Scheduling

Anne Banks Pidduck

Adapted from John Musser

Today

- Network Fundamentals
- Gantt Charts
- PERT/CPM Techniques

WBS

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

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”

Estimation

- History is your best ally
  - Especially when using LOC, function points and so on
- Use multiple methods if possible
  - This reduces your risk
  - If using “experts”, use two or more
- Get buy-in
- Remember: it’s an iterative process!

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 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
- Re-Use Estimation
  - See QSPM links
- U Calgary links

Scheduling

- 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

Terminology

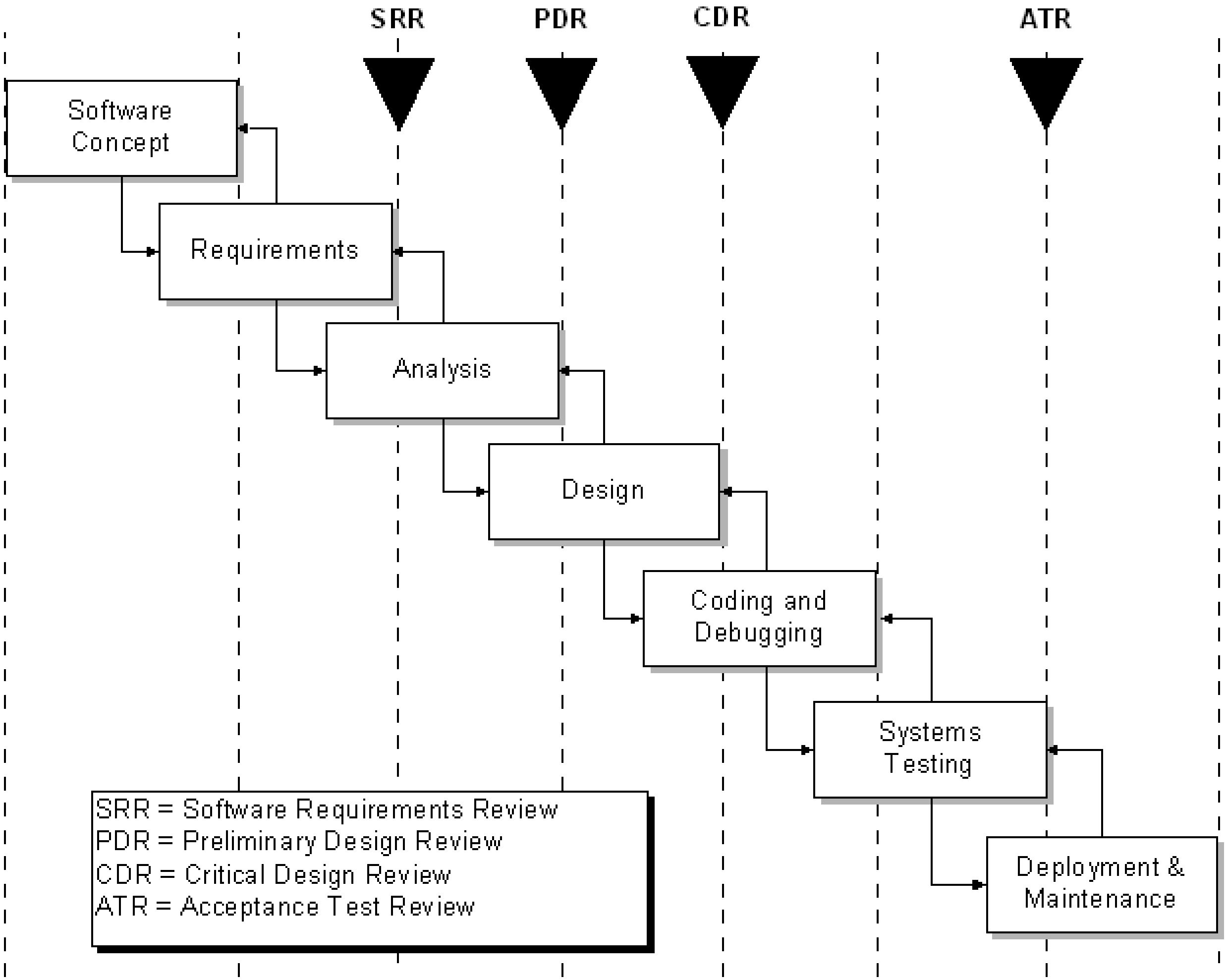
- Precedence:
  - A task that must occur before another is said to have precedence of the other
- Concurrence:
  - Concurrent tasks are those that can occur at the same time (in parallel)
- Leads & Lag Time
  - Delays between activities
  - Time required before or after a given task

Terminology

- 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

Terminology

Example Milestones



SRR = Software Requirements Review
PDR = Preliminary Design Review
CDR = Critical Design Review
ATR = Acceptance Test Review

Terminology

- 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 $T_S = T_L – T_E$
    - T_E = earliest time an event can take place
    - T_L = latest date it can occur w/o extending project's completion date

