Chapter 5:
Project Schedule Management

Information Technology Project Management, Sixth Edition

Note: See the text itself for full citations.

Project Time Management Processes

- **Plan Schedule Management:** establishing policies, procedures, and documentation for planning, developing, managing, executing and controlling the project schedule.
- **Define activities**: identifying and documenting the specific actions to be performed to produce the project deliverables.
- **Sequence activities**: identifying and documenting relationships among the project activities.
- **Estimating activity durations**: estimating the number of work periods needed to complete individual activities with the estimated resources.
- **Estimate Activity Resources:** The process of estimating the types, quantities and characteristics of material, human resources, equipment, or supplies required to perform each activity.
- **Develop the schedule**: analyzing activity sequences, durations, resource requirements, and schedule constraints to create the project schedule model for project execution and monitoring and controlling.
- **Control the schedule**: monitoring the status of the project to update the project schedule and manage changes to the schedule baseline.

6.2 Define Activities

- An **activity** or **task** is an element of work normally found on the work breakdown structure (WBS) that has an expected duration, a cost, and resource requirements

- Things to do:
  - Developing a more detailed WBS
  - 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 that includes:
  - The activity name
  - An activity identifier or number
  - A brief description of the activity
- **Activity attributes** provide more information such as:
  - who is responsible
  - Where: the place where the work has to be performed
  - When: time constraints,
  - Dependency: predecessors, successors
  - Resource and skill requirements
  - Assumptions and constraints related to the activity

Done in 6.3 Sequence activities

Example of Activity List

Activity	Description	duration	predecessor
A	Elicit user requirements	3	
B	Write requirement specification	14	A
C	Requirement validation	2	B
D	High level architectural design	5	C
E	Database design	3	D
F	Interface design	3	C
G	Module design	10	C
H.1	Coding (Module A, B, C) + Unit testing	20	E, F, G
H.2	Coding (Modules D,E) + Unit testing	14	E, F, G
H.3	Coding (Modules F) + Unit testing	8	H.1, H.2
I	Integration	5	H.3
J	Test case design	3	C
K	User Acceptance Test (UAT)	3	I, J
L	Installation & training	10	K

Done in 6.4 Estimate Activity durations

Example of Activity Attributes

E	Database design

Activity ID: 0032
WBS No: 3.2.5
Activity Description:
This activity involves the design of the database for the online shop mobile app.
Location:
All work associated with this activity will take place at the company.
Responsible by:
John Chan
Resources and Skill sets required:
This activity requires the sophisticated database design knowledge with MySQL experience; this workload requires 2 people.
Predecessors:
High level architectural design
Successors:
Coding (Module A, B, C) + Unit testing, Coding (Module D, E) + Unit testing
Assumption:
There will be 2 computers dedicated for this task and each of them will be installed with MySQL version xxx.xx and …
Constraints:
This activity must be finished with full verification by Nov. 5 at latest;

Milestones

- A **milestone** is a significant event that normally has no duration (it is just a point in time.)
- It often takes several activities and a lot of work to complete a milestone
- They're useful tools for setting schedule goals and monitoring progress
- Sometimes, it is required by contract, i.e. intermediate report, certain functions
- Examples include obtaining customer sign-off on key documents or completion of specific products

Milestones

Activity	Description	Milestone (event completed)
A	Elicit user requirements	
B	Write requirement specification	
C	Requirement validation	**Requirement specification**
D	High level architectural design	
E	Database design	
F	Interface design	
G	Module design	**Completion of design**
H.1	Coding (Module A, B, C) + Unit testing	**Intermediate delivery 1**
H.2	Coding (Modules D,E) + Unit testing	
H.3	Coding (Modules F) + Unit testing	
I	Integration	**Program release**
J	Test case design	
K	User Acceptance Test (UAT)	**UAT passed**
L	Installation & training	**Program delivery, O&M documentation**

6.3 Sequence Activities

- Involves reviewing activities and determining dependencies
- A **dependency** or **relationship** is the sequencing of project activities or tasks
- You *must* determine dependencies in order to use **Critical Path Analysis**

Three types of Dependencies

- **Mandatory dependencies (Internal dependencies)**
  - inherent in the nature of the work being performed on a project, sometimes referred to as hard logic.
  - **i.e. coding → testing**
- **Discretionary dependencies**
  - defined by the project team; sometimes referred to as soft logic and should be used with care since they may limit later scheduling options
- **External dependencies**
  - involve relationships between project and non-project activities
  - i.e. delivery of a computer (vendor) → coding

Example of dependencies

Considering 2 activities A and B.

