Project Management

CSE 4407

Md. Bakhtiar Hasan

Assistant Professor
Department of Computer Science and Engineering
Islamic University of Technology

May 21, 2025

The Spark: Origins of Systems Projects

- Project Triggers
  - Problems
  - Opportunities

The Spark: Origins of Systems Projects

- Project Triggers
  - Problems: Things aren't working as they should. Performance gaps, inefficiencies, errors
  - Opportunities

The Spark: Origins of Systems Projects

- Project Triggers
  - Problems: Things aren't working as they should. Performance gaps, inefficiencies, errors
  - Opportunities: Chance to improve, upgrade, or innovate. New tech, changing markets, e-commerce potential

Spotting Troubles: Signs of Underlying Problem

- Where to Look: How do we *detect* these problems?

Okay, We See Symptoms... Now What? Define the Problem!

- Why: It's the critical first step in any structured approach (like SDLC or O-O). Sets the foundation for the entire project

Uncovering the Real Issues: Analyst Detective Work

- How Analysts Identify Issues (During info gathering - interviews, observation, etc.)

Turning Problems into Goals: Setting Objectives

- Linking Issues and Objectives
  - Objectives should directly address the identified Issues (point-by-point if possible)

Case Study: Catherine's Catering Conundrum

- The Business: Small catering company (meals, receptions, banquets). Started small, good reputation led to growth

Demo: Problem Definition - Catherine's Catering

Catherine's Catering: Translating Objectives to User Requirements

- From Objectives to Action: User requirements specify *how* objectives will be met from a user perspective

Catherine's Catering: Thinking Ahead - Testing

- Why
  - Helps ensure requirements are *clear* and *testable*
  - Starts early, evolves over time

Not All Projects Are Created Equal: Selection Criteria

- Reality Check: Organizations have limited resources (time, money, people)

Project Go/No-Go: The Big Questions (1/2)

- Five Key Criteria for Selection

Project Go/No-Go: Practicality and Worth (2/2)

- Five Key Criteria for Selection
  4. Practical in Terms of Resources?
     - Do *we* (analysts/dev team) have the necessary skills and tools?
     - Does the *organization* have the capacity (staff, infrastructure)?
     - Recognize limitations – some projects might require external expertise

Selected the Project... But Is It Doable?

- Recap: We have narrowed down potential projects based on strategic fit, backing, timing, etc.

Can We *Really* Do This? The T.E.O. Test

- **T**echnical Feasibility: Do we have the tech and skills?
- **E**conomic Feasibility: Does it make financial sense?
- **O**perational Feasibility: Will people actually use it effectively?

Can We *Really* Do This? The T.E.O. Test

- **T**echnical Feasibility: Do we have the tech and skills?
  - Add on to present system
  - Technology available to meet users' needs
- **E**conomic Feasibility: Does it make financial sense?
- **O**perational Feasibility: Will people actually use it effectively?

Can We *Really* Do This? The T.E.O. Test

- **T**echnical Feasibility: Do we have the tech and skills?
  - Add on to present system
  - Technology available to meet users' needs
- **E**conomic Feasibility: Does it make financial sense?
  - Systems analysts' time
  - Cost of systems study
  - Cost of employees' time for study
  - Estimated cost of hardware
  - Cost of packaged software or software development
- **O**perational Feasibility: Will people actually use it effectively?

Can We *Really* Do This? The T.E.O. Test

- **T**echnical Feasibility: Do we have the tech and skills?
  - Add on to present system
  - Technology available to meet users' needs
- **E**conomic Feasibility: Does it make financial sense?
  - Systems analysts' time
  - Cost of systems study
  - Cost of employees' time for study
  - Estimated cost of hardware
  - Cost of packaged software or software development
- **O**perational Feasibility: Will people actually use it effectively?
  - Whether the system will operate when put in service
  - Whether the system will be used

Feasibility Deep Dive: Technical Check

- Key Questions

Feasibility Deep Dive: Economic Check

- Key Questions

Feasibility Deep Dive: Operational Check

- Assumes: Tech is possible, Economics make sense

Supporting Feasibility: Estimating Workloads

- Why?
  - Essential for assessing Technical (Hardware needs) and Economic (Processing costs) feasibility
  - Ensures new hardware can handle *current* AND *future* demands (avoids costly early replacement due to growth)

Feasibility Passed! Now, What Gear Do We Need?