Scheduling Techniques

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

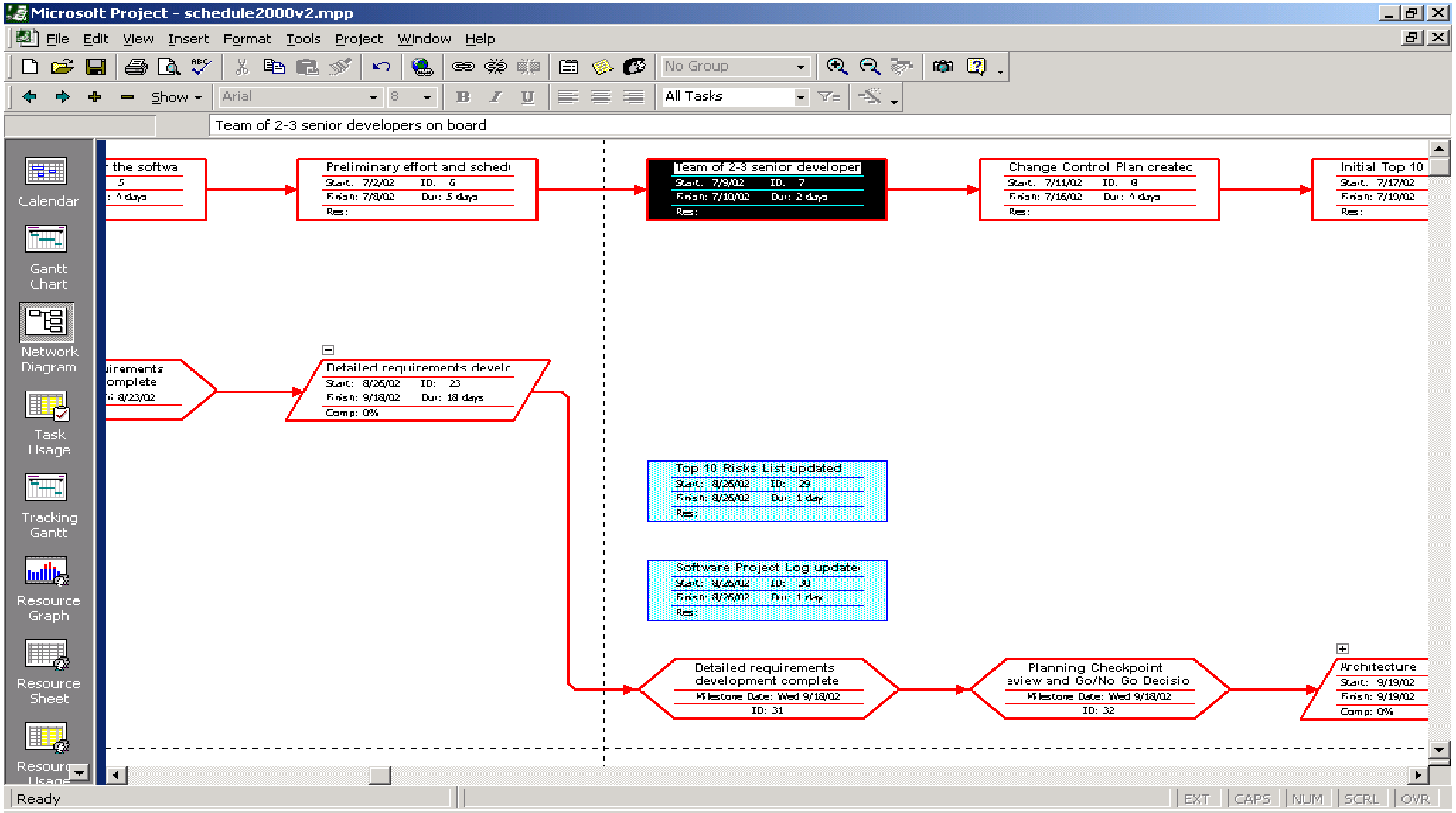
Network Diagrams

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

Mathematical Analysis

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

MS-Project Example



Team of 2-3 senior developers on board

the softwa
5
: 4 days

Preliminary effort and schedu
Start: 7/2/02 ID: 6
Finish: 7/8/02 Dur: 5 days
Res:

Team of 2-3 senior developer
Start: 7/9/02 ID: 7
Finish: 7/10/02 Dur: 2 days
Res:

Change Control Plan createc
Start: 7/11/02 ID: 8
Finish: 7/16/02 Dur: 4 days
Res:

Initial Top 10
Start: 7/17/02
Finish: 7/19/02
Res:

uirements
omplete
i: 8/23/02

Detailed requirements develc
Start: 8/26/02 ID: 23
Finish: 9/18/02 Dur: 18 days
Comp: 0%

Top 10 Risks List updated
Start: 8/26/02 ID: 29
Finish: 8/26/02 Dur: 1 day
Res:

Software Project Log update
Start: 8/26/02 ID: 30
Finish: 8/26/02 Dur: 1 day
Res:

Detailed requirements
development complete
Milestone Date: Wed 9/18/02
ID: 31

Planning Checkpoint
eview and Go/No Go Decisio
Milestone Date: Wed 9/18/02
ID: 32

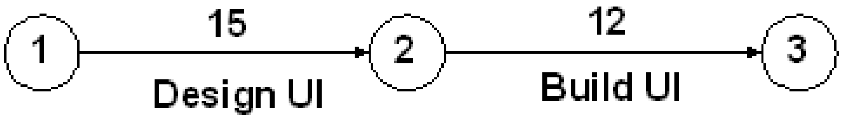
Architecture
Start: 9/19/02
Finish: 9/19/02
Comp: 0%

