Software Project Scheduling

CO3: Student will be able to **Plan, schedule** and **track** the progress of the projects.

Eight Reasons for Late Software Delivery

- An unrealistic deadline established by someone outside the software engineering group and forced on managers and practitioners within the group
- Changing customer requirements that are not reflected in schedule changes
- An honest underestimate of the amount of effort and /or the number of resources that will be required to do the job
- Predictable and/or unpredictable risks that were not considered when the project commenced
- Technical difficulties that could not have been foreseen in advance
- Human difficulties that could not have been foreseen in advance
- Miscommunication among project staff that results in delays
- A failure by project management to recognize that the project is falling behind schedule and a lack of action to correct the problem

Software project scheduling

- Software project scheduling is an activity that distributes estimated effort across the planned project duration by allocating the effort to specific software engineering tasks.

Basic principles that guide software project scheduling

- Compartmentalization
- Interdependency
- Time allocation
- Effort validation
- Defined responsibilities
- Defined outcomes
- Defined milestones.

Basic Principles for Project Scheduling

- Compartmentalization
  - The project must be compartmentalized into a number of manageable activities, actions, and tasks; both the product and the process are decomposed
- Interdependency
  - The interdependency of each compartmentalized activity, action, or task must be determined
  - Some tasks must occur in sequence while others can occur in parallel
  - Some actions or activities cannot commence until the work product produced by another is available
- Time allocation
  - Each task to be scheduled must be allocated some number of work units
  - In addition, each task must be assigned a start date and a completion date that are a function of the interdependencies
  - Start and stop dates are also established based on whether work will be conducted on a full-time or part-time basis

Basic Principles for Project Scheduling (continued)

- Effort validation
  - Every project has a defined number of people on the team
  - As time allocation occurs, the project manager must ensure that no more than the allocated number of people have been scheduled at any given time
- Defined responsibilities
  - Every task that is scheduled should be assigned to a specific team member
- Defined outcomes
  - Every task that is scheduled should have a defined outcome for software projects such as a work product or part of a work product
  - Work products are often combined in deliverables
- Defined milestones
  - Every task or group of tasks should be associated with a project milestone
  - A milestone is accomplished when one or more work products has been reviewed for quality and has been approved

Relationship Between People and Effort

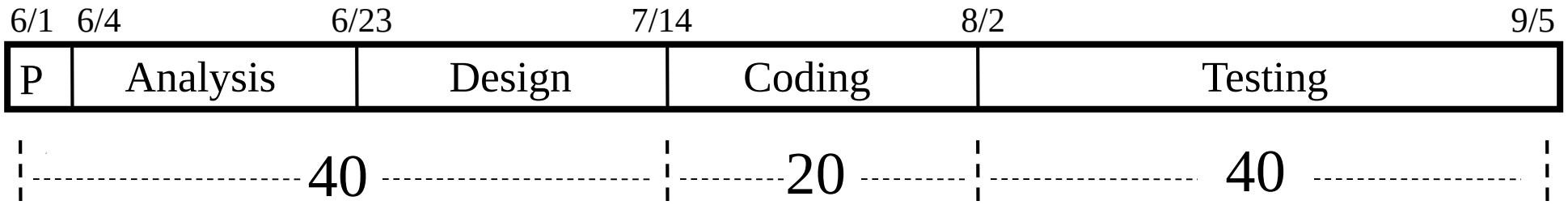
- Common management myth: *If we fall behind schedule, we can always add more programmers and catch up later in the project*
  - This practice actually has a disruptive effect and causes the schedule to slip even further
  - The added people must learn the system
  - The people who teach them are the same people who were earlier doing the work
  - During teaching, no work is being accomplished
  - Lines of communication (and the inherent delays) increase for each new person added

40-20-40 Distribution of Effort

- A recommended distribution of effort across the software process is 40% (analysis and design), 20% (coding), and 40% (testing)
- Work expended on project planning rarely accounts for more than 2 - 3% of the total effort
- Requirements analysis may comprise 10 - 25%
  - Effort spent on prototyping and project complexity may increase this
- Software design normally needs 20 – 25%
- Coding should need only 15 - 20% based on the effort applied to software design
- Testing and subsequent debugging can account for 30 - 40%
  - Safety or security-related software requires more time for testing

40-20-40 Distribution of Effort (continued)

Example: 100-day project



Task Network

Defining a task set for the software project

- A task set is a collection of software engineering work tasks, milestones, work products, and quality assurance filters that must be accomplished to complete a particular project.
- In order to develop a project schedule, a task set must be distributed on the project time line.
- The task set will vary depending upon the project type and the degree of rigor with which the software team decides to do its work.

Most software organizations encounter the following projects

1. **Concept development projects** that are initiated to explore some new business concept or application of some new technology.
2. **New application development projects** that are undertaken as a consequence of a specific customer request.
3. **Application enhancement projects** that occur when existing software undergoes major modifications to function, performance, or interfaces that are observable by the end user.
4. **Application maintenance projects** that correct, adapt, or extend existing software in ways that may not be immediately obvious to the end user.
5. **Reengineering projects** that are undertaken with the intent of rebuilding an existing (legacy) system in whole or in part.

A Task Set Example

Concept development projects are approached by applying the following major tasks:

1. **Concept scoping** determines the overall scope of the project.
2. **Preliminary concept planning** establishes the organization's ability to undertake the work implied by the project scope.
3. **Technology risk assessment** evaluates the risk associated with the technology to be implemented as part of the project scope.
4. **Proof of concept** demonstrates the viability of a new technology in the software context.
5. **Concept implementation** implements the concept representation in a manner that can be reviewed by a customer and is used for "marketing" purposes when a concept must be sold to other customers or management.
6. **Customer reaction** to the concept solicits feedback on a new technology concept and targets specific customer applications.

