Remember

- Quiz 2 this week during the workshop session

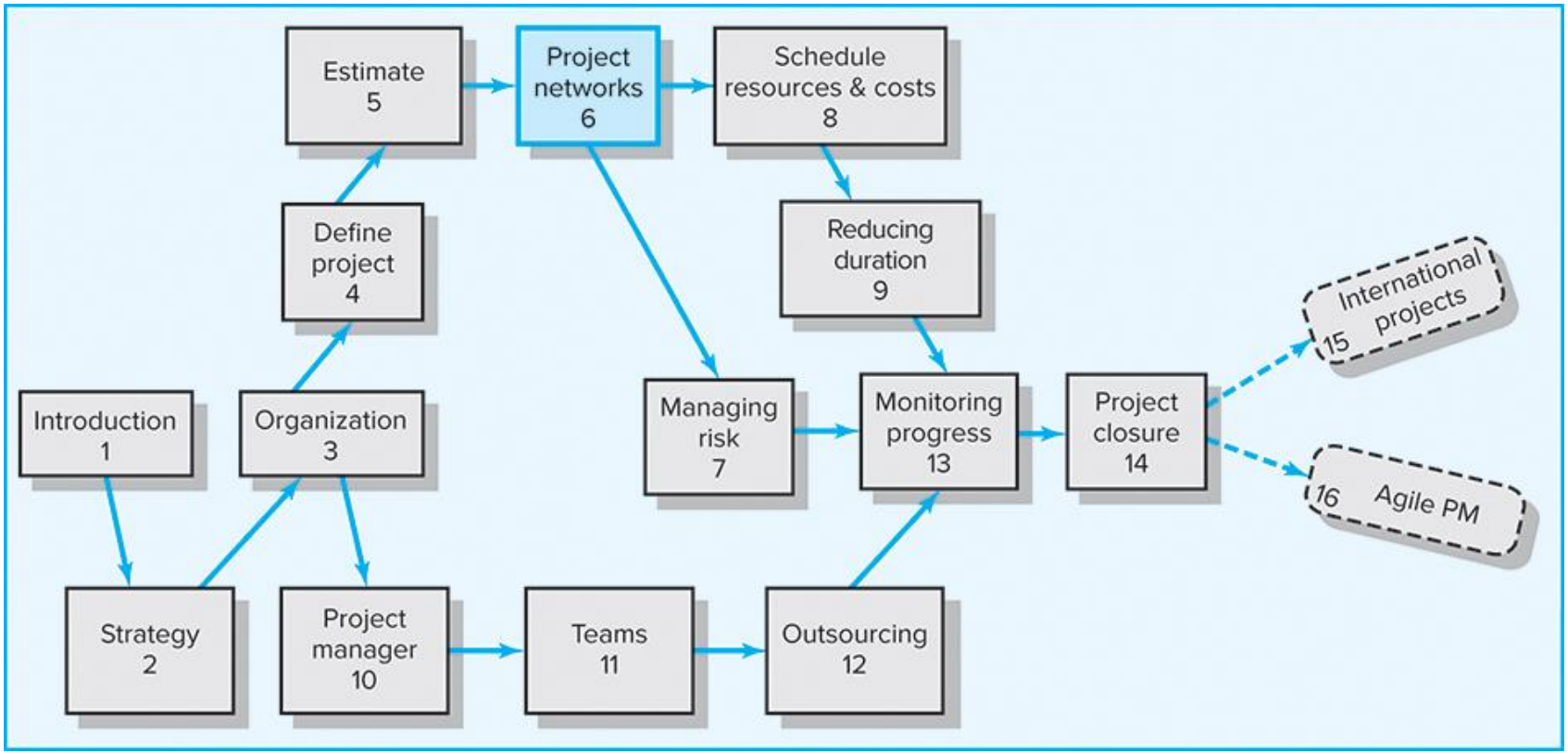
What we learned last week

- Cost estimation including top down and bottom up estimation methods
  - Consensus, Ratio, Apportion, Function Point, and Learning Curves methods
  - Template, parametric procedures, range estimates and phase estimating methods
- Types of project cost
- Refining Estimates
- Creating a Database for Estimating

This week

- Network diagram (AoN) including forward and backward pass
- Critical path methods and Critical path analysis

Where We Are Now



Learning Objectives

1. Understand the linkage between WBS and the project network
2. Diagram a project network using AON methods
3. Calculate early, late, and slack activities times
4. Identify and understand the importance of managing the critical path
5. Distinguish free slack from total slack
6. Demonstrate understanding and application of lags in compressing projects or constraining the start or finish of an activity

Chapter Outline

1 From Work Package to Activity
2 Constructing a Project Network
3 Activity-on-Node (AON) Fundamentals
4 Network Computation Process
5 Using the Forward and Backward Pass Information
6 Extended Network Techniques to Come Closer to Reality

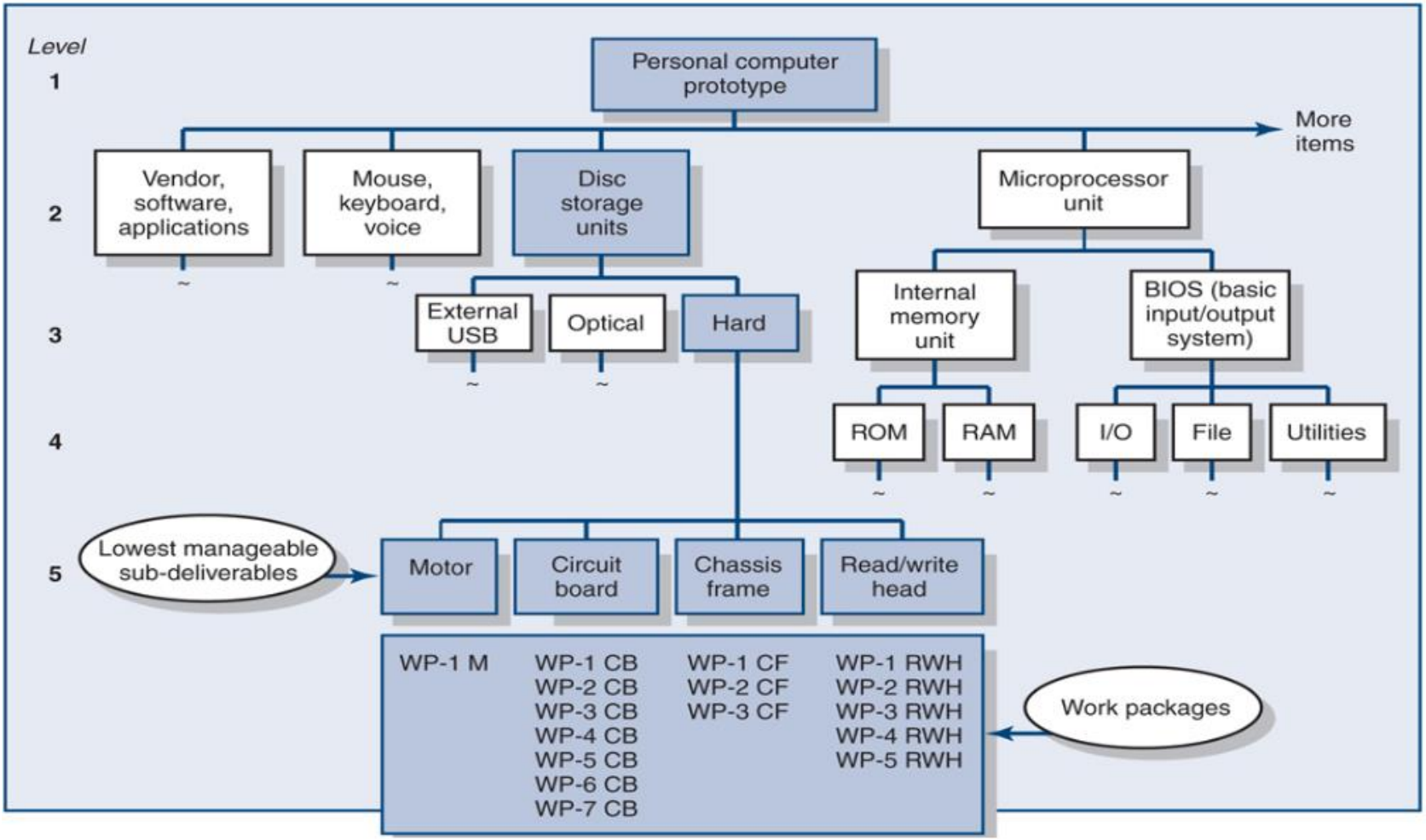
Developing the Project Network

- The Project Network
  - A flow chart that graphically depicts the logical sequences, interdependencies, and start and finish times of the project activities along with **the longest path(s)** through the network—the ***critical path***
    - Provides the basis for **scheduling** labor and equipment.
    - Enhances **communication** among project participants.
    - Provides an **estimate** of the project's **duration**.
    - Provides a basis for **budgeting** cash flow.
    - Highlights activities that are '**critical**' and cannot be delayed.
    - Highlights activities that can be **compressed** to meet a deadline.

From WBS/Work Package to Network

From chapter 4

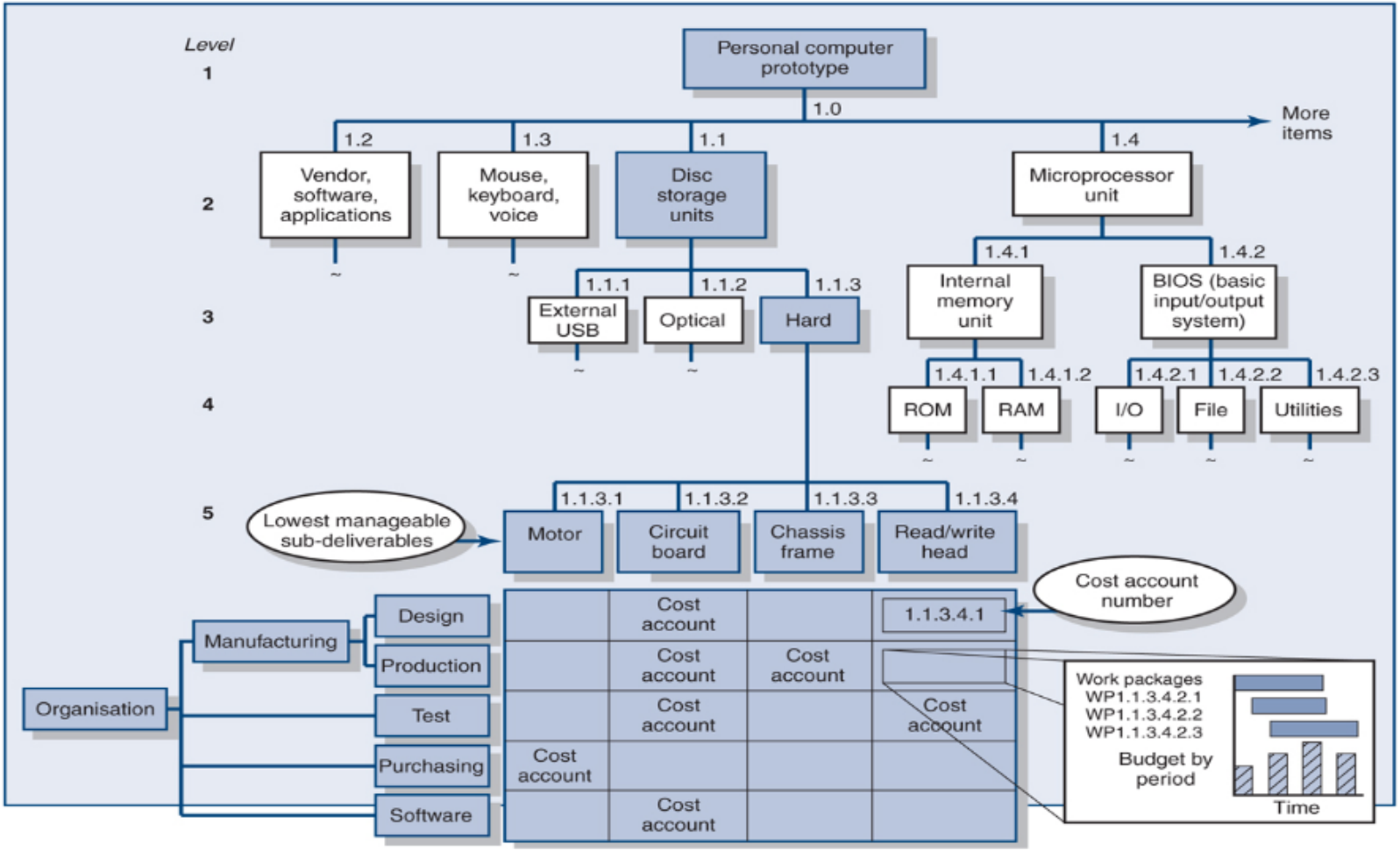
Figure 4.4 WORK BREAKDOWN STRUCTURE (WBS)