Network Diagrams

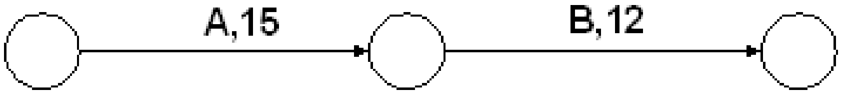
- Two classic formats
  - – AOA: Activity on Arrow
  - – AON: Activity on Node
- Each task labeled with
    - Identifier (usually a letter/code)
    - Duration (in standard unit like days)
- There are other variations of labeling
- There is 1 start & 1 end event
- Time goes from left to right

Node Formats

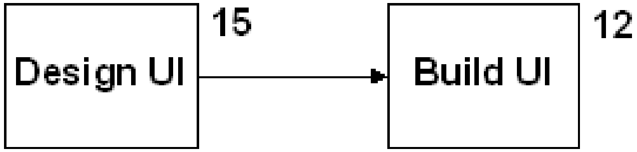
Activity on Arrow (AOA)



or



Activity on Node (AON)



or

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

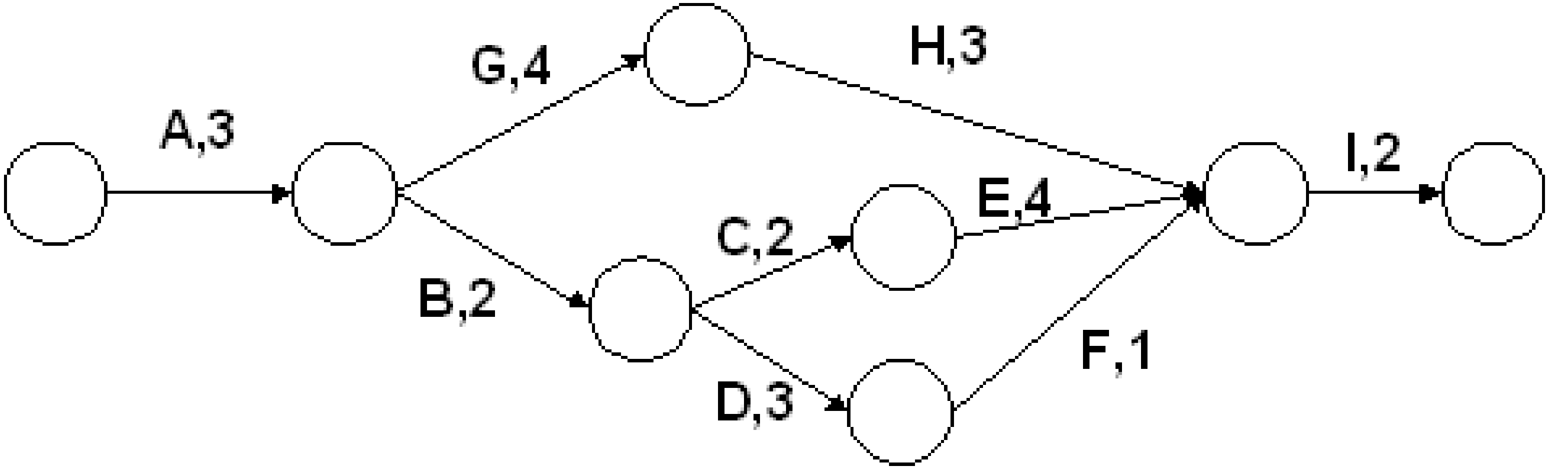
Network Diagrams

- 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)

Critical Path

- “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 Example



CPM

- Critical Path Method
  - The process for determining and optimizing the critical path
- Non-CP tasks can start earlier or later without impacting completion date
- Note: Critical Path may change to another as you shorten the current path
- Should be done in conjunction with the functional manager

4 Task Dependency Types

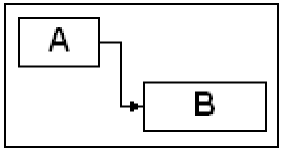
- **Mandatory** Dependencies
  - “Hard logic” dependencies
  - Nature of the work dictates an ordering
  - Ex: Coding has to precede testing
  - Ex: UI design precedes UI implementation
- **Discretionary** Dependencies
  - “Soft logic” dependencies
  - Determined by the project management team
  - Process-driven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

4 Task Dependency Types

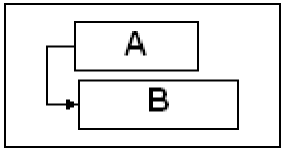
- **External** Dependencies
    - Outside of the project itself
    - Ex: Release of 3rd party product; contract signoff
    - Ex: stakeholders, suppliers, Y2K, year end
- **Resource** Dependencies
    - Two tasks rely on the same resource
    - Ex: You have only one DBA but multiple DB tasks

Task Dependency Relationships

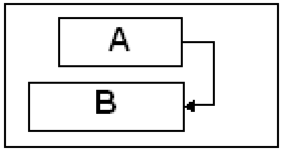
- Finish-to-Start (FS)
  - B cannot start till A finishes
  - A: Construct fence; B: Paint Fence



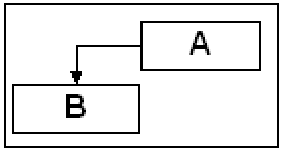
- Start-to-Start (SS)
  - B cannot start till A starts
  - A: Pour foundation; B: Level concrete