Refinement of Major Tasks

Task definition: Task 1.1 Concept Scoping

1.1.1 Identify need, benefits and potential customers;

1.1.2 Define desired output/control and input events that drive the application;

Begin Task 1.1.2

1.1.2.1 TR: Review written description of need[7]

1.1.2.2 Derive a list of customer visible outputs/inputs

1.1.2.3 TR: Review outputs/inputs with customer and revise as required;

endtask Task 1.1.2

1.1.3 Define the functionality/behavior for each major function;

Begin Task 1.1.3

1.1.3.1 TR: Review output and input data objects derived in task 1.1.2;

1.1.3.2 Derive a model of functions/behaviors;

1.1.3.3 TR: Review functions/behaviors with customer and revise as required;

endtask Task 1.1.3

1.1.4 Isolate those elements of the technology to be implemented in software;

1.1.5 Research availability of existing software;

1.1.6 Define technical feasibility;

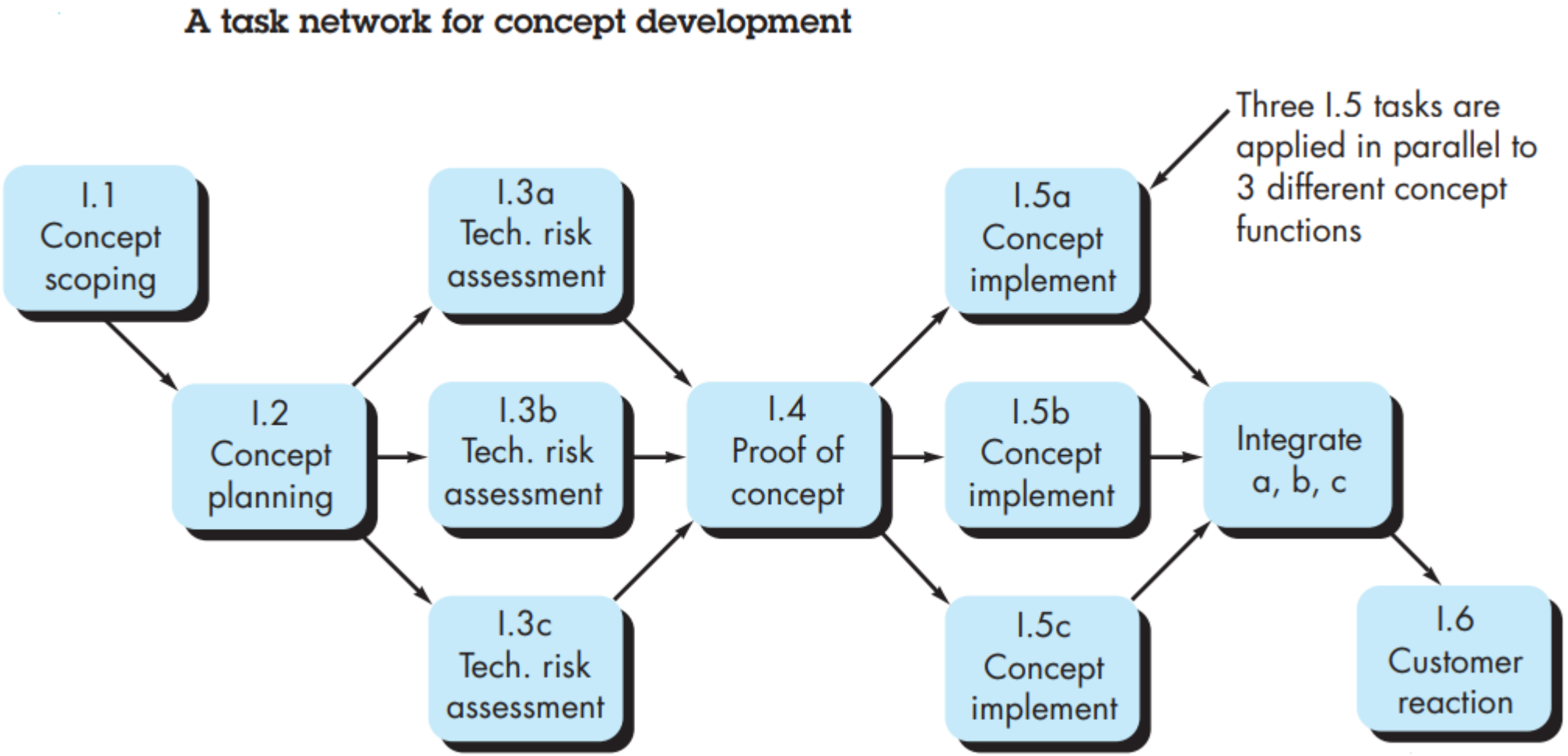
1.1.7 Make quick estimate of size;

1.1.8 Create a Scope Definition;

endtask definition: Task 1.1

Defining a task network

- A task network, also called an activity network, is a graphic representation of the task flow for a project.



