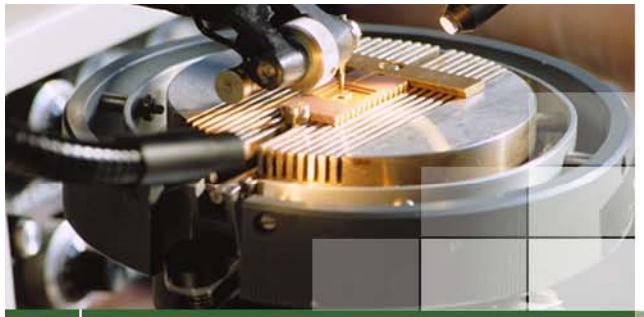


POLITECNICO DI MILANO

Dipartimento di
Elettronica e Informazione

Session 5

Scheduling

Emanuele Della Valle

http://home.dei.polimi.it/dellavalle

Lecturer: Dario Cerizza

Credits

- This slides are partially based on CEFRIEL's slides for PMI Certification and largely based on Prof. John Musser class notes on "Principles of Software Project Management"
- Original slides are available at http://www.projectreference.com/
- Reuse and republish permission was granted

Today

- Session 4 review
- Scheduling Fundamentals
- Scheduling Techniques
  - Network Diagrams
  - Bar Charts
- Schedule Optimization Techniques
- Mythical Man-Month

Session 4 Review

- WBS
- Estimation

WBS

- Types: Process, Product, Hybrid
- Formats: Outline or graphical organizational chart
- High-level WBS does not show dependencies or durations
- WBS becomes input to many things, esp. schedule
- What hurts most is what's missing

Estimation

- "The single most important task of a project: setting realistic expectations. Unrealistic expectations based on inaccurate estimates are the single largest cause of software failure." Futrell, Shafer, Shafer, "Quality Software Project Management"
- Session 4 continuation
  - http://www.emanueledellavalle.org/slides/P&MSP2009_05_WBS-Estimation-&-Scheduling.ppt
    - 55. Effort Estimation

Estimation

- History is your best ally
  - Especially when using Function Points, LOC (Lines of Code), ...
- Use multiple methods if possible
  - This reduces your risk
  - If using “experts”, use two
- Get buy-in
- Remember: it’s an iterative process!
- Know your “presentation” techniques

Estimation

- Bottom-up
  - More work to create but more accurate
  - Often with Expert Judgment at the task level
- Top-down
  - Used in the earliest phases
  - Usually as is the case with Analogy or Expert Judgment
- Analogy
  - Comparison with previous project: formal or informal
- Expert Judgment
  - Via staff members who will do the work
  - Most common technique along with analogy
  - Best if multiple ‘experts’ consulted

Estimation

- Parametric Methods
  - Know the trade-offs of: LOC & Function Points
- Function Points
  - Benefit: relatively independent of the technology used to develop the system
  - We will re-visit this briefly later in semester (when discussing “software metrics”)

Today

- Session 4 review
- **Scheduling Fundamentals**
- Scheduling Techniques
  - Network Diagrams
  - Bar Charts
- Schedule Optimization Techniques
- Mythical Man-Month

Planning, Estimating and Scheduling

- Initial Planning:
  - Why
    - SOW, Charter
  - What/How (partial/1st pass)
    - WBS
    - Other planning documents
      - Software Development Plan, Risk Mgmt., Cfg. Mgmt.
- Estimating
  - How much/How long
    - Size (quantity/complexity) and Effort (duration)
    - It's an iterative process
- Scheduling
  - In which order
    - Precedence, concurrences, lag & lead times, slack & float, …
    - It's an iterative process

What Scheduling is

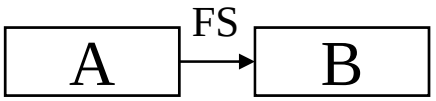
- Once tasks (from the WBS) and size/effort (from estimation) are known: then schedule
  - Define the start and end time of each activity
- Objective: trade-off of six objectives
  - Primary objectives
    1. Best time
    2. Least cost
    3. Least risk
  - Secondary objectives
    4. Evaluation of schedule alternatives
    5. Effective use of resources
    6. Communications

Terminology: Precedence and Concurrence

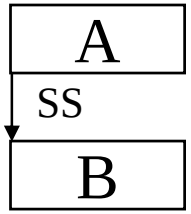
- Precedence:
  - A task that must occur before another is said to have precedence of the other
- Concurrence:
  - Concurrent tasks are those that can occur at the same time (in parallel)

Terminology: Four task relationship types

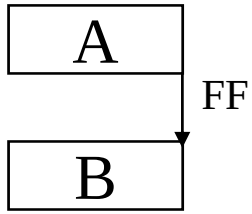
- *Finish-to-start*: "A f-to-s B": the initiation of the successor activity depends upon completion of the predecessor activity



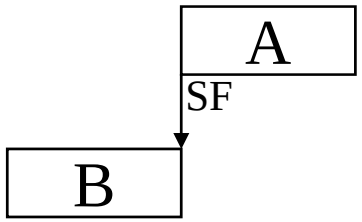
- *Start-to-start*: "A s-to-s B": the initiation of successor activity depends upon initiation of the predecessor activity



- *Finish-to-finish*: "A f-to-f B": the completion of the successor activity depends upon completion of the predecessor activity

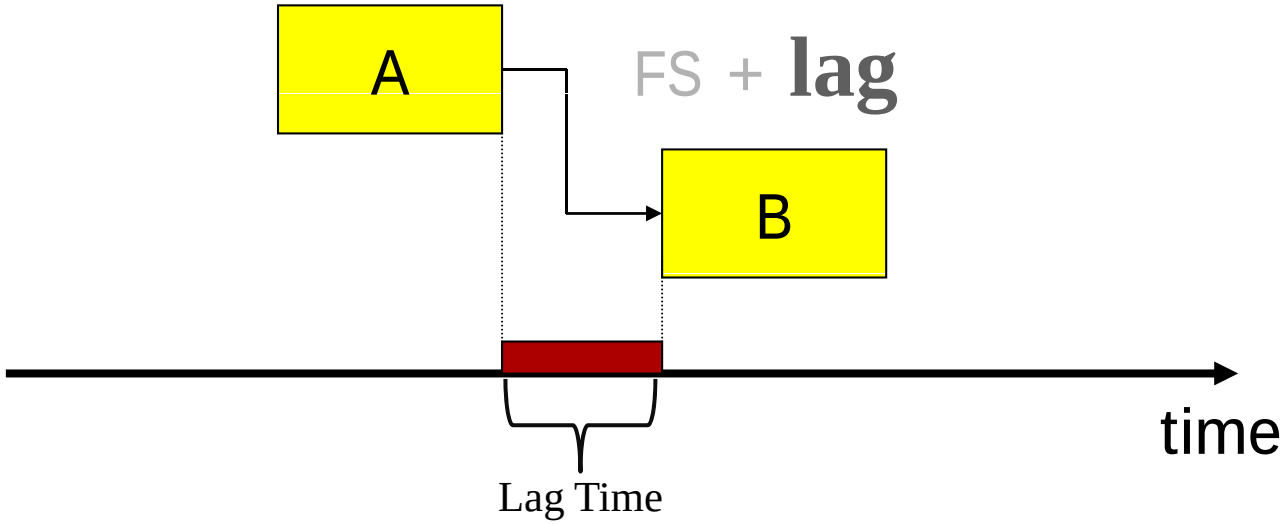


- *Start-to-finish*: "A s-to-f B": the completion of the successor activity depends upon initiation of the predecessor activity



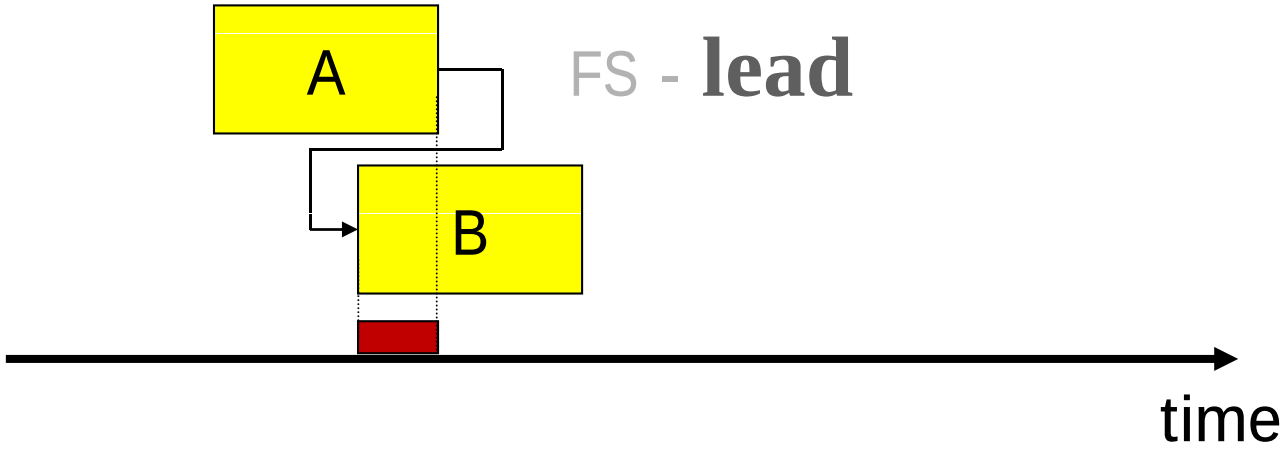
Terminology: Lag Time

- It means a delay between tasks in sequence
- Example: if "A f-to-s B" and lag is equal to 10, B can start only 10 days after A end