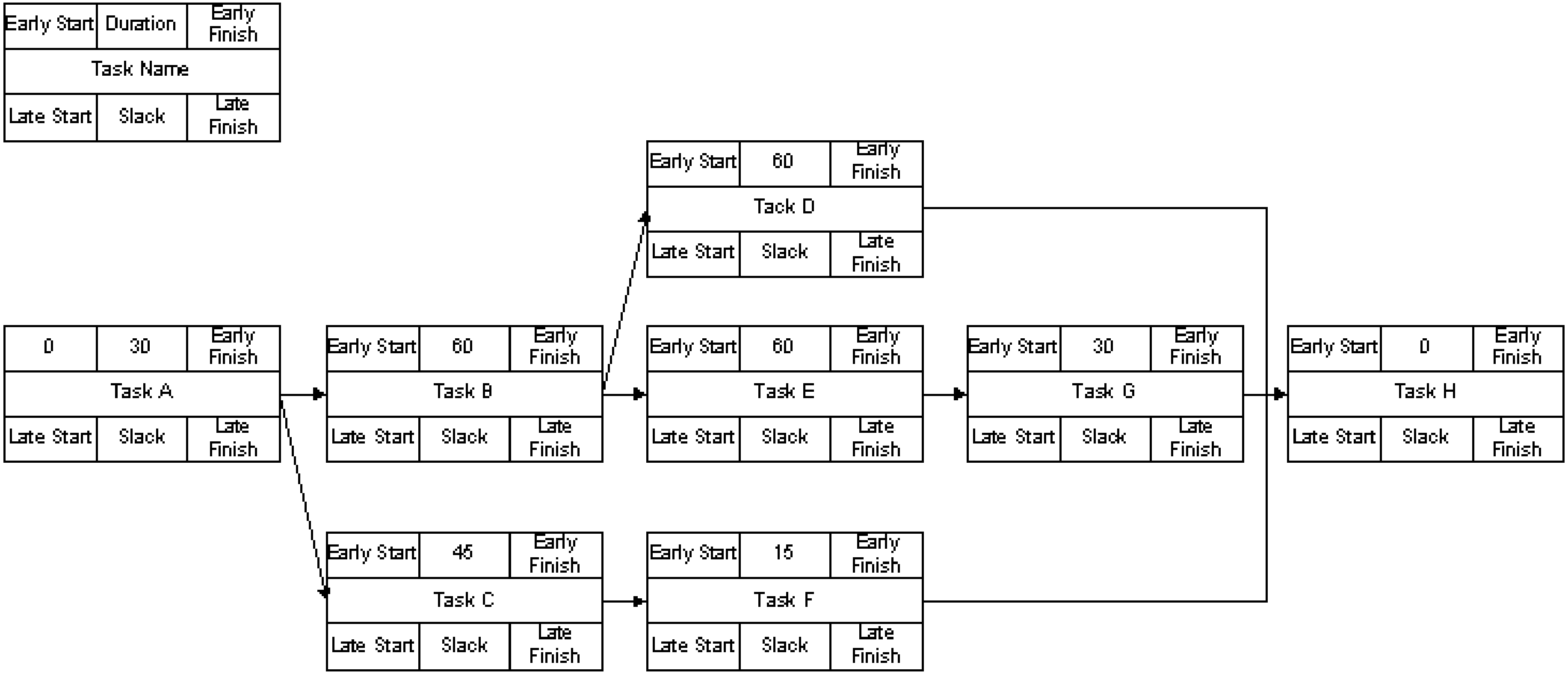
- Finish-to-Finish (FF)
  - B cannot finish till A finishes
  - A: Add wiring; B: Inspect electrical