- Next Step: Dive deeper into the specifics of Hardware and Software requirements
  - Builds upon Technical Feasibility assessment
  - Involves inventory, estimation, and evaluation

Taking Stock: The Hardware Inventory

- Why?
  - Can’t make good decisions without knowing the starting point
  - Identifies usable existing hardware (potential for reuse/upgrades)
  - Reduces guesswork

Choosing the Right Tools: Evaluating Hardware

- Shared Responsibility: Management, Users, and Systems Analysts
  - Analyst oversees objectively
  - Analyst educates others on pros/cons

Own It or Rent It? Buy vs. Cloud Hardware

- Paths for Acquiring Hardware Infrastructure
  - Buy
  - Cloud

Cloud Flavors: SaaS, PaaS, IaaS

- Beyond just Hardware
  - Infrastructure as a Service (IaaS)
  - Platform as a Service (PaaS)
  - Software as a Service (SaaS)

Cloud Flavors: SaaS, PaaS, IaaS

- Beyond just Hardware
  - Infrastructure as a Service (IaaS)
    - Renting the basic building blocks (compute, storage, network)
    - You manage OS, applications
  - Platform as a Service (PaaS)
  - Software as a Service (SaaS)

Cloud Flavors: SaaS, PaaS, IaaS

- Beyond just Hardware
  - Infrastructure as a Service (IaaS) $\rightarrow$ Like renting the land and utilities
    - Renting the basic building blocks (compute, storage, network)
    - You manage OS, applications
  - Platform as a Service (PaaS)
  - Software as a Service (SaaS)

Cloud Flavors: SaaS, PaaS, IaaS

- Beyond just Hardware
  - Infrastructure as a Service (IaaS) $\rightarrow$ Like renting the land and utilities
    - Renting the basic building blocks (compute, storage, network)
    - You manage OS, applications
  - Platform as a Service (PaaS)
    - Renting infrastructure *plus* operating systems, databases, development tools
    - You manage applications and data
  - Software as a Service (SaaS)

Cloud Flavors: SaaS, PaaS, IaaS

- Beyond just Hardware
  - Infrastructure as a Service (IaaS) $\rightarrow$ Like renting the land and utilities
    - Renting the basic building blocks (compute, storage, network)
    - You manage OS, applications
  - Platform as a Service (PaaS) $\rightarrow$ Like renting a workshop with tools provided
    - Renting infrastructure *plus* operating systems, databases, development tools
    - You manage applications and data
  - Software as a Service (SaaS)

Cloud Flavors: SaaS, PaaS, IaaS

- Beyond just Hardware
  - Infrastructure as a Service (IaaS) $\rightarrow$ Like renting the land and utilities
    - Renting the basic building blocks (compute, storage, network)
    - You manage OS, applications
  - Platform as a Service (PaaS) $\rightarrow$ Like renting a workshop with tools provided
    - Renting infrastructure *plus* operating systems, databases, development tools
    - You manage applications and data
  - Software as a Service (SaaS)
    - Renting ready-to-use software applications over the internet (e.g., Google Workspace, Salesforce, Office 365)
    - Provider manages everything

Cloud Flavors: SaaS, PaaS, IaaS

- Beyond just Hardware
  - Infrastructure as a Service (IaaS) $\rightarrow$ Like renting the land and utilities
    - Renting the basic building blocks (compute, storage, network)
    - You manage OS, applications
  - Platform as a Service (PaaS) $\rightarrow$ Like renting a workshop with tools provided
    - Renting infrastructure *plus* operating systems, databases, development tools
    - You manage applications and data
  - Software as a Service (SaaS) $\rightarrow$ Like renting a fully furnished apartment
    - Renting ready-to-use software applications over the internet (e.g., Google Workspace, Salesforce, Office 365)
    - Provider manages everything

Beyond the Box: Evaluating Vendor Support

- Key Areas to Evaluate: What happens *after* the sale?

