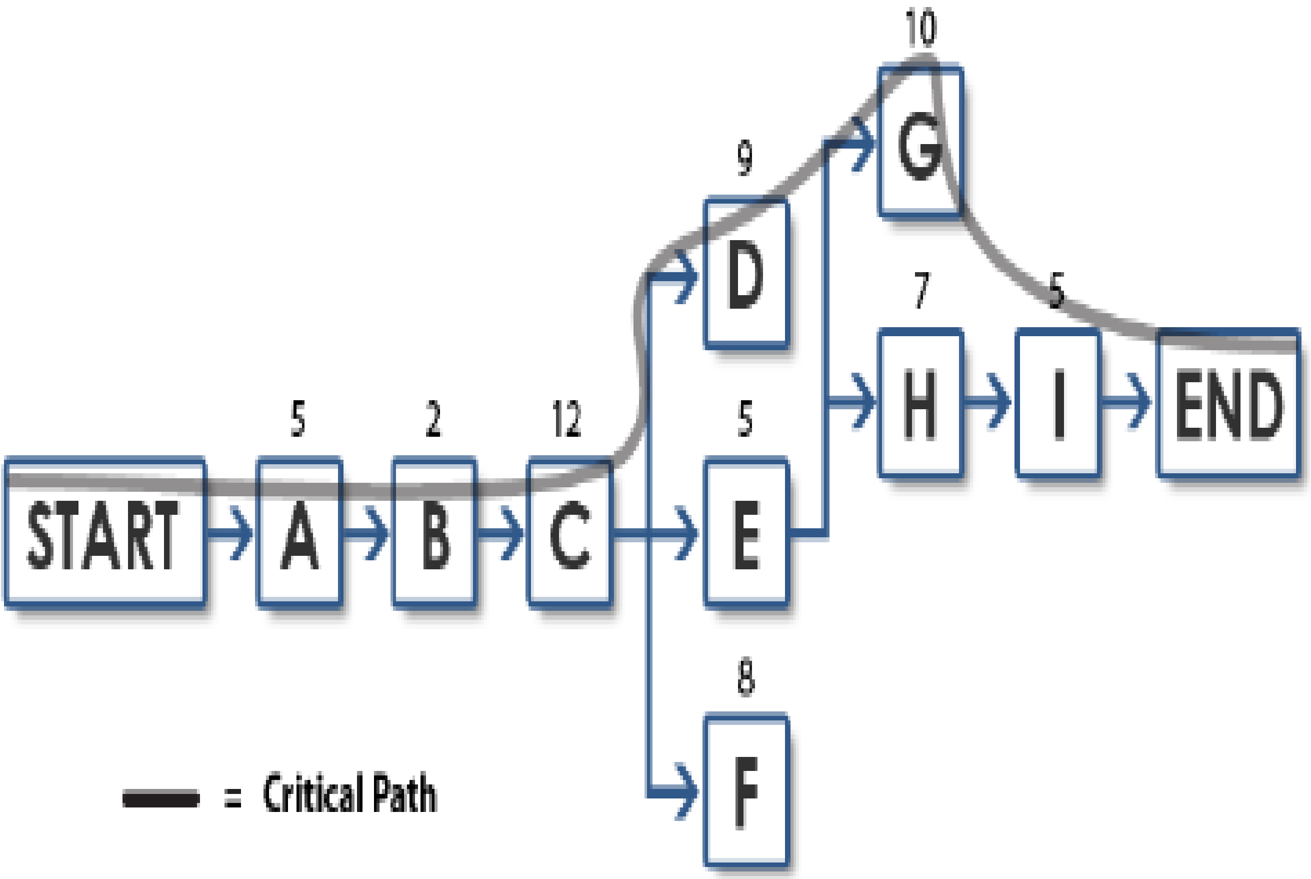
A. Excavate
B. Foundation
C. Frame
D. Electrical
E. Roof
F. Masonry
G. Interior
H. Exterior
I. Landscape

Example

- Some of the process steps (nodes A, B, and C) run in series, while other process steps (nodes D, E, and F) run in parallel.
- Notice that step B cannot happen until step A has been completed.
- Likewise, step C cannot happen until step B has completed.
- Step H cannot happen until steps D, E, and F have completed
- ALL need to be completed before Step H.
- This is important to know because those steps that are running in parallel most likely will have different expected completion times.