- 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

0	30	Early Finish
Task A		
Late Start	Slack	Late Finish

Early Start	60	Early Finish
Task B		
Late Start	Slack	Late Finish

Early Start	45	Early Finish
Task C		
Late Start	Slack	Late Finish

Early Start	60	Early Finish
Tack D		
Late Start	Slack	Late Finish

Early Start	60	Early Finish
Task E		
Late Start	Slack	Late Finish

Early Start	15	Early Finish
Task F		
Late Start	Slack	Late Finish

Early Start	30	Early Finish
Task G		
Late Start	Slack	Late Finish

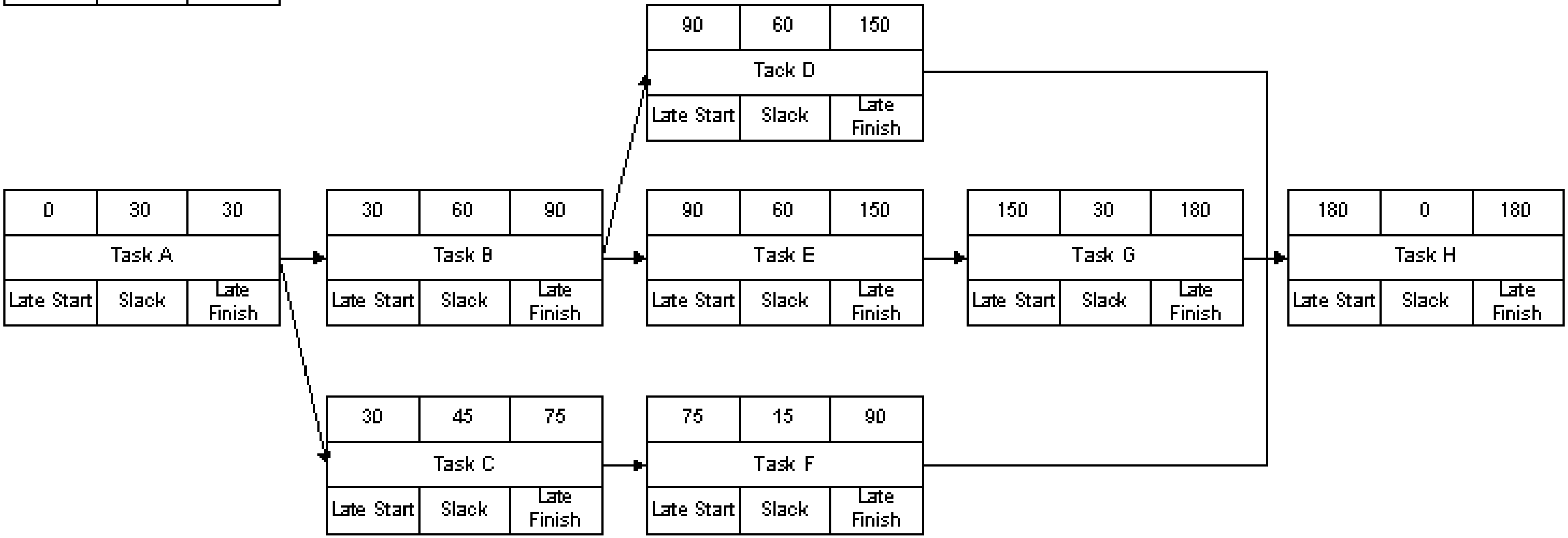
Early Start	0	Early Finish
Task H		
Late Start	Slack	Late Finish

Forward Pass

- 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

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



0	30	30
Task A		
Late Start	Slack	Late Finish

30	60	90
Task B		
Late Start	Slack	Late Finish

30	45	75
Task C		
Late Start	Slack	Late Finish

90	60	150
Tack D		
Late Start	Slack	Late Finish

90	60	150
Task E		
Late Start	Slack	Late Finish

75	15	90
Task F		
Late Start	Slack	Late Finish

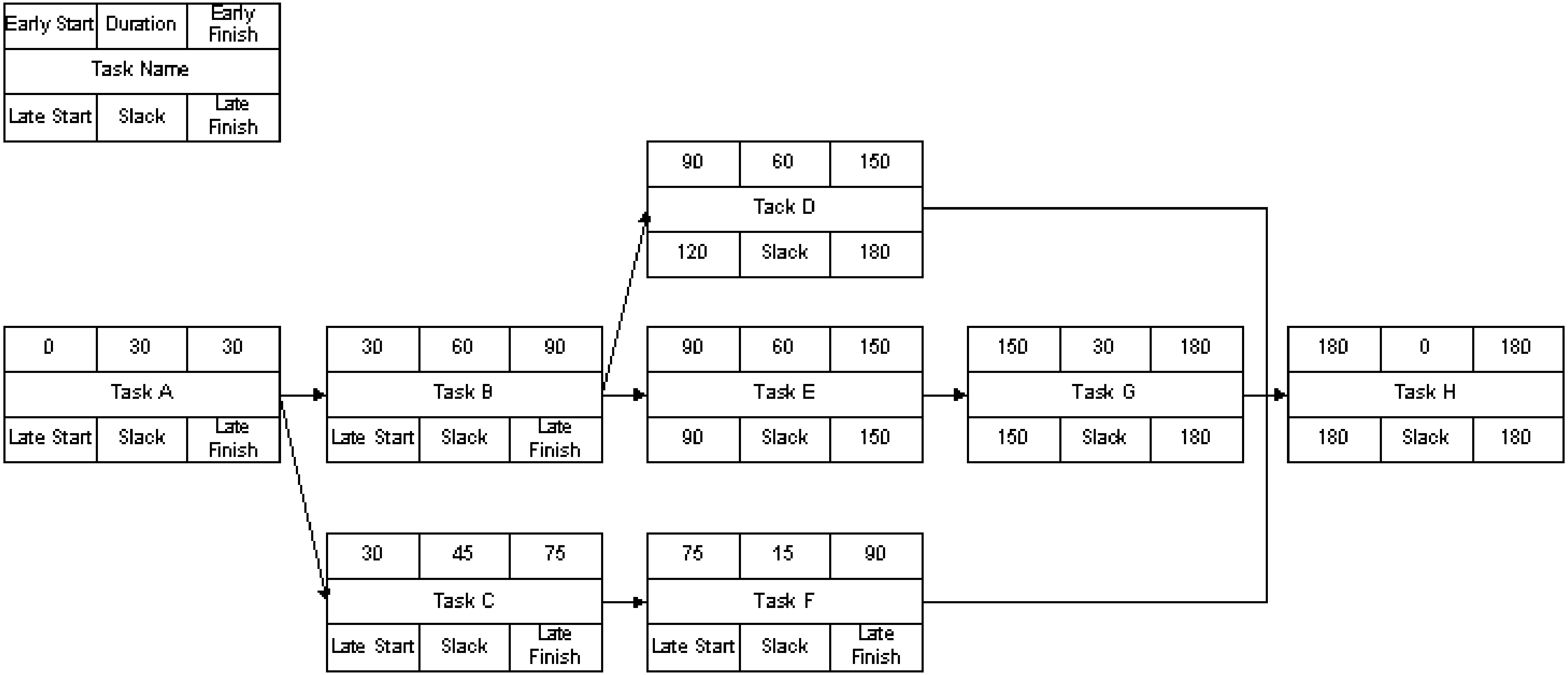
150	30	180
Task G		
Late Start	Slack	Late Finish