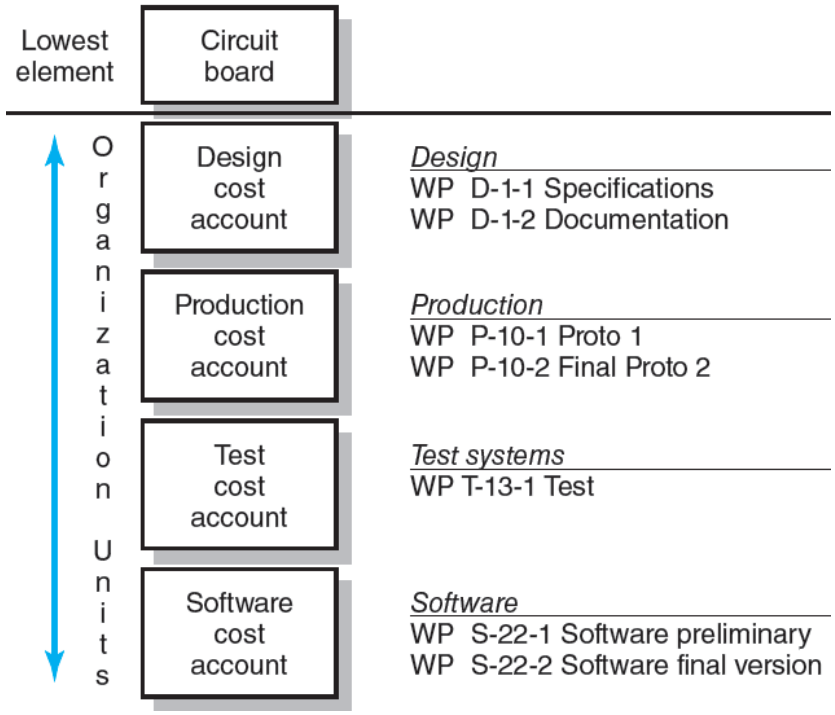
From WBS/Work Package to Network

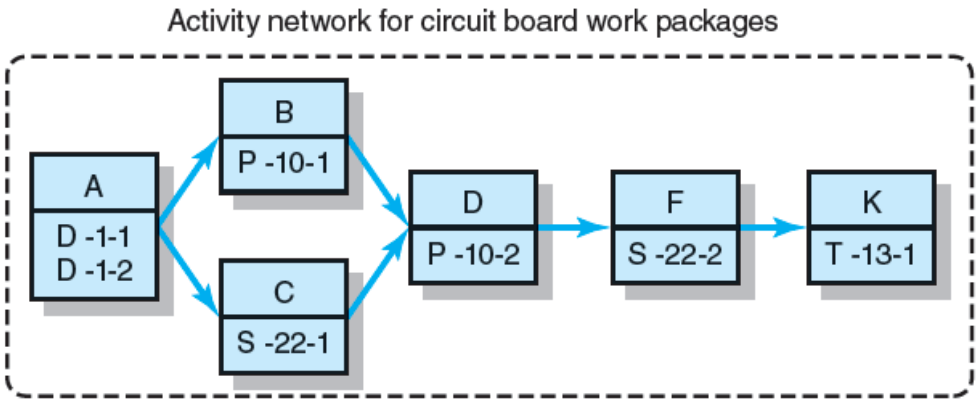
From chapter 4

Figure 4.5 INTEGRATION OF WBS AND OBS



From WBS/Work Package to Network





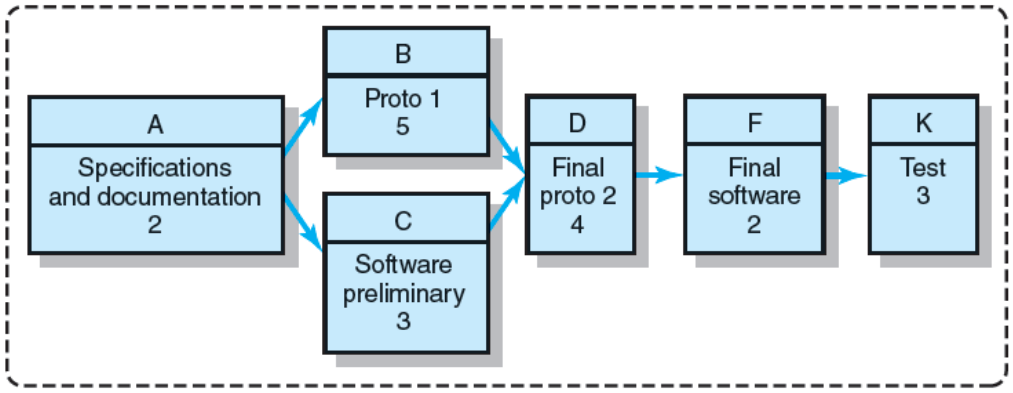
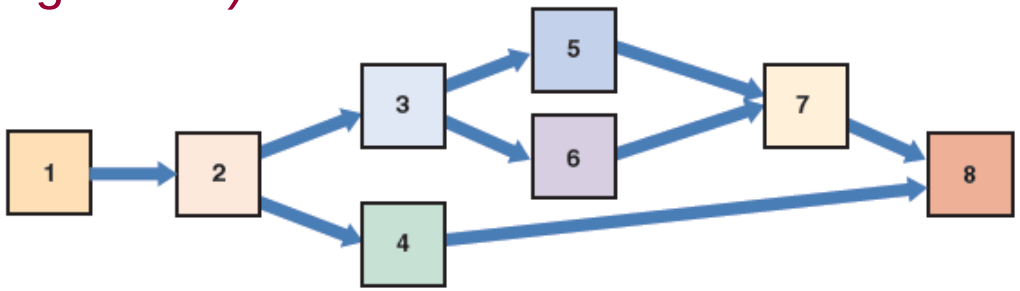


FIGURE 6.1

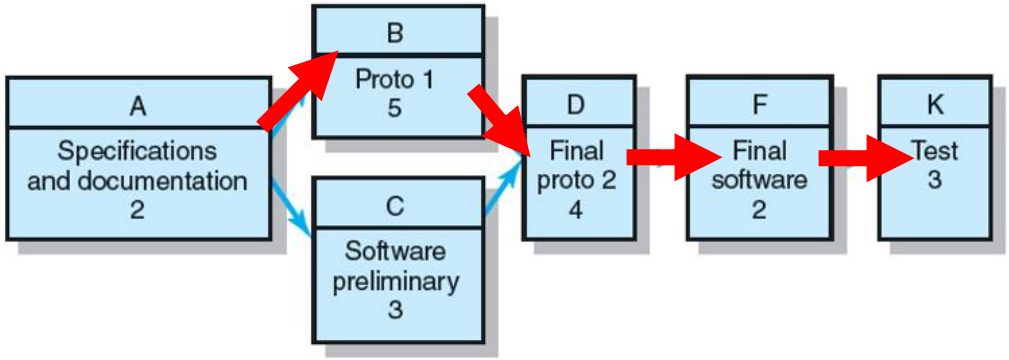
Constructing a Project Network

- Terminology
  - **Activity:** an element of the project that requires time
  - **Parallel Activities:** Activities that can occur independently and, if desired, not at the same time
  - **Merge Activity:** an activity that has two or more preceding activities on which it depends (more than one dependency arrow flowing into it)
  - **Burst Activity:** an activity that has more than one activity immediately following it (more than one dependency arrow flowing from it)



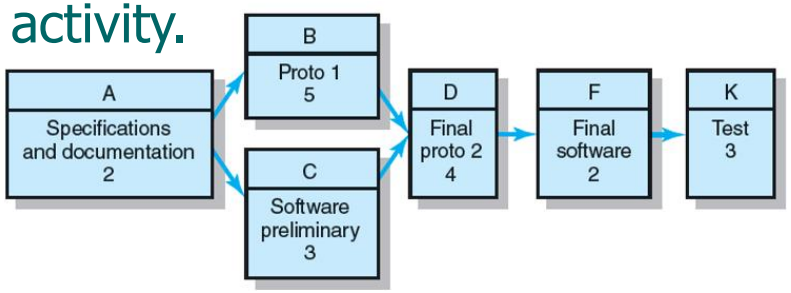
Constructing a Project Network (cont'd)

- Terminology
  - **Path:** a sequence of connected, dependent activities
  - **Critical Path:**
    - The longest path through the activity network that allows for the completion of all project-related activities
    - The shortest expected time in which the entire project can be completed.
    - Delays on the critical path will delay completion of the entire project.

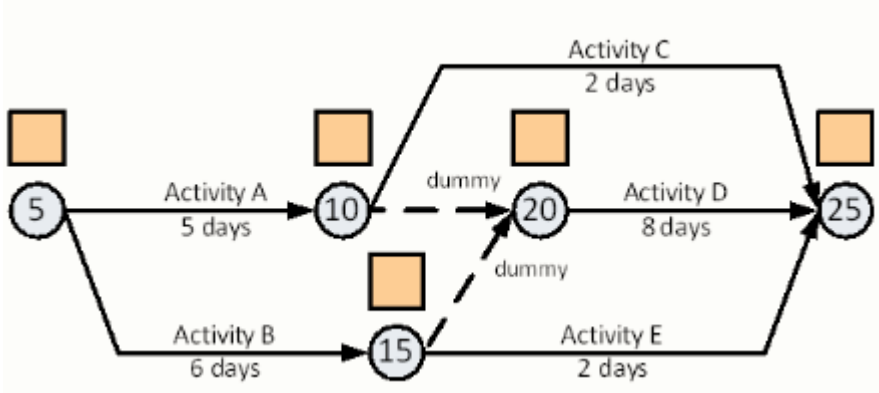


Constructing a Project Network (cont'd)

- Two Approaches
  - Activity-on-Node (AON)
    - Uses a node to depict an activity.

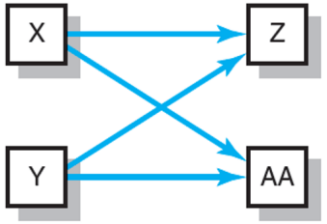


  - Activity-on-Arrow (AOA)
    - Uses an arrow to depict an activity.



Basic Rules to Follow in Developing Project Networks

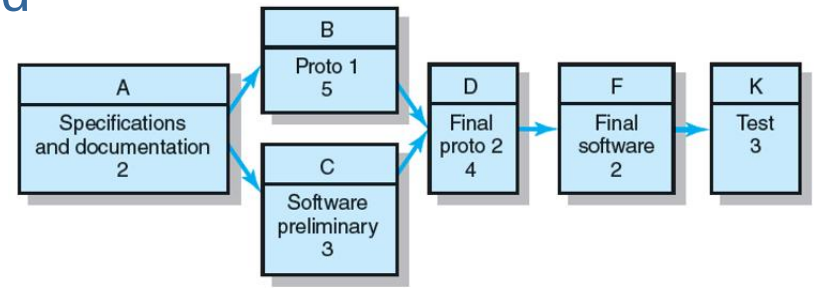
1. Networks typically flow from left to right.
2. An activity cannot begin until all preceding connected activities are complete.
3. Arrows indicate precedence and flow and can cross over each other.



4. Each activity must have a unique identify number.
5. An activity identification number must be greater than that of any predecessor activities.
6. Looping is not allowed.
7. Conditional statements are not allowed

   If successful, do something; if not, do nothing
8. Use common start and stop nodes.

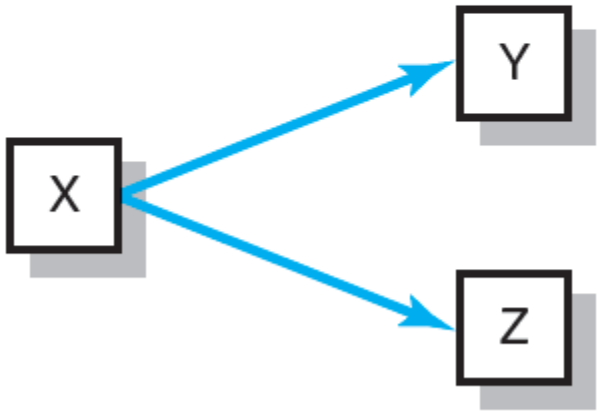
   For multiple starts and stops



Activity-on-Node Fundamentals

A is preceded by nothing
B is preceded by A
C is preceded by B

(A)



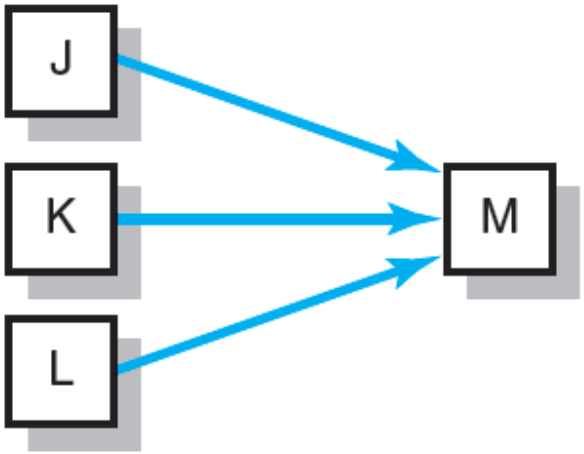
Y and Z are preceded by X

Y and Z can begin at the same time, if you wish

(B) X is a burst activity

FIGURE 6.2

Activity-on-Node Fundamentals (cont'd)

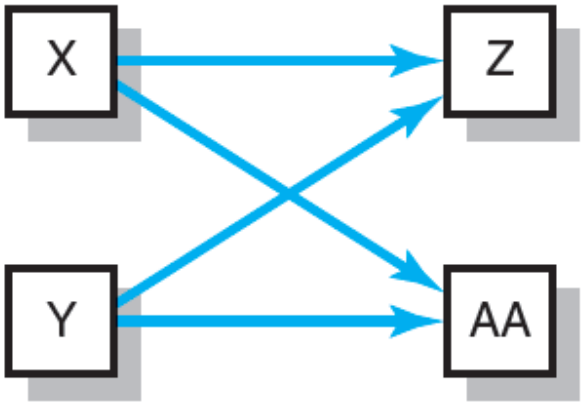


J, K, & L can all begin at the same time, if you wish (they need not occur simultaneously)

but

All (J, K, L) must be completed before M can begin

(C) M is a merge activity



Z is preceded by X and Y

AA is preceded by X and Y

(D)

FIGURE 6.2 (cont'd)

AON Example

- Lets Do it!!

