

 POLITECNICO DI MILANO

Dipartimento di
Elettronica e Informazione

Session 5

Scheduling

Emanuele Della Valle

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

- This slides are 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

3

- Session 4 review
- Network Fundamentals
- Gantt Charts
- PERT/CPM Techniques
- Mid-term review

Session 4 Review

4

- Planning
- WBS
- Estimation
- (Note: NPV, ROI, etc will be covered later in the term)

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

- “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

- History is your best ally
 - Especially when using LOC, function points, etc.
- 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

- 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 with/as 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 w/analogy
 - Best if multiple 'experts' consulted

- 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”)

- Initial Planning:
 - Why
 - SOW, Charter
 - What/How (partial/1st pass)
 - WBS
 - Other planning documents
 - Software Development Plan, Risk Mgmt., Cfg. Mgmt.

- Estimating
 - Size (quantity/complexity) and Effort (duration)
 - Iterates

- Scheduling
 - Begins along with 1st estimates
 - Iterates

- Once tasks (from the WBS) and size/effort (from estimation) are known: then schedule
- Primary objectives
 - Best time
 - Least cost
 - Least risk
- Secondary objectives
 - Evaluation of schedule alternatives
 - Effective use of resources
 - Communications

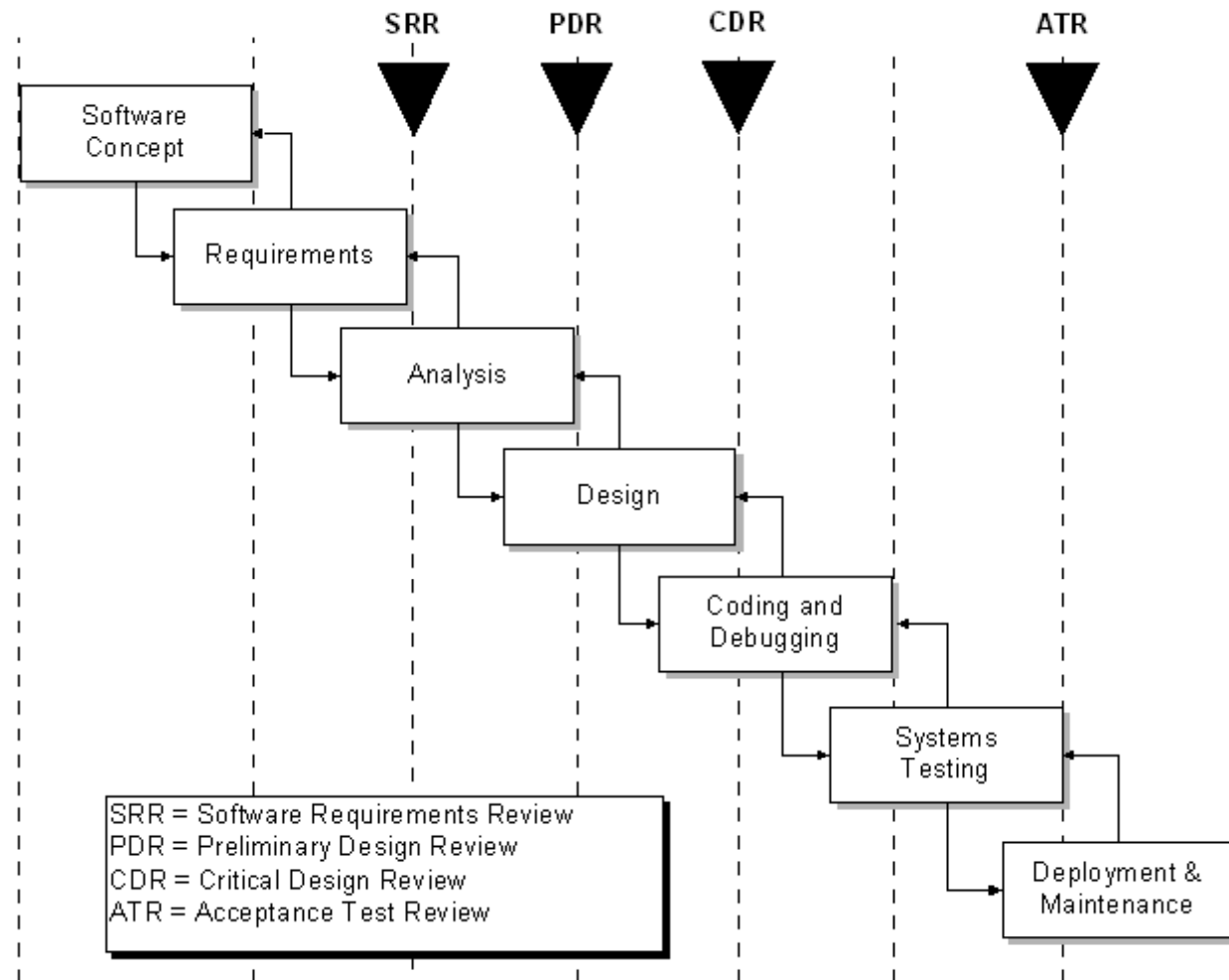
- Precedence:
 - A task that must occur before another is said to have precedence of the other

- Concurrency:
 - Concurrent tasks are those that can occur at the same time (in parallel)

- Leads & Lag Time
 - Delays between activities
 - Time required before or after a given task

- Milestones
 - Have a duration of zero
 - Identify critical points in your schedule
 - 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

- Example
- Milestones



- Slack & Float
 - Float & Slack: synonymous terms
 - Free Slack
 - Slack an activity has before it delays next task
 - Total Slack
 - Slack an activity has before delaying whole project
 - Slack Time $TS = TL - TE$
 - TE = earliest time an event can take place
 - TL = latest date it can occur w/o extending project's completion date

