GSOE9820 – Engineering Project Management

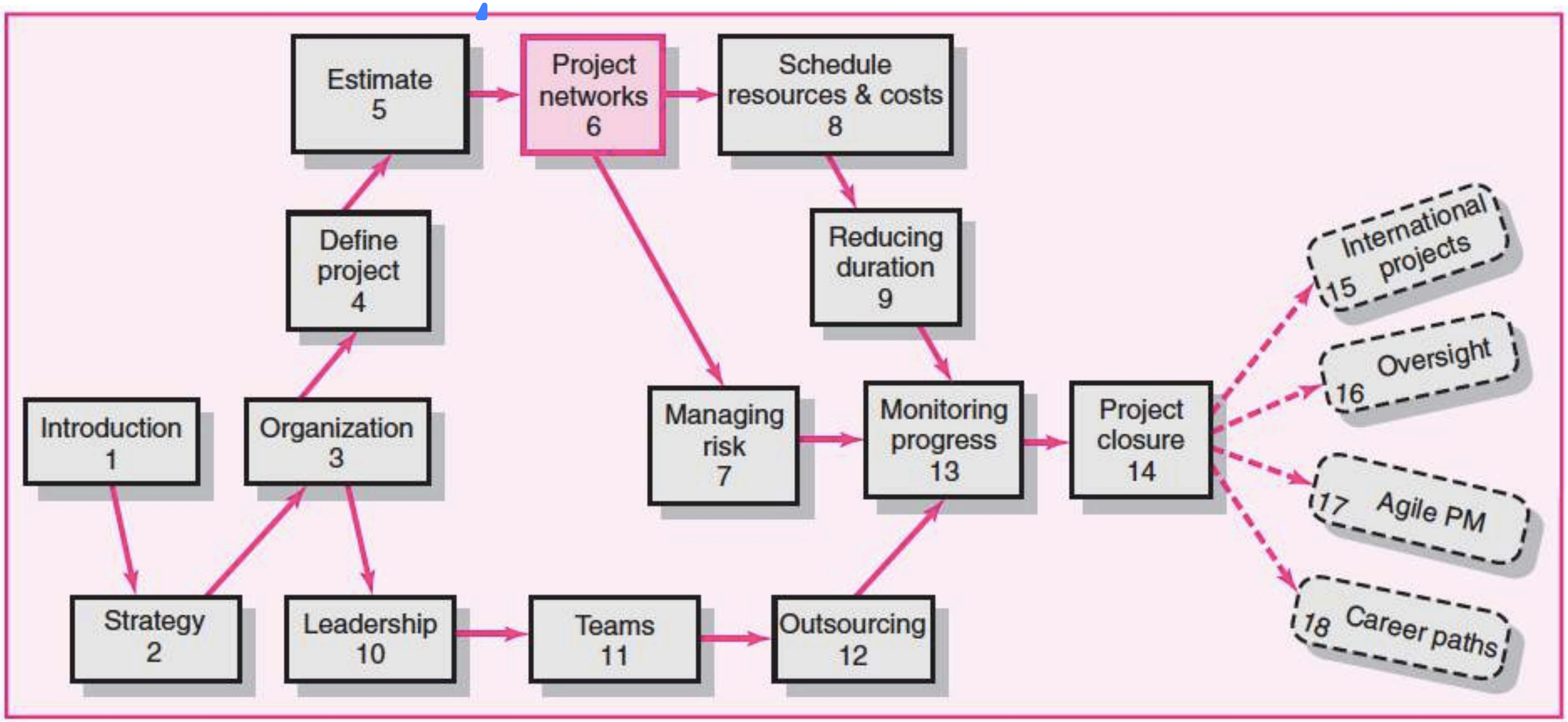
Corey Martin

UNSW AUSTRALIA

Never Stand Still | Faculty of Engineering | School of Mechanical and Manufacturing Engineering