180	0	180
Task H		
Late Start	Slack	Late Finish

Backward Pass

- 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



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

0	30	30
Task A		
Late Start	Slack	Late Finish

30	60	90
Task B		
Late Start	Slack	Late Finish

30	45	75
Task C		
Late Start	Slack	Late Finish

90	60	150
Tack D		
120	Slack	180

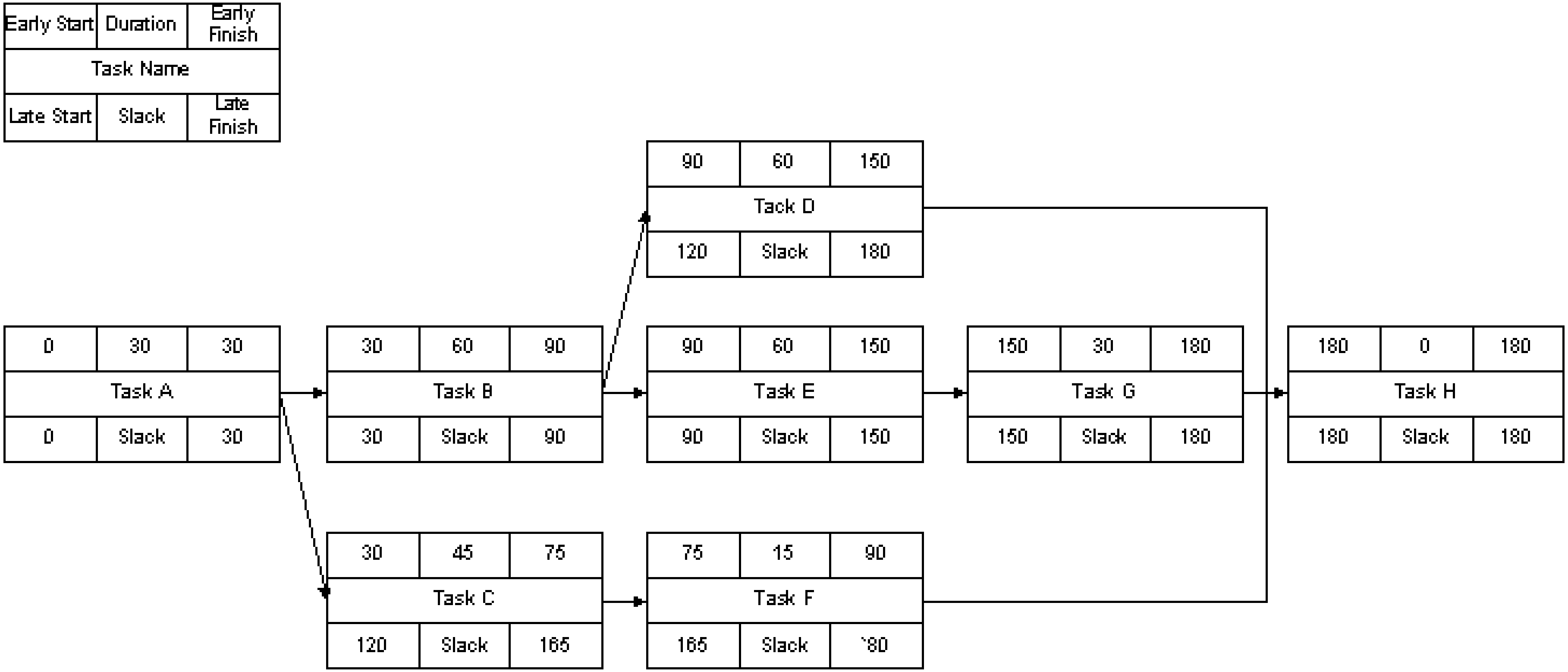
90	60	150
Task E		
90	Slack	150

75	15	90
Task F		
Late Start	Slack	Late Finish

150	30	180
Task G		
150	Slack	180

180	0	180
Task H		
180	Slack	180

Example Step 4



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

0	30	30
Task A		
0	Slack	30

30	60	90
Task B		
30	Slack	90

30	45	75
Task C		
120	Slack	165

90	60	150
Tack D		
120	Slack	180

90	60	150
Task E		
90	Slack	150

75	15	90
Task F		
165	Slack	`80

150	30	180
Task G		
150	Slack	180

180	0	180
Task H		
180	Slack	180

Network Diagrams

- 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

- **P**rogram **E**valuation and **R**eview **T**echnique
- 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
  - 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

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

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)

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

- For the whole critical path use

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

PERT Example

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

- 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)