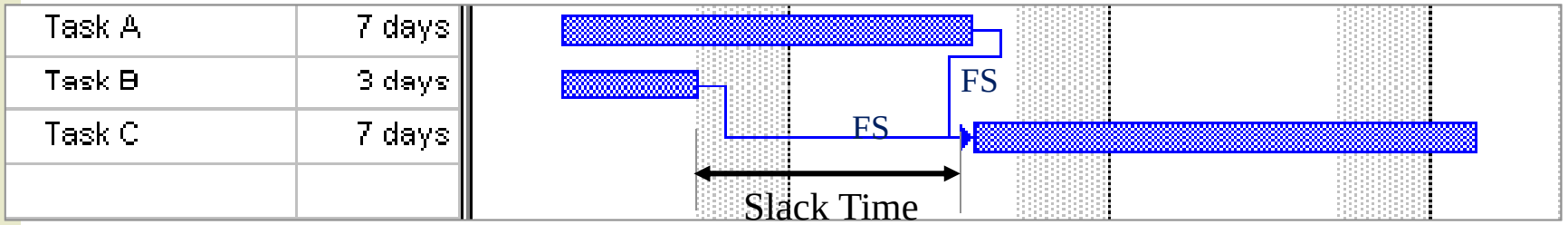
Terminology: Lead Time

- It means an advance between tasks in sequence
- Example: if "A f-to-s B" and lead is equal to 10, B can start 10 days before A end



Terminology: Slack & Float

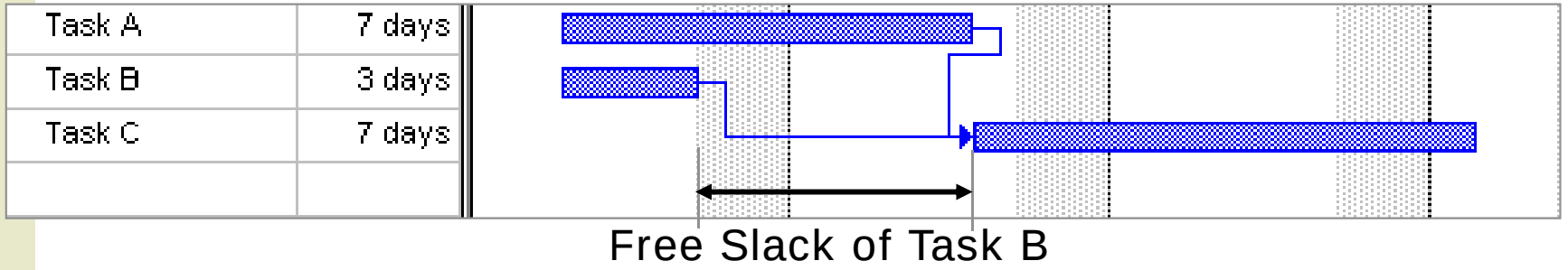
- When then schedule contains several tasks, it may happen that there is some “free” time between tasks



- Slack & Float: synonymous terms
  - We’ll use Slack
- Two types of slack time: Free Slack and Total Slack

Terminology: Free Slack

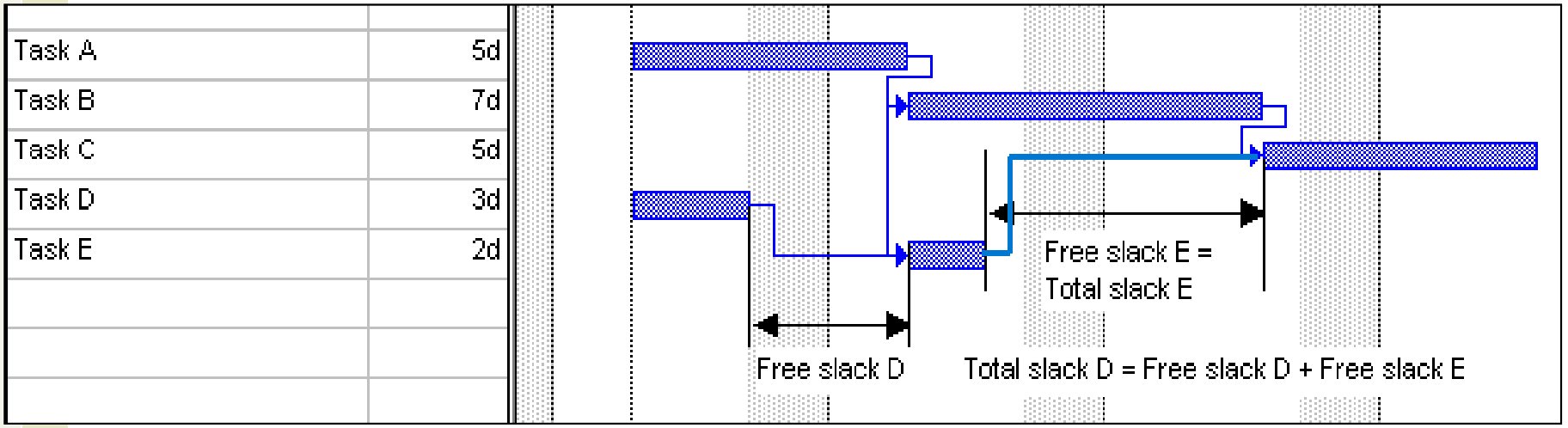
- Free Slack: Maximum delay of a task without causing a delay for any downstream task
- Example
  - A and B can be activated as soon as possible
  - Relationships
    - "A finish-to-start C"
    - "B finish-to-start C"



- Free Slack of Task B
  = Late Finish of B – Early Finish of B
  = Late Start of B – Early Start of B

Terminology: Total Slack

- Total Slack: Maximum delay of a task without causing a delay for the total project (it can cause delays to other tasks)



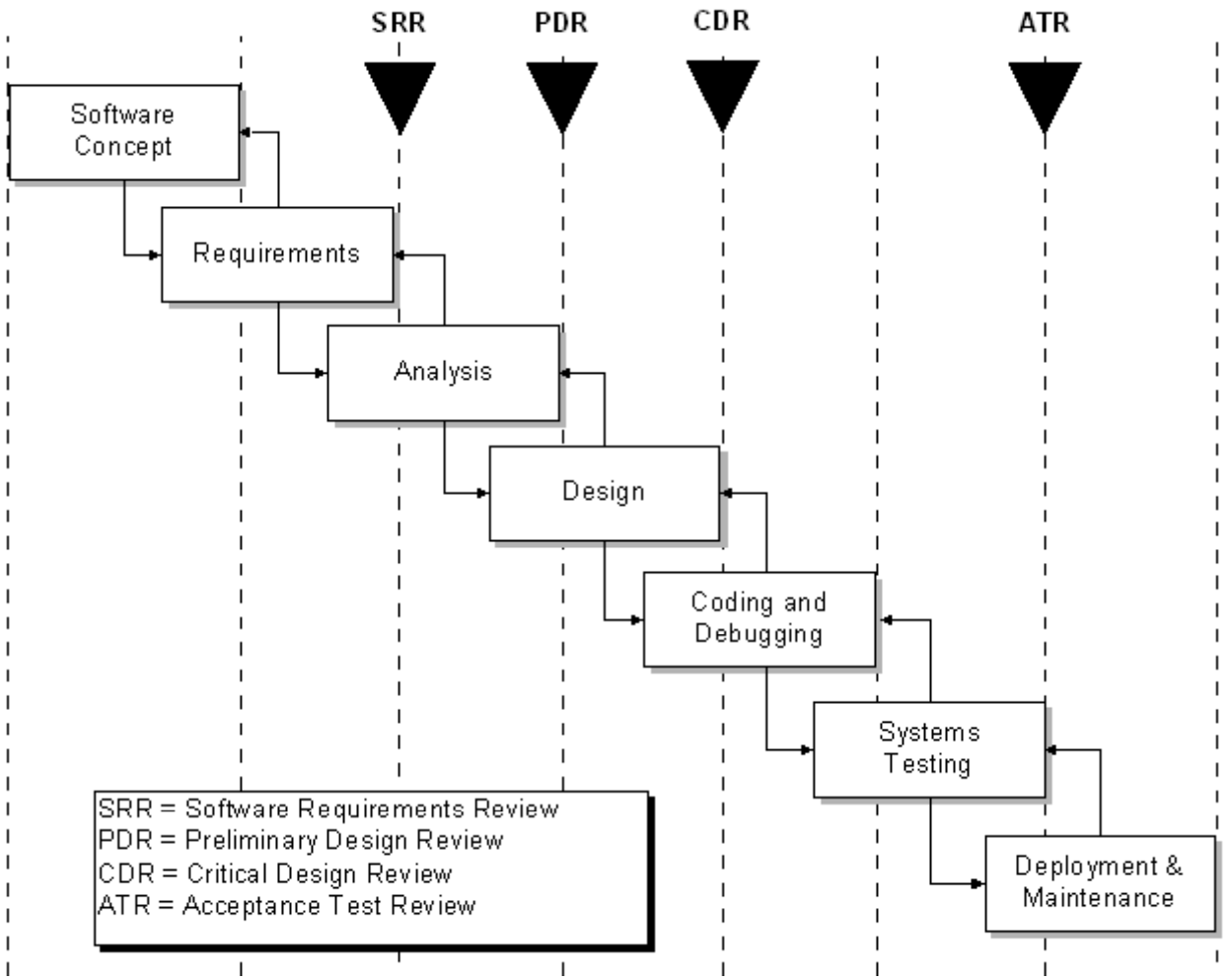
- Normally, if not specified, **slack** stands for **total slack**