Trend Watch: Bring Your Own Device (BYOD)

- What: Employees using their *personal* devices for work

Now for Software: Build, Buy, or Rent?

- Paths for Acquiring Software
  - Create Custom Software
  - Purchase COTS
  - Use SaaS Provider

Now for Software: Build, Buy, or Rent?

- Paths for Acquiring Software
  - Create Custom Software: Develop it in-house or hire developers
  - Purchase COTS: Buy Commercial-Off-The-Shelf packages (e.g., Microsoft Office, SAP)
  - Use SaaS Provider: Subscribe to software delivered over the cloud (e.g., Salesforce, Google Workspace)
- Summary Trade-offs

Option	Advantages	Disadvantages
Custom Software	Customization, Innovation, In-house support, Ownership	High cost, Dev team, Maintenance

Now for Software: Build, Buy, or Rent?

- Paths for Acquiring Software
  - Create Custom Software: Develop it in-house or hire developers
  - Purchase COTS: Buy Commercial-Off-The-Shelf packages (e.g., Microsoft Office, SAP)
  - Use SaaS Provider: Subscribe to software delivered over the cloud (e.g., Salesforce, Google Workspace)
- Summary Trade-offs

Option	Advantages	Disadvantages
Custom Software	Customization, Innovation, In-house support, Ownership	High cost, Dev team, Maintenance
COTS Software	Low cost, Functionality, Reliability, Proven, Documentation	Generic, Rigid, Vendor risk, Common, Integration
SaaS Provider	No IT, Focus, Quick setup, Scalability, Auto-updates	Less control, Security, Provider risk, Lock-in, Missing features

Judging Software Quality and Support

- Objective Evaluation: Don't rely solely on vendor claims your demos! Use *your* data, involve users

Weighing the Scales: Costs vs. Benefits

- The Deciding Factor
  - While meeting requirements is key, the final ‘Go/No-Go’ often hinges on **Cost-Benefit Analysis**
  - Does the value justify the expense?

Predicting the Future (Sort Of): Forecasting Basics

- Why: Needed for credible cost-benefit analysis over the system's life

What's the Upside? Identifying Benefits

- Two Types of Benefits
  - Tangible Benefits
  - Intangible Benefits

What's the Upside? Identifying Benefits

- Two Types of Benefits
  - Tangible Benefits: Measurable in dollars
    - Increased processing speed $\rightarrow$ Reduced labor time
    - Access to new information $\rightarrow$ Better decisions leading to profit/savings
    - Timelier information $\rightarrow$ Faster response to opportunities/threats
    - Superior calculation power $\rightarrow$ Complex analysis possible
    - Decreased employee time on tasks $\rightarrow$ Labor cost savings
    - Reduced errors $\rightarrow$ Lower correction costs
  - Intangible Benefits

What's the Upside? Identifying Benefits

- Two Types of Benefits
  - Tangible Benefits: Measurable in dollars
    - Increased processing speed $\rightarrow$ Reduced labor time
    - Access to new information $\rightarrow$ Better decisions leading to profit/savings
    - Timelier information $\rightarrow$ Faster response to opportunities/threats
    - Superior calculation power $\rightarrow$ Complex analysis possible
    - Decreased employee time on tasks $\rightarrow$ Labor cost savings
    - Reduced errors $\rightarrow$ Lower correction costs
  - Intangible Benefits: Difficult to quantify, but still important!
    - Improved decision-making process
    - Enhanced data accuracy
    - Improved customer service
    - Reputation
    - Increased employee job satisfaction