- If B has a Mandatory Dependency on A then it means action on B cannot be performed until Action on A has been completed.
  - A – Coding; B – Testing
  - A – Build car prototype; B – Perform crash testing
- If B has a Discretionary Dependency on A, then it means action on B can be performed even before Action on A has been completed for some reasons (better use of resources).
  - A – Develop module A,B,C ; B – Develop module D,E
  - A – Book airline ticket; B – Buy insurance

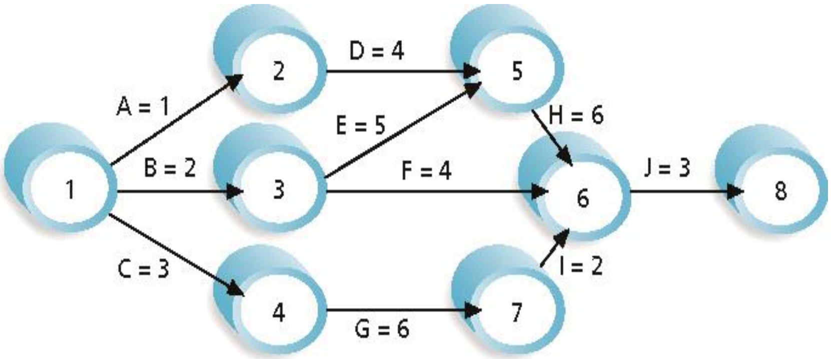
Example of dependencies (cont.)

- B has an External Dependency on A. It means B is a project activity while A is a non-project activity.
  - A – Delivery of computer equipment;
    B – Coding
  - A – Delivery of raw materials;
    B – Build the product
  - A – Licensed by government;
    B – Sell the slot machines to casinos

Network Diagrams

- Network diagrams are the preferred technique for showing activity sequencing
- A **network diagram** is a schematic display of the logical relationships among, or sequencing of, project activities
- *Two main formats*:
  - **Activity-on-Arrow diagramming method**
  - **Precedence Diagramming Method (PDM)**

Figure 6-2. 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.

Arrow Diagramming Method (ADM)

- Also called **activity-on-arrow (AOA)** network diagrams
- Activities are represented by arrows
- Nodes or circles are the starting and ending points of activities
- Can only show *finish-to-start* dependencies

Precedence Diagramming Method (PDM)

- Activities are represented by boxes
- Arrows show relationships between activities
- More popular than ADM method and used by project management software
- Better at showing different types of dependencies

PDM

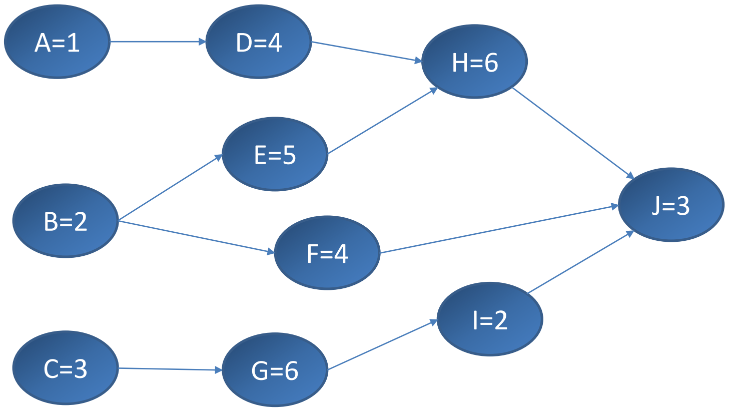
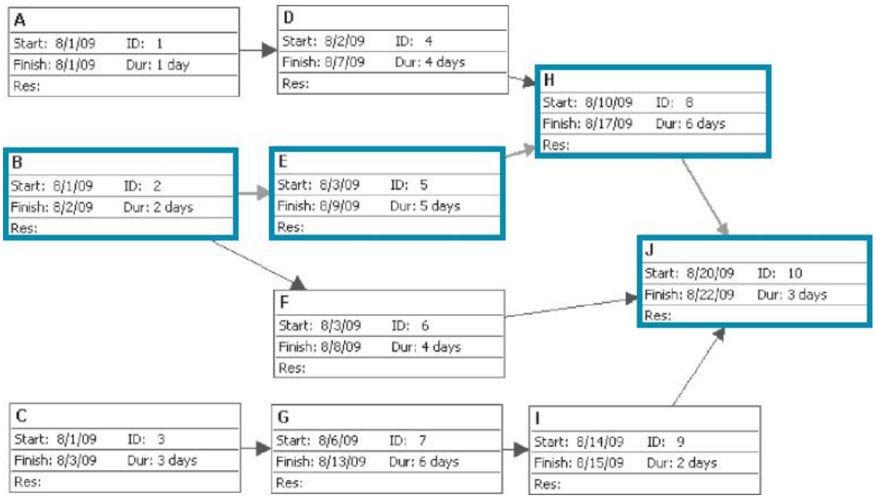


Figure 6-4. Sample PDM Network Diagram



A
Start: 8/1/09 ID: 1
Finish: 8/1/09 Dur: 1 day
Res:

D
Start: 8/2/09 ID: 4
Finish: 8/7/09 Dur: 4 days
Res:

H
Start: 8/10/09 ID: 8
Finish: 8/17/09 Dur: 6 days
Res:

B
Start: 8/1/09 ID: 2
Finish: 8/2/09 Dur: 2 days
Res:

E
Start: 8/3/09 ID: 5
Finish: 8/9/09 Dur: 5 days
Res:

J
Start: 8/20/09 ID: 10
Finish: 8/22/09 Dur: 3 days
Res:

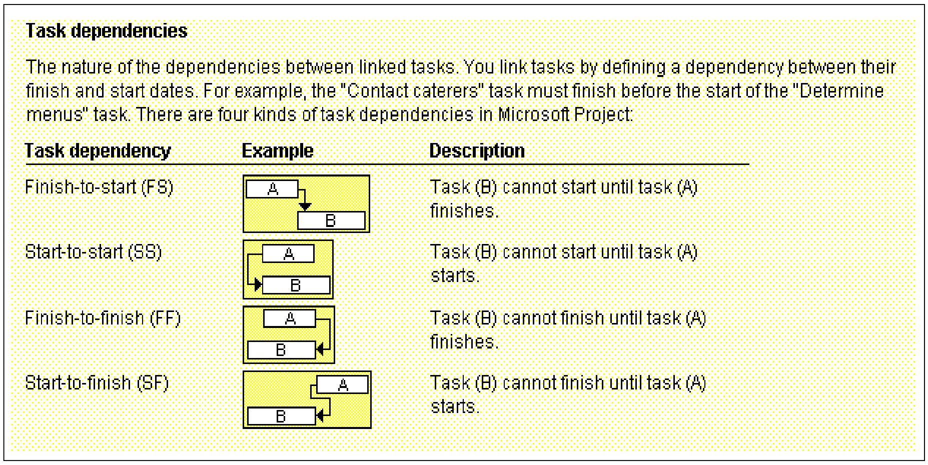
F
Start: 8/3/09 ID: 6
Finish: 8/8/09 Dur: 4 days
Res:

C
Start: 8/1/09 ID: 3
Finish: 8/3/09 Dur: 3 days
Res:

G
Start: 8/6/09 ID: 7
Finish: 8/13/09 Dur: 6 days
Res:

I
Start: 8/14/09 ID: 9
Finish: 8/15/09 Dur: 2 days
Res:

Task Dependency Types

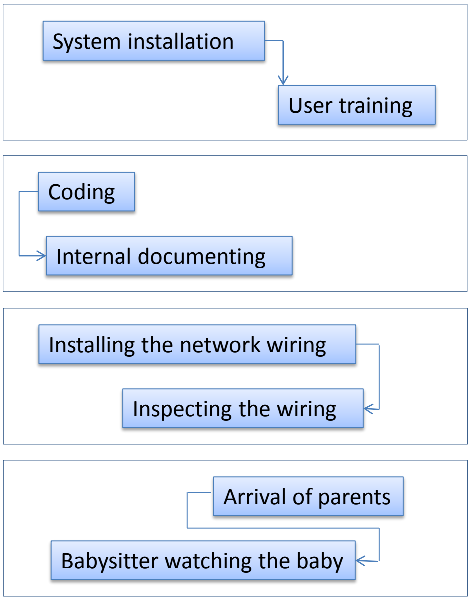


Task dependencies

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.

Examples of dependency types



FS

System installation

User training

Most common

SS

Coding

Internal documenting

FF

Installing the network wiring

Inspecting the wiring

SF

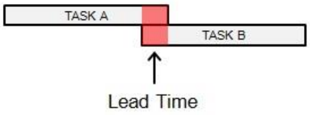
Arrival of parents

Babysitter watching the baby

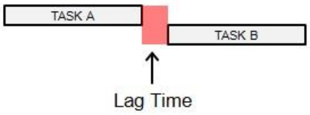
seldom used

Lead and Lag

- Lead
  - The period of time that a task starts before the predecessor finishes.
  - Ex.
    - FS-5D (starts 5 days before the predecessor finishes)
    - The coding starts 5 days before the module design is completed

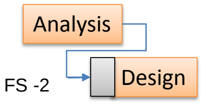


- Lag
  - The period of time that a task starts after a predecessor finishes.
  - Ex.
    - FS+3D (starts 3 days after the predecessor finishes)
    - The UAT test will start 3 days after the completion of the test plan design

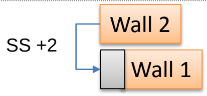


Examples

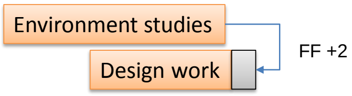
Design can start 2 days before completion of analysis.



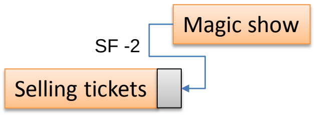
Wall 1 construction can start 2 days after wall 2 construction starts.