PERT

- 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

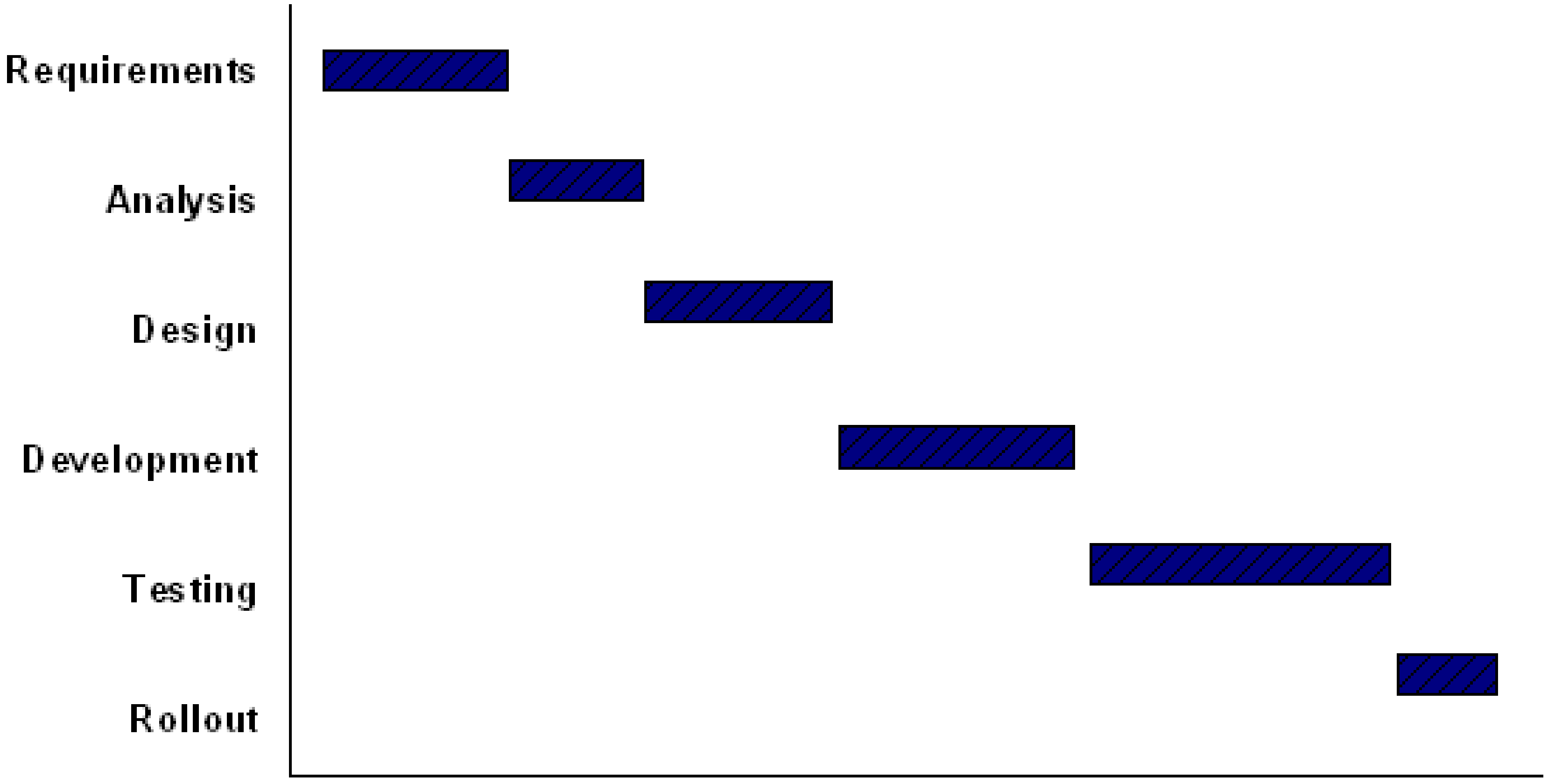
CPM vs. PERT

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

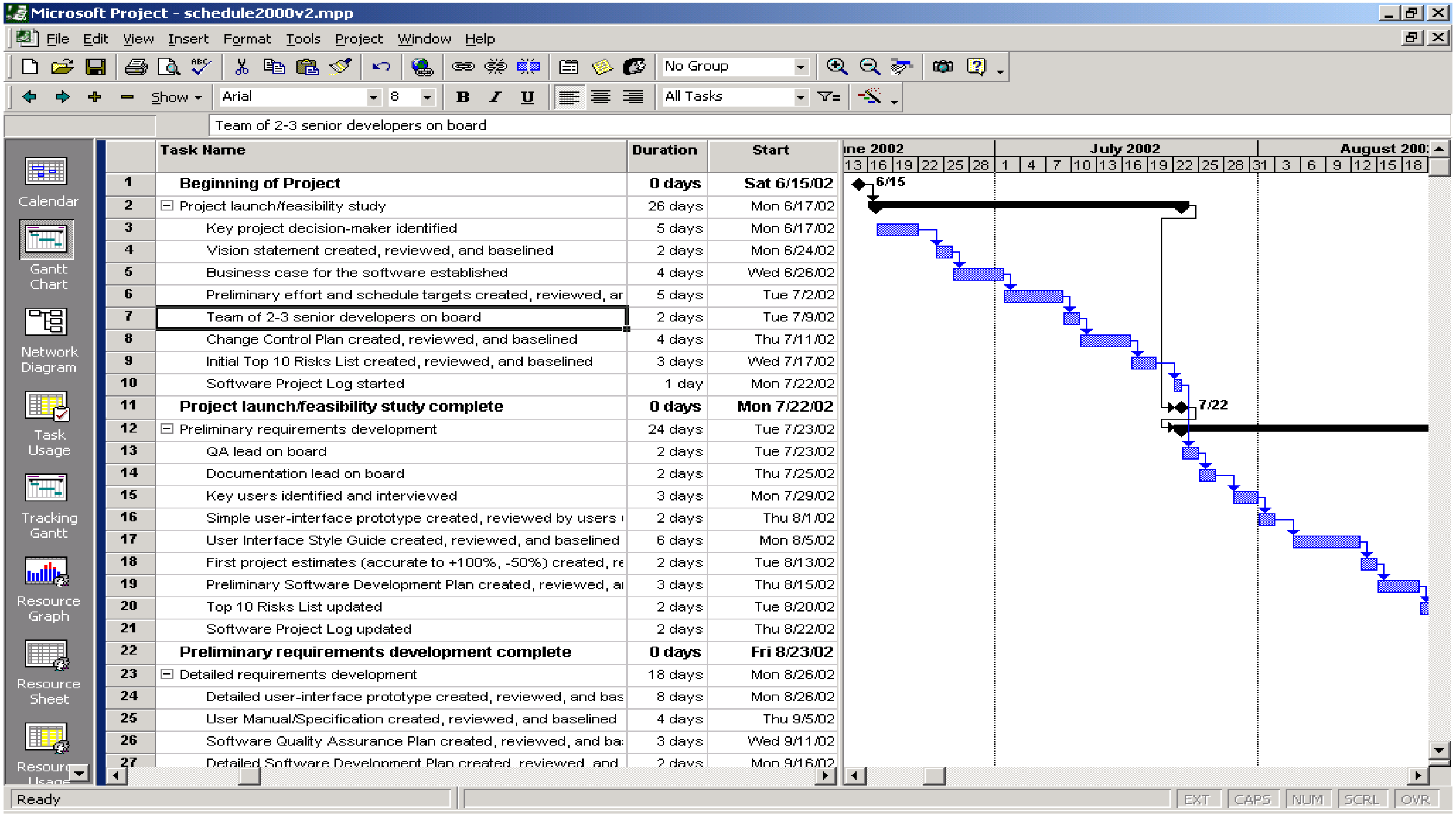
Milestone Chart

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

Bar Chart


Requirements
Analysis
Design
Development
Testing
Rollout

Gantt Chart



Team of 2-3 senior developers on board

	Task Name	Duration	Start
1	**Beginning of Project**	**0 days**	**Sat 6/15/02**
2	⊟ Project launch/feasibility study	26 days	Mon 6/17/02
3	Key project decision-maker identified	5 days	Mon 6/17/02
4	Vision statement created, reviewed, and baselined	2 days	Mon 6/24/02
5	Business case for the software established	4 days	Wed 6/26/02
6	Preliminary effort and schedule targets created, reviewed, ar	5 days	Tue 7/2/02
7	Team of 2-3 senior developers on board	2 days	Tue 7/9/02
8	Change Control Plan created, reviewed, and baselined	4 days	Thu 7/11/02
9	Initial Top 10 Risks List created, reviewed, and baselined	3 days	Wed 7/17/02
10	Software Project Log started	1 day	Mon 7/22/02
11	**Project launch/feasibility study complete**	**0 days**	**Mon 7/22/02**
12	⊟ Preliminary requirements development	24 days	Tue 7/23/02
13	QA lead on board	2 days	Tue 7/23/02
14	Documentation lead on board	2 days	Thu 7/25/02
15	Key users identified and interviewed	3 days	Mon 7/29/02
16	Simple user-interface prototype created, reviewed by users (	2 days	Thu 8/1/02
17	User Interface Style Guide created, reviewed, and baselined	6 days	Mon 8/5/02
18	First project estimates (accurate to +100%, -50%) created, re	2 days	Tue 8/13/02