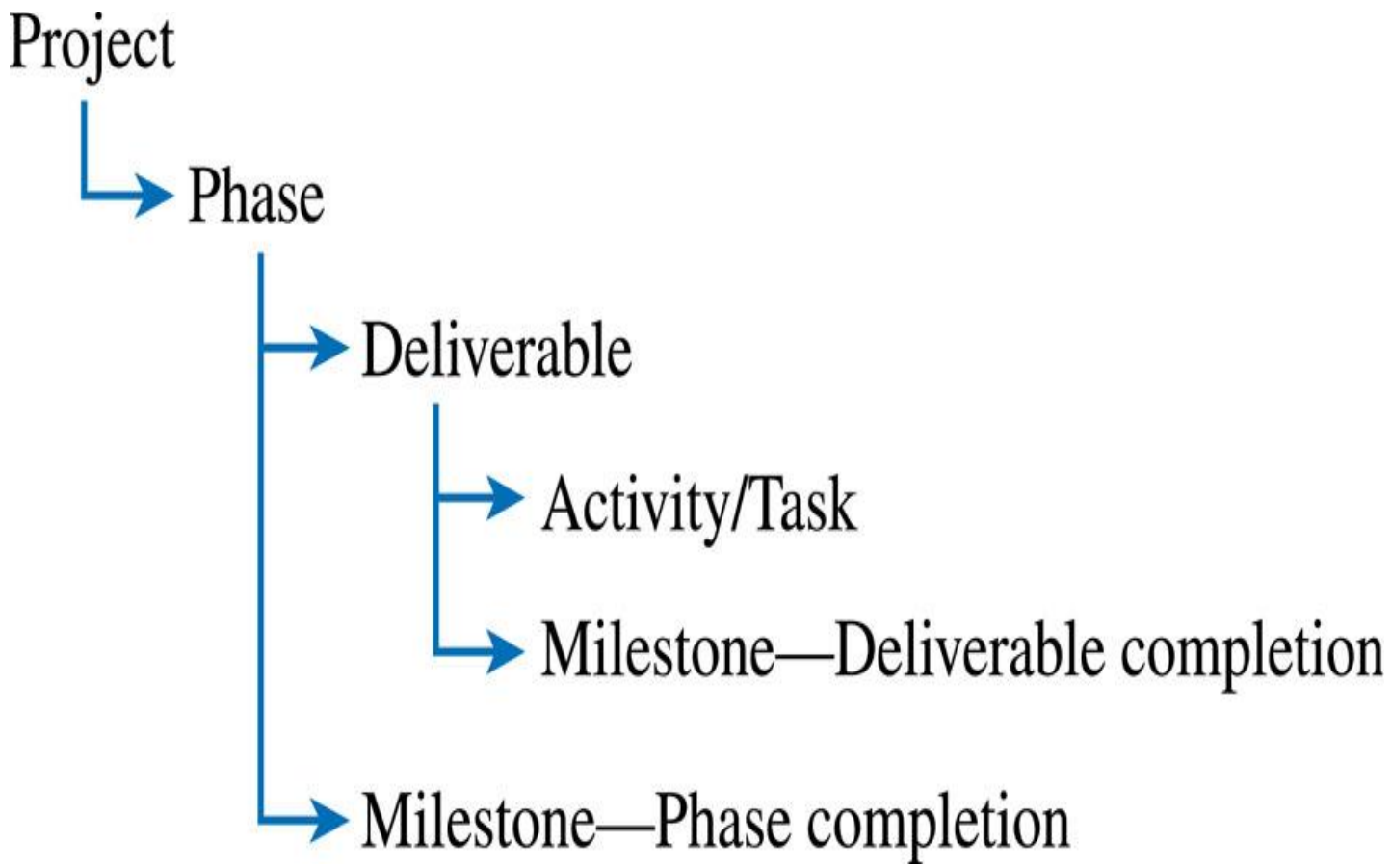
Scheduling Methods

- Work Breakdown Structure (WBS)
- Critical Path Method (CPM)
- Program Evaluation and Review Technique (PERT)

The Work Breakdown Structure (WBS)

- The WBS represents a logical decomposition of the work to be performed and focuses on how the product, service, or result is naturally subdivided. It is an outline of what work is to be performed.

Work Package



Deliverables and Milestones

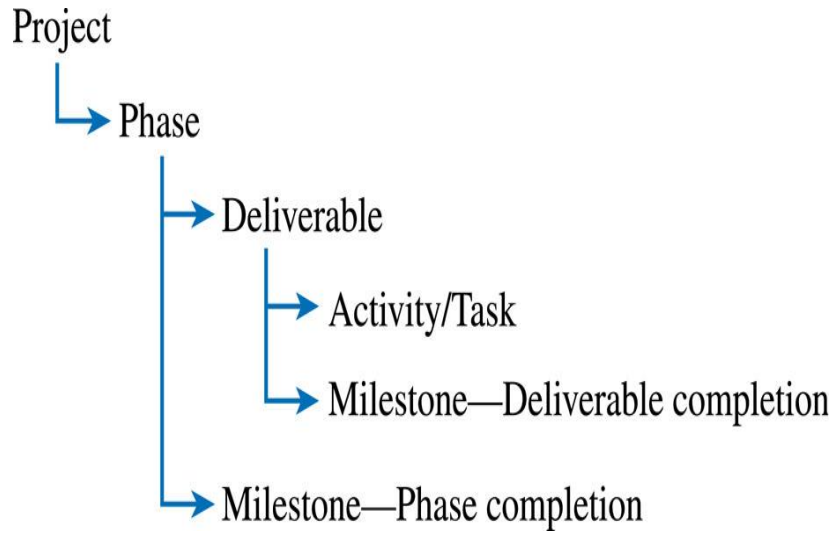
- Deliverables
  - Tangible, verifiable work products
  - Reports, presentations, prototypes, etc.
- Milestones
  - Significant events or achievements
  - Acceptance of deliverables or phase completion
  - Cruxes (proof of concepts)
  - Quality control
  - Keeps team focused

Example Work Breakdown Schedule

–0.0 EC Bank Project
- +1.0 Conceptualize & initialize project
- +2.0 Develop charter & plan
- +3.0 Analysis
- +4.0 Design
- +5.0 Construction
- –6.0 Testing
  - +6.1 Test plan
  - –6.2 Test results report
    - 6.2.1 Review test plan with client
    - 6.2.2 Carry out test plan
    - 6.2.3 Analyze results
    - 6.2.4 Prepare test results report and presentation
    - 6.2.5 Present test results to client
    - 6.2.6 Address any software issues or problems
    - 6.2.7 **Milestone:** client signs off on test results
  - +6.3 **Milestone:** testing completed
- +7.0 Implementation
- +8.0 Close project
- +9.0 Evaluate project success

The WBS Should Follow the Work Package Concept

–0.0 EC Bank Project
- +1.0 Conceptualize & initialize project
- +2.0 Develop charter & plan
- +3.0 Analysis
- +4.0 Design
- +5.0 Construction
- –6.0 Testing
  - +6.1 Test plan
  - –6.2 Test results report
    - 6.2.1 Review test plan with client
    - 6.2.2 Carry out test plan
    - 6.2.3 Analyze results
    - 6.2.4 Prepare test results report and presentation
    - 6.2.5 Present test results to client
    - 6.2.6 Address any software issues or problems
    - 6.2.7 **Milestone:** client signs off on test results
  - +6.3 **Milestone:** testing completed
- +7.0 Implementation
- +8.0 Close project
- +9.0 Evaluate project success