Design work cannot finish until 2 days after environmental studies are complete



Magic show tickets finishes 2 hours before start of show

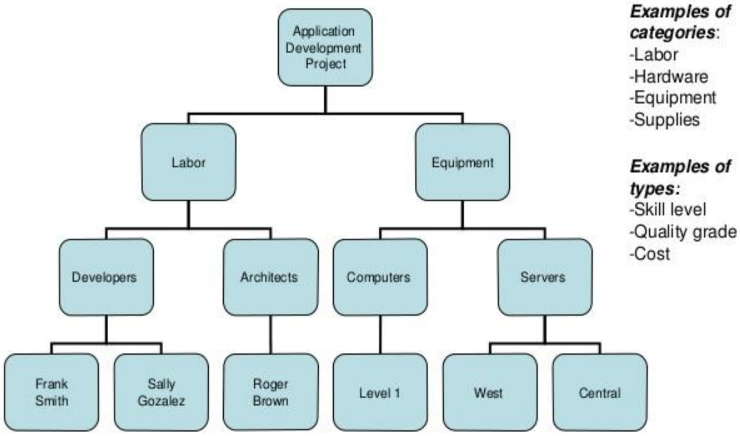


We must wait 3 days for the cement to be congealed before we cover it with carpet

6.4 Estimate Activity Resources

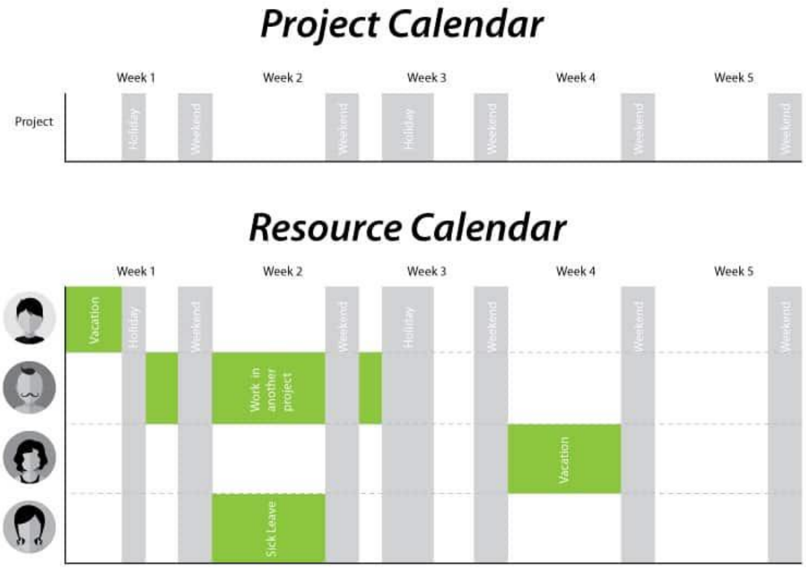
- The process of estimating the types, quantities and characteristics of material, human resources, equipment, or supplies required to perform each activity.

Resource Breakdown Structure



Examples of categories:
-Labor
-Hardware
-Equipment
-Supplies

Examples of types:
-Skill level
-Quality grade
-Cost



6.5 Estimate Activity Duration

- **Duration** includes the actual amount of time worked on an activity *plus* elapsed time
- **Effort** is the number of workdays or work hours required to complete a task
- **Effort** does not normally equal to **Duration**
- People doing the work should help create estimates, and an expert should review them

Duration Estimations

- Tools and Techniques used in “Estimate Activity Durations”
  - **Analogous Estimating**
    - based on historical data from a similar activity
    - Easier, but less accurate
  - **Parametric Estimating**
    - Based on statistical relationship between historical data and other variables (e.g., square footage in construction, meter/hour)
    - e.g. if the assigned resource is capable of installing 25 meters of cable/hour, the duration required to install 1,000 meters is 40 hours.
  - **Three-point Estimating**
    - **PERT** (Program Evaluation & Review Technique) uses three-point estimates to define an approximate range for an activity's duration: **Most likely**, **Optimistic** and **Pessimistic**
- These techniques can also be applied to cost estimations.

Program Evaluation and Review Technique (PERT)

- **PERT** is an 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 **=**

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

- Example:

PERT weighted average =

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

where optimistic time = 8 days
most likely time = **10 days**, and
pessimistic time = 24 days

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

Activity Duration Estimates

- Quantitative assessments of the likely number of time periods, that are required to complete an activity.

6.6 Develop Schedule

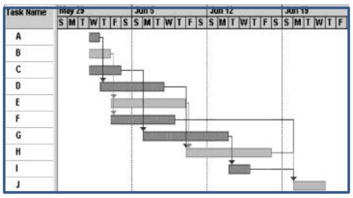
- Uses results of the other time management processes to determine the start and end date of the project
- Ultimate goal is to create a realistic project schedule that provides a basis for monitoring project progress for the time dimension of the project
- Important tools and techniques include Gantt charts, critical path analysis, and critical chain scheduling, and PERT analysis

Outputs of "Develop Schedule"

- **Project Schedule** – an output of a schedule model that presents linked activities with planned dates, durations, milestones and resources. (GANTT Chart)
- **Schedule Baseline** – the final approved version of the Project Schedule.
- **Project Calendars** – identifies working days and shifts that are available for scheduled activities. (with the consideration of holidays, shift time of workers etc.)