- Mathematical Analysis
 - Network Diagrams
 - PERT
 - CPM
 - GERT
- Bar Charts
 - Milestone Chart
 - Gantt Chart

Network Diagrams

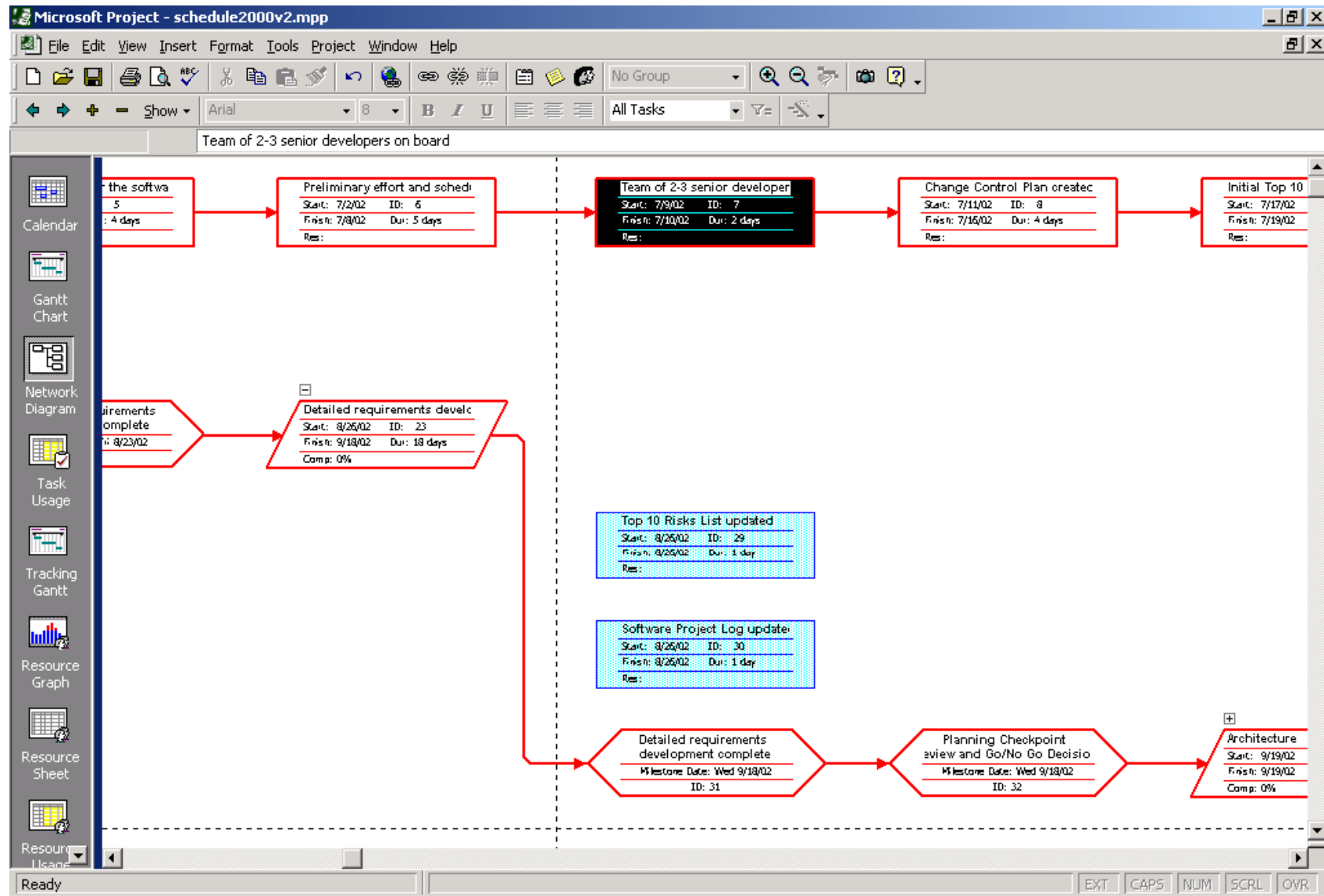
17

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

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

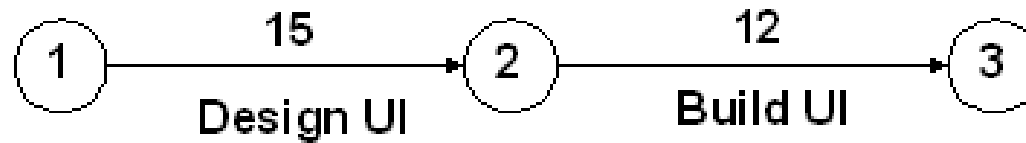
MS-Project Example

19

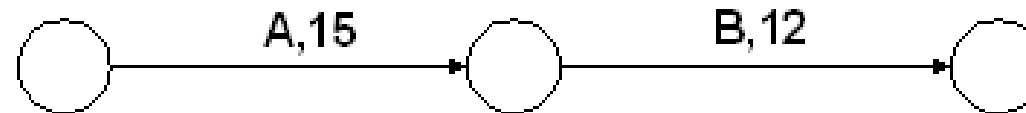


- Two classic formats
 - AOA: Activity on Arrow
 - AON: Activity on Node
- Each task labeled with
 - Identifier (usually a letter/code)
 - Duration (in standard unit like days)
- There are other variations of labeling
- There is 1 start & 1 end event
- Time goes from left to right