Week 6
Developing a project network

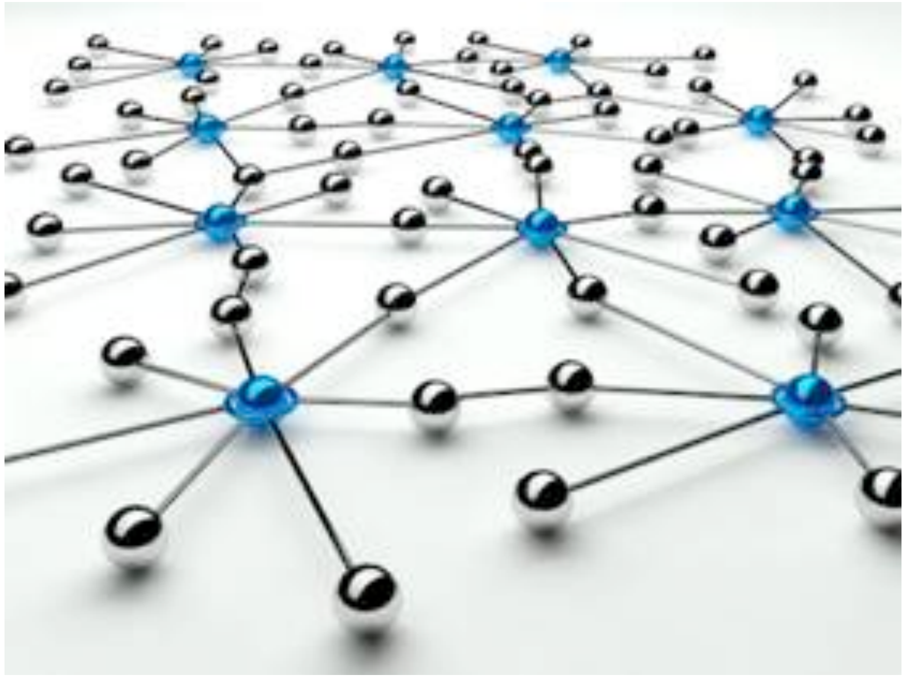
Course Roadmap



Reference: Gray, C & Larson, E, Project Management, 5th Ed. McGraw-Hill

The project network

Is a flow chart that graphically depicts the sequence, interdependencies and start and finish times of the project job plan of activities.



Benefits of developing the project network

- Provides the basis for scheduling labour 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;
- Help managers get and stay on plan.

Time elements of a project

Activity

- Is some action which requires time

Event

- It does not consume time.
- Is a point in time when an activity is started or completed.
- May also be known as a “milestone”

Project network approaches

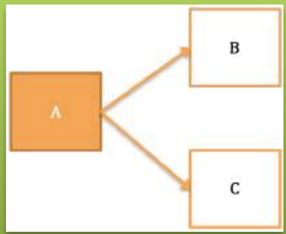
Activity-On-Node (AON)

- Uses a node to depict an activity

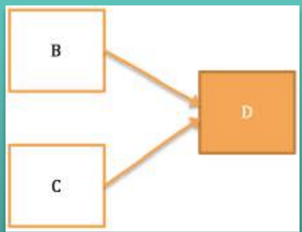
Activity-On-Arrow (AOA)

- Uses an arrow to depict an activity.

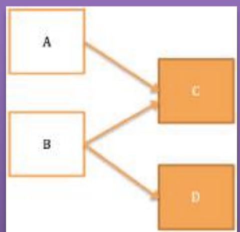
Types of activities



Burst - an activity that has more than one activity immediately following it



Merge - an activity that has two or more preceding activities on which it depends



Parallel - activities that can occur independently and, if desired, not at the same time.

Project network work flow

Path

– a sequence of connected, dependent activities.

Critical path

– the **longest path** through the activity network that allows for the completion of all activities;

– the **shortest expected time** in which the entire project can be completed.