Terminology: Milestones

- Identify crucial points in your schedule

- Have a duration of zero
- Shown as inverted triangle or a diamond
- Often used at “review” or “delivery” times
  - Or at end or beginning of phases
  - Ex: Software Requirements Review (SRR)
  - Ex: User Sign-off
- Can be tied to contract terms

Terminology: Milestones examples



Terminology: Four Dependency Types

1. Mandatory Dependencies
   - “Hard logic” dependencies
   - Nature of the work dictates an ordering
   - Ex: UI design precedes UI implementation
   - Ex: Coding has to precede testing
2. Discretionary Dependencies
   - “Soft logic” dependencies
   - Determined by the project management team
   - Process-driven
   - Ex: Discretionary order of creating certain modules

- NOTE: substantial process innovation often take place when “hard logic” dependencies are shown to be wrong
  - Ex: Test first approaches

Terminology: Four Task Dependency Types

3. External Dependencies
   - Outside of the project itself
   - Ex: Release of 3rd party product; contract signoff
   - Ex: stakeholders, suppliers, year end

4. Resource Dependencies
   - Two task rely on the same resource
   - Ex: You have only one DBA but multiple DB tasks

Today

- Session 4 review
- Scheduling Fundamentals
- **Scheduling Techniques**
  - Network Diagrams
  - Bar Charts
- Schedule Optimization Techniques
- Mythical Man-Month

Scheduling Techniques

- Network Diagrams
  - CPM
  - PERT
- Bar Charts
  - Gantt Chart
  - Milestone Chart

Network Diagrams

- Developed in the 1950's
- A graphical representation of the tasks necessary to complete a project
- Visualizes the flow of tasks & relationships

CPM & PERT

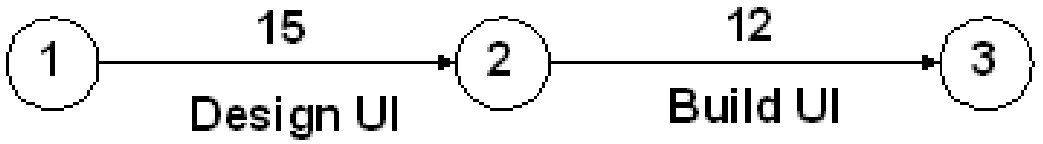
- CPM
  - Critical Path Method
- PERT
  - Program Evaluation and Review Technique
- Sometimes treated synonymously
- All are models using network diagrams

Two classic formats

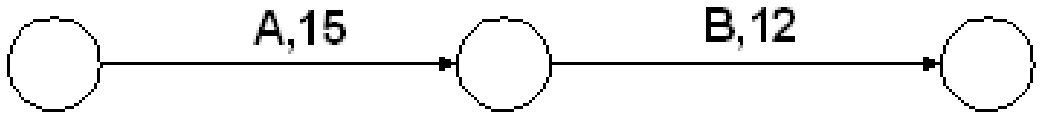
- Two classic formats
  - AOA: Activity on Arrow
  - AON: Activity on Node
- Each activity labeled with
  - Identifier (usually a letter/code)
  - Duration (in standard unit like days)
- There are other variations of labeling
- There is 1 start & 1 end event
- Time goes from left to right

Formats

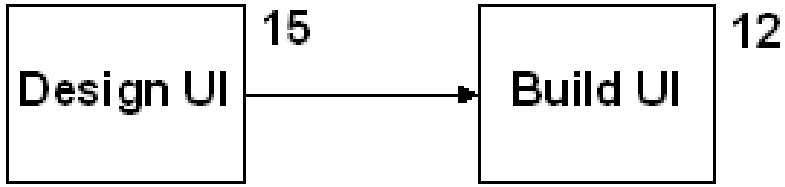
Activity on Arrow (AOA)



or



Activity on Node (AON)



or

Early Start	Duration	Early Finish
Task Name		
Late Start	Slack	Late Finish

Formats

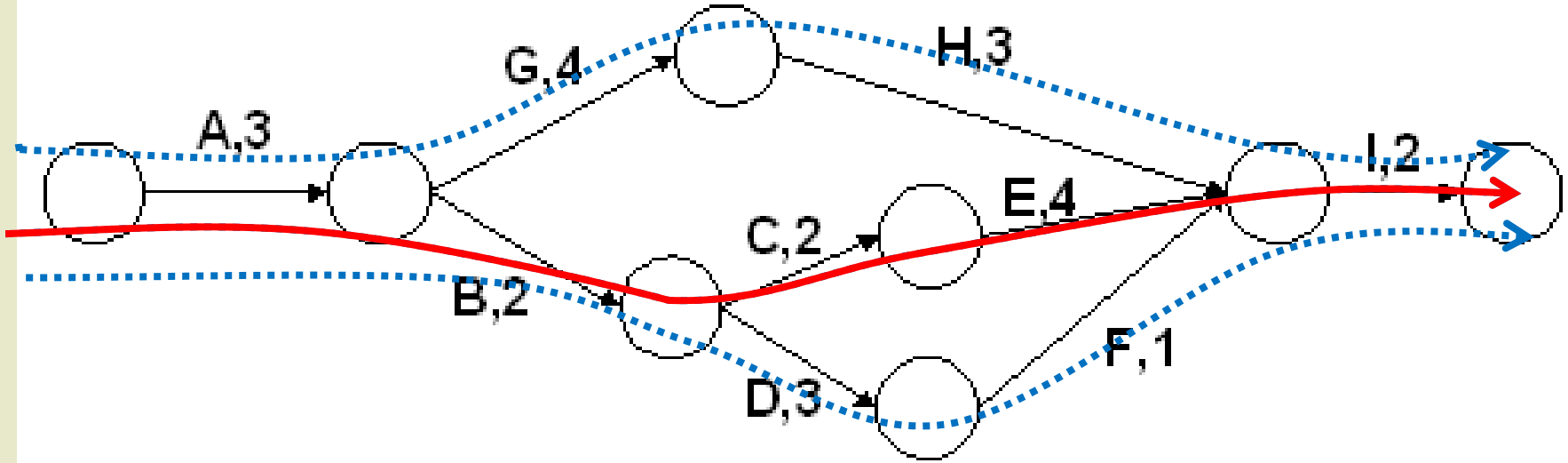
- AOA
  - Activities on Arrows
  - Circles representing Events
    - Such as ‘start’ or ‘end’ of a given task
  - Lines representing Tasks
    - Thing being done ‘Build UI’
  - a.k.a. Arrow Diagramming Method (ADM)

- AON
  - Activities on Nodes
    - Nodes can be circles or rectangles (usually latter)
    - Task information written on node
  - Arrows are dependencies between tasks
  - a.k.a. Precedence Diagramming Method (PDM)

Critical Path

- "The specific set of sequential tasks upon which the project completion date depends"
- All the tasks on the critical path have Free and Total Slack time = 0
- All projects have a Critical Path
- Accelerating non-critical tasks do not directly shorten the schedule

Critical Path: an Example



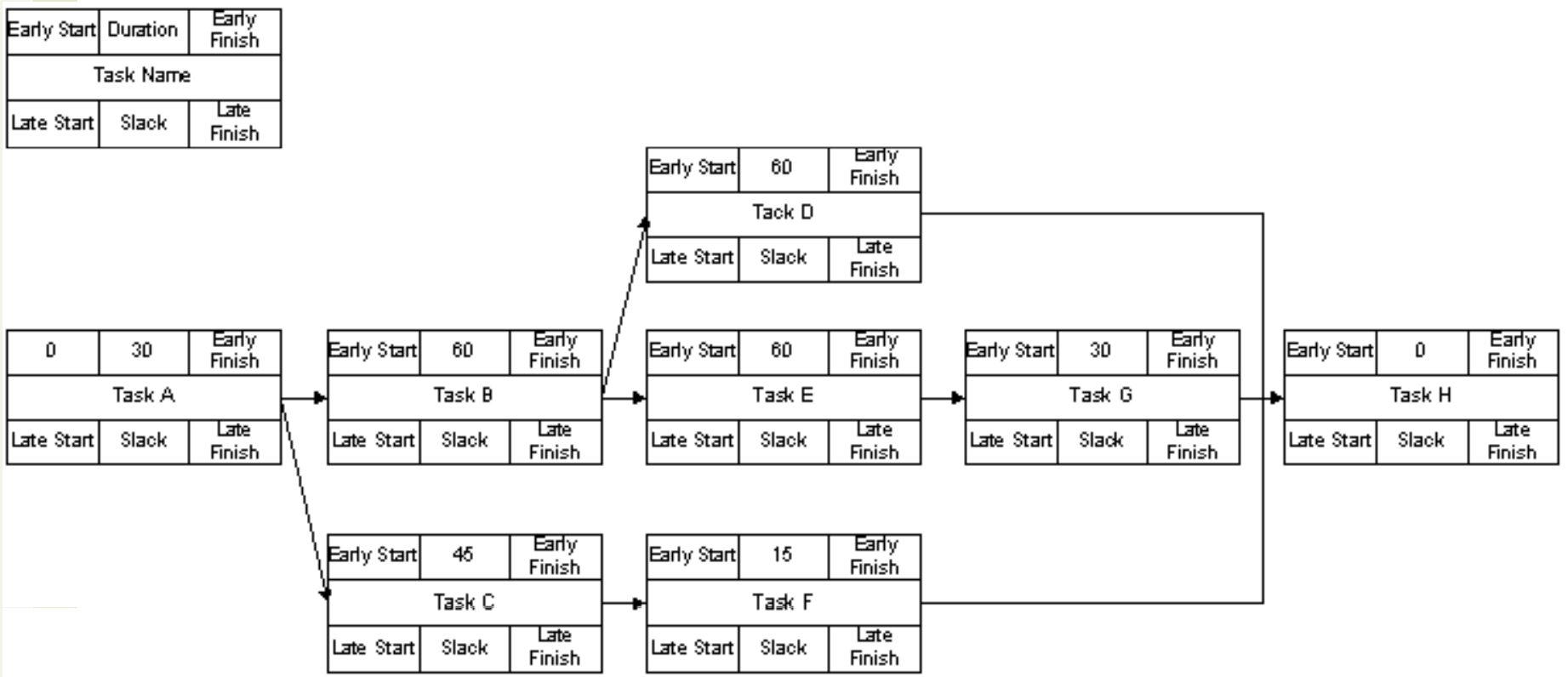
Critical Path Method (CPM)