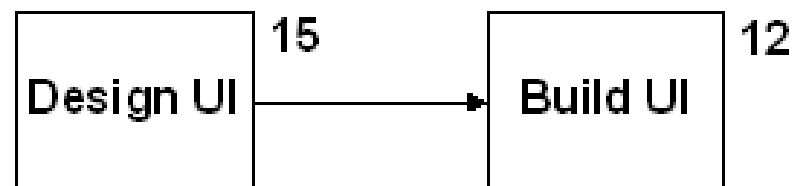
Activity on Arrow (AOA)



or



Activity on Node (AON)



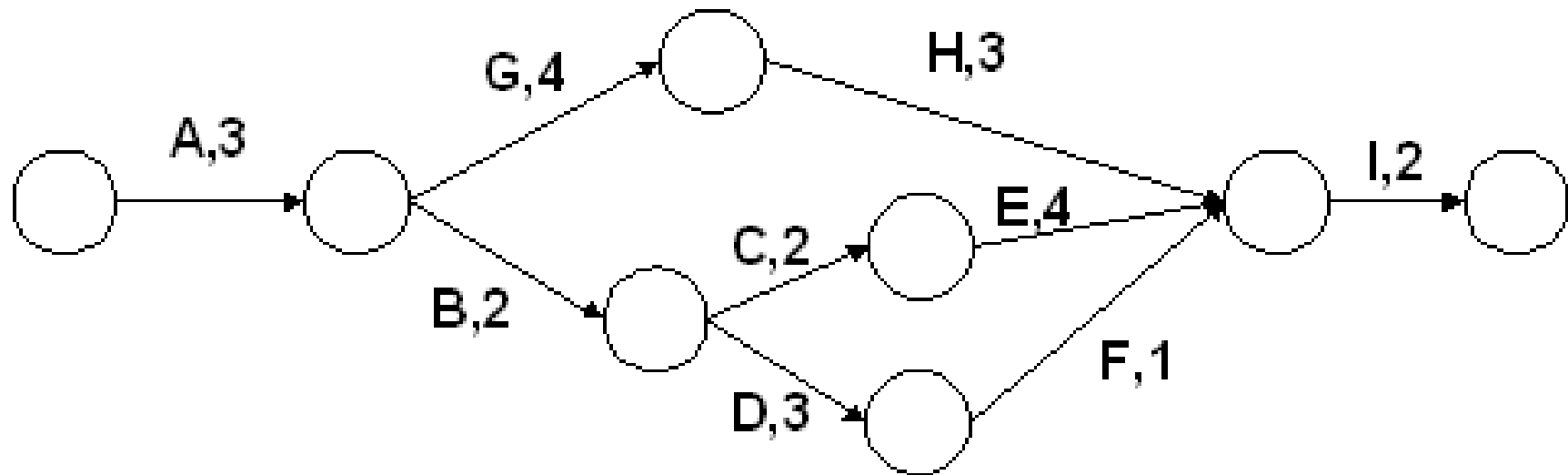
or

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

- AOA consists of
 - 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
 - Tasks 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)

- “The specific set of sequential tasks upon which the project completion date depends”
 - or “the longest full path”
- All projects have a Critical Path
- Accelerating non-critical tasks do not directly shorten the schedule



- Critical Path Method
 - The process for determining and optimizing the critical path
- Non-CP tasks can start earlier or later w/o impacting completion date
- Note: Critical Path may change to another as you shorten the current
- Should be done in conjunction with the you & the functional manager

4 Task Dependency Types

26

- Mandatory Dependencies
 - “Hard logic” dependencies
 - Nature of the work dictates an ordering
 - Ex: Coding has to precede testing
 - Ex: UI design precedes UI implementation

- 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

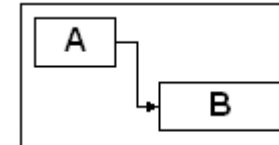
4 Task Dependency Types

27

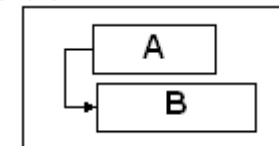
- External Dependencies
 - Outside of the project itself
 - Ex: Release of 3rd party product; contract signoff
 - Ex: stakeholders, suppliers, Y2K, year end

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

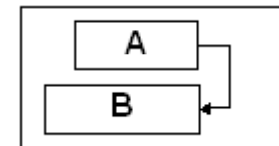
- Finish-to-Start (FS)
 - B cannot start till A finishes
 - A: Construct fence; B: Paint Fence



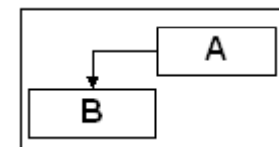
- Start-to-Start (SS)
 - B cannot start till A starts
 - A: Pour foundation; B: Level concrete



- Finish-to-Finish (FF)
 - B cannot finish till A finishes
 - A: Add wiring; B: Inspect electrical