Basic rules for developing a project network

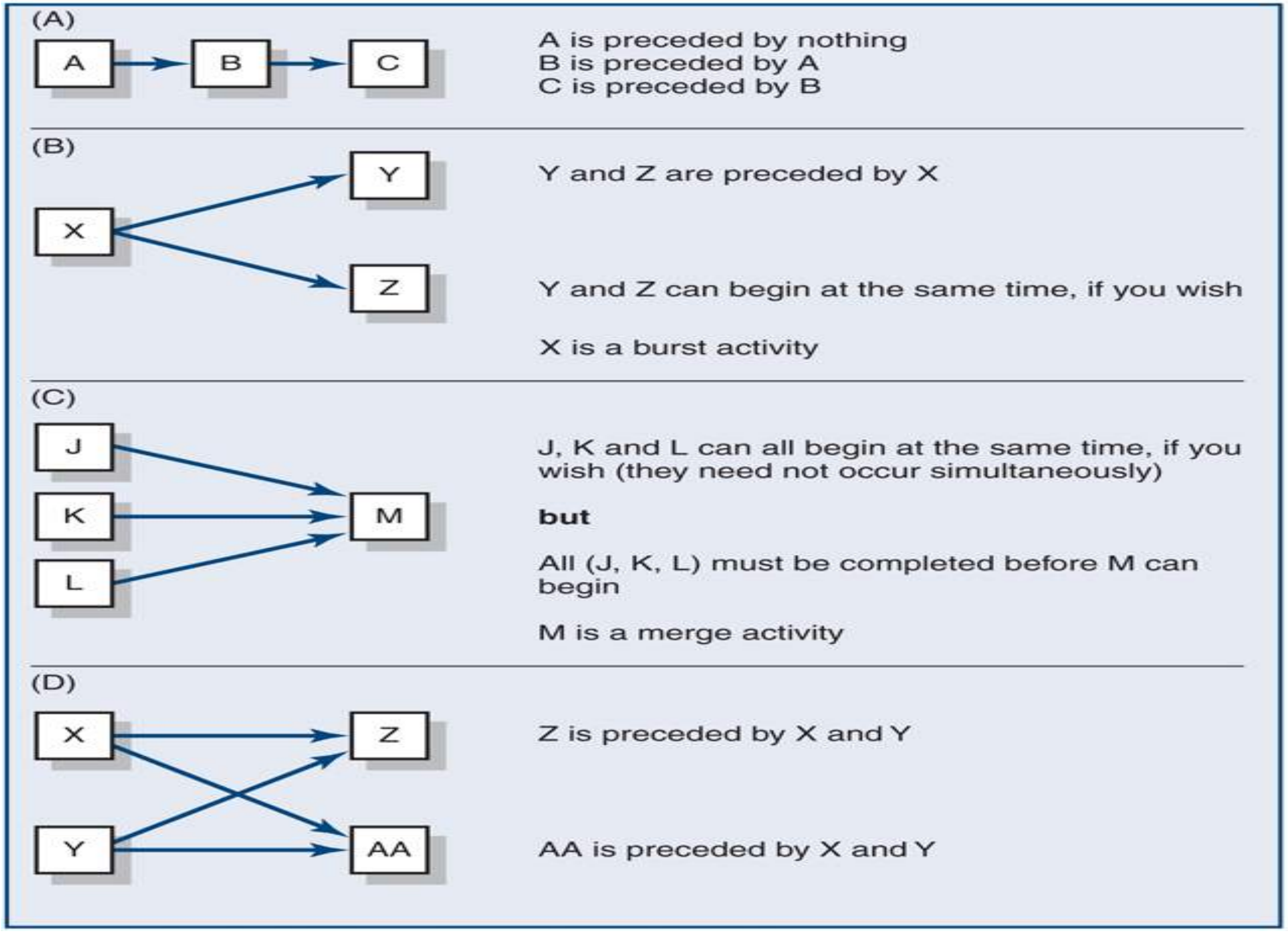
- Networks flow left to right
- An activity cannot begin until all preceding connected activities are complete
- Arrows on networks indicate precedence and flow
- Each activity should have a unique identification number
- An activity identification number must be larger than that of any preceding activities
- Looping is not allowed
- Conditional statements are not allowed
- Use common start and stop nodes

Activity-on-node fundamentals

- Activity-on-node (AON) or precedence diagram method.
- An activity is represented by a ***node*** (box).
- The dependencies are represented by ***arrows*** to indicate relationship and sequence.
- ***Predecessor*** activities are ones to be completed before.
- ***Successor*** activities are ones to be completed after.

Activity-on-node networks

Figure 6.2 ACTIVITY-ON-NODE NETWORK FUNDAMENTALS



Recap

Project Definition (WBS)

- identifies all the work elements involved in a project.

Project Network

- Places the activities in the right sequence

Network computation process

Forward pass

- 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

- How late can the activity start? (late start—LS)
- How late can the activity finish? (late finish—LF)

Key times of an activity

Term	Acronym	Description	Formula
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

Project network activity legend

Early Start (ES)	Activity identifier	Early Finish (EF)
Start Slack (SL-Start)	Description	Finish Slack (SL-Finish)
Late Start (LS)	Duration	Late Finish (LF)

Forward pass computation process

1. Add activity times along each path in the network (ES + Duration = EF).
2. Carry the early finish (EF) to the next activity where it becomes its early start (ES) ……. ***unless***
3. the succeeding activity is a merge activity, in which case select the largest EF of all preceding activities

Backward pass computation process

1. Subtract activity times along each path in the network (LF – Duration = LS).
2. Carry the late start (LS) to the next activity where it becomes its late finish (LF) ……. ***unless***
3. the succeeding activity is a burst activity, in which case select the smallest LF of all preceding activities.

Slack / Float

- Is the amount of time that a task in a project network can be delayed without causing a delay.
- Can be used to balance the schedule
- Allows flexibility in scheduling scarce resources.

Types of Slack

Total slack	Free slack
• Shared by activities along a path • Affects project completion date	• Owned by the activity • Affects subsequent tasks

Determining total slack (or float)

- The amount of time an activity can be delayed and not delay the overall project
- The amount of time an activity can exceed its early finish date without affecting the project end date or an imposed completion date

Total Start Slack = LS – ES, or

Total Finish Slack = LF – EF

- Total slack can change as the project progresses

Total Slack values

Total Slack value	Interpretation
TS > 0	Activity delay is possible without delaying the project completion
TS = 0	**Critical situation.** Any delay in zero float activities will cause the project completion date to slip. Identifies the critical path.
TS < 0	You are behind schedule. You can get negative slack if you put a constraint on your completion date

Ownership of total slack

Although total slack is calculated for each activity, it is NOT owned by that activity

Total slack is **shared by ALL** activities in a path

E.g. If the first activity in the path uses up the total slack, the total slack for the remaining activities becomes zero.