What's the Downside? Identifying Costs

- Two Types of Costs (Parallel to Benefits)
  - Tangible Costs
  - Intangible Costs

What's the Downside? Identifying Costs

- Two Types of Costs (Parallel to Benefits)
  - Tangible Costs: Can be accurately projected and quantified
    - Hardware/Software purchase cost
    - Analyst and Developer time (salaries/fees)
    - Business user time (participation in study, training)
    - Operational costs (maintenance contracts, cloud subscription fees, electricity)
  - Intangible Costs

What's the Downside? Identifying Costs

- Two Types of Costs (Parallel to Benefits)
  - Tangible Costs: Can be accurately projected and quantified
    - Hardware/Software purchase cost
    - Analyst and Developer time (salaries/fees)
    - Business user time (participation in study, training)
    - Operational costs (maintenance contracts, cloud subscription fees, electricity)
  - Intangible Costs: Difficult to estimate, but represent real risks
    - Losing competitive edge $\rightarrow$ If the system fails or is delayed
    - Damaged company image $\rightarrow$ Due to system errors/outages
    - Reduce employee morale $\rightarrow$ Due to difficulty/frustrating system
    - Ineffective decisions $\rightarrow$ Due to system providing poor/untimely information

Technique 1: Break-Even Analysis

- Purpose: Determines the point where a new system becomes more *cost-effective* than the current system based on volume

Technique 2: Payback Analysis (and Break-Even Limits)

- Break-Even Analysis Limitations
  - Primarily focuses on *costs*, assuming benefits remain constant (often not true!)
  - Doesn't explicitly show *when* the initial investment is recouped

From Big Idea to Done Deal: Managing Time and Activities

- Challenge: Systems projects, especially large ones, can get complex and unwieldy

Divide and Conquer: The Work Breakdown Structure (WBS)

- What: A hierarchical decomposition of the total scope of work to be carried out by the project team

Ways to Structure the WBS

- Two Common Approaches
  - Product-Oriented WBS
  - Process-Oriented WBS

Ways to Structure the WBS

- Two Common Approaches
  - Product-Oriented WBS
    - Breaks down the work based on the *components* of the final product
    - Example: Website $\rightarrow$ Home Page, Product Pages, FAQ Page, Contact Page, E-commerce Module
    - Each component has sub-tasks
  - Process-Oriented WBS

Ways to Structure the WBS

- Two Common Approaches
  - Product-Oriented WBS
    - Breaks down the work based on the *components* of the final product
    - Example: Website → Home Page, Product Pages, FAQ Page, Contact Page, E-commerce Module
    - Each component has sub-tasks
  - Process-Oriented WBS
    - Breaks down the work based on the *phases* or *processes* involved (e.g., SDLC phases)
    - Example: Website → Emphasizes on what to do in Initiation, Planning, Analysis, Design, and Launch

Demo: Sample Process Oriented WBS

How Long Will It *Really* Take?

- Goal: Arrive at realistic estimates for each task in the WBS

Estimation Techniques: Drawing From The Past

- **Relying on Experience**
  - Best approach if you have done the same tasks before
  - Providers estimates based on real-world knowledge (including potential pitfalls)
  - Gives “most likely” and “pessimistic” estimates

Estimation Techniques: Three-Point Method

- Concept
  - Combines optimistic, pessimistic, and most likely estimates to get a weighted average
  - Accounts for uncertainty

Estimation Techniques: Specialized

- **Function Point Analysis (FPA)**
  - Estimates effort based on the system's *functional size* and complexity, NOT lines of code initially
  - Measures five components: External Inputs, External Outputs, External Queries, Internal Logical Files, External Interface Files
  - Complexity ratings are applied
  - Can be used to compare estimated effort across different Programming languages
  - Resource: International Function Point Users Group (IFPUG)