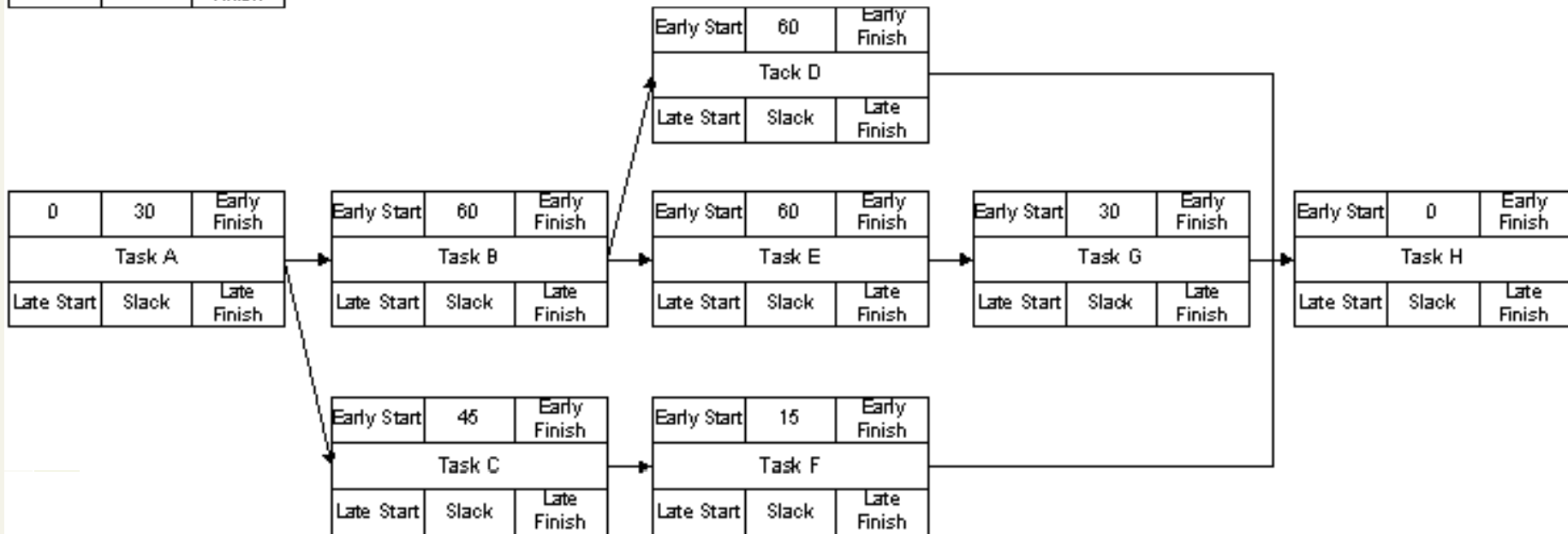


- Start-to-Finish (SF)
 - B cannot finish till A starts (rare)



Example Step 1

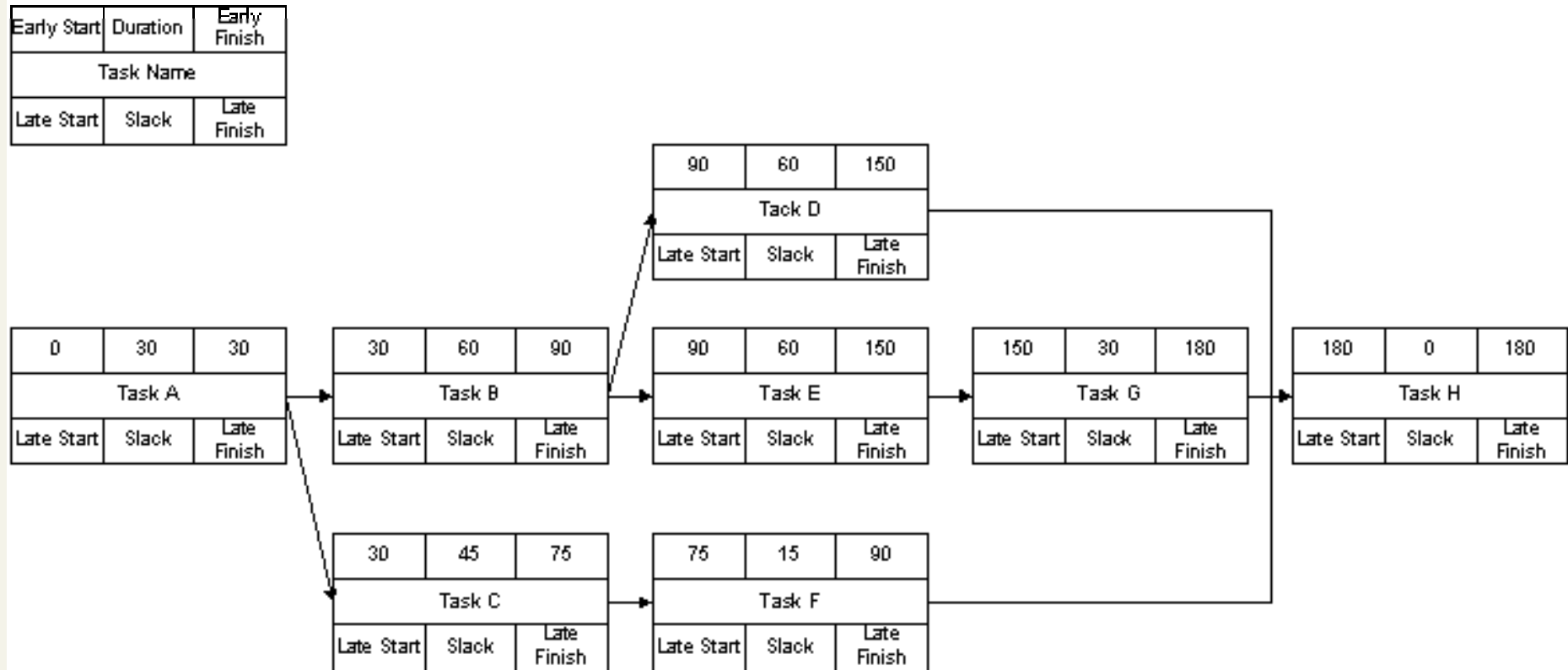
Early Start	Duration	Early Finish
Task Name		
Late Start	Slack	Late Finish