Determining Free slack (or float)

- Is owned by the activity
- Can never be negative
- Is the amount of time that an activity can be delayed without delaying the early start (ES) of any successive activities

$$\mathbf{FS = ES_{\text{(of successor activity)}} - EF_{\text{(of current activity)}}}$$

Sensitivity

Is the likelihood the original critical path(s) will change once the project is initiated.

Typical rules of thumb:

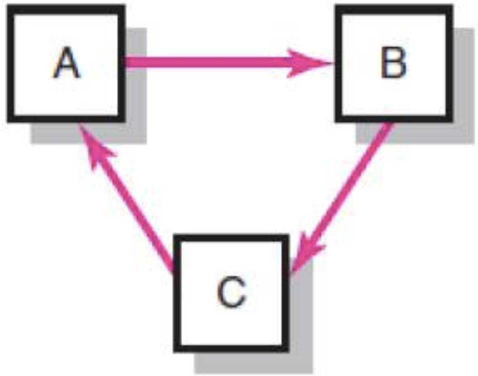
Very little slack and lots of critical paths

→ MORE sensitive

Lots of slack and only one critical path

→LESS sensitive

Practical considerations



Network logic errors
- Looping
- Conditional statements (e.g. if-then) are invalid

Activity numbering
- Each activity to have an unique identifier
- Number in ascending order
- Leave gaps to add missing or new activities

Use of computers to develop networks
- E.g. Gantt charts

Calendar dates
- Assign actual dates (include non-workdays)

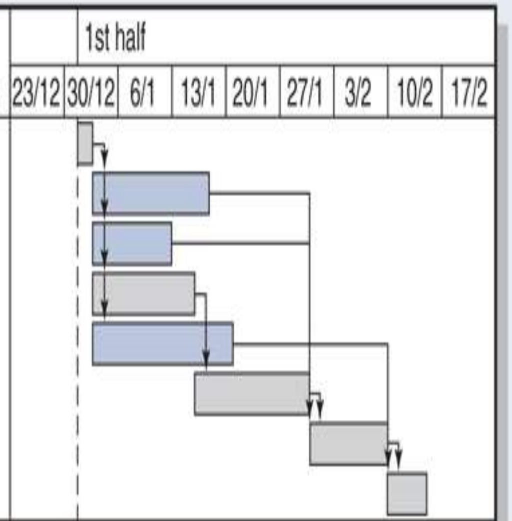
Multiple starts and multiple projects
- Use a common start and finish event to ensure project has a clear beginning and end

Note: ⇧Detail → ⇧Accuracy → ⇧**Overhead/Costs**

Example: Gantt Chart of an Air Control Project

Figure 6.12 AIR CONTROL PROJECT—GANTT CHART

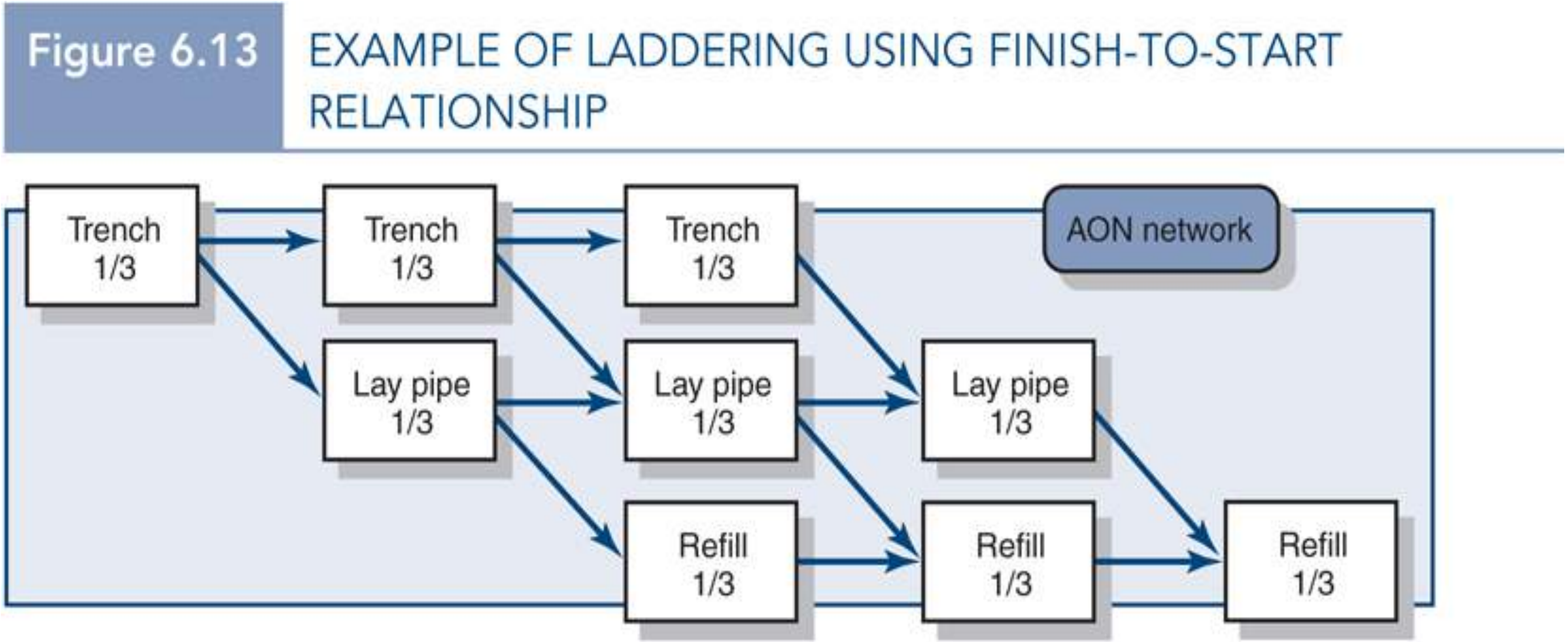
ID	Duration	Task name	Start	Finish	Late start	Late finish	Free slack	Total slack
1	2 days	Order review	Tue 1/1	Wed 2/1	Tue 1/1	Wed 2/1	0 days	0 days
2	15 days	Order vendor parts	Thu 3/1	Thu 17/1	Wed 16/1	Wed 30/1	13 days	13 days
3	10 days	Produce other standard parts	Thu 3/1	Sat 12/1	Mon 21/1	Wed 30/1	18 days	18 days
4	13 days	Design custom parts	Thu 3/1	Tue 15/1	Thu 3/1	Tue 15/1	0 days	0 days
5	18 days	Software development	Thu 3/1	Sun 20/1	Wed 23/1	Sat 9/2	20 days	20 days
6	15 days	Manufacture custom hardware	Wed 16/1	Wed 30/1	Wed 16/1	Wed 30/1	0 days	0 days
7	10 days	Assemble	Thu 31/1	Sat 9/2	Thu 31/1	Sat 9/2	0 days	0 days
8	5 days	Test	Sun 10/2	Thu 14/2	Sun 10/2	Thu 14/2	0 days	0 days