Scheduling the Work: Planning and Control

- Planning: Selecting the team, assigning tasks, estimating time, *creating the schedule*

Scheduling the Work: Planning and Control

- Planning: Selecting the team, assigning tasks, estimating time, *creating the schedule*
- Control: Monitoring progress against the plan, using feedback, taking corrective action (expediting, rescheduling), motivating the team
- Foundation: A detailed WBS

Phase	Activity
Analysis	Data gathering Data flow and decision analysis Proposal preparation
Design	Data entry design Input design Output design Data organization
Implementation	Implementation Evaluation

Scheduling the Work: Planning and Control

- Planning: Selecting the team, assigning tasks, estimating time, *creating the schedule*
- Control: Monitoring progress against the plan, using feedback, taking corrective action (expediting, rescheduling), motivating the team
- Foundation: A detailed WBS
- Key
  - Detail must be sufficient for scheduling and control
  - Time estimates are added

Phase	Activity	Detailed Activity	Wk.
Analysis	Data gathering	Conduct interviews	3
		Administer questionnaires	4
		Read company reports	4
		Introduce prototype	5
		Observe reactions to prototype	3
	Data flow and decision analysis	Analyze data flow	8
	Proposal preparation	Perform cost-benefit analysis	3
		Prepare proposal	2
		Present proposal	2
Design	Data entry design	-	-
	Input design	-	-
	Output design	-	-
	Data organization	-	-
Implementation	Implementation	-	-
	Evaluation	-	-

Visualizing the Time: Gantt Charts

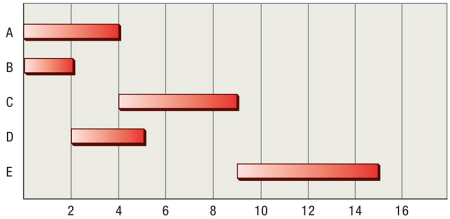
- What: A horizontal bar chart showing project tasks against a timeline

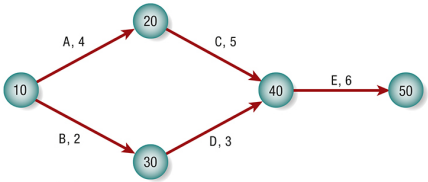
Gantt Chart Limits and A More Powerful Tool: PERT

- Gantt Chart Weakness Doesn't clearly show **dependencies** or **precedence**
  - Can't easily tell *why* a task starts after another one finishes. Is it required, or just coincidence?
  - Doesn't highlight which tasks are *critical* to the overall project duration

Visualizing Dependencies: Gantt vs. PERT

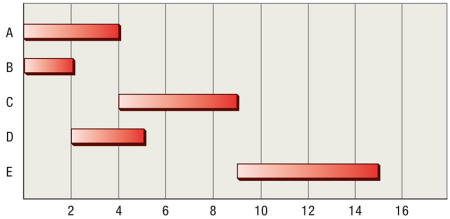
- Gantt Chart: Represents the tasks as bars against time
- PERT Diagram: Represents the same tasks as a network

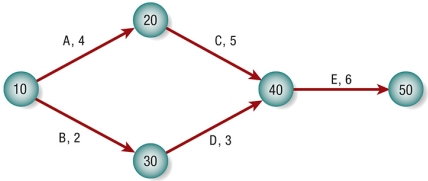




PERT Concepts: Critical Path

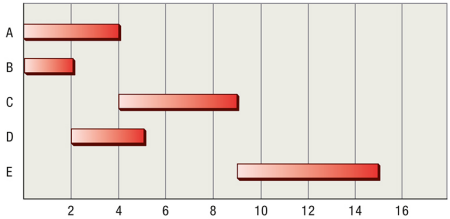
- Path: A sequence of connected activities from the start event to the end event
- Path Length: Sum of the durations of all activities on a path