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

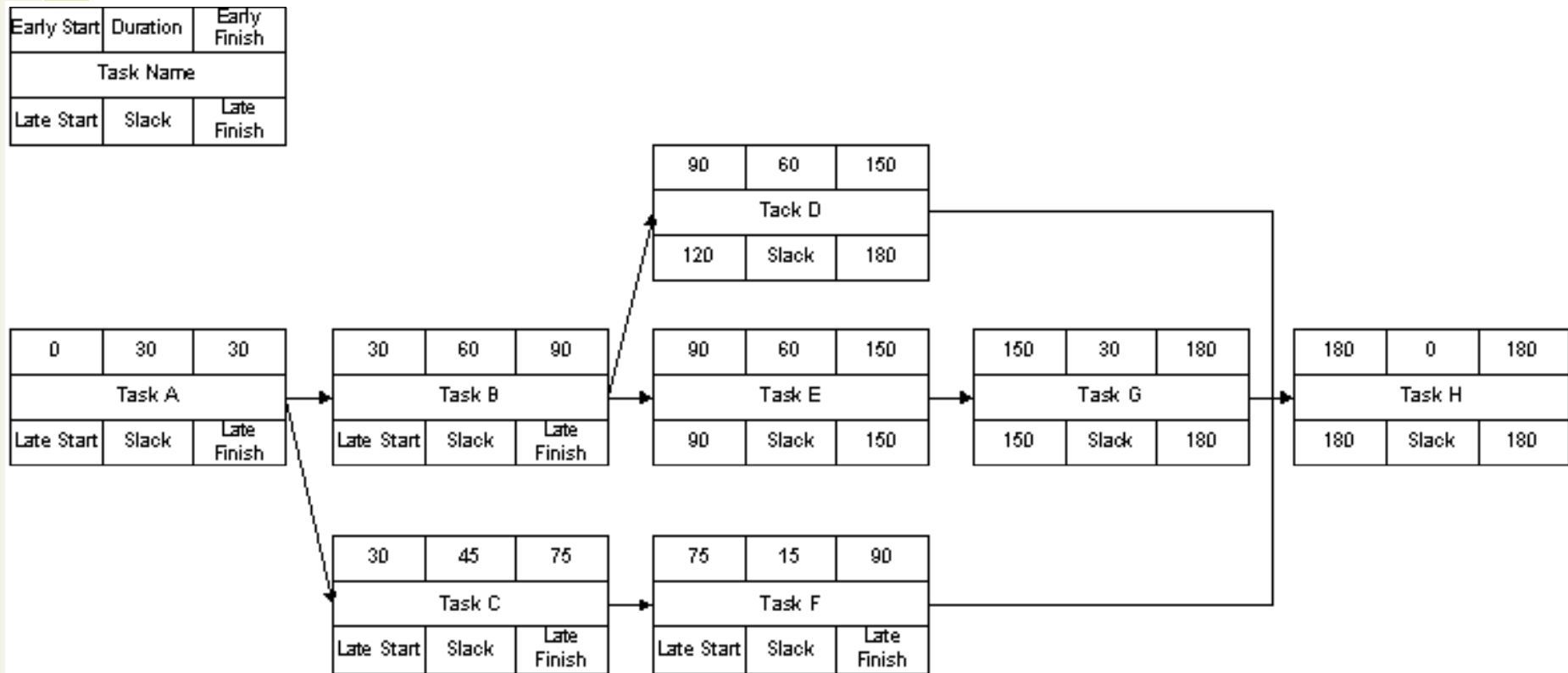
Example Step 2

31



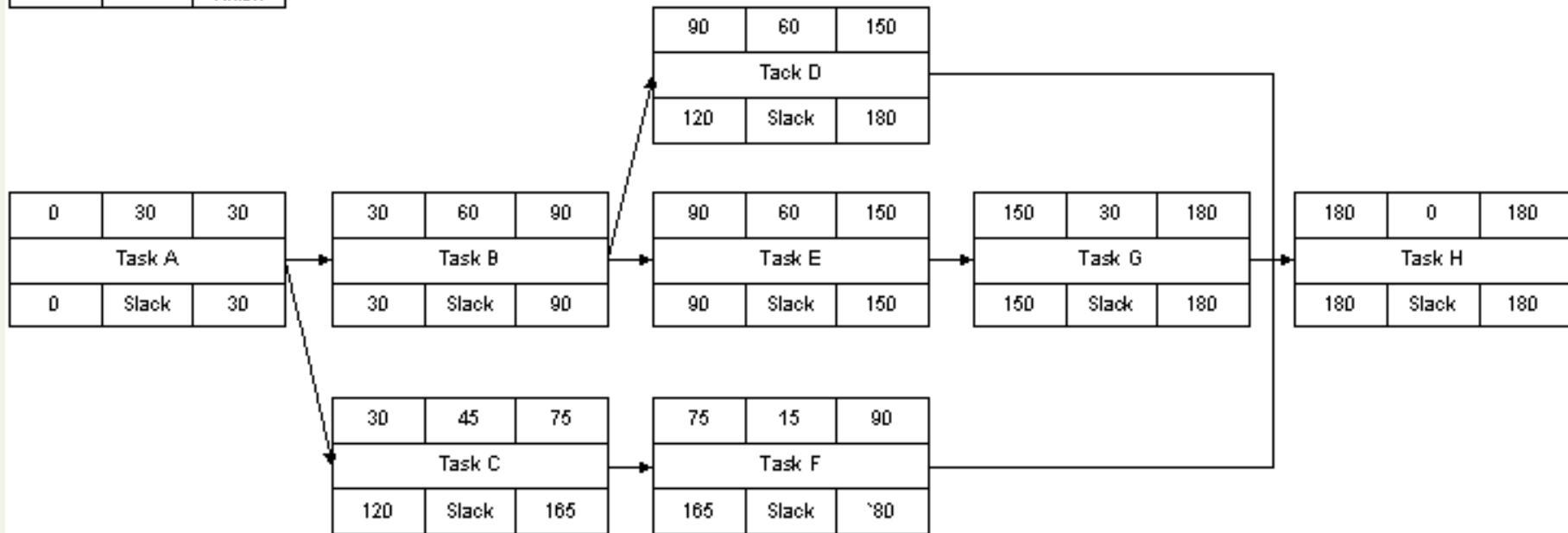
- To determine the last finish (LF) and last start (LS) times
- Start at the end node
- Compute the bottom pair of numbers
- Subtract duration from connecting node's earliest start time