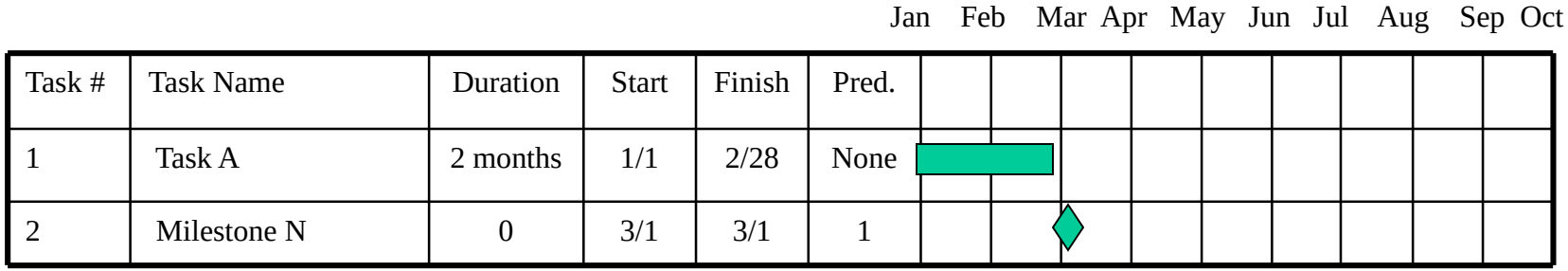
Example WBS with Estimated Task Durations

6.2 Test Results Report	
6.2.1 Review test plan with client	1 day
6.2.2 Carry out test plan	5 days
6.2.3 Analyze results	2 days
6.2.4 Prepare test results report and presentation	3 days
6.2.5 Present test results to client	1 day
6.2.6 Address any software issues or problems	5 days

Timeline Chart

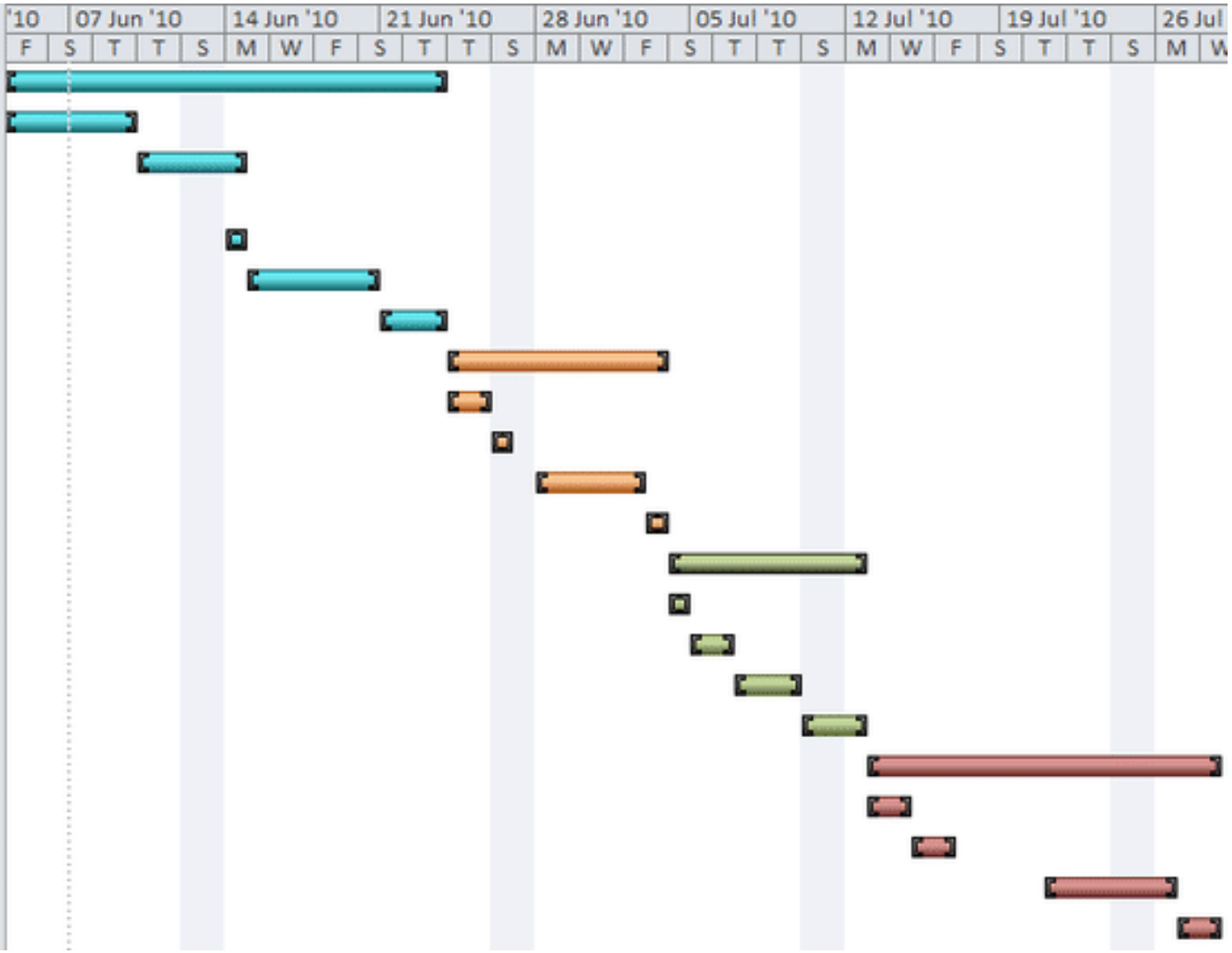
Mechanics of a Timeline Chart

- Also called a Gantt chart; invented by Henry Gantt, industrial engineer, 1917
- All project tasks are listed in the far left column
- The next few columns may list the following for each task: projected start date, projected stop date, projected duration, actual start date, actual stop date, actual duration, task inter-dependencies (i.e., predecessors)
- To the far right are columns representing dates on a calendar
- The length of a horizontal bar on the calendar indicates the duration of the task
- When multiple bars occur at the same time interval on the calendar, this implies task concurrency
- A diamond in the calendar area of a specific task indicates that the task is a milestone; a milestone has a time duration of zero