EV-6

Activity	Predecessor
A	
B	A
C	A
D	B
E	B,C
F	D
G	E, F

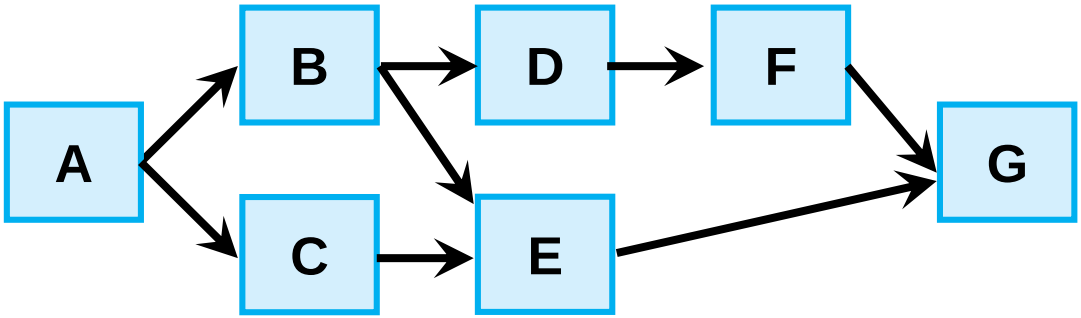
DRAW AN ACTIVITY-ON-NODE NETWORK FOR THE ABOVE PROJECT

AON Example

- Lets Do it!!

EV-6

Activity	Predecessor
A	
B	A
C	A
D	B
E	B,C
F	D
G	E, F



DRAW AN ACTIVITY-ON-NODE NETWORK FOR THE ABOVE PROJECT

Network example: Automated warehouse

Network Information

AUTOMATED WAREHOUSE
Order Picking System

Activity	Description	Preceding Activity
A	Define Requirements	None
B	Assign Team	A
C	Design Hardware	A
D	Code Software	B
E	Build and Test Hardware	C
F	Develop Patent Request	C
G	Test Software	D
H	Integrate Systems	E, F, G

At this beginning stage, the project information contains activities with predecessors.

TABLE 6.1

Network Information

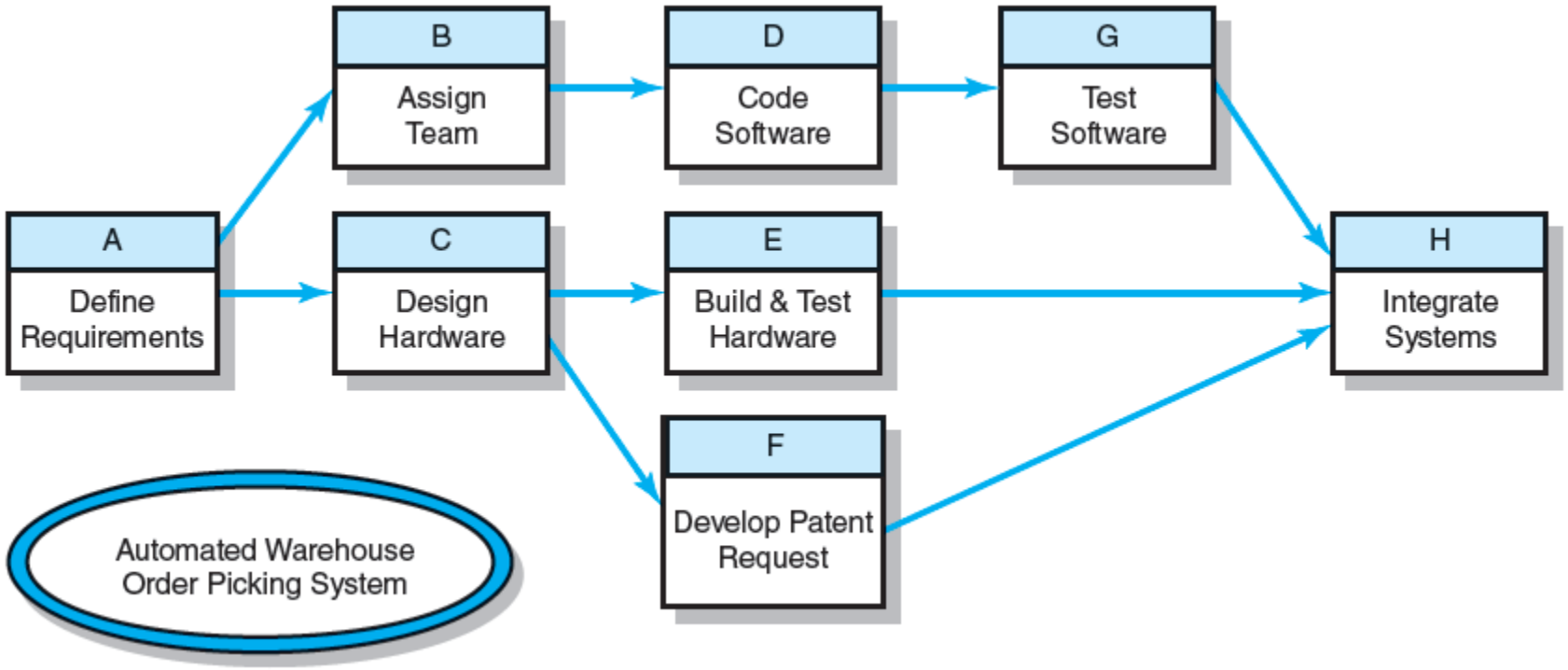
AUTOMATED WAREHOUSE
Order Picking System

Activity	Description	Preceding Activity
A	Define Requirements	None
B	Assign Team	A
C	Design Hardware	A
D	Code Software	B
E	Build and Test Hardware	C
F	Develop Patent Request	C
G	Test Software	D
H	Integrate Systems	E, F, G

Automated Warehouse—Complete Network

AUTOMATED WAREHOUSE
Order Picking System

Activity	Description	Preceding Activity
A	Define Requirements	None
B	Assign Team	A
C	Design Hardware	A
D	Code Software	B
E	Build and Test Hardware	C
F	Develop Patent Request	C
G	Test Software	D
H	Integrate Systems	E, F, G

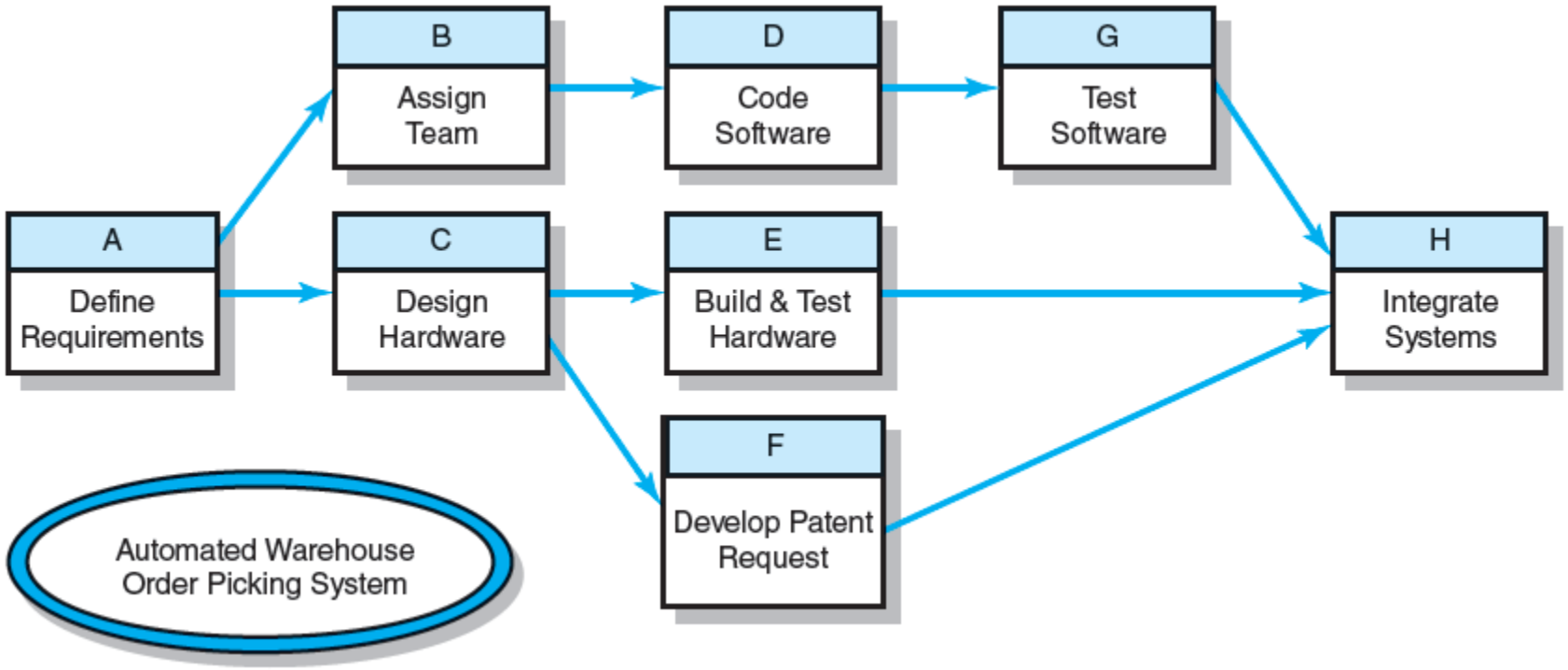


Automated Warehouse—Complete Network

AUTOMATED WAREHOUSE
Order Picking System

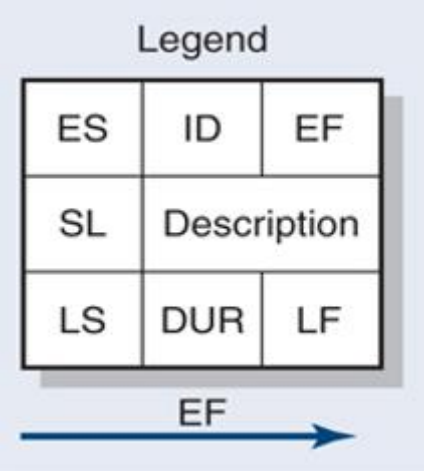
Activity	Description	Preceding Activity
H	Integrate Systems	E, F, G

At this stage a project network has been developed depicting the activities with sequences and dependencies via arrows. **The next step** is to estimate duration of activities through ***the network computation process.***



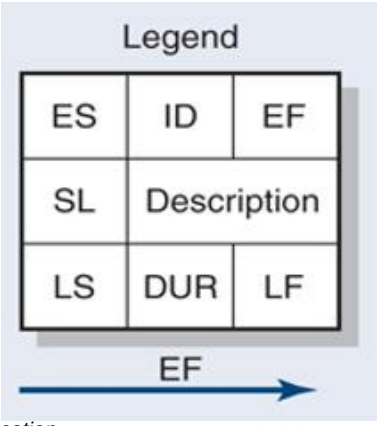
Network Computation Process

Term	Acronym	Description	Abbreviation
Late finish	LF	The latest an activity can finish and not delay a following activity	LF = LS + DUR
Late start	LS	The latest an activity can start and not delay a following activity	LS = LF – DUR
Early finish	EF	The earliest an activity can finish if all preceding activities are finished by their early finish times	EF = ES + DUR
Early start	ES	The earliest an activity can start. It is the largest early finish of all its immediate predecessors	ES = EF – DUR



Network Computation Process

- Forward Pass—Earliest Times
  - How soon can the activity start? (early start—ES)
  - How soon can the activity finish? (early finish—EF)
  - How soon can the project finish? (expected time—ET)
- Backward Pass—Latest Times
  - How late can the activity start? (late start—LS)
  - How late can the activity finish? (late finish—LF)
  - Which activities represent the critical path?
  - How long can the activity be delayed?
    - (slack or float—SL)