- The process for determining and optimizing the critical path
- Non-Critical Path tasks can start earlier or later without impacting completion date
- Note: Critical Path may change to another as you shorten the current
- Should be done in conjunction with the project manager & the functional manager
- Based upon a 2-passes approach
  - Forward and Backward

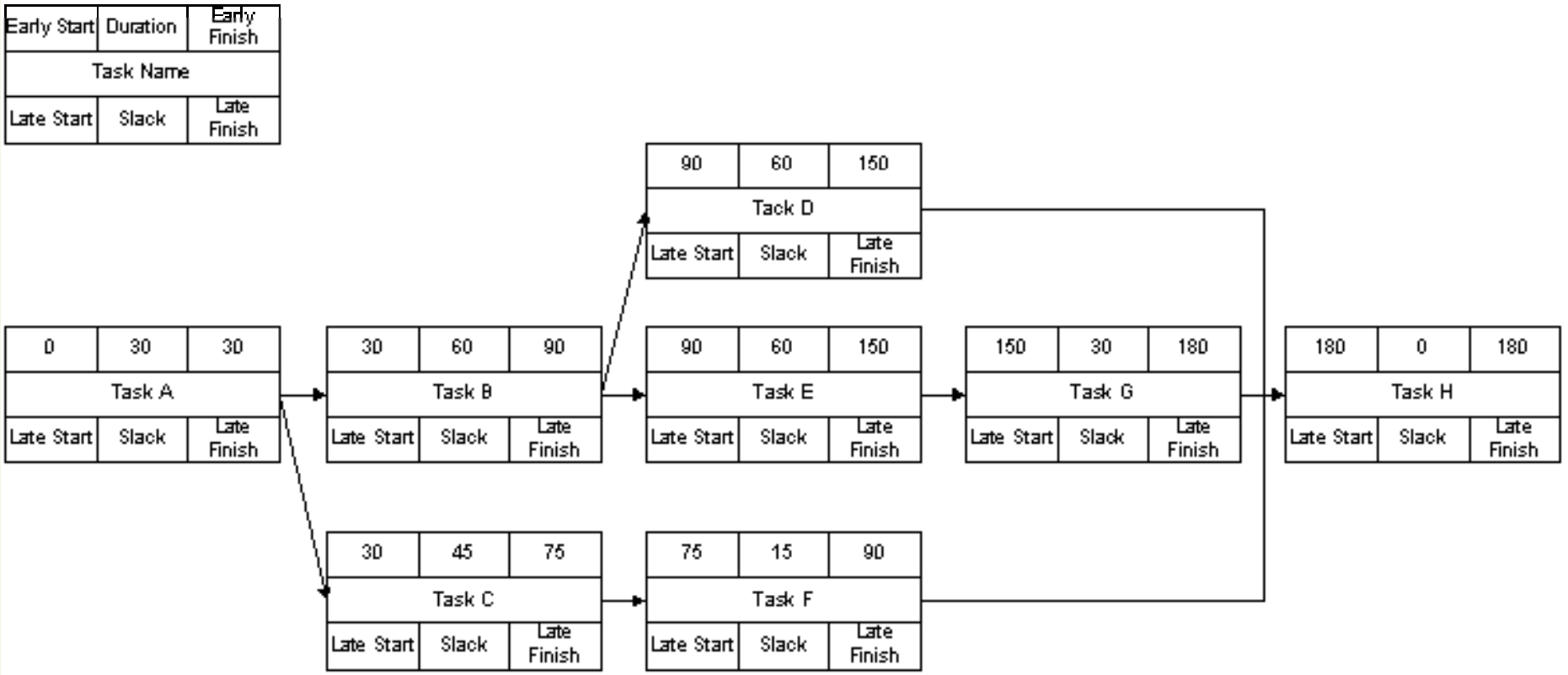
CPM: Forward Pass

- To determine early start (ES) and early finish (EF) times for each task
- Work from left to right
- Adding times to each node and each path
- Rule: when several tasks converge, the ES for the next task is the **largest** of preceding EF times

CPM: Example: Forward (part 1)



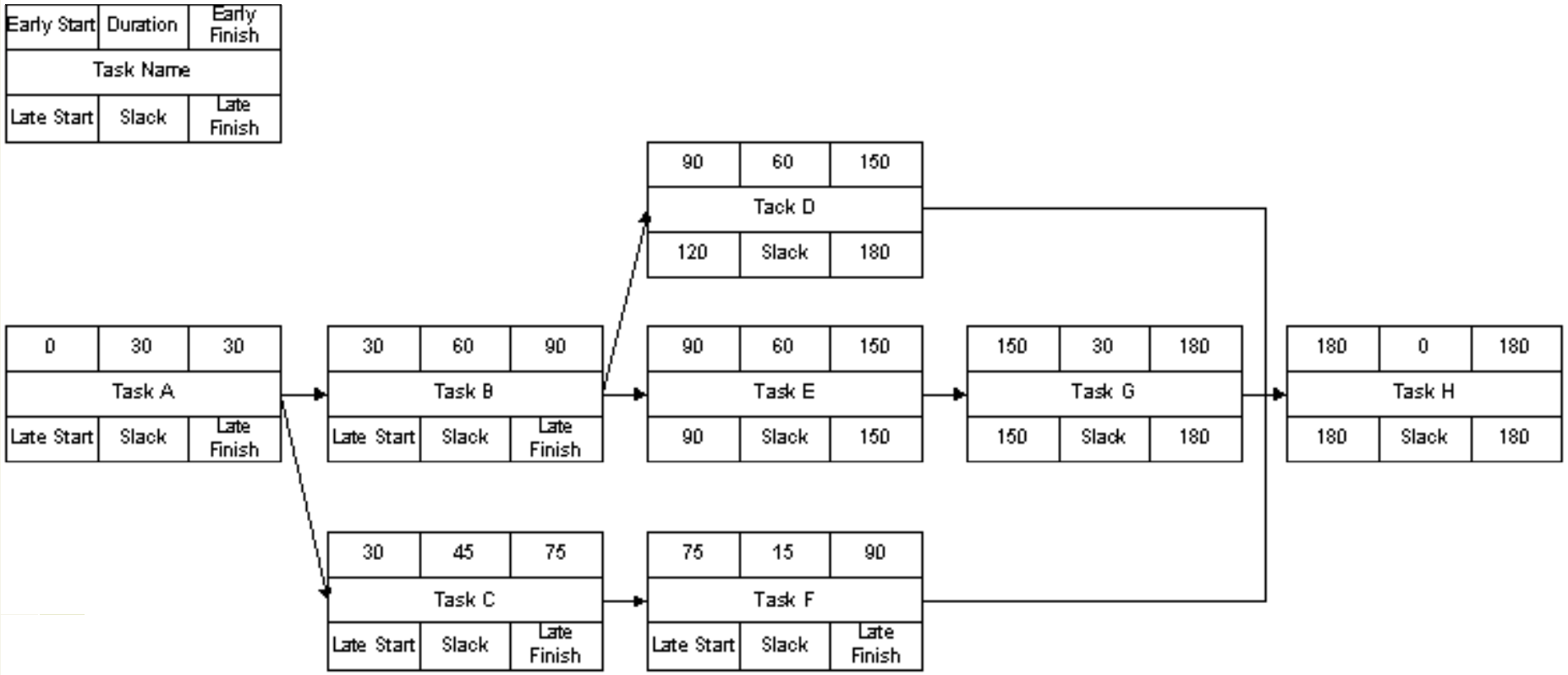
CPM: Example: Forward (part 2)