Gantt Chart Example

Task Name	Duration
Iteration 1 - Server Component	14 days
Initial Research on Protocols	4 days
Investigate suitable Communication Protocol	3 days
GUI Design	1 day
Coding Phase	5 days
Debug & Testing	3 days
Iteration 2 - Client Component	8 days
Study Java Socket API	2 days
Design Client GUI - Register	1 day
Coding Phase	5 days
Debug & Testing	1 day
Iteration 3 - Asynchronous Draw	7 days
Design GUI	1 day
Design Custom Shapes	2 days
Coding Phase	3 days
Debug & Testing	2 days
Iteration 4 - Synchronous Draw	12 days
Design Client & Server	2 days
Design Synchronisation Mechanisms	2 days
Coding Phase	4 days
Debuging & Testing	2 days

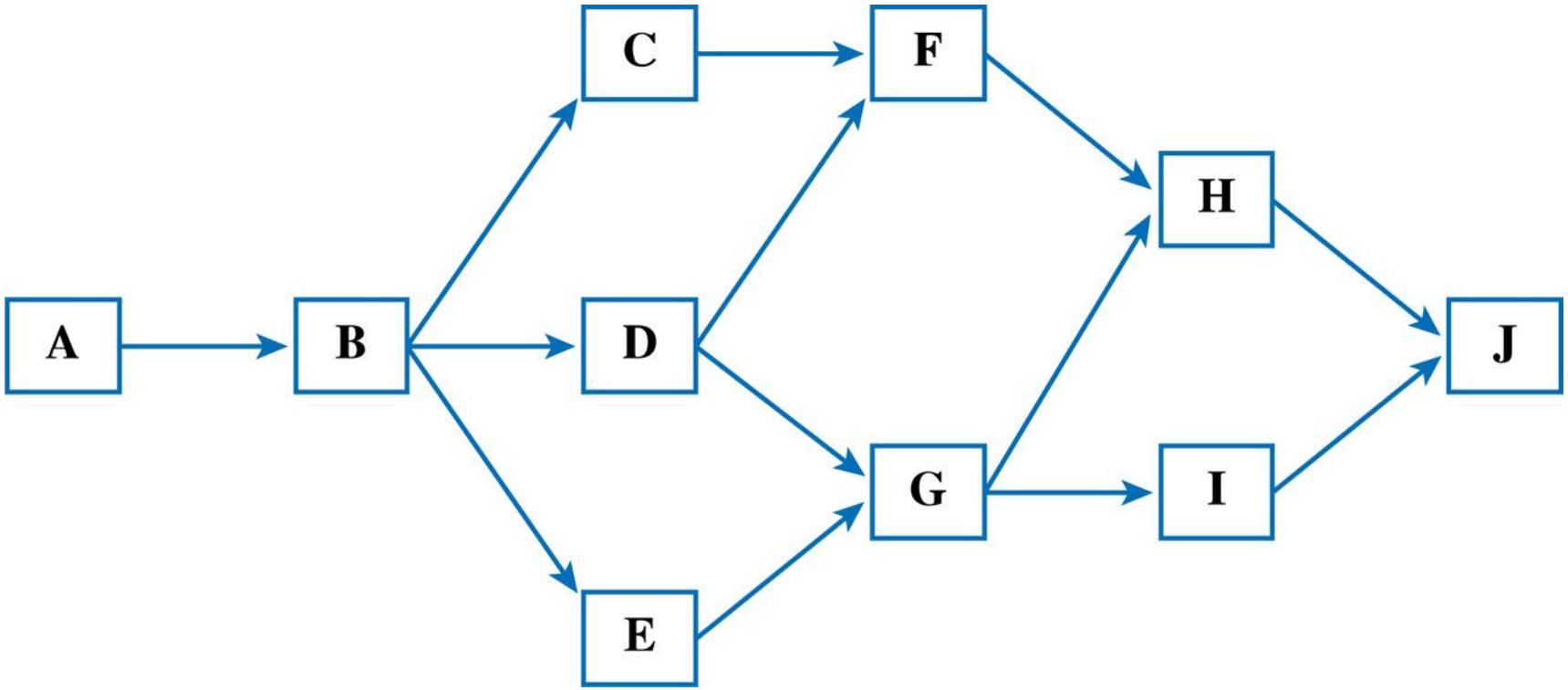


Project Network Diagram

Activity Analysis for AON

Activity	Description	Estimated Duration (Days)	Predecessor
A	Evaluate current technology platform	2	None
B	Define user requirements	5	A
C	Design Web page layouts	4	B
D	Set-up Server	3	B
E	Estimate Web traffic	1	B
F	Test Web pages and links	4	C,D
G	Move web pages to production environment	3	D,E
H	Write announcement of intranet for corp. newsletter	2	F,G
I	Train users	5	G
J	Write report to management	1	H,I

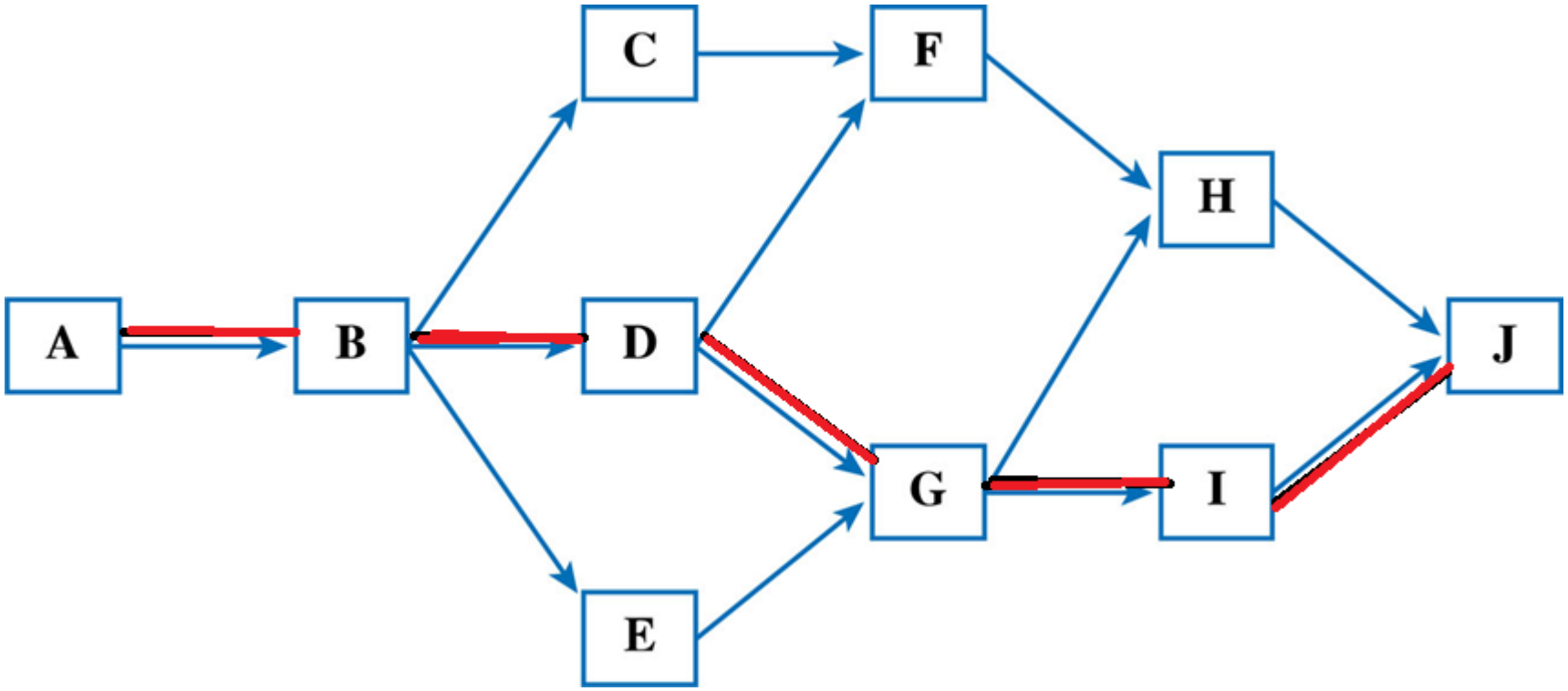
Activity on the Node (AON) Network Diagram