Example Step 3

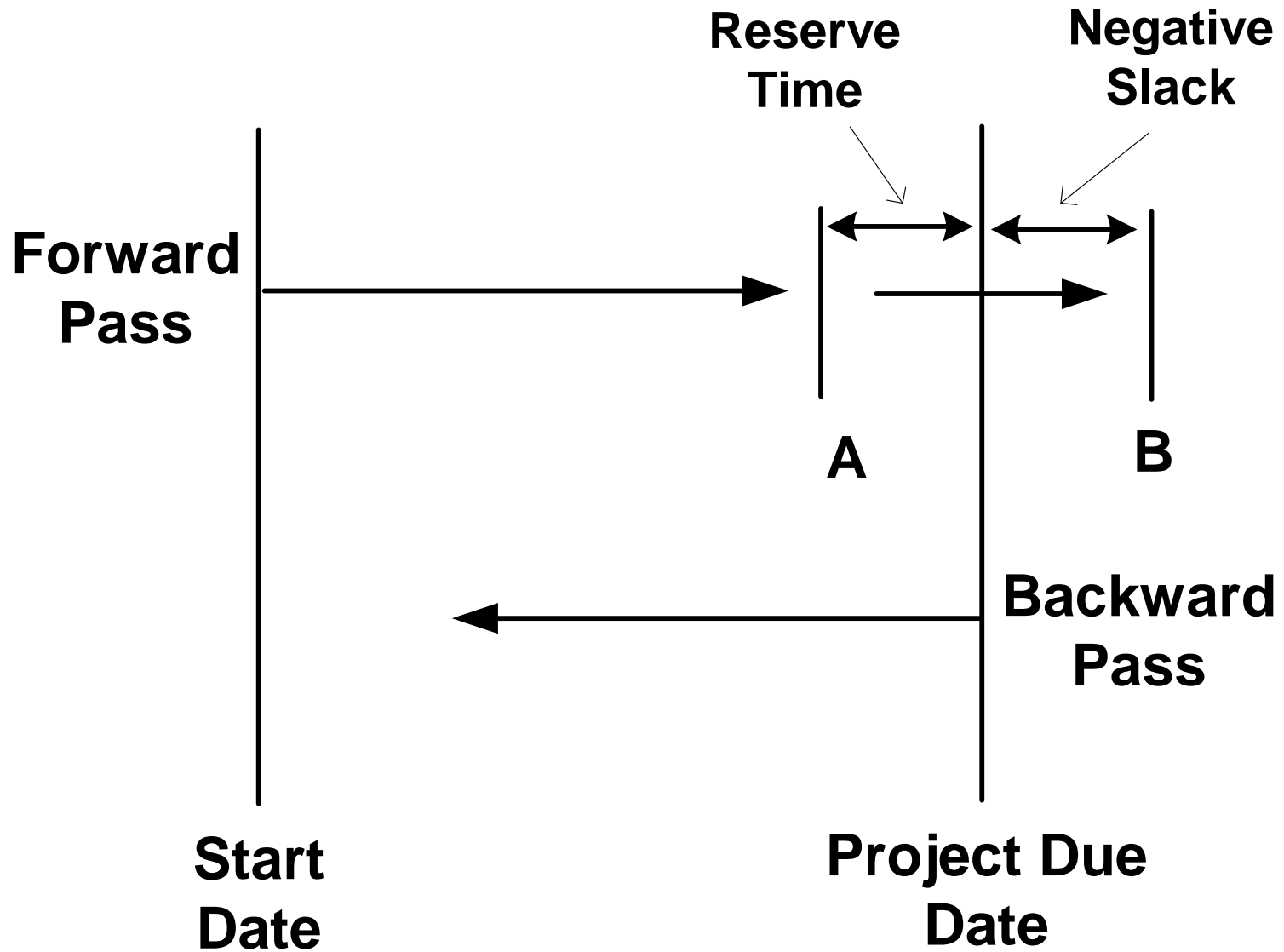


Example Step 4

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



- How can slack be negative?
- What does that mean?
- How can you address that situation?