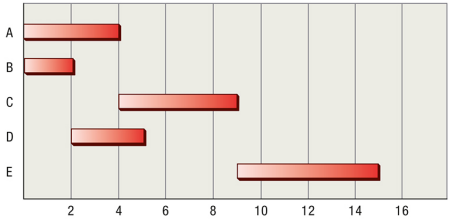
PERT Concepts: Critical Path

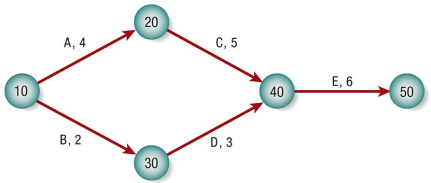
- Path: A sequence of connected activities from the start event to the end event
- Path Length: Sum of the durations of all activities on a path
- Critical Path: The **longest path** through the PERT network
  - Determines the **shortest possible completion time** for the entire project
  - **Any delay** on a critical path activity directly delays the project completion date
- Example
  - Path 1 (A-C-E): 10 → 20 → 40 → 50 (15 weeks)
  - Path 2 (B-D-E): 10 → 30 → 40 → 50 (11 weeks)



PERT Concepts: Critical Path

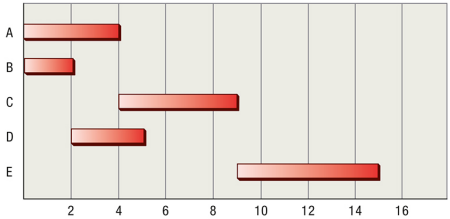
- Path: A sequence of connected activities from the start event to the end event
- Path Length: Sum of the durations of all activities on a path
- Critical Path: The **longest path** through the PERT network
  - Determines the **shortest possible completion time** for the entire project
  - **Any delay** on a critical path activity directly delays the project completion date
- Example
  - Path 1 (A-C-E): 10 → 20 → 40 → 50 **(15 weeks)**
  - Path 2 (B-D-E): 10 → 30 → 40 → 50 (11 weeks)

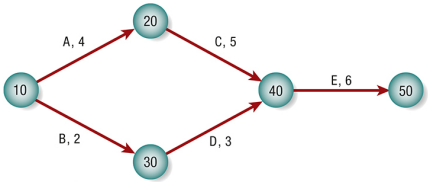




PERT Concepts: Slack Time

- The amount of time a task or path can be delayed *without* delaying the entire project
- Exists only on non-critical paths





PERT Concept: Dummy Activities

- What: Activities with ZERO duration, usually shown as dashed lines

PERT Example: Data Gathering and Proposal

- Scenario: Scheduling the initial phases of a systems project

Activity		Predecessor(s)	Duration (Weeks)
A	Conduct interviews	None	3
B	Administer questionnaires	A	4
C	Read company reports	None	4
D	Analyze data flow	B, C	8
E	Introduce prototype	B, C	5
F	Observe reactions prototype	E	3
G	Perform cost-benefit	D	3
H	Prepare proposal	F, G	2
I	Present proposal	H	2

PERT Example: Constructing the Network

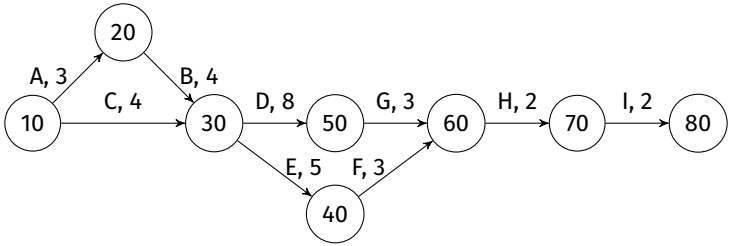
- Process: Start with activities having no predecessors. Add activities sequentially based on their listed predecessors. Ensure all dependencies are represented.