Possible Activity Paths

Possible Paths	Path	Total
Path 1	A+B+C+F+H+J	18
	2+5+4+4+2+1	
Path 2	A+B+D+F+H+J	17
	2+5+3+4+2+1	
Path 3	A+B+D+G+H+J	16
	2+5+3+3+2+1	
Path 4	A+B+D+G+I+J	19*
	2+5+3+3+5+1	
Path 5	A+B+E+G+I+J	17
	2+5+1+3+5+1	

Activity on the Node (AON) Network Diagram



Critical Path

- Longest path
- Shortest time project can be completed
  - Zero slack (or float)
    - The amount of time an activity can be delayed before it delays the project
- Must be monitored and managed!
  - Project manager can expedite or crash by adding resources
  - Fast tracking – running activities in parallel which were originally planned as sequential
  - The CP can change
  - Can have multiple CPs

PERT

- Program Evaluation and Review Technique
- Developed in 1950s to help manage the Polaris Submarine Project
- Developed about the same time as the Critical Path Method
  - Often combined as PERT/CPM
- Employs both a project network diagram with a statistical distribution

Activity Analysis for PERT

Activity	Predecessor	Optimistic Estimates (Days) (a)	Most Likely Estimates (Days) (b)	Pessimistic Estimates (Days) (c)	Expected Duration $\frac{(a+4b+c)}{6}$
A	None	1	2	4	2.2
B	A	3	5	8	5.2
C	B	2	4	5	3.8
D	B	2	3	6	3.3
E	B	1	1	1	1.0
F	C,D	2	4	6	4.0
G	D,E	2	3	4	3.0
H	F,G	1	2	5	2.3
I	G	4	5	9	5.5
J	H,I	.5	1	3	1.3

Possible PERT Activity Paths

Possible Paths	Path	Total
Path 1	A+B+C+F+H+J	18.8
	2.2+5.2+3.8+4.0+2.3+1.3	
Path 2	A+B+D+F+H+J	18.3
	2.2+5.2+3.3+4.0+2.3+1.3	
Path 3	A+B+D+G+H+J	18.6
	2.2+5.2+3.3+3.0+2.3+1.3	
Path 4	A+B+D+G+I+J	20.5*
	2.2+5.2+3.3+3.0+5.5+1.3	
Path 5	A+B+E+G+I+J	18.2
	2.2+5.2+1.0+3.0+5.5+1.3	

Methods for Tracking the Schedule

- Qualitative approaches
  - Conduct periodic project status meetings in which each team member reports progress and problems
  - Evaluate the results of all reviews conducted throughout the software engineering process
  - Determine whether formal project milestones (i.e., diamonds) have been accomplished by the scheduled date
  - Compare actual start date to planned start date for each project task listed in the timeline chart
  - Meet informally with the software engineering team to obtain their subjective assessment of progress to date and problems on the horizon
- Quantitative approach
  - Use earned value analysis to assess progress quantitatively

> "The basic rule of software status reporting can be summarized in a single phrase: No surprises." Capers Jones

Earned Value Analysis

Description of Earned Value Analysis

- Earned value analysis is a measure of progress by assessing the percent of completeness for a project
- It gives accurate and reliable readings of performance very early into a project
- It provides a common value scale (i.e., time) for every project task, regardless of the type of work being performed
- The total hours to do the whole project are estimated, and every task is given an earned value based on its estimated percentage of the total

Earned Value

- Suppose you just signed a contract with a consulting firm called Dewey, Cheatem, and Howe for developing an IS.
- Project Budget, Schedule, Tasks
  - $40,000
  - 4 months
  - 20 Tasks (evenly divided over 4 months)
    - $2,000 per task
    - 5 tasks per month

The Planned Project Schedule And Budget

Task	Month 1	Month 2	Month 3	Month 4
1	$2,000			
2	$2,000			
3	$2,000			
4	$2,000			
5	$2,000			
6		$2,000		
7		$2,000		
8		$2,000		
9		$2,000		
10		$2,000		
11			$2,000	
12			$2,000	
13			$2,000	
14			$2,000	
15			$2,000	
16				$2,000
17				$2,000
18				$2,000
19				$2,000
20				$2,000
Total	**$10,000**	**$10,000**	**$10,000**	**$10,000**

Earned Value Concepts

- Planned Value (PV)
  - The planned or budgeted cost of work scheduled for an activity or component of the WBS
    - In our case, our planned value for each task is $2,000
    - The planned value for each month is $10,000
- Budgeted At Completion (BAC)
  - The total budget for our project
    - In our case, $40,000 is our BAC since this is what we expect to pay for the completed project
    - The BAC is the total cumulative planned value

At the end of Month 1, we received the following invoice…

Invoice

Dewey, Cheatem, and Howe

Amount Due: $8,000.00

Payment Due: Immediately

Page 1 of 2

This Looks Like Good News!