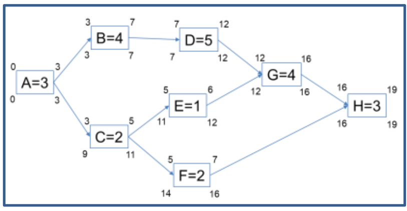
Schedule Baseline

Project Schedule



Gantt Chart with allocated resources

Milestone list

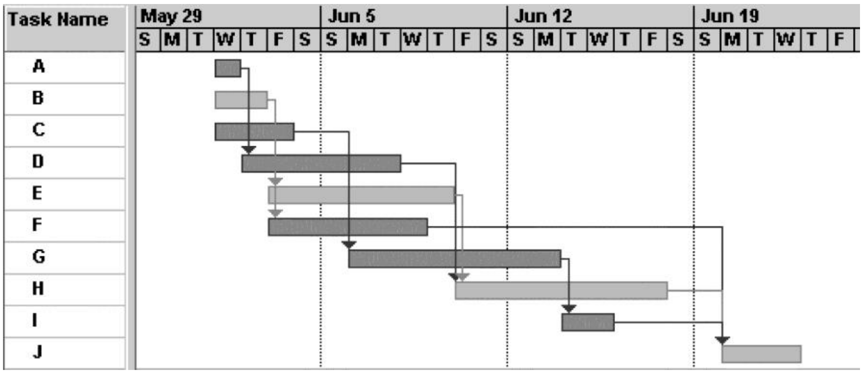


Schedule network diagram

Gantt Charts

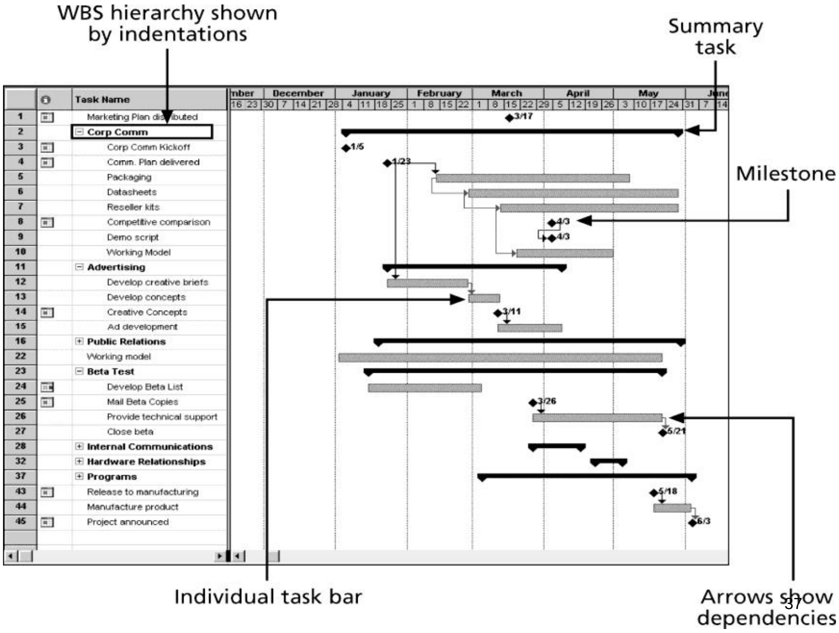
- **Gantt charts** provide a standard format for displaying project schedule information by listing project activities and their corresponding start and finish dates in a calendar format
- Symbols include:
  - Black diamonds: milestones
  - Thick black bars: summary tasks
  - Lighter horizontal bars: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

Figure 6-5. Gantt Chart for Project X



Note: Darker bars would be red in Project 2007 to represent critical tasks.

Figure 6-6. Gantt Chart for Software Launch Project


WBS hierarchy shown by indentations

Summary task

Milestone

Individual task bar

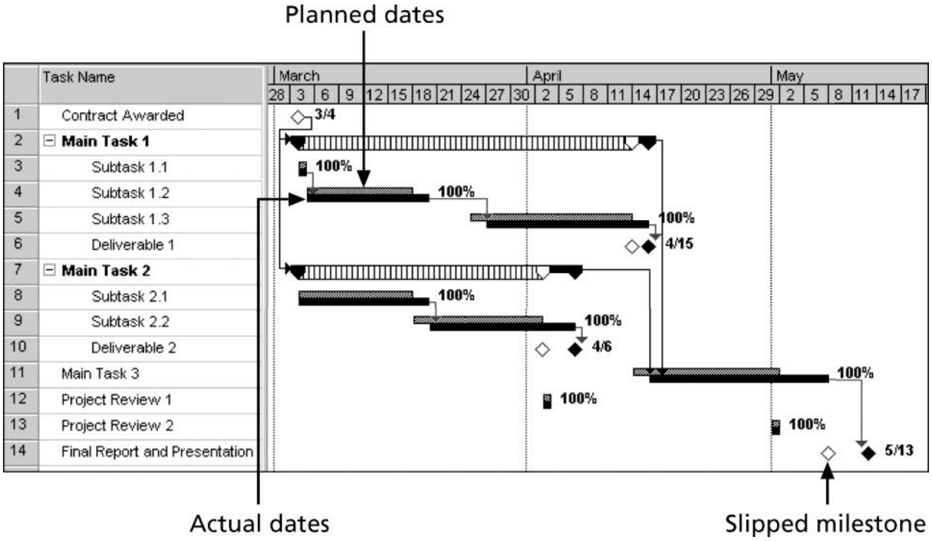
Arrows show dependencies

	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

Adding Milestones to Gantt Charts

- Many people like to focus on meeting milestones, especially for large projects
- Milestones emphasize important events or accomplishments on projects
- Normally create milestone by entering tasks with a zero duration, or you can mark any task as a milestone