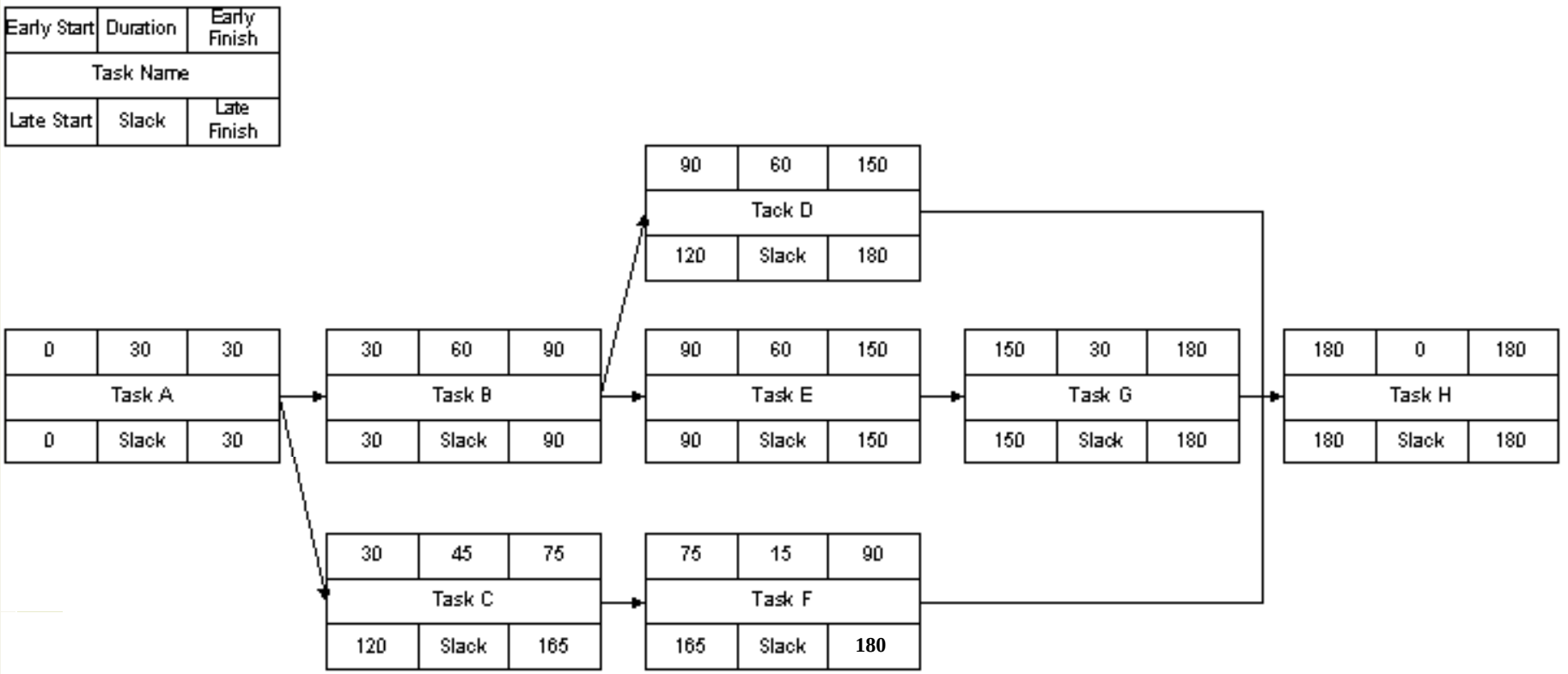
CPM: Backward Pass

- To determine the last finish and last start times
- Start at the end node and move backward left
- Subtract duration from connecting node's earliest start time
- Rule: when several tasks converge, the last finish for the previous task is the smallest of following last start times

CPM: Example: Backward (part 1)



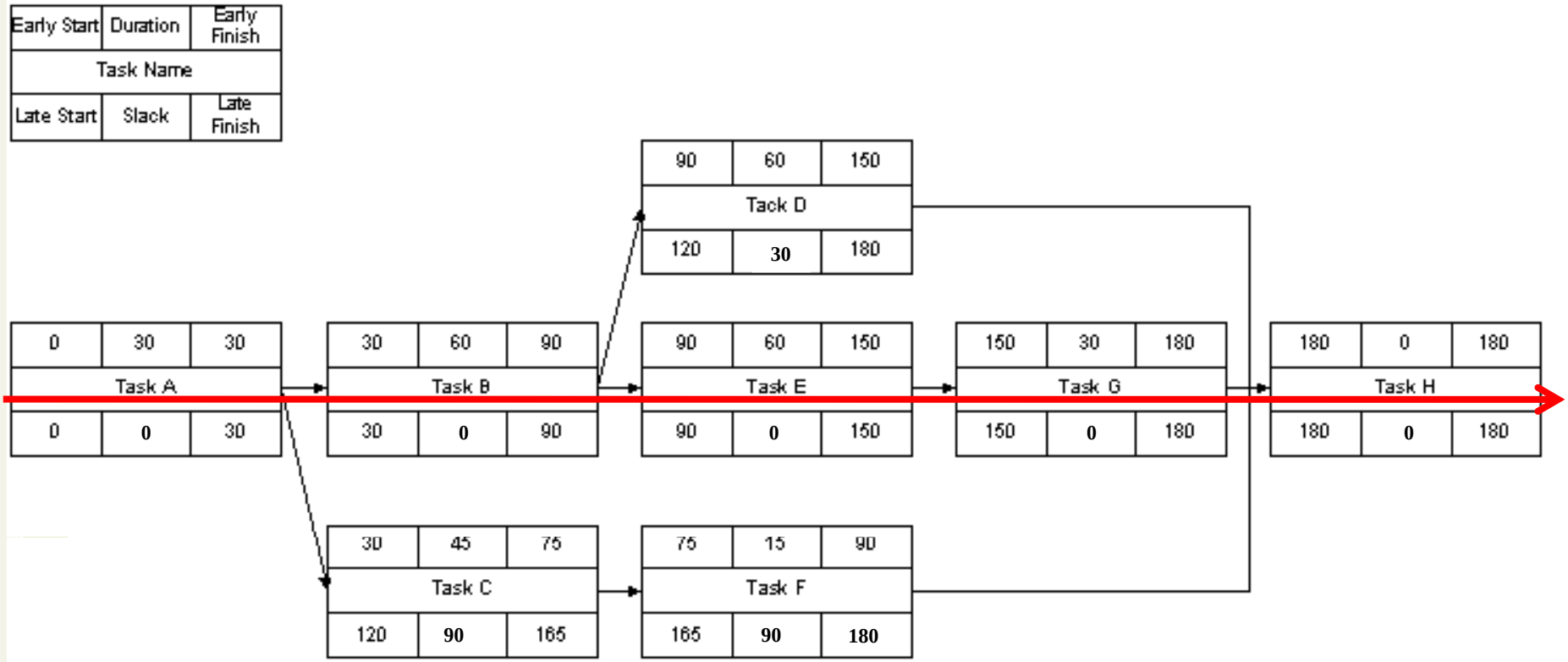
CPM: Example: Backward (part 2)