- We expected to pay $10,000 but we're only being billed for $8,000
  - Are we really ahead of our budgeted or planned value by $2,000?
- It depends on what work was accomplished for the $8,000 that is due

Therefore, we need to look at the rest of the invoice to be sure

It appears that only three of the five tasks scheduled to be completed in Month 1 were completed as planned. In fact, two of the tasks cost more to complete than originally estimated.

Maybe things are not as good as we thought!

Invoice

Dewey, Cheatem, and Howe

Work Completed for Month 1

Task 1: $2,000
Task 2: $3,000
Task 3: $3,000

Page 2 of 2

Some More Earned Value Concepts

- Actual Cost (AC)
  - The actual cost incurred for completing an activity or component of the WBS
    - For example, the actual cost for completing task 2 is $3,000
    - Or, we have to pay $8,000 for the three tasks that were completed in Month 1
- Earned Value (EV)
  - A performance measurement that tells us how much of the budget we really should have spent for the work that was completed
  - We need to pay our consultants $8,000 in actual costs even though we should be paying them only $6,000
  - This $6,000 is called the earned value

Planned, Actual, & Earned Values for Month 1

Task	Planned	Actual	Earned
1	$2,000	$2,000	$2,000
2	$2,000	$3,000	$2,000
3	$2,000	$3,000	$2,000
4	$2,000		
5	$2,000		
Cumulative	$10,000	$8,000	$6,000

What we *planned* to pay

What we *have* to pay

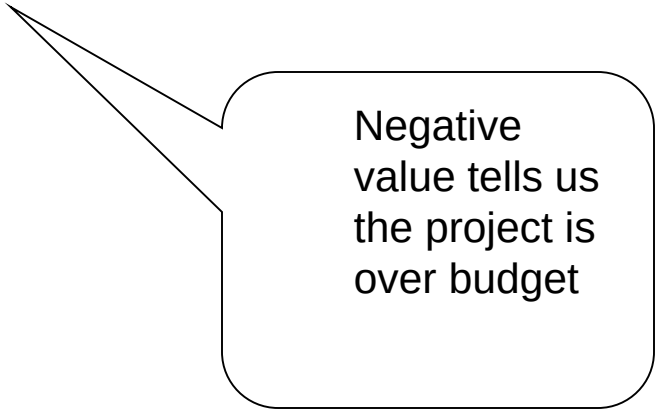
What we *should* pay

Cost Metrics

- Cost Variance (CV)-the difference between a task's or WBS component's estimated cost and its actual cost:
  - CV = EV - AC
    - Negative Value = over budget
    - Positive Value = under budget
    - Value = 0 means project is right on budget
- Cost Performance Index (CPI)-percentage of work completed per dollar spent
  - CPI = EV ÷ AC
    - ratio > 1 = ahead of budget
    - ratio < 1 = behind budget (cost overrun)
    - Ratio = 1 means project is right on budget

Cost Metrics

Cost Variance (CV) = EV – AC
= $6,000 - $8,000
= (-$2,000)