Figure 6-7. Sample Tracking Gantt Chart



Planned dates

Actual dates

Slipped milestone

Critical Path Method (CPM)

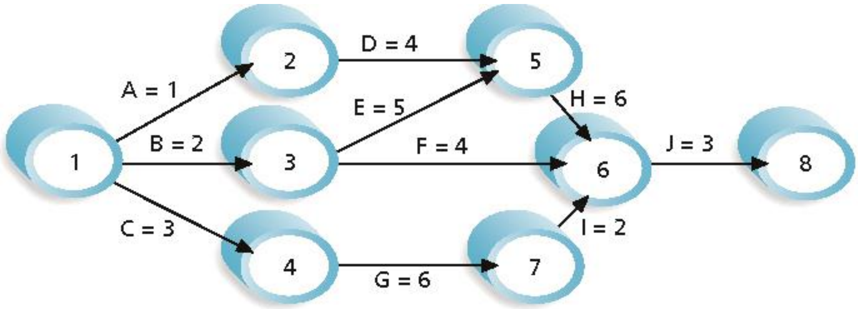
- **CPM** is a network diagramming technique used to predict total project duration

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

Calculating the Critical Path

- First develop a network diagram
- Add the duration estimates for all activities on each path through the network diagram
- The longest path is the critical path
- If one or more of the activities on the critical path takes longer than planned, the whole project schedule will slip *unless* the project manager takes corrective action

Figure 6-8. 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.

More on the Critical Path

- The critical path is *not* the one with all the critical activities; it only accounts for time
- There can be more than one critical path if the lengths of two or more paths are the same.
- The critical path can change as the project progresses

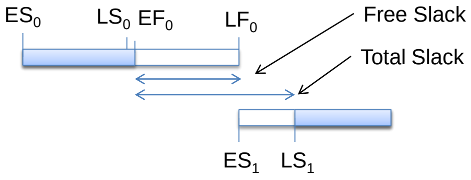
Using Critical Path Analysis to Make Schedule Trade-offs

- **Slack** or **float** is the amount of time an activity may be delayed without delaying a succeeding activity or the project finish date.
  - **Free slack** or **free float** is the amount of time an activity can be delayed without delaying the early start of any immediately following activities (successor).
  - **Total slack** or **total float** is the amount of time an activity may be delayed from its early start without delaying the planned project finish date.

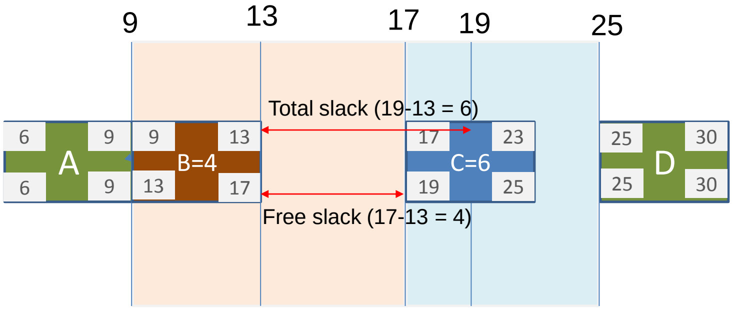
Calculation of Free Slack and Total Slack

- Free Slack
  - (Min) Free Slack = $ES_1 - ES_0 - Duration_0$ $= ES_1 - EF_0$
- Total Slack
  - (Min) Total Slack = $LS_1 - ES_0 - Duration_0$ $= LS_1 - EF_0$

0 – current
1 – successor

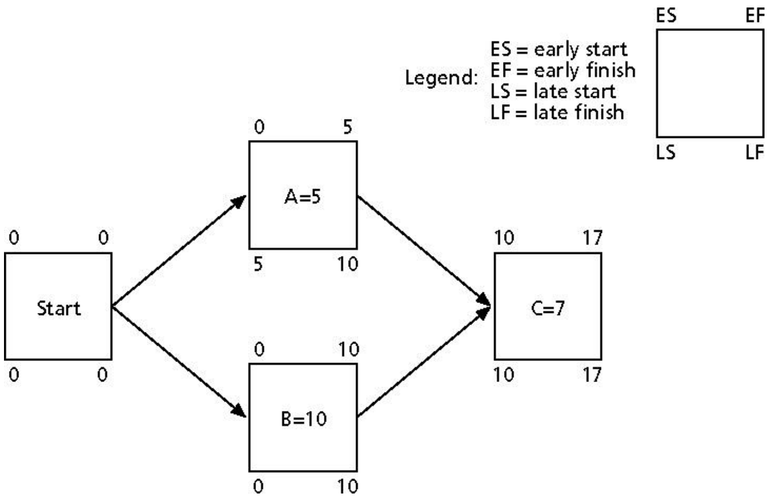


Free Slack and Total Slack



With Free slack, B has 8 days to be accomplished.
With Total slack, B has 10 days to be accomplished.