- 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

- 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

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

- Combined to estimate a task duration

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

where

t_e = expected time

a = optimistic time estimate

m = most likely time estimate

b = pessimistic time estimate

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

- For the whole critical path use

$$s = \frac{b - a}{6}$$

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

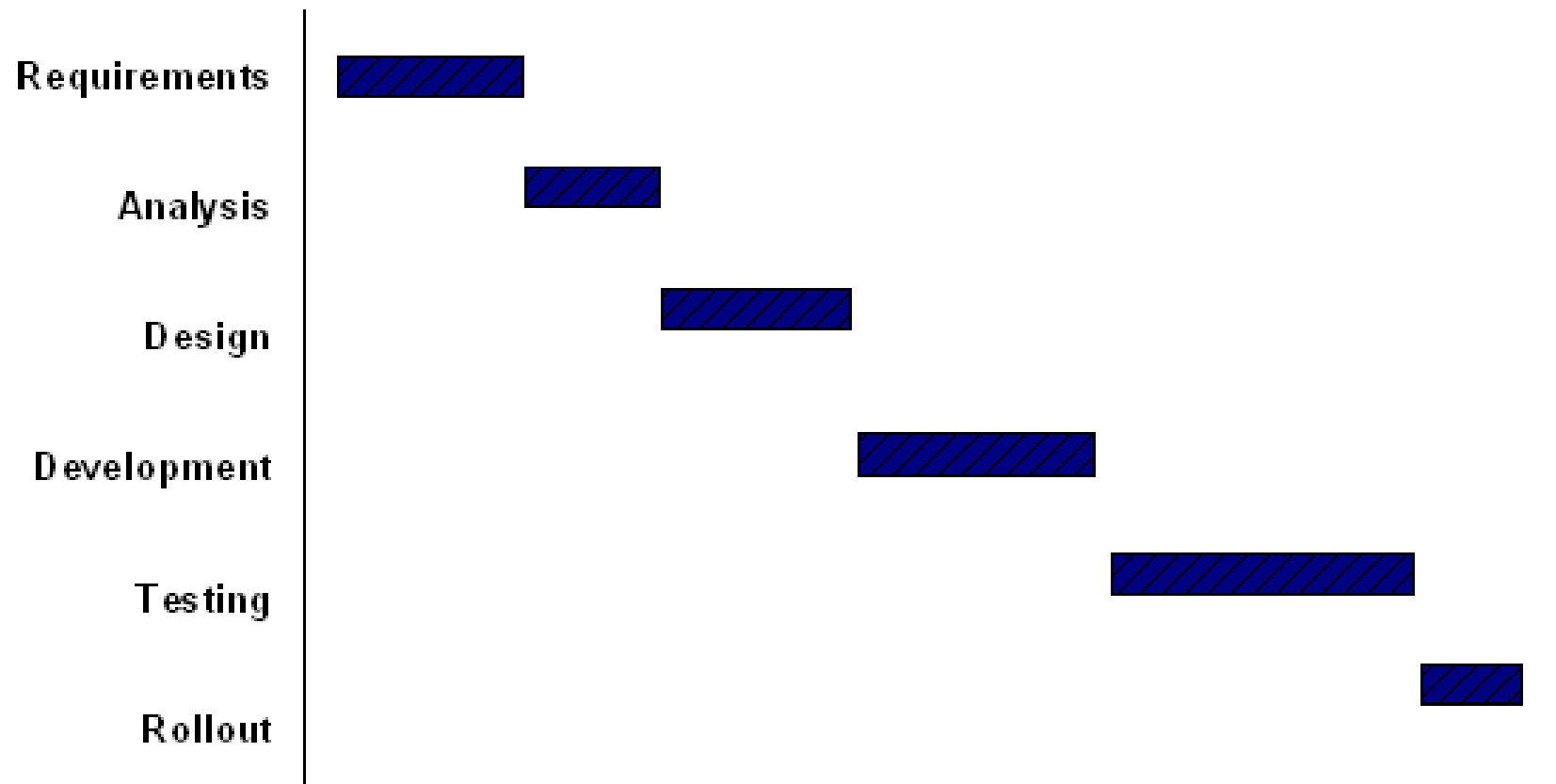
- Planner 1 (P1) and Planner 2 (P2) are asked to estimate m , a and b
- Confidence interval for P2 is 4 times wider than P1 for a given probability
- Ex: 68% probability of 9.7 to 11.7 days (P1) vs. 9.5-13.5 days (P2)

Description	Planner 1	Planner 2
m	10d	10d
a	9d	9d
b	12d	20d
PERT time	10.16d	11.5d
Std. Dev.	0.5d	1.8d

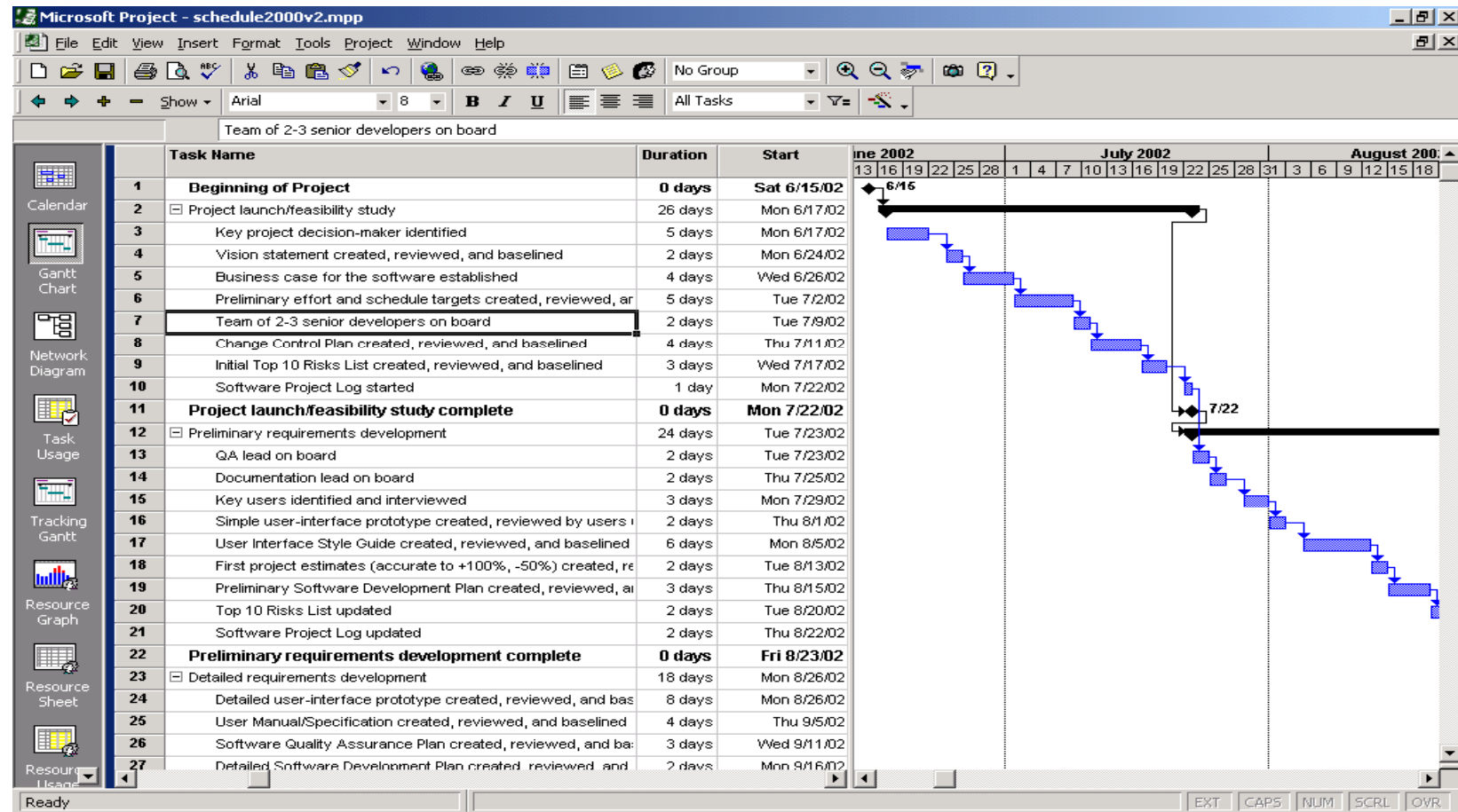
- Advantages
 - Accounts for uncertainty
- Disadvantages
 - Time and labor intensive
 - Assumption of unlimited resources is big issue
 - Lack of functional ownership of estimates
 - Mostly only used on large, complex project
- Get PERT software to calculate it for you