Extended network techniques
(a dose of reality)

- Laddering
- Lags
- Hammock

Example: Laddering using finish-to-start relationship



Lag

Is the minimum amount of time a dependent activity must be delayed to begin or end

The relationship between start and/or finish of a project activity and the start and/or finish of another activity

Most common lag relationships

- finish-to-start
- finish-to-finish
- start-to-start
- start-to-finish

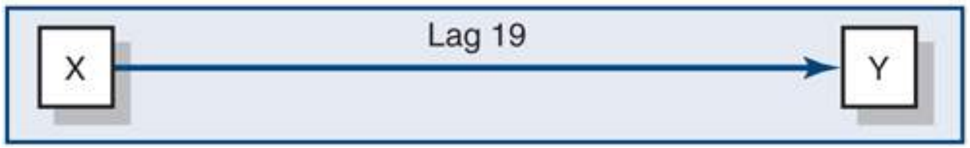
Finish-to-start lag relationship

- Finish-to-start lags are often used when ordering materials. E.g. 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

Example:

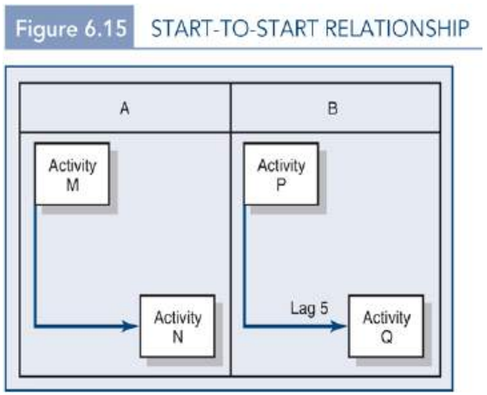
Figure 6.14 FINISH-TO-START RELATIONSHIP



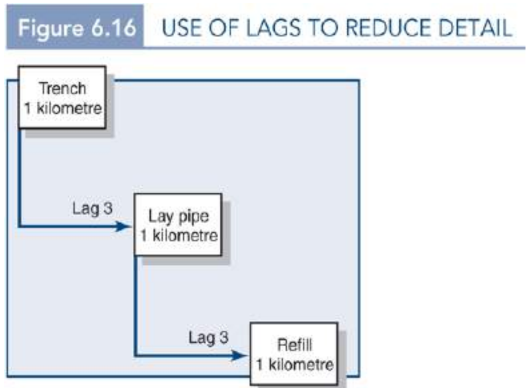
Start-to-start lag relationship

The start of an activity depends on the start of another activity.

Examples:



Can reduce network detail and project delays



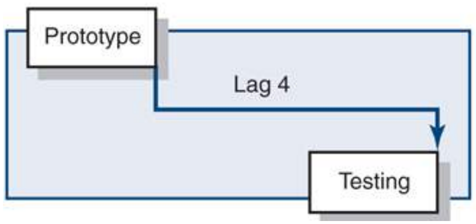
Often used in concurrent engineering

Finish-to-finish lag relationship

- The finish of one activity depends on the finish of another activity.