Calculating Early and Late Start and Finish Dates



Determining the ES,EF,LS,LF

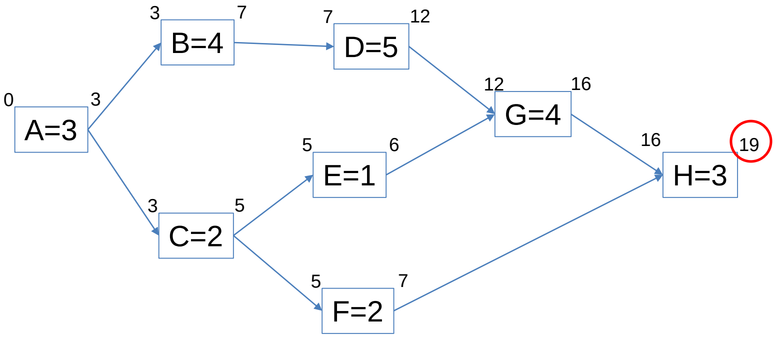
- A forward pass through the network diagram determines the **early start (ES)** and **early finish (EF)** dates

- A backward pass determines the **late start (LS)** and **late finish (LF)** dates

Demo of forward pass and backward pass

Activity	Predecessor	Duration
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

Forward pass – finding the ES, EF

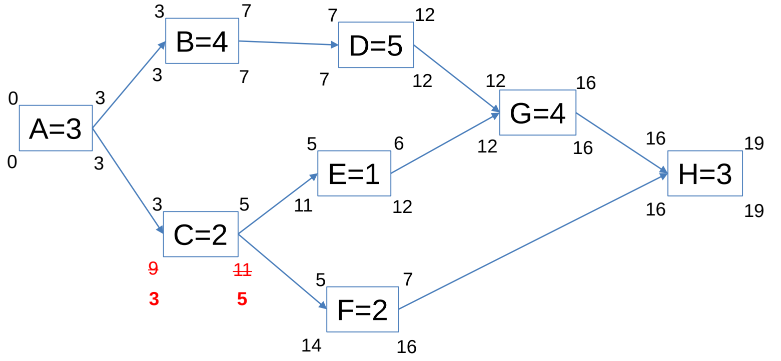


Without considering Leads and Lags.

It takes 19 days to finish the project.

√

Backward pass – finding the LS, LF



When you find that your schedule is lagging behind or your boss push you to deliver your product much earlier.

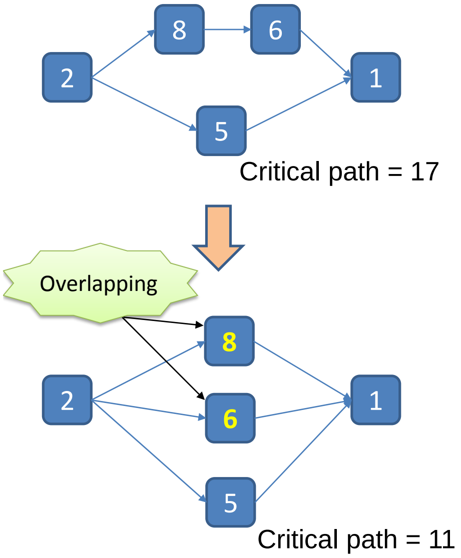
What should you do?

Shortening (Compressing) a Project Schedule

- Two main techniques for shortening schedules
  - **Fast tracking** activities by doing them in parallel or overlapping them (i.e. adding lead time)
  - **Crashing** activities by obtaining the greatest amount of schedule compression for the least incremental cost