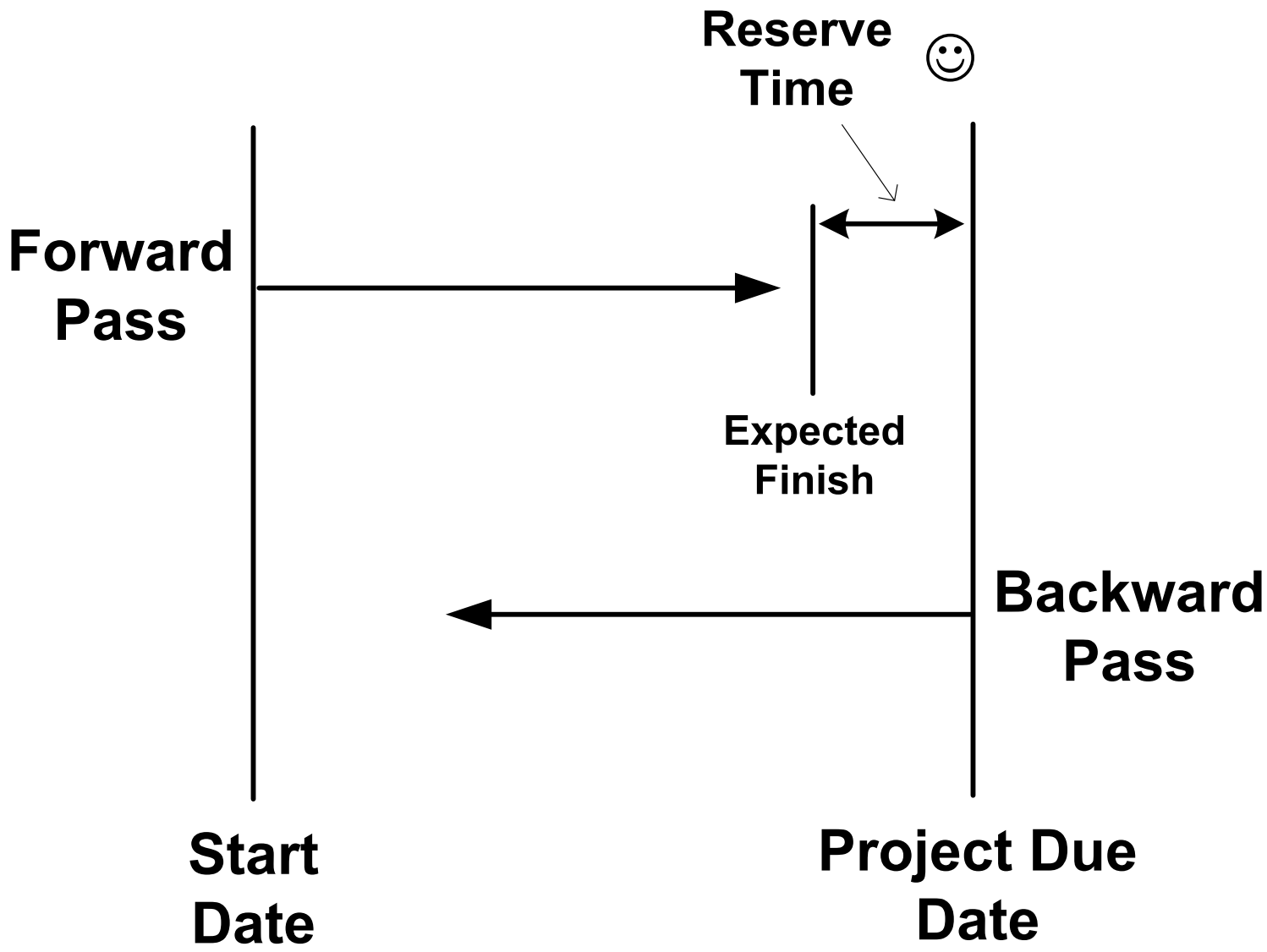
CPM: Example: Compute Slacks

Slack = Late Finish – Early Finish = Late Start – Early Start

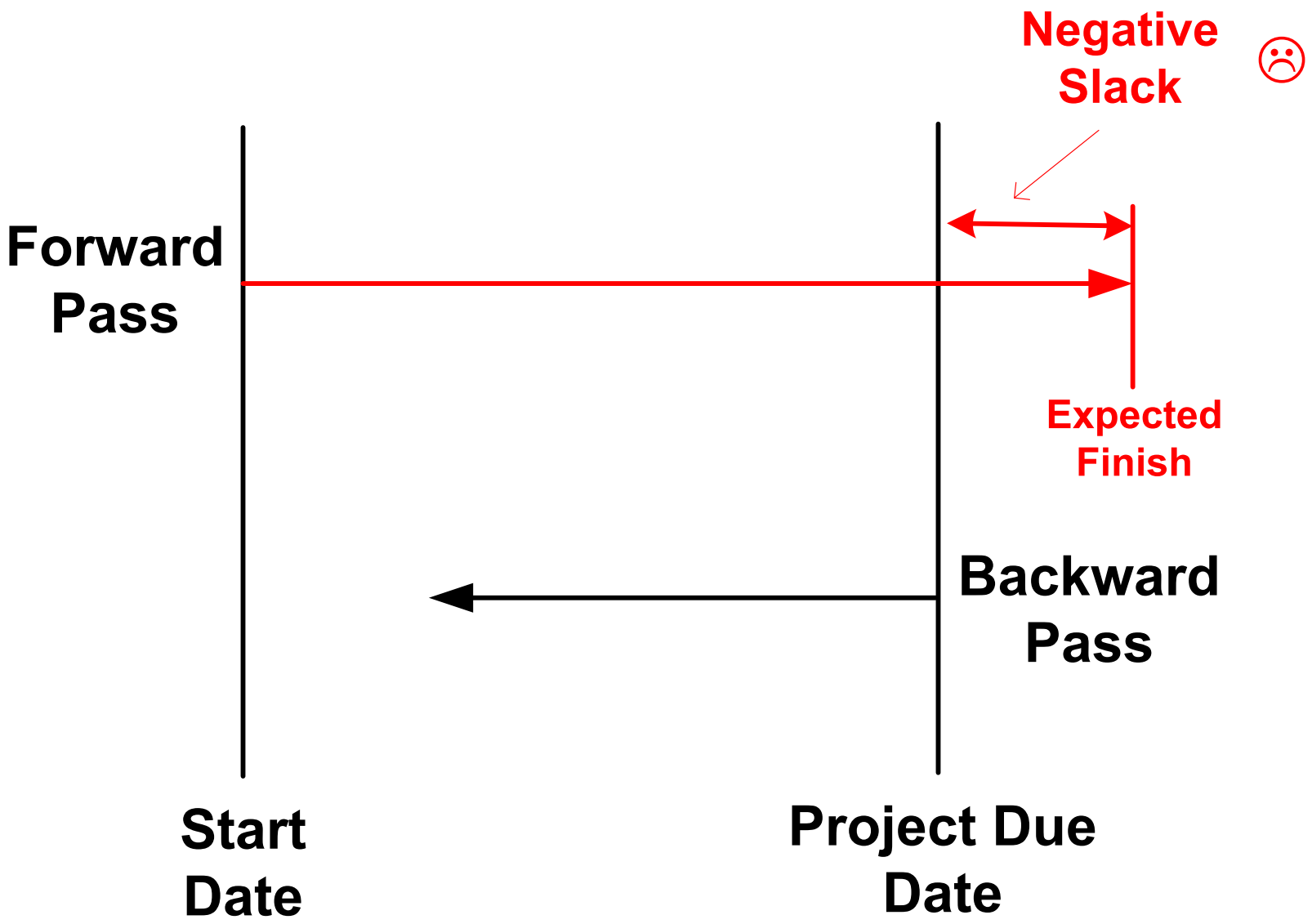
It's the Total Slack since the early and late dates are computed considering all the dependencies until the end of the project



CPM and Slack & Reserve



CPM and Slack & Reserve



CPM: Advantages and Disadvantages

- Advantages
  - Show precedence well
  - Reveal interdependencies not shown in other techniques
  - Ability to calculate critical path
  - Ability to perform “what if” exercises

- Disadvantages
  - Default model assumes resources are unlimited
    - You need to incorporate this yourself (Resource Dependencies) when determining the “real” Critical Path
  - Difficult to follow on large projects

PERT

- Program Evaluation and Review Technique
- Based on idea that estimates are uncertain
  - Therefore uses duration ranges
  - And the probability of falling to a given range
- Uses an “expected value” (or weighted average) to determine durations
- Use the following methods to calculate the expected durations, then use as input to your network diagram

PERT

- Start with 3 estimates for each task
  - Optimistic
    - Would likely occur 1 time in 20
  - Most likely
    - Modal value of the distribution
  - Pessimistic
    - Would be exceeded only one time in 20

PERT Formula

- Combined to estimate a task duration to be used in the network diagram

$$t_e = \frac{a + 4m + b}{6}$$

where

t_e = expected time

a = optimitistic time estimate

m = most likely time estimate

b = pessimistic time estimate

PERT Formula

- Confidence Interval can be determined
- Based on a standard deviation of the expected time
  - Using a bell curve (normal distribution)

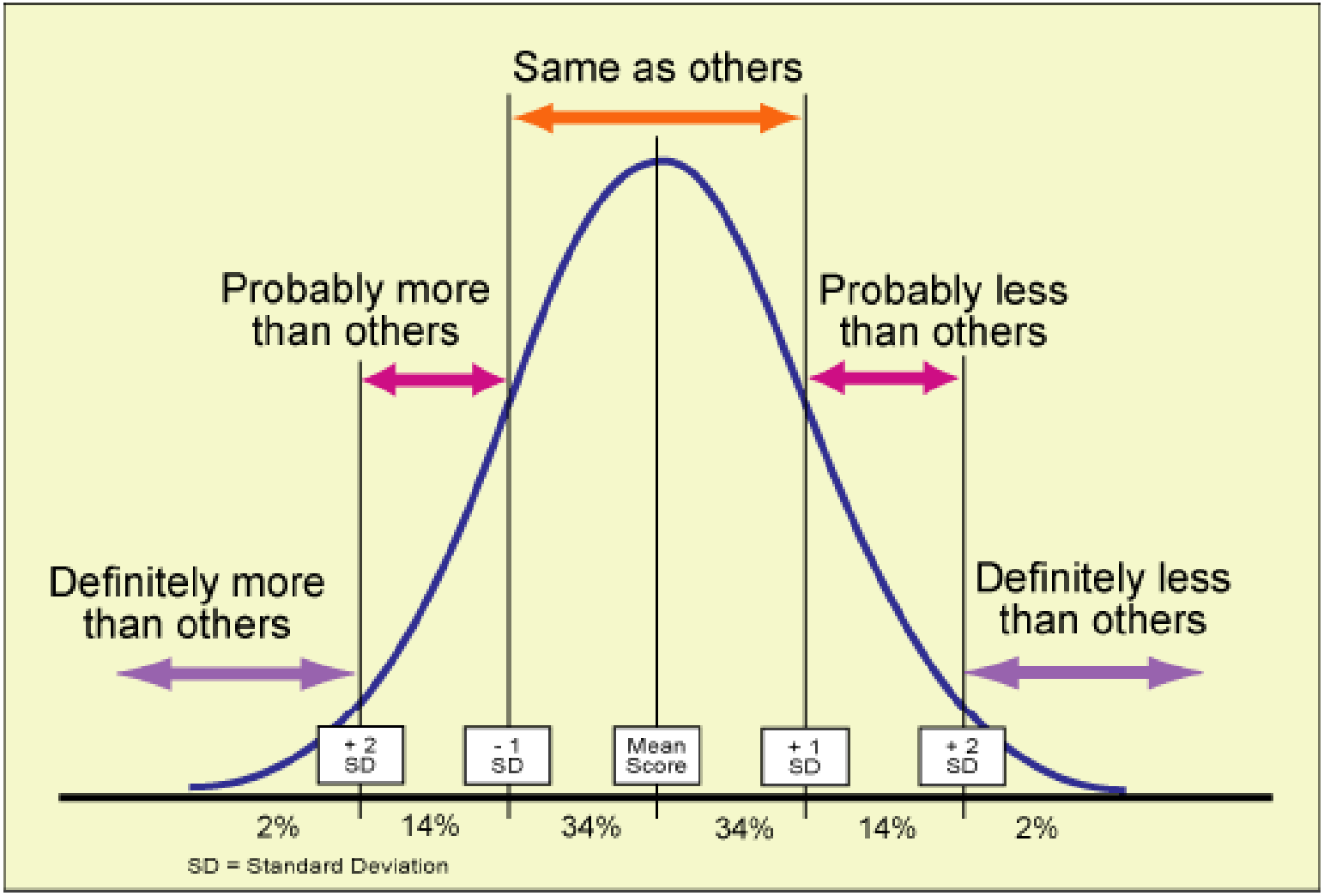
- For each task use

$$s_i = \frac{b_i - a_i}{6}$$

- For the whole critical path use

$$s_{cp} = \sqrt{s_1^2 + s_2^2 + \ldots + s_n^2}$$

MEMO: bell curve (normal distribution)



[source : http://www.fontys.nl/lerarenopleiding/tilburg/engels/Toetsing/bell_curve2.gif]

PERT Example

- Planner 1 (P1) and Planner 2 (P2) are asked to estimate m, a and b
- Confidence interval for P2 is 4 times wider than P1 for a given probability