Example: Testing cannot be completed any earlier than four days after the prototype is complete. It cannot be finish-to-start because testing of subcomponents does not qualify as complete system testing.

Figure 6.18 FINISH-TO-FINISH RELATIONSHIP

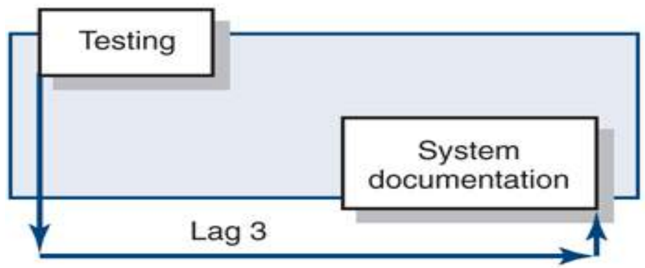


Start-to-finish lag relationship

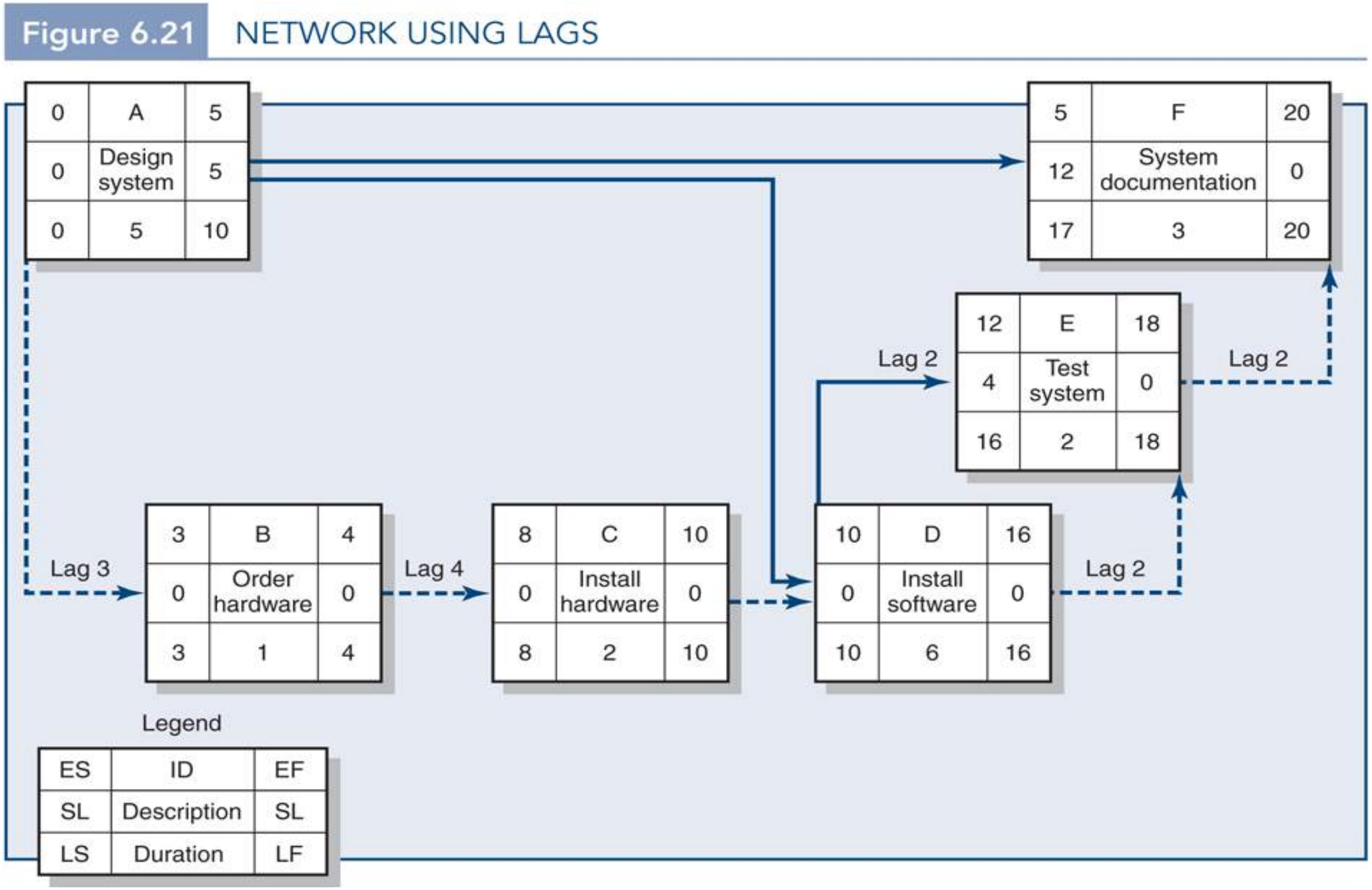
- The finish of an activity depends on the start of another activity.