Example: Activity Network

A. Excavate	5 days
B. Foundation	2 days
C. Frame	12 days
D. Electrical	9 days
E. Roof	5 days
F. Masonry	8 days
G. Interior	10 days
H. Exterior	7 days
I. Landscape	5 days



Critical Path

- ✧ The critical path is a line that goes through all of the nodes that have the **longest expected completion times**.
- ✧ Nodes A, B, and C run in series, so the critical path is straightforward.
- ✧ Notice that between the three nodes that run in parallel (nodes D, E, and F), node D is expected to take the longest to complete as compared to the other two nodes. The critical path would run through nodes D and G because those particular nodes have the longest expected completion times.

Most Likely Time (Real Time)

By looking at the Activity Network Diagram, the team can easily see that the expected completion time as defined by the critical path is 50 days.

(5+2+12+9+10+7+5 = 50 days)

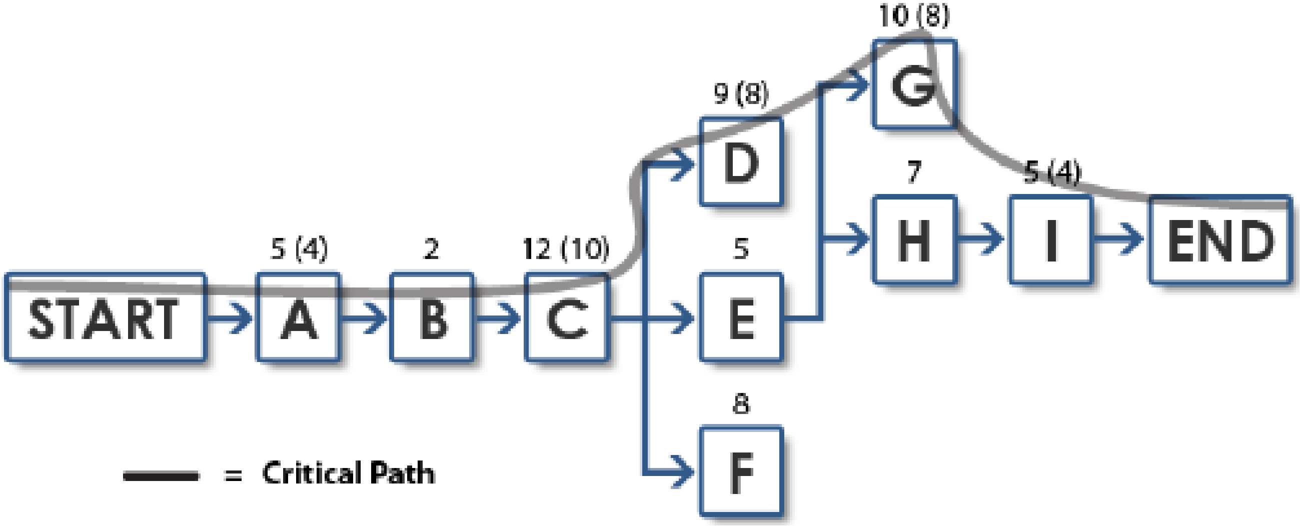
That’s the **MOST LIKELY** time.

Optimistic Time

The team might want to know what the best case (Optimistic Time), in terms of time, would be.

To determine the best case, the critical path line must be followed.

To come up with that number, the team would decide upon the shortest possible time for each of the nodes, and then add those up. The numbers in parenthesis are the most optimistic times (4+2+10+8+8+7+4 = 43)