Description	Planner 1	Planner 2
m	10d	10d
a	9d	9d
b	12d	20d
PERT time	10.2d	11.5d
Std. Dev.	0.5d	1.8d

- Ex: 68% probability of 9.7 to 10.7 days (P1) vs. 9.7-13.3 days (P2)

PERT: Advantages and Disadvantages

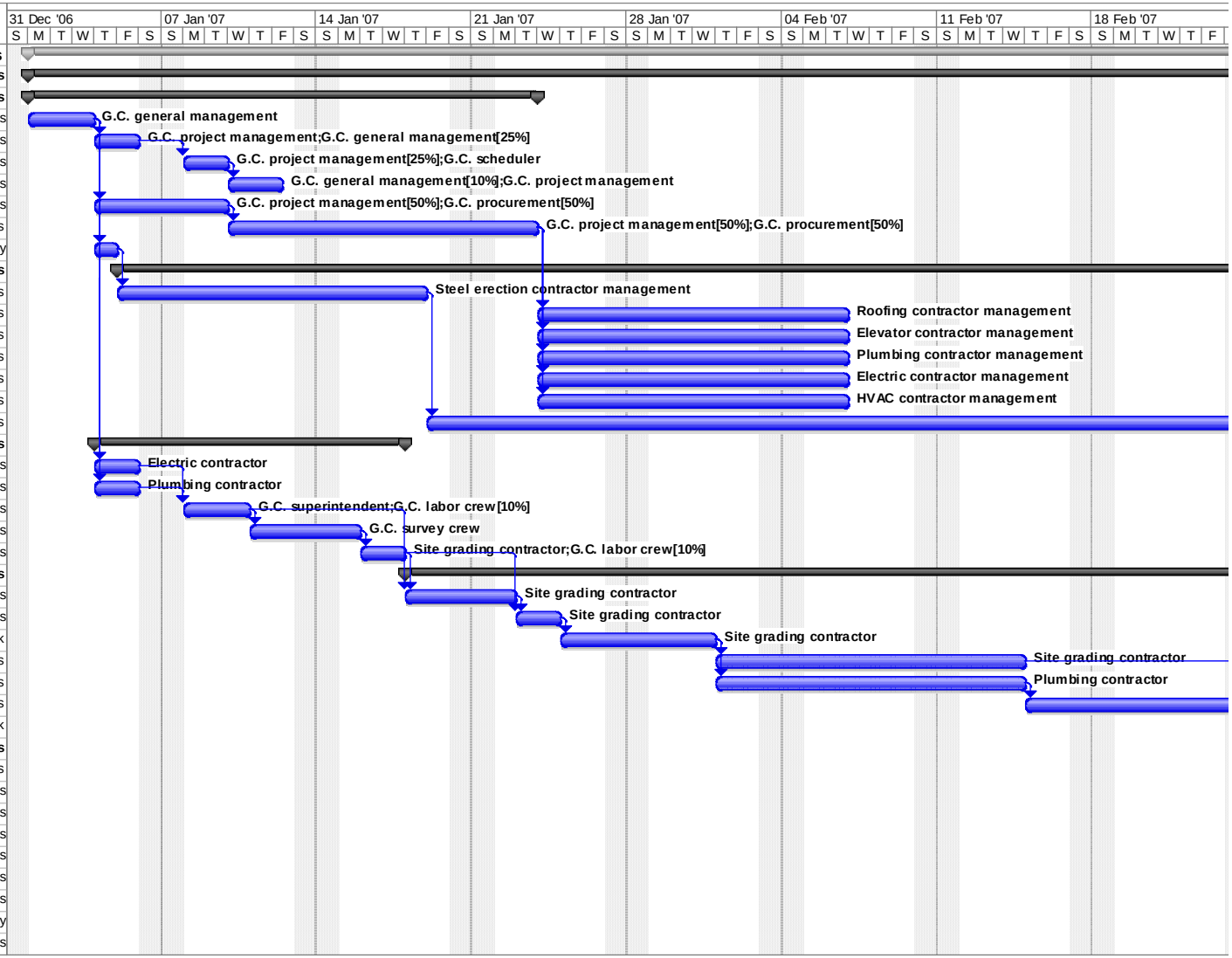
- Advantages
  - Accounts for uncertainty
- Disadvantages
  - Time and labor intensive
  - Assumption of unlimited resources is big issue
  - Lack of functional ownership of estimates
  - Mostly only used on large, complex project
- Get PERT software to calculate it for you

CPM vs. PERT

- Both use Network Diagrams
- CPM: deterministic
- PERT: probabilistic
- CPM: one estimate, PERT, three estimates
- PERT is infrequently used

Gantt Chart

ID	Task Name	Duration
0	**Commercial Construction**	**355 days**
1	**Three-story Office Building (7**	**355 days**
2	**General Conditions**	**17 days**
3	Receive notice to proc	3 days
4	Submit bond and insu	2 days
5	Prepare and submit pr	2 days
6	Prepare and submit sc	2,33 days
7	Obtain building permits	4 days
8	Submit preliminary sh	2 wks
9	Submit monthly reque	1 day
10	**Long Lead Procurement**	**70 days**
11	Submit shop drawings	2 wks
12	Submit shop drawings	2 wks
13	Submit shop drawings	2 wks
14	Submit shop drawings	2 wks
15	Submit shop drawings	2 wks
16	Submit shop drawings	2 wks
17	Detail, fabricate, and c	12 wks
18	**Mobilize on Site**	**10 days**
19	Install temporary powe	2 days
20	Install temporary wate	2 days
21	Set up site office	3 days
22	Set line and grade ber	3 days
23	Prepare site - lay dow	2 days
24	**Site Grading and Utilities**	**35 days**
25	Clear and grub site	3 days
26	Stone site access and	2 days
27	Rough grade site (cut	1 wk
28	Install storm drainage	2 wks
29	Install exterior fire line	2 wks
30	Perform final site gradi	2 wks
31	Erect building batter b	1 wk
32	**Foundations**	**39 days**
33	Excavate foundations	2 wks
34	Excavate elevator pit	2 days
35	Form column piers and	4 days
36	Rough-in electric and p	4 days
37	Form elevator pit walls	4 days
38	Set reinforcing and and	4 days
39	Pour column piers and	5 days
40	Pour concrete elevator	1 day
41	Cure elevator wall con	7 days



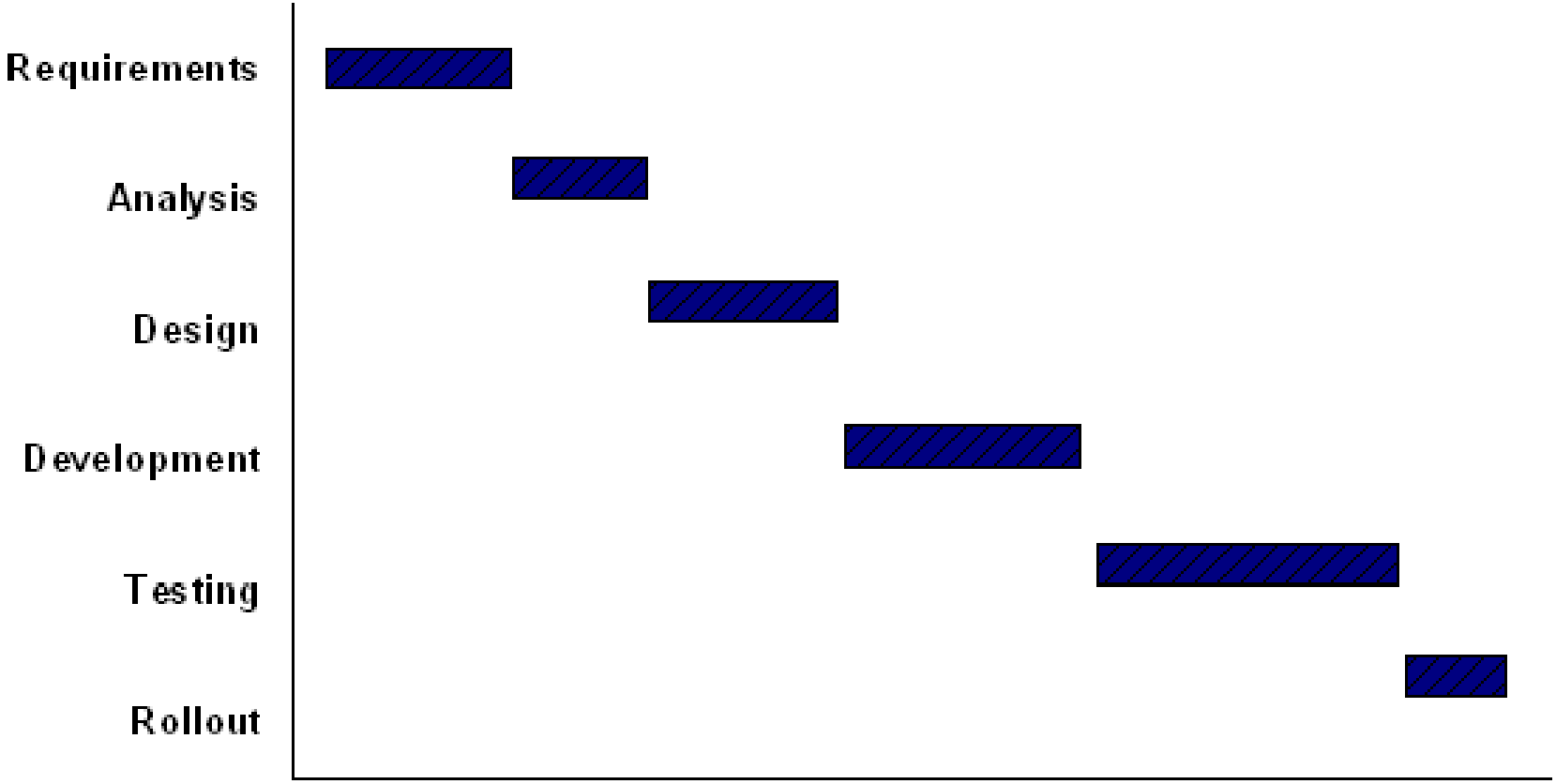
Gantt Chart

- Advantages
  - Easily understood
  - Easily created and maintained

  → Largely used
- Disadvantages
  - It does not show uncertainty of a given activity (as does PERT)
  - It has difficulties to show complex relationships among tasks