Network Information

AUTOMATED WAREHOUSE
Order Picking System

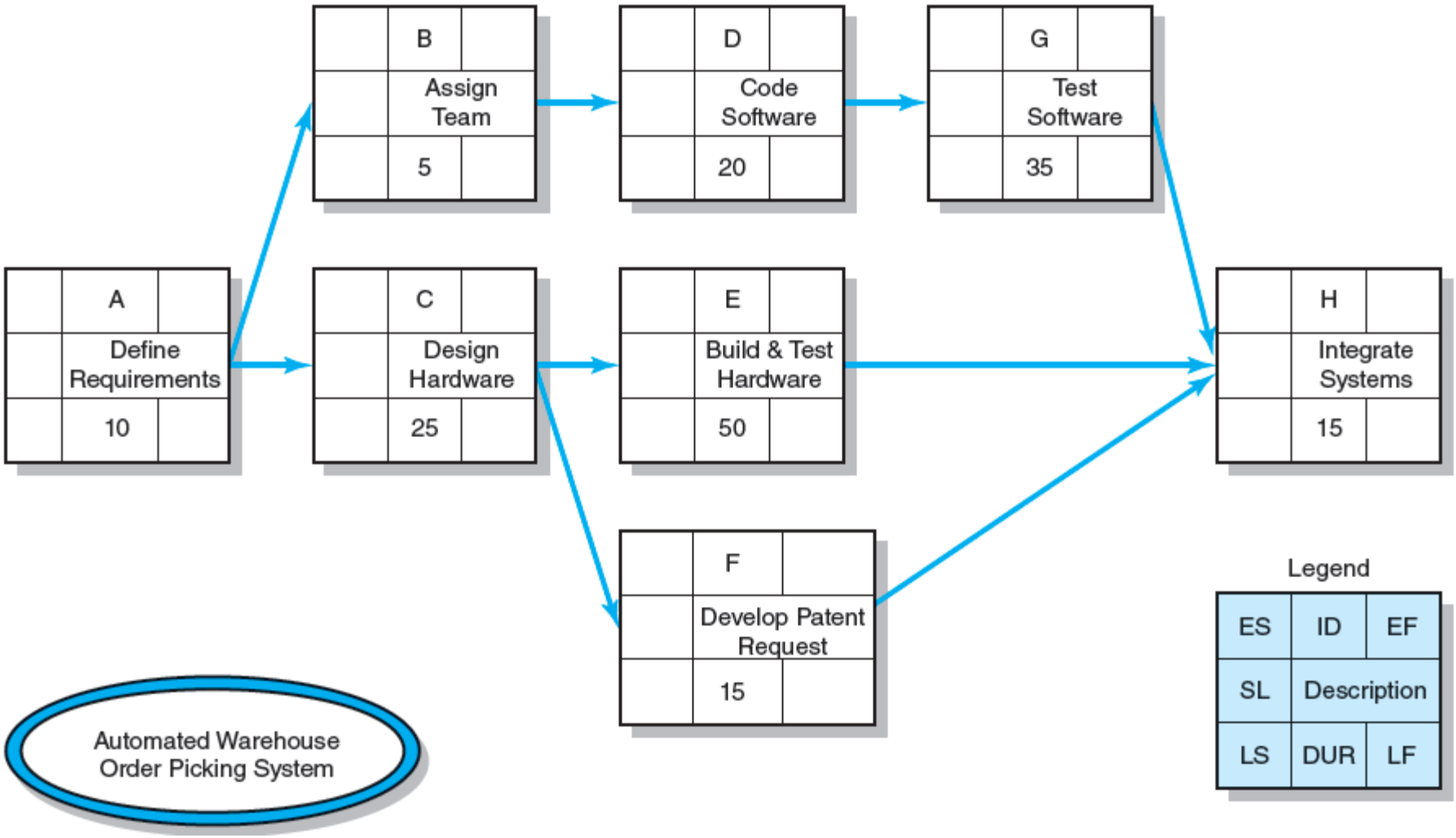
Activity	Description	Preceding Activities	Activity Time
A	Define Requirements	None	10 workdays
B	Assign Team	A	5
C	Design Hardware	A	25
D	Code Software	B	20
E	Build and Test Hardware	C	50
F	Develop Patent Request	C	15
G	Test Software	D	35
H	Integrate Systems	E, F, G	15

The activity times are added to the network information from the work packages.

AON Example

- Lets do it!

Activity-on-Node Network



This stage shows the network with the activity time estimate found in the node. The next steps are to complete the forward and backward pass.

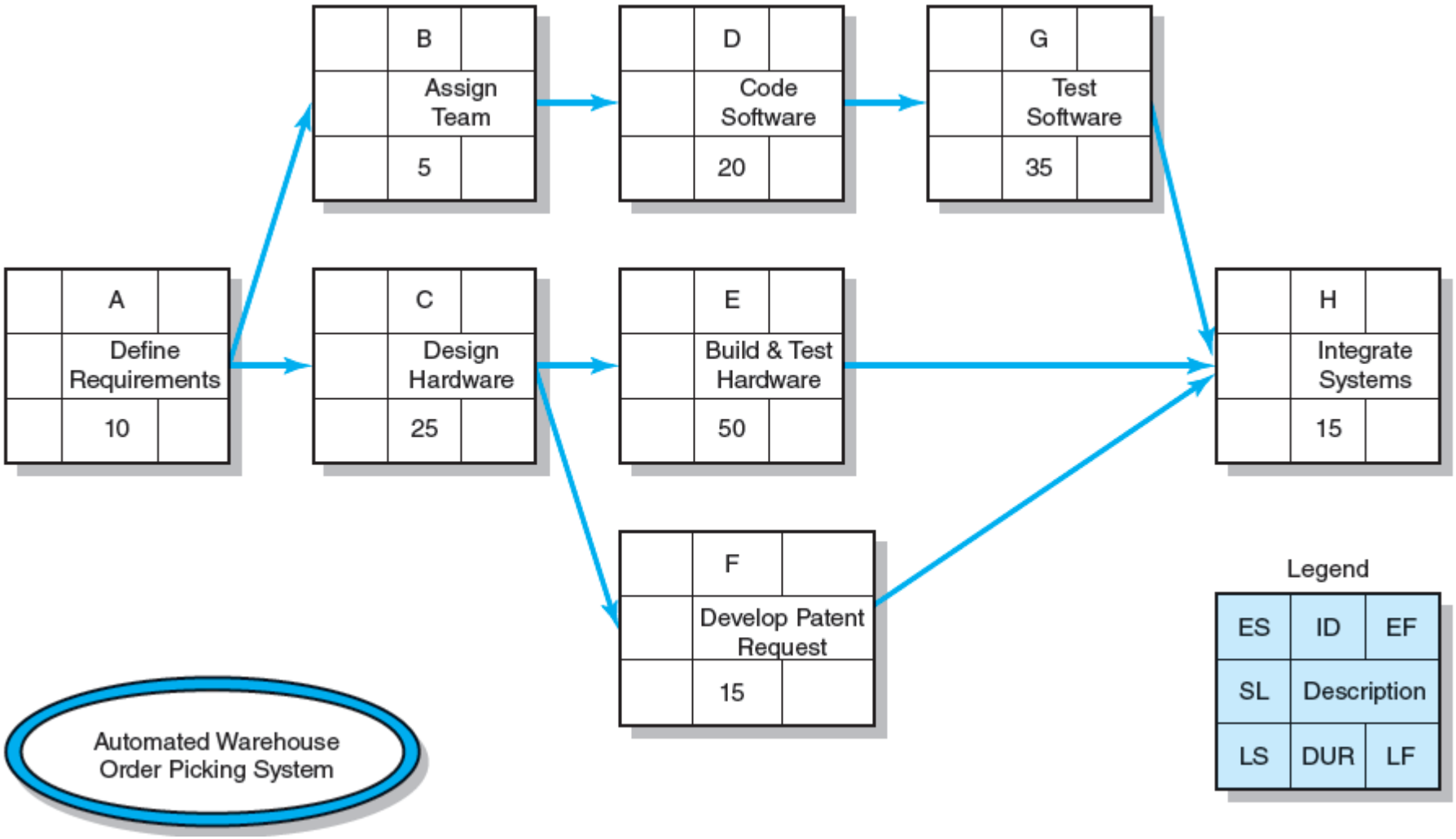
FIGURE 6.5

Forward Pass Computation

- Add activity times along each path in the network (ES + Duration = EF).
- Carry the early finish (EF) to the next activity where it becomes its early start (ES) ***unless…***
  - The next succeeding activity is a *merge* activity, in which case the ***largest*** early finish (EF) number of all its immediate predecessor activities is selected.

Activity-on-Node Network

- Lets do it!