19	Preliminary Software Development Plan created, reviewed, ai	3 days	Thu 8/15/02
20	Top 10 Risks List updated	2 days	Tue 8/20/02
21	Software Project Log updated	2 days	Thu 8/22/02
22	**Preliminary requirements development complete**	**0 days**	**Fri 8/23/02**
23	⊟ Detailed requirements development	18 days	Mon 8/26/02
24	Detailed user-interface prototype created, reviewed, and bas	8 days	Mon 8/26/02
25	User Manual/Specification created, reviewed, and baselined	4 days	Thu 9/5/02
26	Software Quality Assurance Plan created, reviewed, and ba:	3 days	Wed 9/11/02
27	Detailed Software Development Plan created, reviewed, and	2 days	Mon 9/16/02

Gantt Chart

- Disadvantages
  - Does not show interdependencies well
  - No 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

Reducing Project Duration

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

Compression Techniques

- 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

Mythical Man-Month

- Book: "The Mythical Man-Month"
  - Author: Fred Brooks
- "*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”

Mythical Man-Month

- Why is software project disaster so common?
  - 1. Estimation techniques are poor and 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
  - 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)
  - His formula: n(n-1)/2
- Don't assume more 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 (the most linear process)
- Why is this particularly bad?
    - Occurs late in process and without warning
    - Higher costs: primary and secondary
- Fix: Allocate more test time
    - Understand task dependencies

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