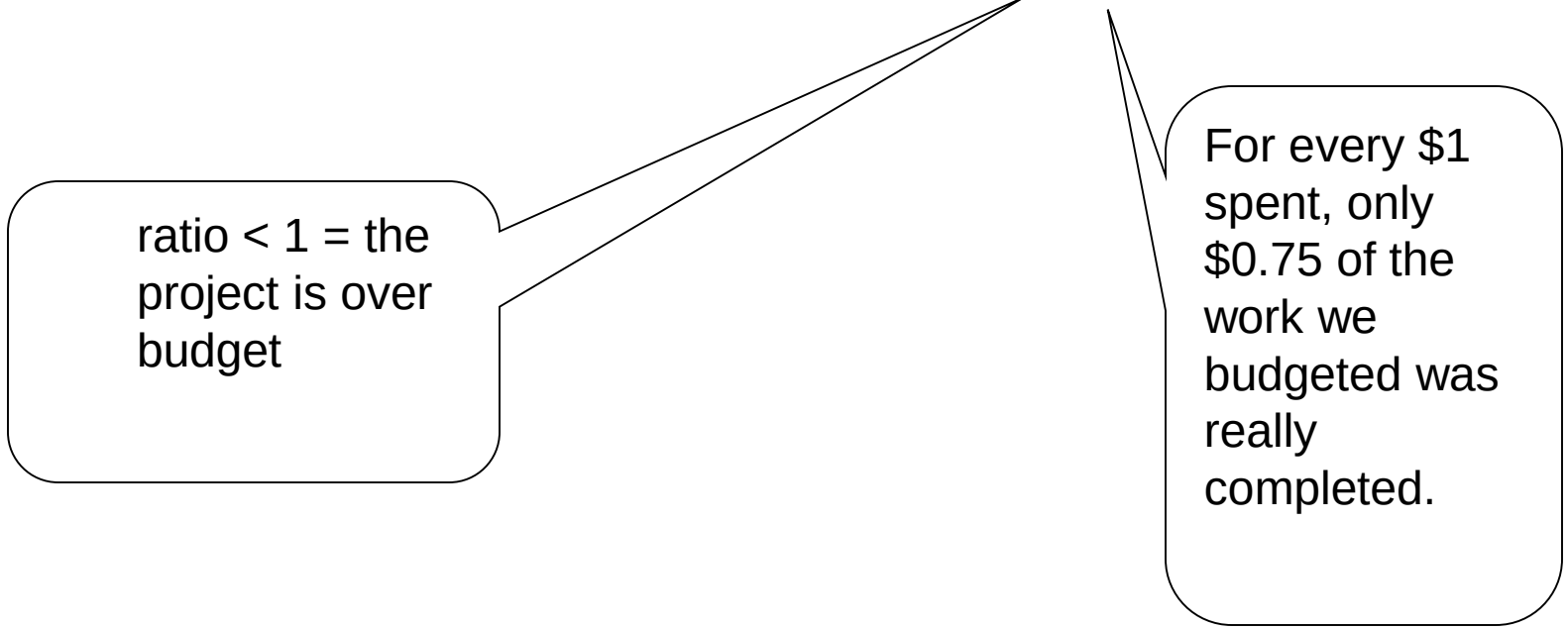


Cost Metrics

Cost Performance Index (CPI) = EV / AC

= $6,000 / $8,000

= .75

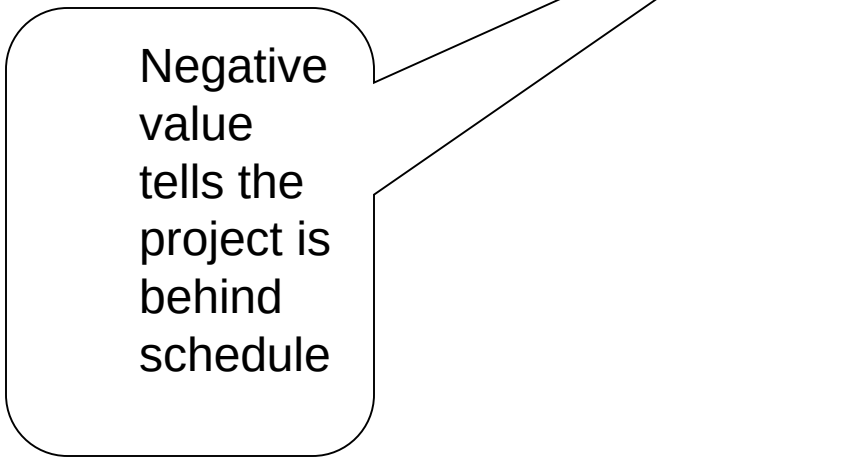


Schedule Metrics

- Schedule Variance (SV) – the difference between the current progress of the project and its original or planned schedule
  - SV = EV – PV
    - Negative Value = behind schedule
    - Positive Value = ahead of schedule
    - Value = 0 means project is right on schedule
- Schedule Performance Index (SPI) – a ratio of the work performed to the work scheduled.
  - SPI = EV ÷ PV
  - ratio > 1 = ahead of schedule
  - ratio < 1 = behind schedule
  - Ratio = 1 means our project is right on schedule

Schedule Metrics

Schedule Variance (SV) = EV – PV
= $6,000 - $10,000
= (-$4,000)

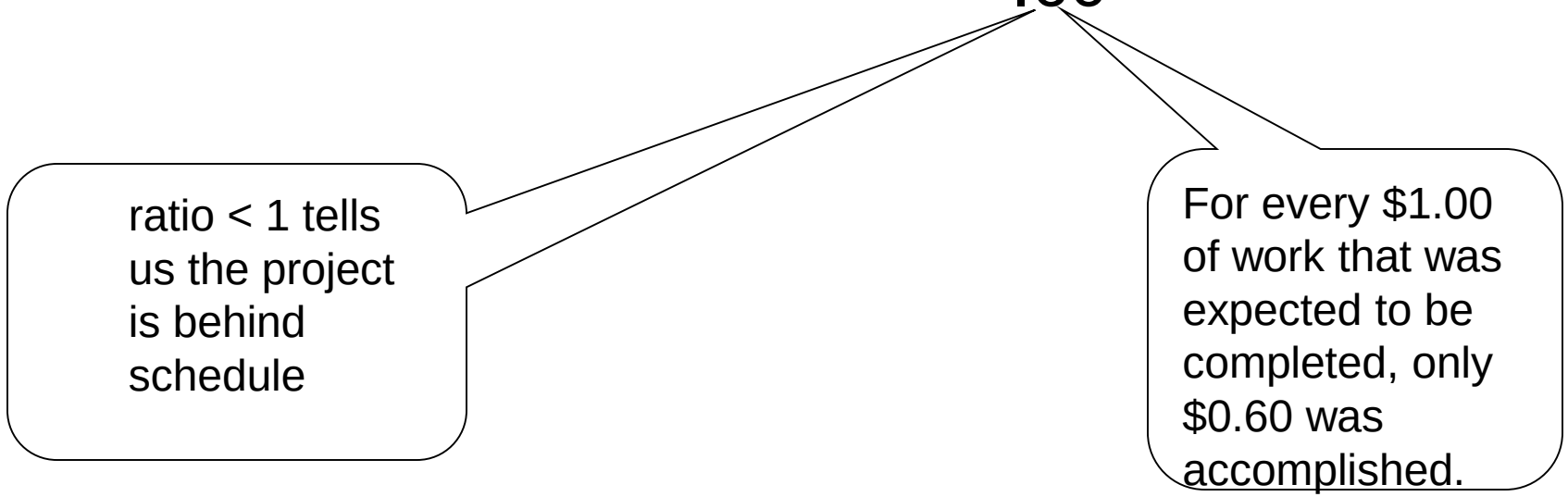


Schedule Metrics

Schedule Performance Index (SPI) = EV/PV

= $6,000 / $10,000

= .60



Summary of Project Performance Metrics

Task	Planned Value PV	Actual Cost AC	Earned Value EV	Cost Variance CV	Schedule Variance SV	Cost Performance Index CPI	Schedule Performance Index SPI
1	$2,000	$2,000	$2,000	-0-	-0-	1.00	1.00
2	$2,000	$3,000	$2,000	($1,000)	-0-	0.67	1.00
3	$2,000	$3,000	$2,000	($1,000)	-0-	0.67	1.00
4	$2,000				($2,000)	-	0.00
5	$2,000				($2,000)	-	0.00
Cumulative	**$10,000**	**$8,000**	**$6,000**	**($2,000)**	**($4,000)**	**0.75**	**0.60**

Notations & formulae from John Nicholas book

- BCWS : Budgeted cost of the work scheduled (PV, BAC)
- ACWP : Actual cost of the work performed (AC)
- BCWP : Budgeted cost of the work performed (EV)

- AV=BCWS-ACWP (Difference between current budget & current actual expenditure)
- SV=BCWP-BCWS
- CV=BCWP-ACWP
- SPI=BCWP/BCWS
- CPI=BCWP/ACWP