Activity-on-Node Network Forward Pass

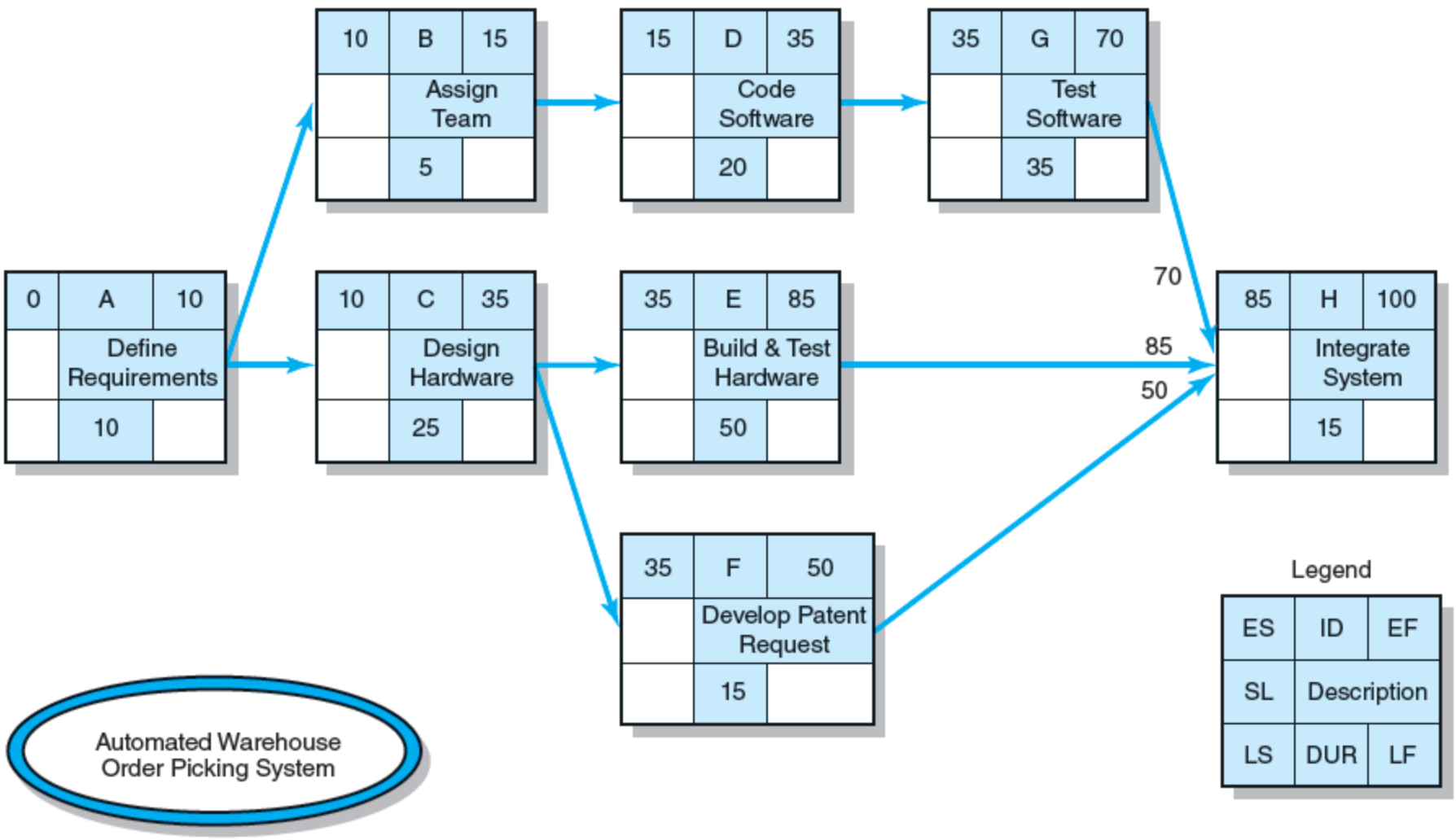


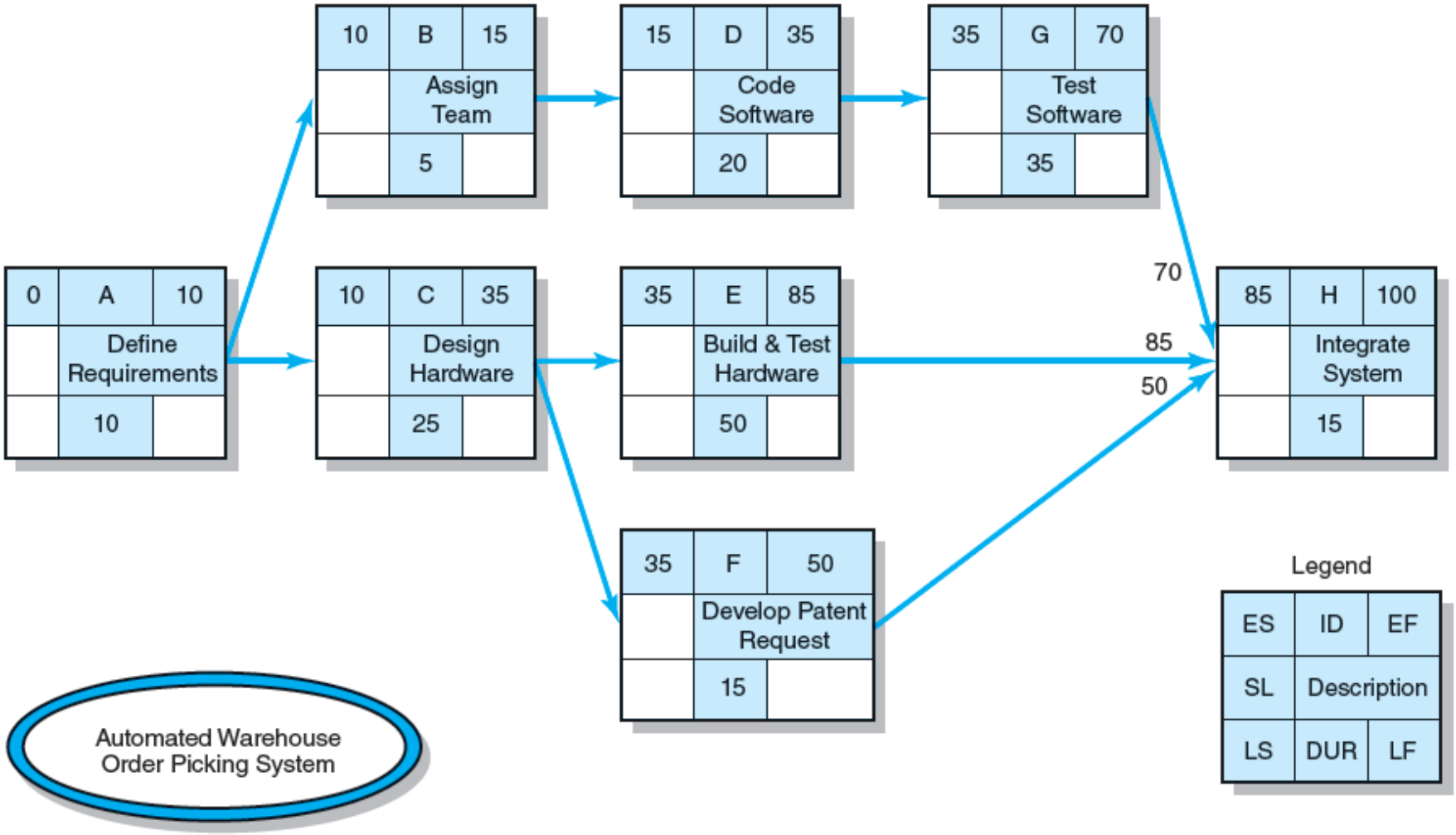
FIGURE 6.6

Backward Pass Computation

- Subtract activity times along each path starting with the project end activity (LF - Duration = LS).
- Carry the late start (LS) to the next preceding activity where it becomes its late finish (LF) ***unless*** *there is any imposed ending date.*
- The next succeeding activity is a *burst* activity, in which case the ***smallest*** late start (LS) number of all its immediate successor activities is selected.

Activity-on-Node Network

- Lets do it.



Activity-on-Node Network Backward Pass

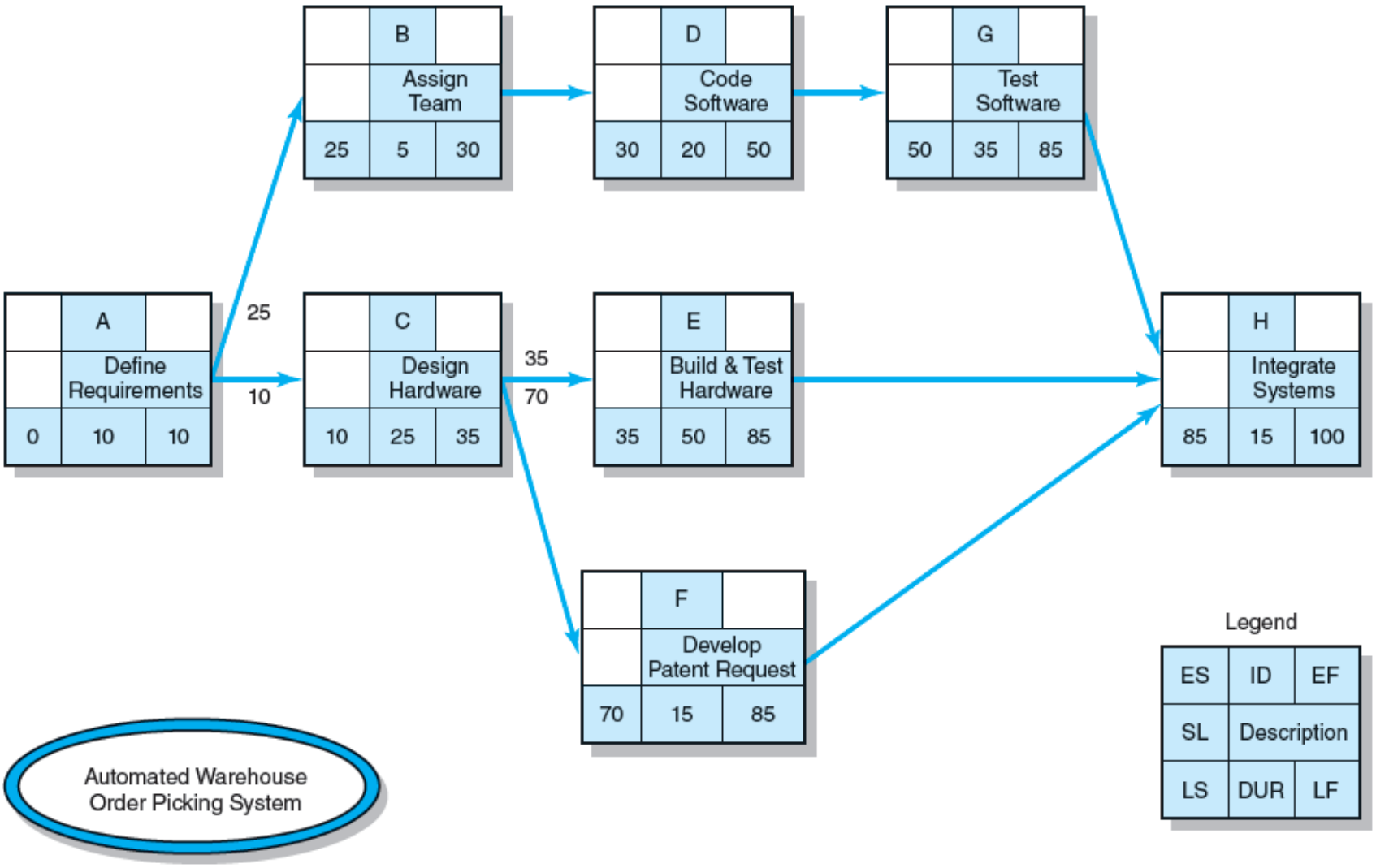


FIGURE 6.7

Determining Total Slack (TS)

- Total Slack (or Float)
  - Tells us the amount of time an activity can be delayed without delaying the project.
  - Is how long an activity can exceed its early finish date without affecting the project end date or an imposed completion date.
  - Is simply the difference between the LS and ES (LS – ES = SL) or between LF and EF (LF – EF = SL).
- After slack is computed the critical path can be identified

Forward and Backward Passes Completed with Slack Times

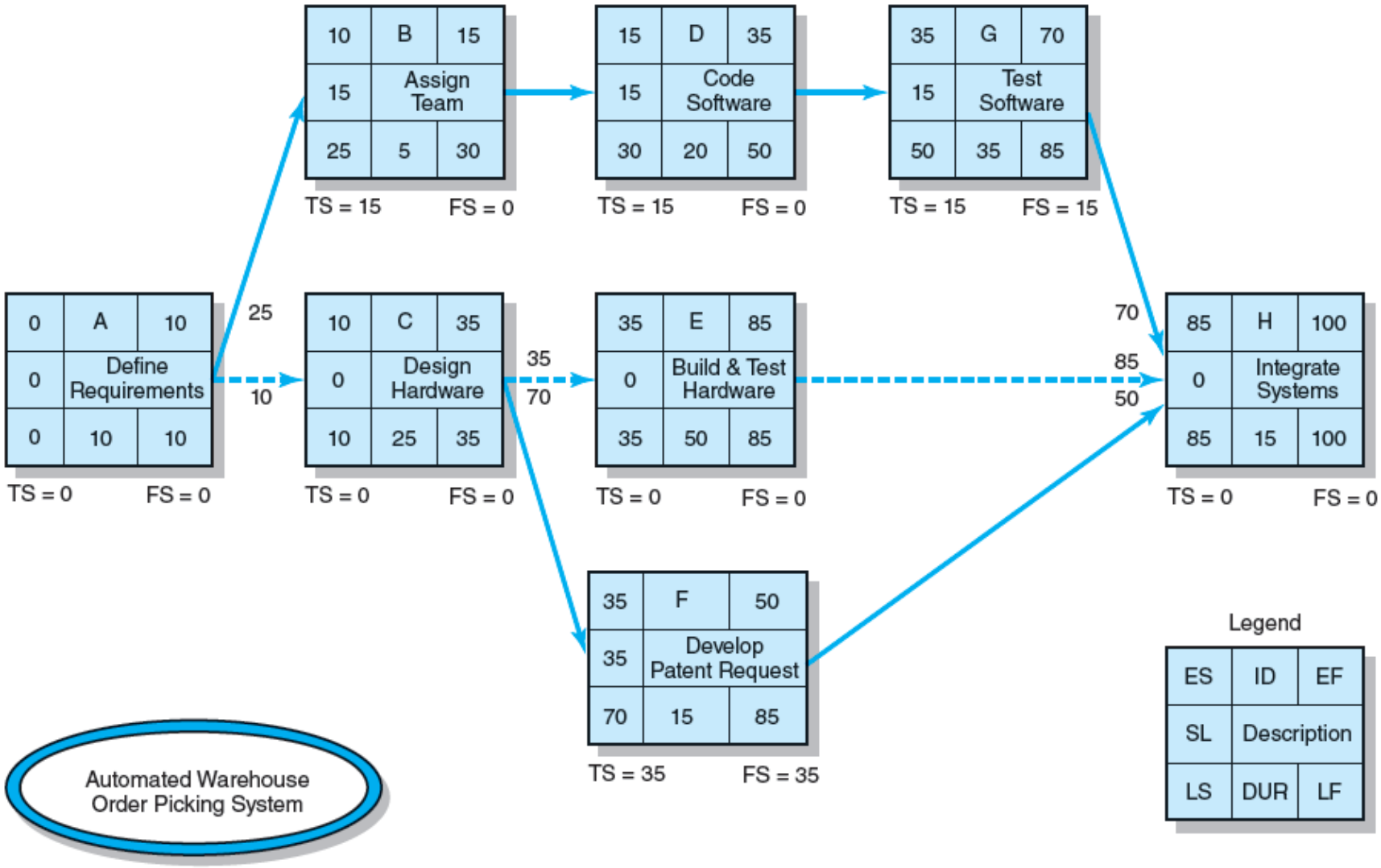
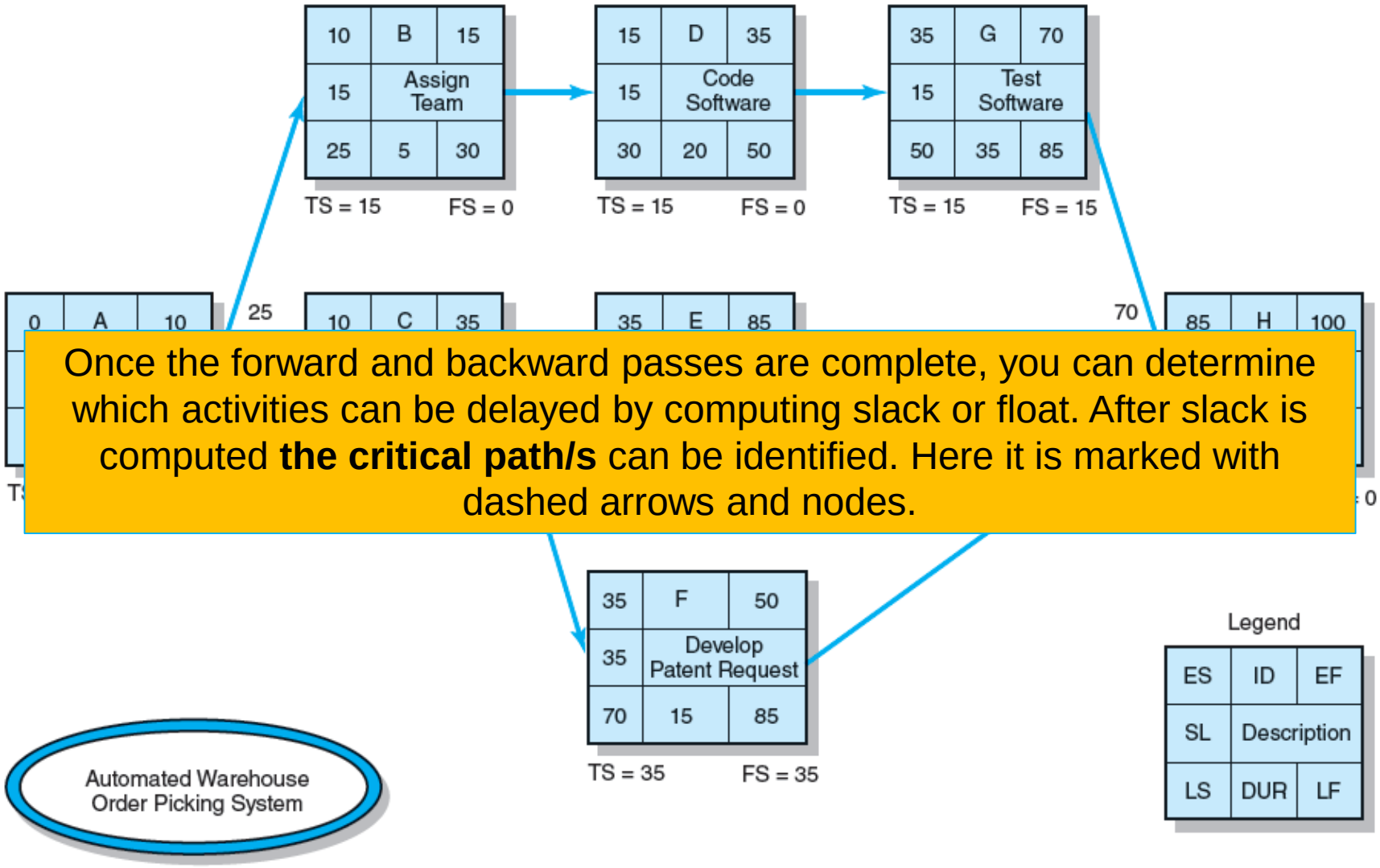


FIGURE 6.8

Forward and Backward Passes Completed with Slack Times



Once the forward and backward passes are complete, you can determine which activities can be delayed by computing slack or float. After slack is computed **the critical path/s** can be identified. Here it is marked with dashed arrows and nodes.