Example: The system documentation cannot end until three days after testing has started. Here all the relevant information is produced after three days of testing.

Figure 6.19 START-TO-FINISH RELATIONSHIP



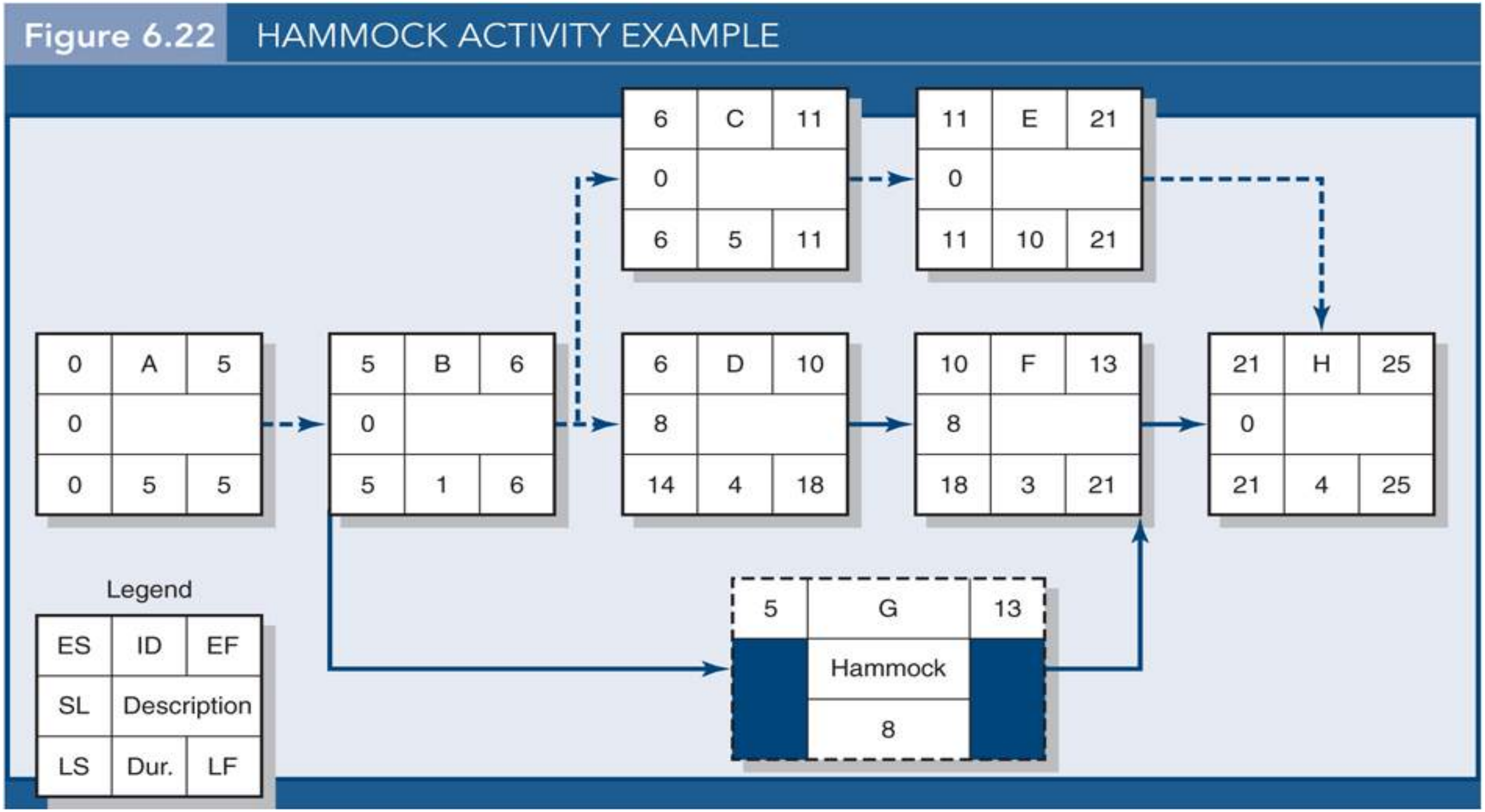
Example: Network using lags



Hammock activities

- An activity that spans over a segment of a project.
- 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.

Example: Hammock activity



Applying statistics

PERT

- A project network gives a ‘*point estimate*’ on duration of each activity.
- PERT focus on the likelihood that the project will be completed on ‘time’ and ‘within budget’.
- PERT assumes a statistical distribution for each activity’s duration.
- Using a pre-defined distribution, the probability that the project will be completed in a certain time frame can be determined.

Program Evaluation and Review Technique (PERT)

Originally developed in 1958 for US Navy Polaris submarine project

Assumes that each activity duration has a range that follows a statistical distribution.

Each activity has duration can range from an optimistic time to a pessimistic time.