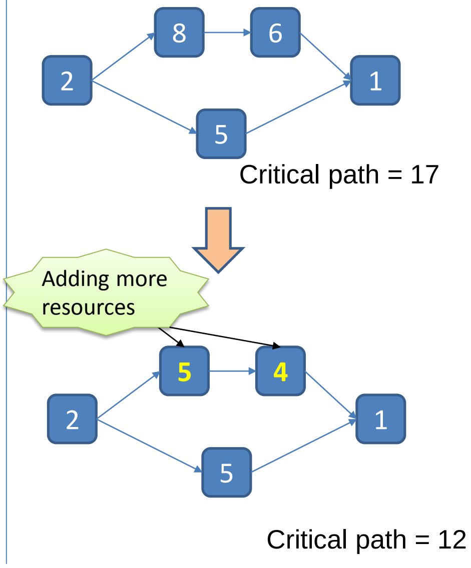
√

Fast Tracking



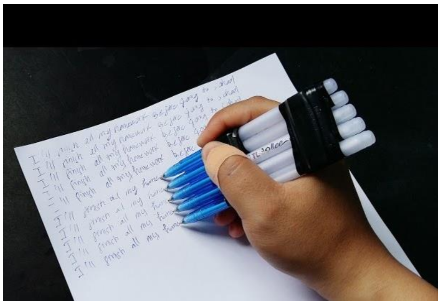
Risk&Quality-Schedule trade-offs

Crashing



Cost-Schedule trade-offs

Fast Tracking vs Crashing



Better coordination required and Higher risk

Better coordination required and Higher cost

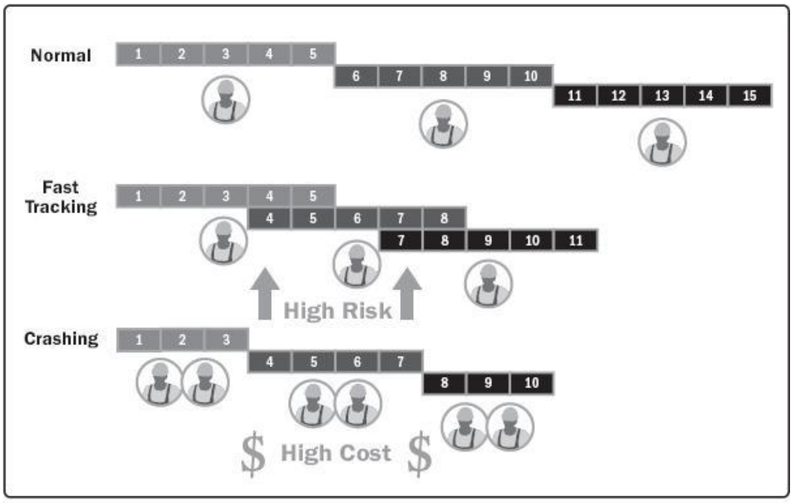


Figure 6-19. Schedule Compression Comparison

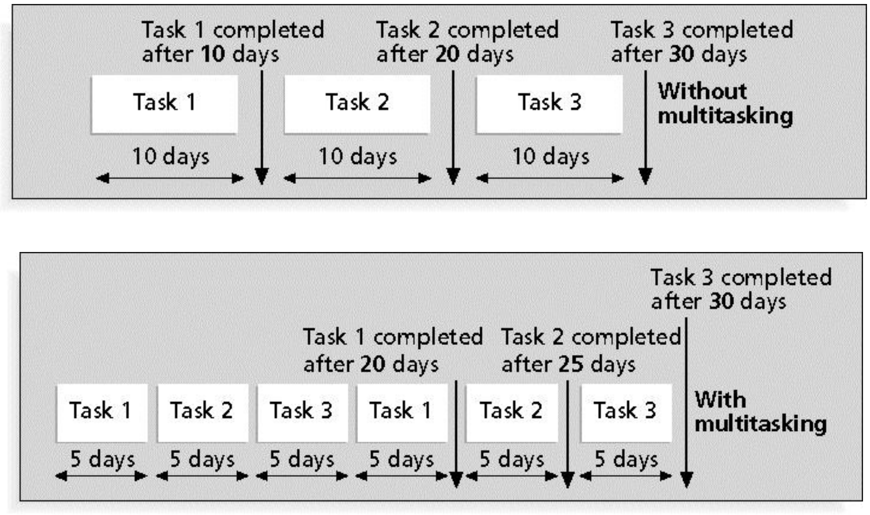
√

Phenomena pointed out by Goldratt

- **Parkinson Law** states that work expands to fill the time allowed
- **Self-Protection** – there is no incentive for early submission; but could be "punished" to shorten the time for next similar assigned tasks.
- **Student Syndrome** – student will procrastinate until the last moment to start doing the assigned tasks.
- **Gold-plating** – When tasks are done early, they may spend some time to beautify the works, which is unnecessary.
- **Multitasking** – minimizing multitasking since it can slow down the completion. We should concentrate one task at a time.

The management philosophy developed by Eliyahu M. Goldratt and introduced in his book "The Goal".

Figures 6-10a and 6-10b. Multitasking Example



Task 1 completed after **10** days

Task 2 completed after **20** days

Task 3 completed after **30** days

Task 1 | Task 2 | Task 3

10 days | 10 days | 10 days

Without multitasking

Task 3 completed after **30** days

Task 1 completed after **20** days

Task 2 completed after **25** days

Task 1 | Task 2 | Task 3 | Task 1 | Task 2 | Task 3

5 days | 5 days | 5 days | 5 days | 5 days | 5 days

With multitasking

Adding Buffers

- 50% reduction on the initial estimates – the other 50% are allocated for buffers.
- A **buffer** is additional time to complete a task
- Critical chain scheduling removes buffers from individual tasks and instead creates:
  - **Project buffers** or additional time added before the project’s due date
  - **Feeding buffers** or additional time added before tasks on the critical path

6.7 Control Schedule

- Goals are to know the status of the schedule, influence factors that cause schedule changes, determine that the schedule has changed, and manage changes when they occur
- Tools and techniques include:
  - Progress reports
  - A schedule change control system
  - Project management software, including schedule comparison charts like the tracking Gantt chart
  - Variance analysis, such as analyzing float or slack
  - Performance management, such as earned value (Chapter 7)

Schedule Control Suggestions

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

Chapter Summary

- Project time management is often cited as the main source of conflict on projects, and most IT projects exceed time estimates
- Main processes include:
  - Define activities
  - Sequence activities
  - Estimate activity resources
  - Estimate activity durations
  - Develop schedule
  - Control schedule