FIGURE 6.8

The Critical Path

- Is the network path(s) that has (have) the least slack in common.
- Is the longest path through the activity network.
- Is the shortest expected time in which the entire project can be completed.
- Is important because it impacts completion time.
- Is where you put best people on.
- Is where you pay extra extension when doing risk assessment.
- Is where you look at when other managers asking to ‘borrow’ people or equipment.
- Is where you look at when you don’t have time to monitor all activities.

Determining Free Slack (FS)

- Free Slack
  - Is how long an activity can exceed its early finish date without affecting early start dates of any successor(s).
  - Allows flexibility in scheduling scarce resources.
  - Only activities that occur at the end of a chain of activities, where ***you have a merge activity***, can have free slack.

F…ses Completed with Slack Times

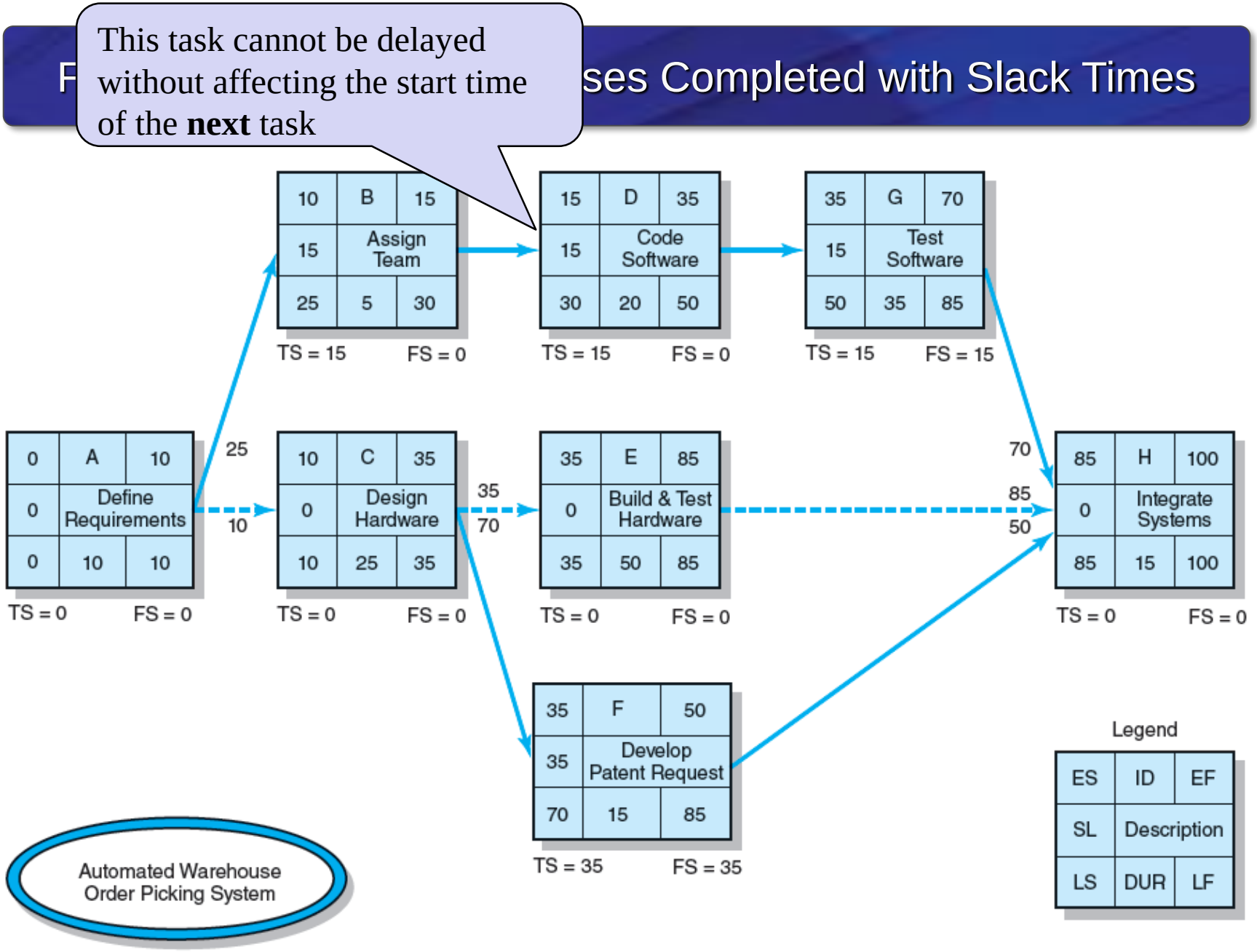


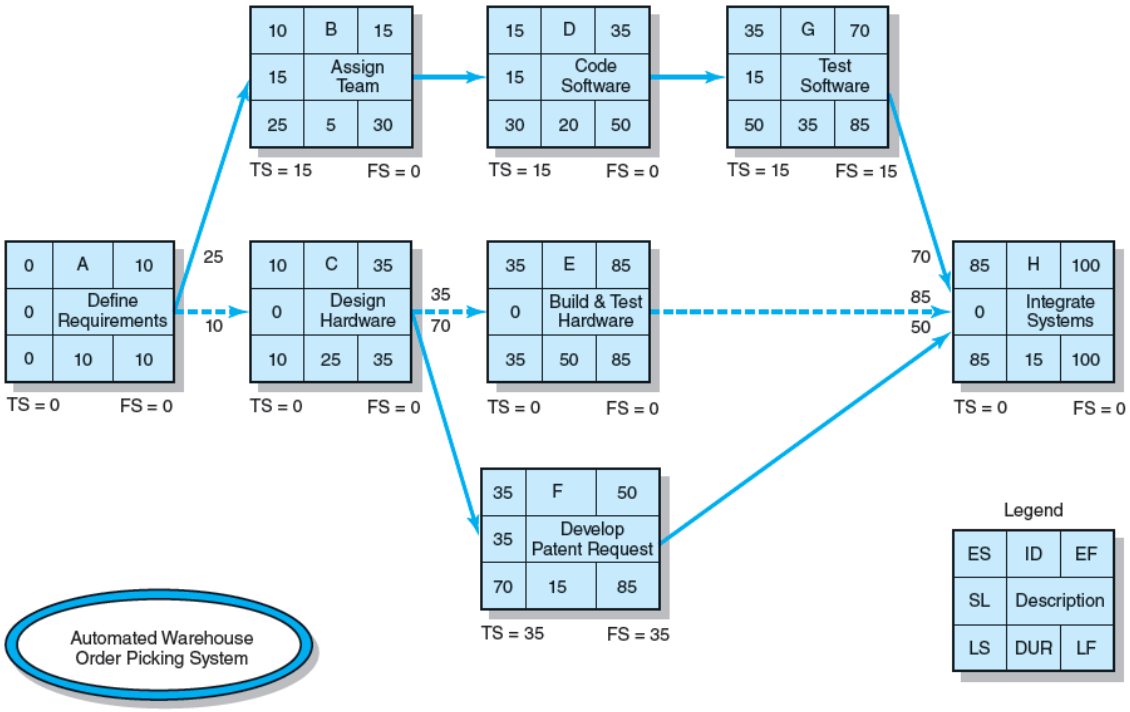
FIGURE 6.8

Network Sensitivity

- Sensitivity
  - the likelihood the original critical path(s) will change once the project is initiated.
- A sensitive network: more than one critical path and/or noncritical activities with very little slack.
- An insensitive network: only one critical path and noncritical activities with significant slack.

Network Sensitivity

How sensitive is the Automated warehouse schedule?

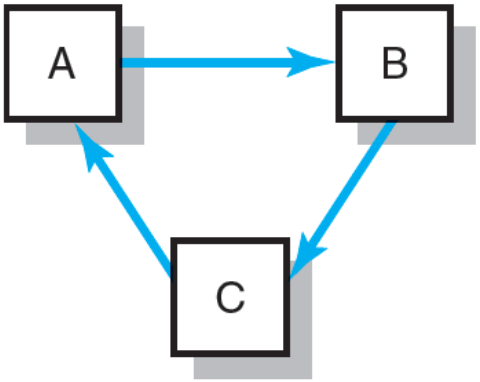


Not very, since there is only one critical path and the two other noncritical paths have 15 and 35 days of slack

➔ considerable flexibility

Network: Practical Considerations

- Network Logic Errors
  - Conditional statements
    - If test successful build proto
    - If failure redesign
  - an illogical loop depicted



- Activity Numbering
  - A unique identification code
  - Ascending order
- Use of Computers to Develop Networks (and Gantt Chart)

Automated Order Warehouse Picking System Bar Chart

Automated Warehouse Order Picking System Schedule Table

ID	Duration	Task Name	Start	Finish	Late Start	Late Finish	Free Slack	Total Slack
2	10	Define Req.s	Tue 1/1	Mon 1/14	Tue 1/1	Tue 1/15	0 days	0 days
3	5	Assign Team	Tue 1/15	Mon 1/21	Tue 2/5	Mon 2/11	0 days	15 days
4	25	Design Hardware	Tue 1/15	Mon 2/18	Tue 1/15	Mon 2/18	0 days	0 days
5	20	Code Software	Tue 1/22	Mon 2/18	Tue 2/12	Mon 3/11	0 days	15 days
6	50	Build and Test HW	Tue 2/19	Mon 4/29	Tue 2/19	Mon 4/29	0 days	0 days
7	15	Dev. Patent Request	Tue 2/19	Mon 3/11	Tue 4/9	Mon 4/29	35 days	35 days
8	35	Test Software	Tue 2/19	Mon 4/8	Tue 3/12	Mon 4/29	15 days	15 days
9	15	Integrate Systems	Tue 4/30	Mon 5/20	Tue 4/30	Mon 5/20	0 days	0 days

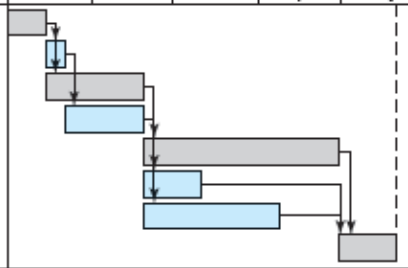
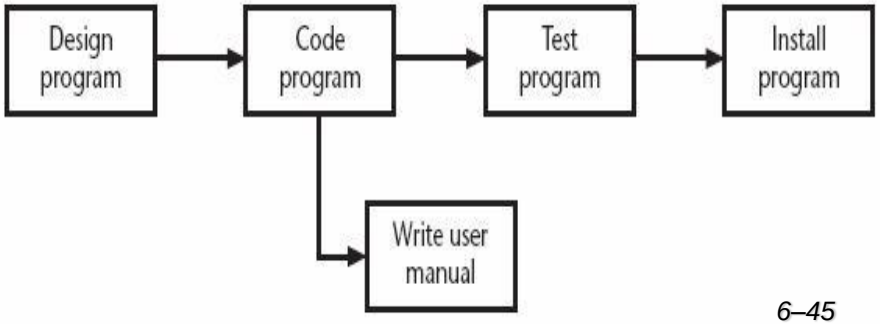


FIGURE 6.11

Network: Practical Considerations

- Network Logic Errors
  - Conditional statements
  - an illogical loop depicted
- Activity Numbering
- Use of Computers to Develop Networks (and Gantt Chart)

- Calendar Dates
- Multiple Starts and Multiple Projects
  - Require a common start and finish activity
  - Network should not contain dangles



Extended Network Techniques to Come Close to Reality

- Laddering
  - Activities are broken into segments so the following activity can begin sooner and not delay the work.