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

- Sometimes called a “bar charts”
- Simple Gantt chart
 - Either showing just highest summary bars
 - Or milestones only



Gantt Chart



- Disadvantages
 - Does not show interdependencies well
 - Does not uncertainty of a given activity (as does PERT)
- Advantages
 - Easily understood
 - Easily created and maintained

- Note: Software now shows dependencies among tasks in Gantt charts
 - In the “old” days Gantt charts did not show these dependencies, bar charts typically do not

- How can you shorten the schedule?
- Via
 - Reducing scope (or quality)
 - Adding resources
 - Concurrency (perform tasks in parallel)
 - Substitution of activities

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

- 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)

- “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”

- 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, manager lack courteous stubbornness
 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.

- 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)
 - His formula: $n(n-1)/2$
 - 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 = $12/4 + 4(4-1)/2 = 3 + 6 = 9$
 - **Fix: don't assume adding people will solve the problem**

- 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 (the most linear process)
- Why is this particularly bad?
 - Occurs late in process and w/o warning
 - Higher costs: primary and secondary
- **Fix: Allocate more test time**
 - **Understand task dependencies**

- 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"

- 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 was programming
- Fixes
 - No “fuzzy” milestones (get the “true” status)
 - Reduce the role of conflict
 - Identify the “true status”

- Next week:
 - OpenProj introduction (bring your laptops!)
 - Q&A about contents covered in sessions 1-5
- The week after: exam
 - You will have rest of class to complete
 - Likely Format: 10-15 questions, most 3 points each

- Projects, programs, products
- McConnell's four dimensions
- Classic mistakes
 - Know a set of these
 - Remember by "type"
 - People, process, product, technology related
 - http://www.emanueledellavalle.org/slides/P&MSP2009_01_Introduction-Fundamentals-Classic-Mistakes.pdf
 - Slides 44-52

- Trade-offs & constraints
 - The triangle
 - Cost, Time, Scope
 - (And Quality)
- PMI processes and knowledge areas
 - Process groups
- Organizational structures
 - Advantages & disadvantages of each form

- Classic project phases
 - http://www.emanueledellavalle.org/slides/P&MSP2009_04_Planning.pdf
 - Slide 5

- Key documents at each phase
 - SOW, Charter
 - Project Management Plan
 - Identify key planning documents
 - http://www.emanueledellavalle.org/slides/P&MSP2009_04_Planning.pdf
 - Slide 9

- Methodologies
 - Trade-offs
 - Basic Pros & Cons
- Given a specific scenario decide what SDLC is most appropriate
- Waterfall process
- Challenges of each phase
- Requirements
 - Criticality
 - Issues
 - Functional & Non-functional

- Wherever I gave you say 5 types of 'things'
 - Like estimation techniques or methodologies
- I may ask 'Tell me two approaches to X'
- Especially interested in the pros & cons
- You did your Class slides readings, yes?

- Four primary steps
 - Define work to be done (WBS)
 - Estimate size
 - Estimate effort
 - Build schedule

- Types: Process, product, hybrid
- Formats: Outline or graphical org chart
- Shows hierarchical task relationships
- High-level version does not show dependencies or durations
- What hurts most is what's missing
- Becomes input to many things, esp. schedule

- Size Estimation Techniques
 - Bottom-up vs. Top-down
 - Analogy
 - Expert Judgment
 - Parametric (not the formulas, just the gist)
 - Function Points
 - LOC

- Schedule presentation techniques
 - Q3, 6-8 months, best/worst case

- Dependencies
 - Types: mandatory, etc.
 - Relationships: FS, SF, etc.
- Network Diagrams
 - CPM
 - PERT
- Schedule Optimization Techniques

- Ambiguous terminology
 - Tasks vs. Activities vs. Work Packages
 - WBS: does it start at 0 or 1
 - Details of Function Point calculation method

- No homework, just exam study
- Install OpenProj (or MS Project) on your laptops
 - <http://openproj.org/openproj>

Questions?

70