Milestone Chart

- Simple Gantt chart
  - Either showing just highest summary bars
  - Or milestones only



Today

- Session 4 review
- Scheduling Fundamentals
- Scheduling Techniques
  - Network Diagrams
  - Bar Charts
- **Schedule Optimization Techniques**
- Mythical Man-Month

Schedule Optimization Techniques

- How can you shorten the schedule?
- With one or more of the following approaches:
  - Reducing scope
    - Doing less
  - Reducing quality
    - Doing faster (or worse)
    - Pay attention that poorly tested software may cost more due to dependencies with other parts
  - Adding resources:
    - Having more persons working
    - See “Critical Man Months” section
  - Crashing
  - Fast Tracking
  - ...

Crashing and Fast Tracking

- Crashing
  - Looks at cost and schedule tradeoffs
    - Gain greatest compression with least cost
  - Add resources to critical path tasks
  - Changing the sequence of tasks
    - Reducing slack times
  - Limit or reduce requirements (scope)
- Fast Tracking
  - Overlapping of phases, activities or tasks that would otherwise be sequential
  - Involves some risk
  - May cause rework

Today

- Session 4 review
- Scheduling Fundamentals
- Scheduling Techniques
  - Network Diagrams
  - Bar Charts
- Schedule Optimization Techniques
- **Mythical Man-Month**

Mythical Man-Month

- Book: “The Mythical Man-Month”
  - Author: Fred Brooks
  - http://www.amazon.com/exec/obidos/ASIN/0201835959/qid%3D1022856693/sr%3D1-1/ref%3Dsr%5F1%5F1/103-4280067-9687806
  - http://my.safaribooksonline.com/0201835959
- “The classic book on the human elements of software engineering”
- First two chapters are full of terrific insight (and quotes)

Mythical Man-Month

- “Cost varies as product of men and months, progress does not.”
- “Hence the man-month as a unit for measuring the size of job is a dangerous and deceptive myth”
- “Good cooking fakes time. If you are made to wait, it is to serve you better, and to please you - Menu of Restaurant Antoine, New Orleans -”

Mythical Man-Month

- Why is software project disaster so common?
  1. Estimation techniques are poor & assume things will go well (an ‘unvoiced’ assumption)
  2. Estimation techniques fallaciously confuse effort with progress, hiding the assumption that men and months are interchangeable
  3. Because of estimation uncertainty, managers lack courteous stubbornness of Antoine's chef
  4. Schedule progress is poorly monitored
  5. When schedule slippage is recognized, the natural response is to add manpower. Which, is like dousing a fire with gasoline

Mythical Man-Month

- Optimism
  - "All programmers are optimists"
  - 1st false assumption: "all will go well" or "each task takes only as long as it 'ought' to take"
  - **The Fix: Consider the larger probabilities**
- Cost (overhead) of communication (and training)
  - Overhead: n(n-1)/2
  - How long does a 12 month project take?
    - 1 person: 12 month
    - 2 persons = 12/2 + 2(2-1)/2 = 6+1 = 7
      - 2 man-month extra
    - 3 persons = 12/3 + 3(3-1)/2 = 4 + 3 = 7
      - 9 man-months extra
    - 4 persons = 12/4 + 4(4-1)/2 = 3 + 6 = 9
  - **The Fix: don't assume adding people will solve the problem**

Mythical Man-Month

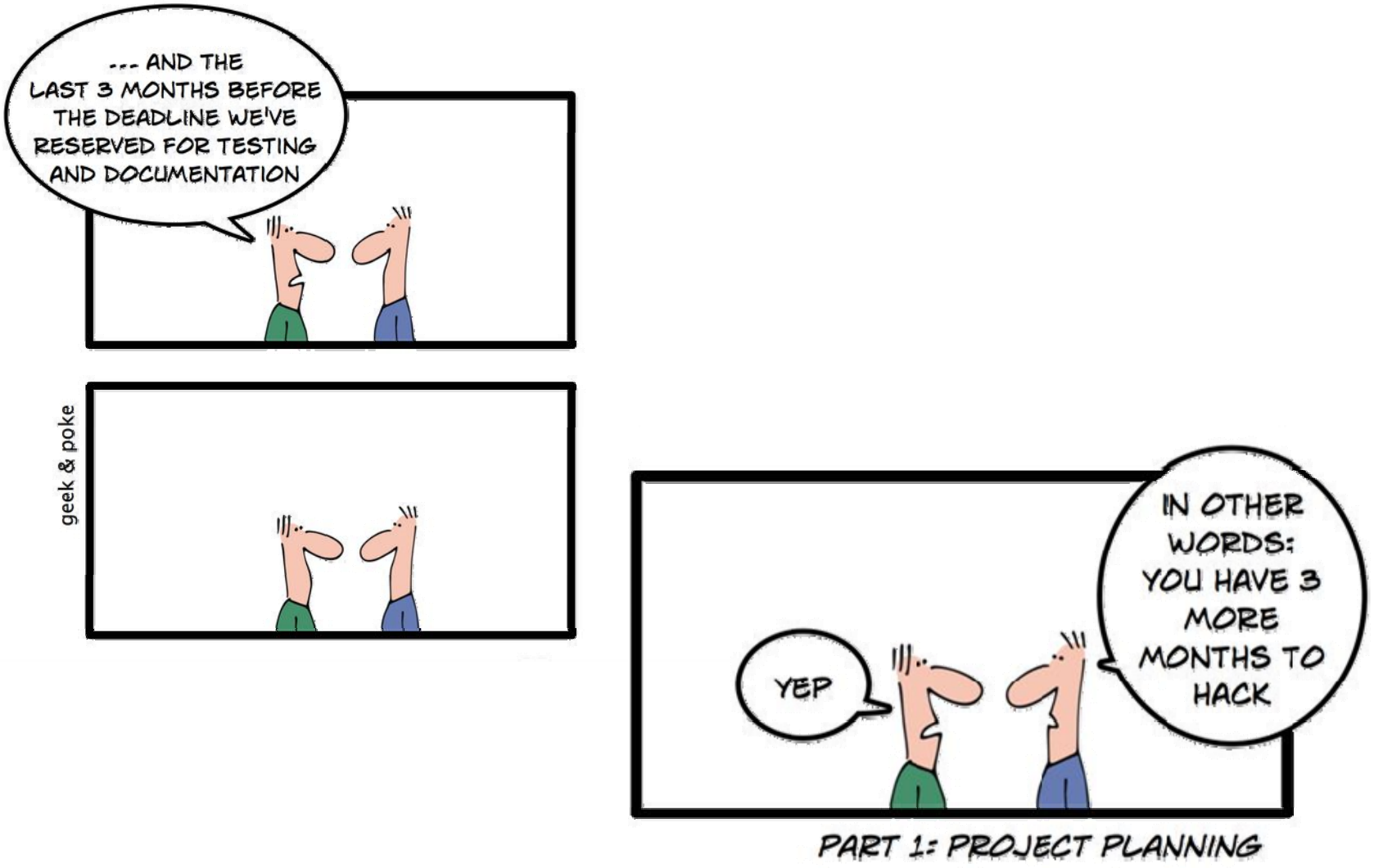
- Sequential nature of the process
  - “The bearing of a child takes nine months, no matter how many women are assigned”
- What is the most mis-scheduled part of process?
  - Testing
- Why is this particularly bad?
  - Occurs late in process and without early-warning
  - Higher costs
- **The Fix: Allocate more test time**
  - **Understand task dependencies, test and fix before use**

Mythical Man-Month

- Reliance on hunches and guesses
  - What is ‘gutless estimating’?
    - Urgency of Client causes Optimistic Estimates
      - E.g., omelet and chef analogy
      - http://my.safaribooksonline.com/0201835959/ch02lev1sec4
    - Regardless of Urgency, tasks require the same amount of time
- The myth of additional manpower
  - Brooks Law
  - “Adding manpower to a late project makes it later”
    - http://en.wikipedia.org/wiki/Brooks%27s_law

Mythical Man-Month

- Q: “How does a project get to be a year late”?
  - A: “One day at a time”
- Studies
  - Each task: twice as long as estimated
  - Only 50% of work week is real coding
    - The rest 50% is communication, negotiation, documentation, …
- **The Fixes**
  - Consider the 50% not-coding time
  - Define clearly measurable milestones
    - No “fuzzy” milestones
  - Reduce the role of conflict among persons
  - Identify the “true status” of a task
    - It’s impressive how much effort is needed to move a 90% done task to a 100% done task


--- AND THE LAST 3 MONTHS BEFORE THE DEADLINE WE'VE RESERVED FOR TESTING AND DOCUMENTATION
geek & poke
YEP
IN OTHER WORDS: YOU HAVE 3 MORE MONTHS TO HACK
PART 1: PROJECT PLANNING

Questions?