Example of Laddering Using Finish-to-Start Relationship

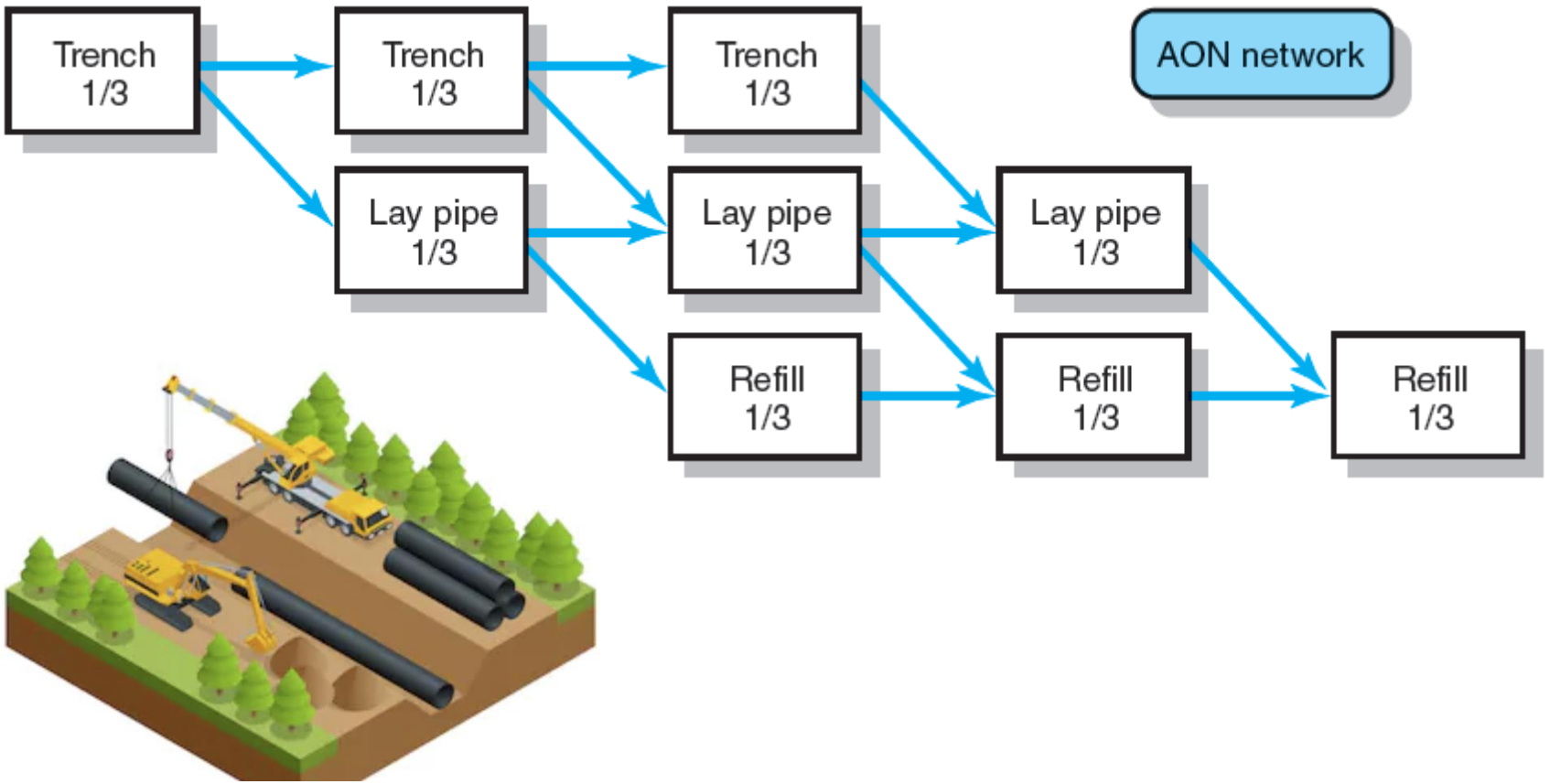


FIGURE 6.12

Extended Network Techniques to Come Close to Reality

- Laddering
  - Activities are broken into segments so the following activity can begin sooner and not delay the work.
- Lags
  - The minimum amount of time a dependent activity **must be delayed** to begin or end.
    - Lengthy activities are broken down to reduce the delay in the start of successor activities.
    - Lags can be used to constrain finish-to-start, start-to-start, finish-to-finish, start-to-finish, or combination relationships.

Finish-to-start lag relationship

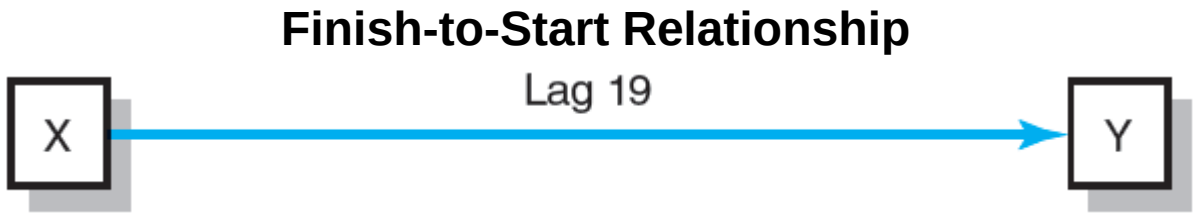
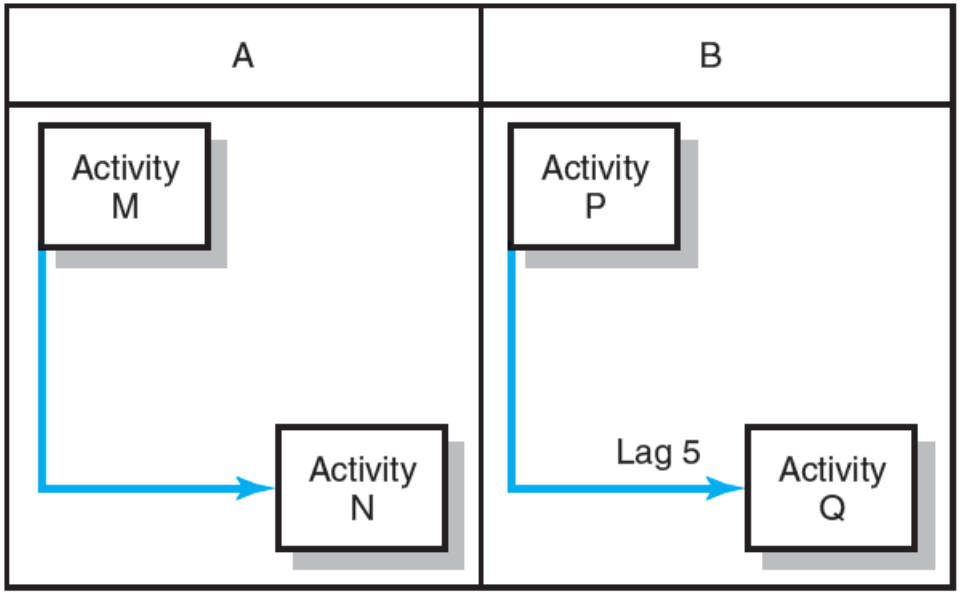


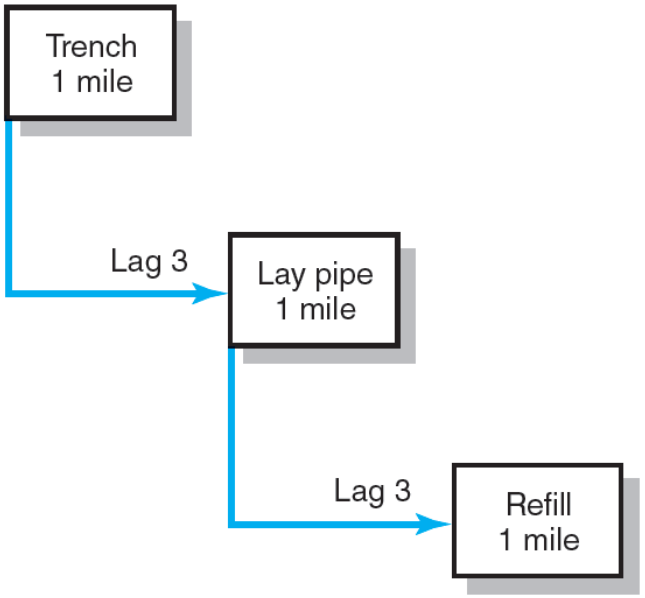
FIGURE 6.13

- Finish-to-start lags are often used when ordering materials. One day to place order and 19 days to receive the goods.
- Use of finish-to-start lags should be justified and approved to avoid unnecessary buffering.

Start-to-start lag relationship



Use of Lags to Reduce Schedule Detail and Project Duration



Often used in concurrent engineering

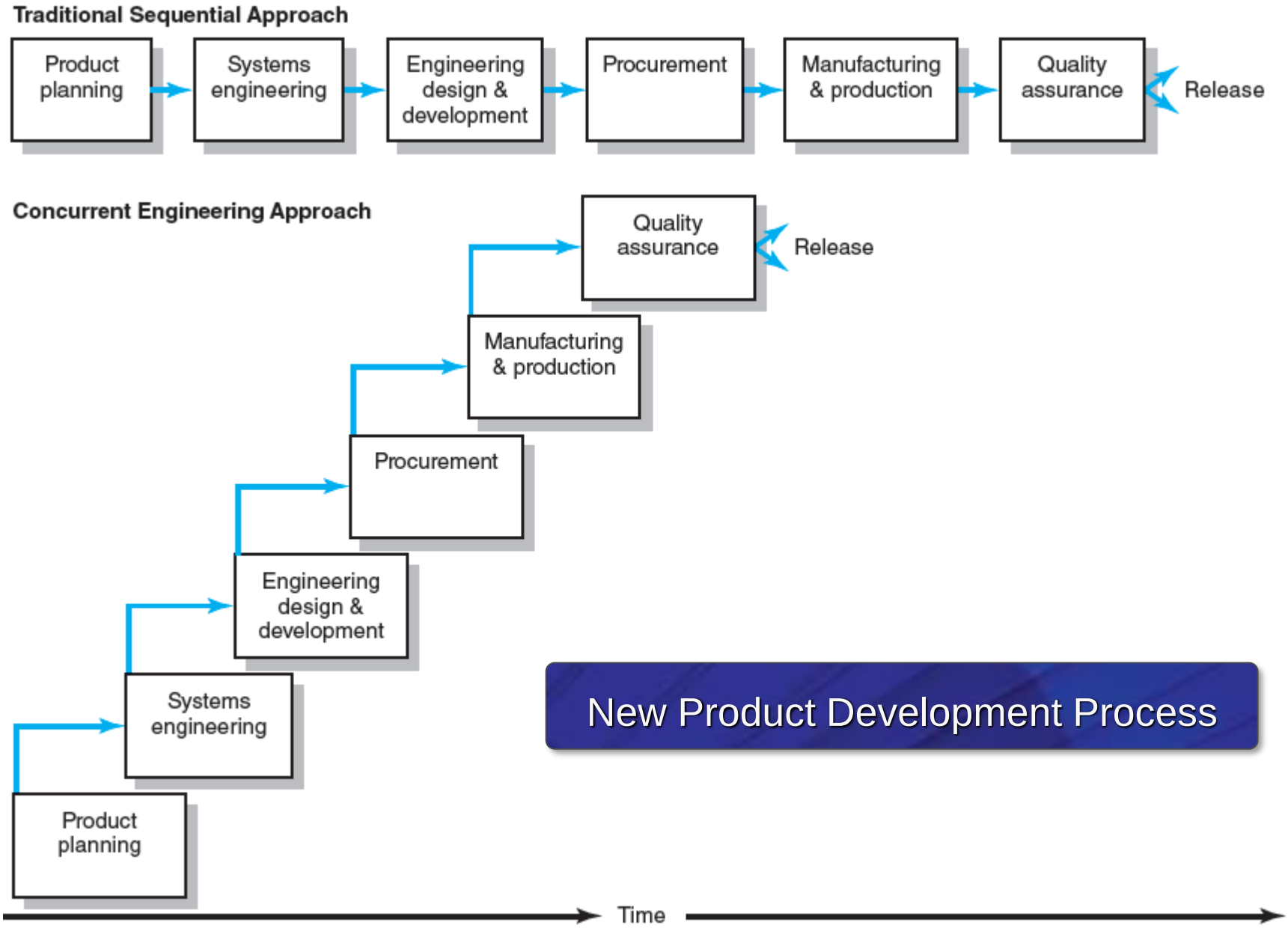
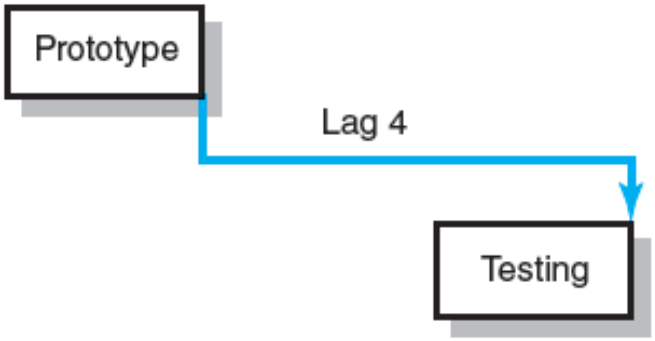


FIGURE 6.16

Finish-to-finish lag relationship

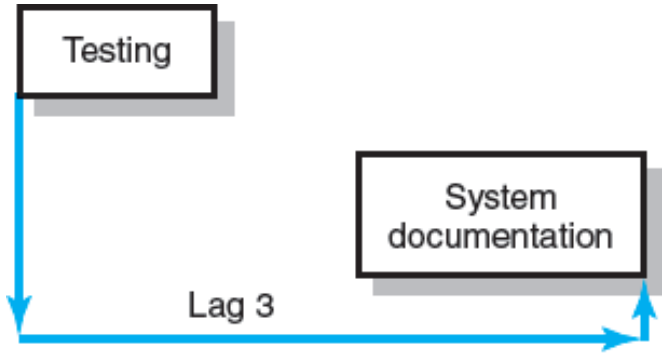


Finish-to-Finish Relationship

FIGURE 6.17

- The finish of one activity depends on the finish of another activity.
- As an example, testing cannot be completed any earlier than four days after the prototype is complete.