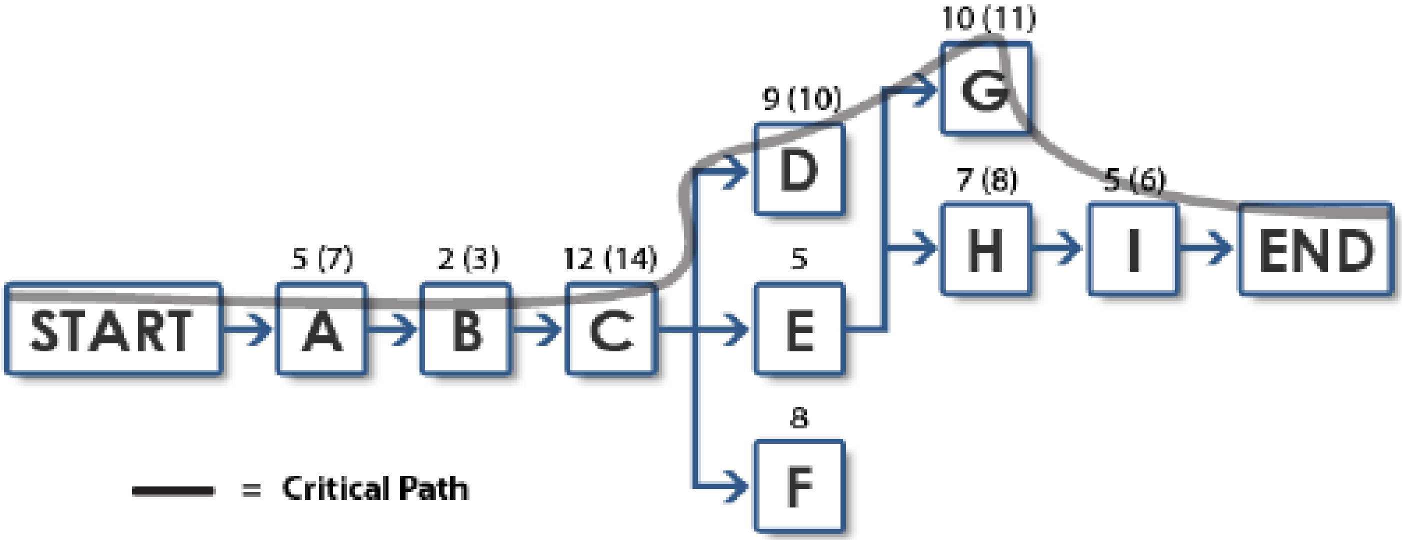


Pessimistic Time

The team also might want to know what the worst case (Pessimistic Time), in terms of time, would be.

To determine the worst case, the critical path line must be followed.

To come up with that number, the team would decide upon the longest possible time for each of the nodes, and then add those up. The numbers in parentheses are the most pessimistic times (7+3+14+10+11+8+6 = 59)



Expected Time

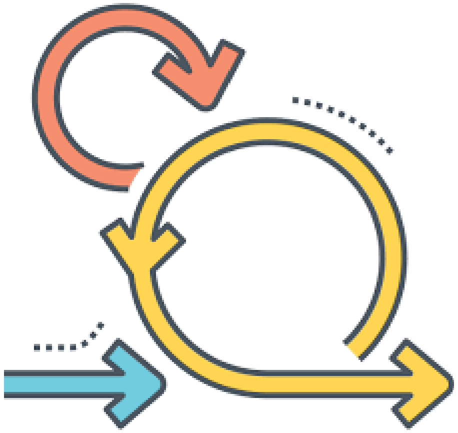
What does all of this mean?

It means the project most likely will take 50 days, but it could take 59 days, or it can be done as soon as 43 days.

$$\text{Expected Time} = \frac{\text{Optimistic} + [4\ (\text{Most Likely})] + \text{Pessimistic}}{6} =$$

$$\text{Expected Time} = \frac{43 + 200 + 59}{6} = 50.3 \text{ days}$$

Agile planning



Agile planning

- Agile methods of software development are iterative approaches where the software is developed and delivered to customers in increments.
- Unlike plan-driven approaches, the functionality of these increments is **not planned in advance but is decided during the development.**
- The decision on what to include in an increment depends on progress and on the customer's priorities.
- The customer's priorities and requirements change so it makes sense to **have a flexible plan that can accommodate these changes.**

Agile planning difficulties

- ✧ Agile planning is reliant on customer involvement and availability.
- ✧ This can be difficult to arrange, as customer representatives sometimes have to prioritize other work and are not available for the planning game.
- ✧ Furthermore, some customers may be more familiar with traditional project plans and may find it difficult to engage in an agile planning process.

Replace with Scientific Term

- ✧ Document that describes how the software and associated hardware will be installed in the customer's environment.
- ✧ Document used to communicate how the work will be done to the project team and customers, and to help assess progress on the project.
- ✧ Actions introduced to reduce the risks of project failure if there are serious problems with the development work that are likely to lead to significant delays.
- ✧ Points in the schedule against which you can assess progress, for example, the handover of the system for testing.
- ✧ Work products that are delivered to the customer, e.g. a requirements document for the system.

List

- Factors affecting software pricing.
- Reasons for overpricing and underpricing.
- Project plan elements.
- Project scheduling activities.
- Project scheduling problems.
- Agile planning stages.

Questions

- ✧ Project planning is conducted during the project proposal and startup only. Comment.
- ✧ What is the rationale behind development by following a plan?
- ✧ Elaborate on the need for plans supplementary to the project plan.
- ✧ You should include ---- in your plan so that if things go wrong, then your delivery schedule is not seriously disrupted. Complete.
- ✧ Give two reasons for inevitable change to plans.
- ✧ What is the course of actions required if there are serious problems with the development work that are likely to lead to significant delays?

STUDY SMART