A weighted average for duration can be calculated

Two types of distributions are used:

- Beta distribution for activities (because usually work tends to stay behind once it gets behind)
- Normal distribution for projects

Is the sum of weighted average of the activities duration on the critical path.

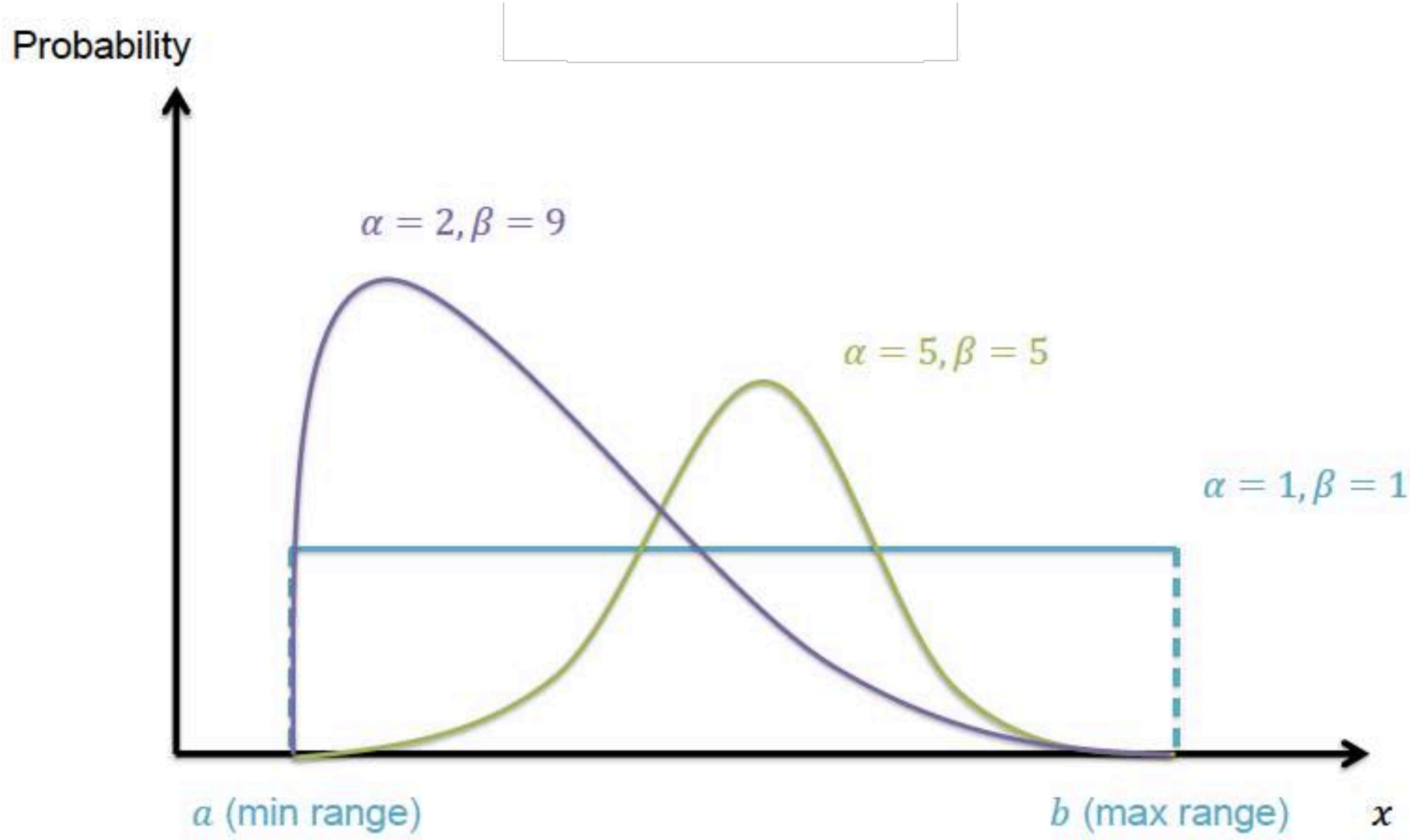
Properties of Beta Distribution

- PERT often uses the ‘beta distribution’ as the statistical distribution for each activity’s duration.
- The beta distribution can take different shapes – symmetrical/asymmetrical.
- The beta distribution is bounded by a minimum range and maximum range, this makes it very useful to model the duration of activities in a project network.

Normal Distribution Versus Beta Distribution

Distribution	Normal	Beta
Number of parameters	Two - μ (mean) and σ^2 (variance)	Four - α, β (shape parameters), a (min of range), b (max of range)
Symmetrical property	Always symmetrical about μ	Symmetrical if $\alpha = \beta$ Asymmetrical if $\alpha \neq \beta$
Mean	μ	$\mu = \frac{a + 4m + b}{6}$ m is the mode
Variance	σ^2	$\sigma^2 = \left(\frac{b-a}{6}\right)^2$

Beta Distribution Probability Density Function



Activity and Project frequency distributions