Start-to-finish lag relationship

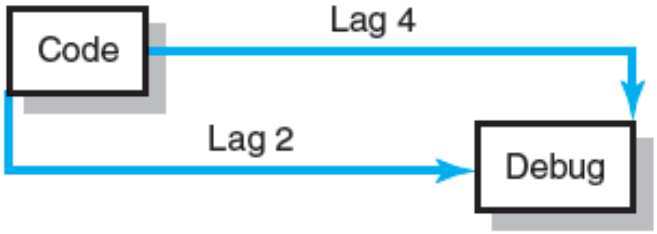


Start-to-Finish Relationship

FIGURE 6.18

- The finish of an activity depends on the start of another activity.
- As an example, product documentation cannot end until three days after testing has started. Here all the relevant information is produced after three days of testing.

Combination relationship



Combination Relationships

FIGURE 6.19

- These relationships are usually start-to-start and finish-to-finish combinations tied to two activities.
- Debug cannot begin until two time units after coding has started. Coding must be finished four days before debug can be finished.

Network Using Lags

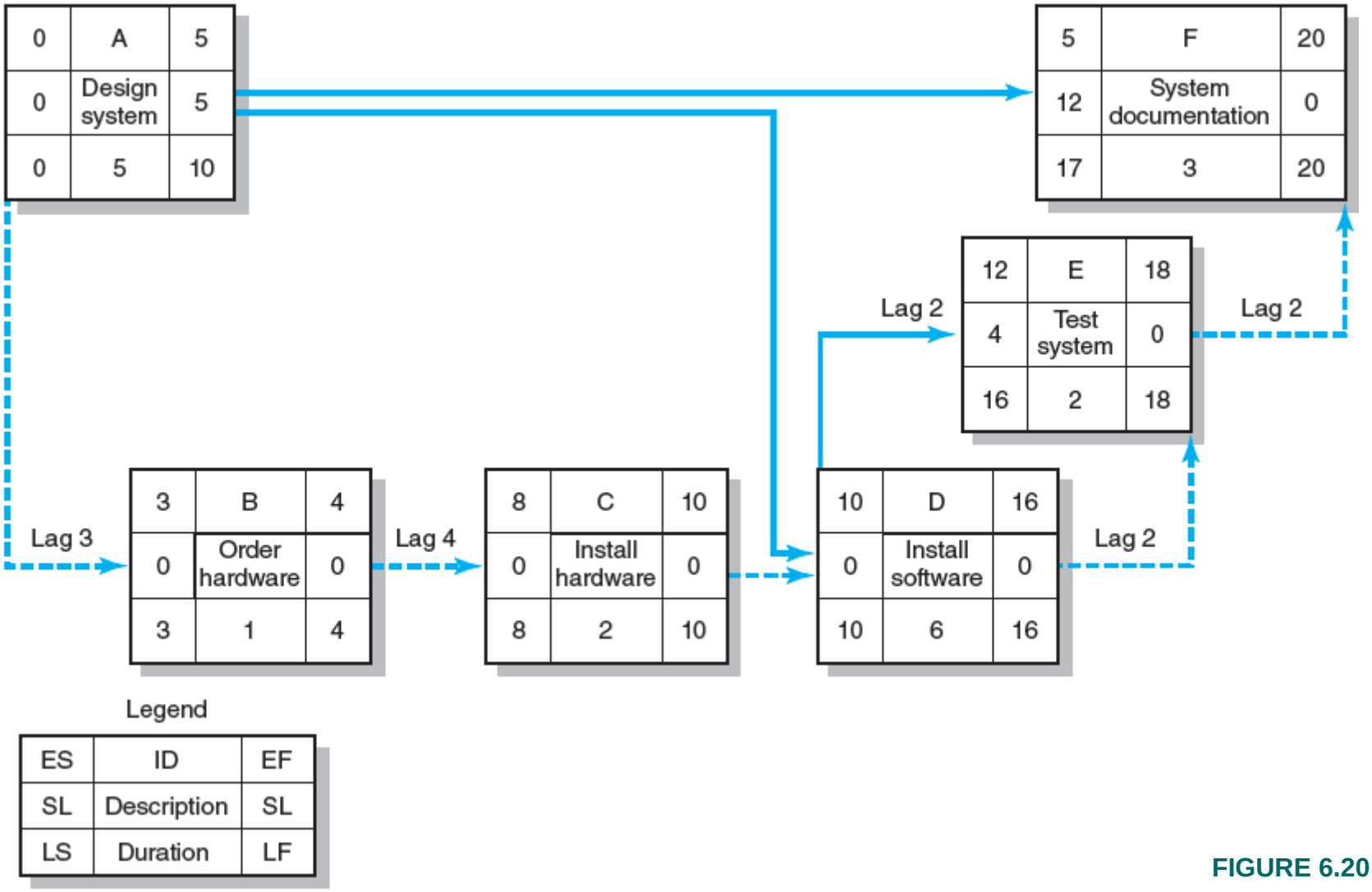


FIGURE 6.20

Hammock Activities

- An activity that spans over a segment of a project.
- The *duration* of hammock activities is determined after the network plan is drawn.
- Hammock activities are used to aggregate sections of the project to facilitate getting the right amount of detail for specific sections of a project.
  - Can be useful to control the costing of a special resource used over several tasks, such as a special crane in a construction site, lighting controller hired for a stage play, etc.

Hammock Activity Example

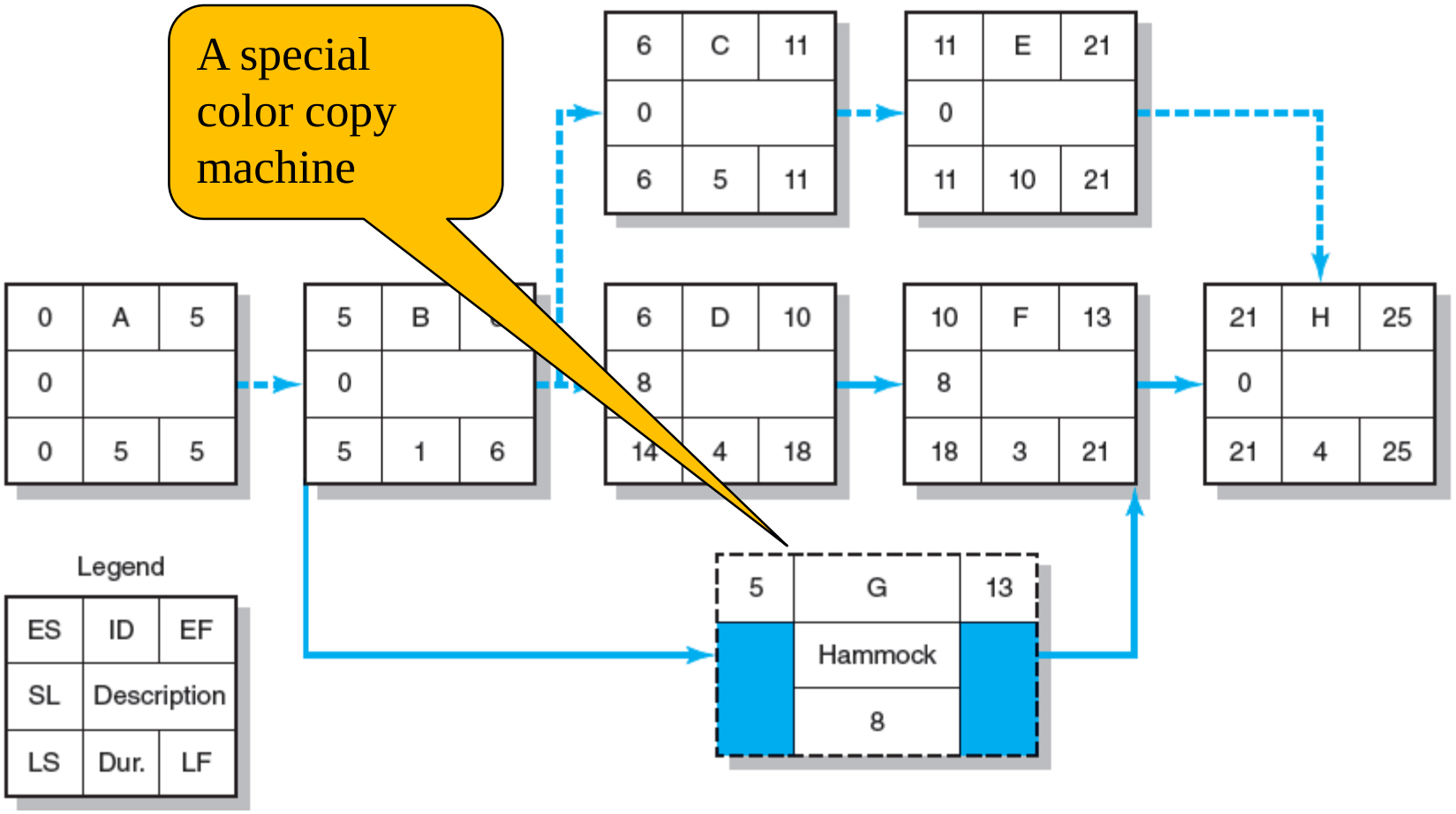
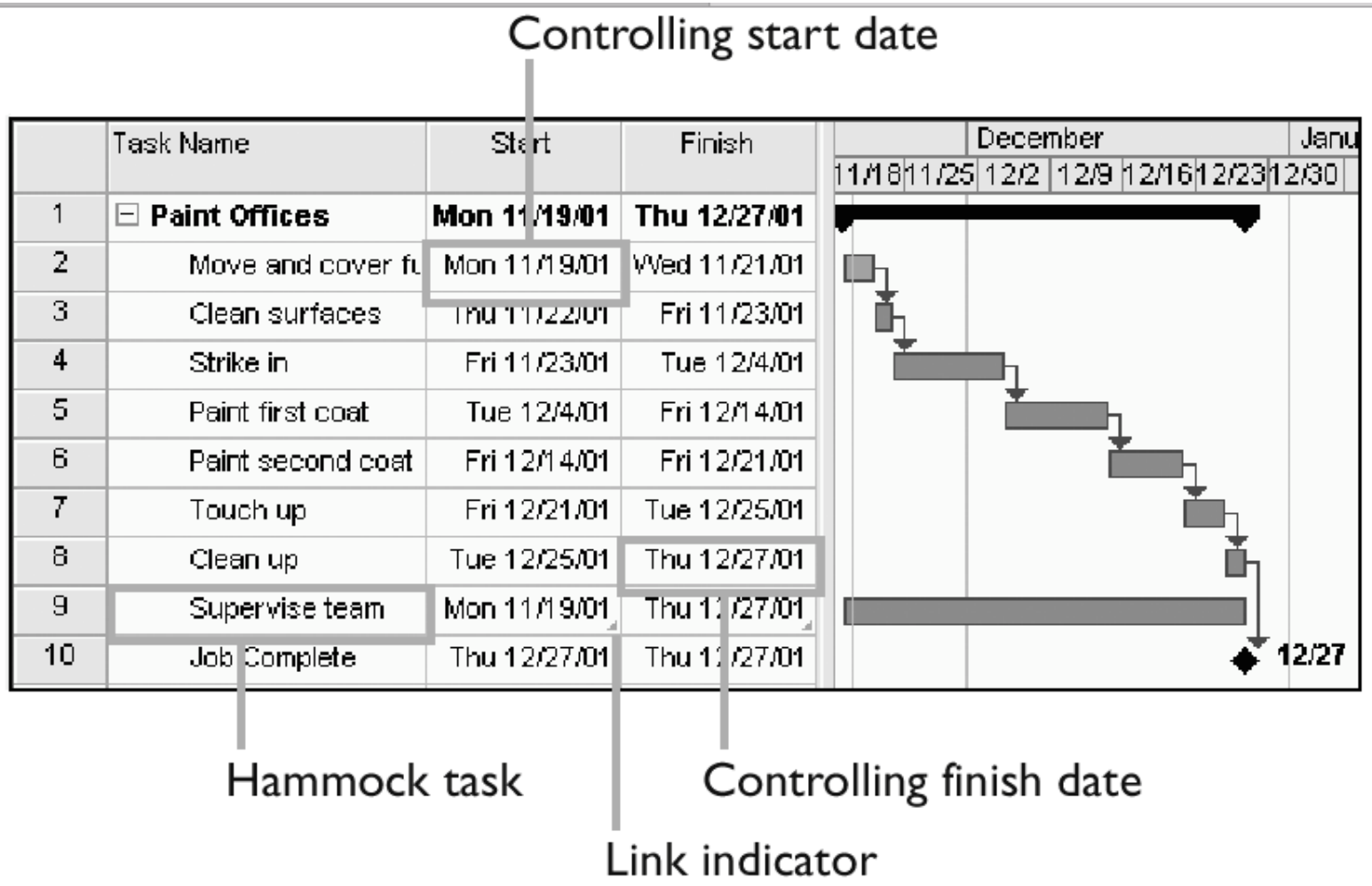


FIGURE 6.21

Hammock Activity



Next Week

- We will look at:
  - Risk Management
  - Contingency Funds