10

Activity		Predecessor(s)	Duration (Weeks)
A	Conduct interviews	None	3
B	Administer questionnaires	A	4
C	Read company reports	None	4
D	Analyze data flow	B, C	8
E	Introduce prototype	B, C	5
F	Observe reactions prototype	E	3
G	Perform cost-benefit	D	3
H	Prepare proposal	F, G	2
I	Present proposal	H	2

PERT Example: Identifying the Critical Path

- Method: Calculate the total duration of *every possible path* from start (10) to end (80). The longest one is critical.



Staying on Track: Controlling the Project

- Reality Check: Things go wrong! Scope changes, delays happen, costs fluctuate

Controlling Costs: Estimation

- Builds on WBS and Schedule: Need cost estimates for each activity

Controlling Costs: Estimation

- Builds on WBS and Schedule: Need cost estimates for each activity
- Main Resource Cost: Project team time! (Also special equipment/tools)
- Cost Estimation Approaches: Similar to Time Estimation
  - Top-Down
    - Base estimates on similar past projects (experience driven)
    - Adjust for known differences
  - Bottom-Up
  - Parametric Modeling

Controlling Costs: Estimation

- Builds on WBS and Schedule: Need cost estimates for each activity
- Main Resource Cost: Project team time! (Also special equipment/tools)
- Cost Estimation Approaches: Similar to Time Estimation
  - Top-Down
    - Base estimates on similar past projects (experience driven)
    - Adjust for known differences
  - Bottom-Up
    - Get estimates from team members responsible for each WBS task
    - Analyst reviews/aggregates
    - Can be time-consuming, variable
  - Parametric Modeling

Controlling Costs: Estimation

- Builds on WBS and Schedule: Need cost estimates for each activity
- Main Resource Cost: Project team time! (Also special equipment/tools)
- Cost Estimation Approaches: Similar to Time Estimation
  - Top-Down
    - Base estimates on similar past projects (experience driven)
    - Adjust for known differences
  - Bottom-Up
    - Get estimates from team members responsible for each WBS task
    - Analyst reviews/aggregates
    - Can be time-consuming, variable
  - Parametric Modeling
    - Use parameters/formulas (e.g., cost per line of code, cost per hour) + Project size estimates
    - Software like COCOMO II can assist

Controlling Costs: Pitfalls and Budgeting

- Why Cost Estimates Often Fail
  - Over-Optimism
  - Rushing

Demo: Sample Budget

Controlling Costs: Pitfalls and Budgeting

- Why Cost Estimates Often Fail
  - Over-Optimism
    - Believing everything will go perfectly
    - Underestimating effort/complexity (e.g., lines of code)
    - “Happy path” estimating
  - Rushing

Demo: Sample Budget

Managing Risk: Looking Out for Trouble

- Best Defense: Thorough initial analysis, feasibility studies, understanding motivations, experience!

Need for Speed? Expediting Activities

- Crashing: Speeding up project activities to finish earlier
- Why
  - Potential bonus for early completion
  - Free up resources/team members for other projects sooner
  - Recover from earlier delays

Need for Speed? Expediting Activities

- Crashing: Speeding up project activities to finish earlier
  - Crash Time: Absolute minimum time an activity *can* take
  - Cost/Week: The additional cost incurred to reduce the activity duration by one week
- Why
  - Potential bonus for early completion
  - Free up resources/team members for other projects sooner
  - Recover from earlier delays

How to Expedite: The Analysis

- Rule #1: Only expedite critical-path activities
- Rule #2: Pick the cheapest critical-path activity per time saved
- Rule #3: Never expedite an activity below its minimum (crash) duration

How to Expedite: The Analysis

- Rule #1: Only expedite critical-path activities
- Rule #2: Pick the cheapest critical-path activity per time saved
- Rule #3: Never expedite an activity below its minimum (crash) duration

Eligible Activities	Activity Chosen	Time for Each Path (22)	19	19	16	Cost	Cu. Cost
A, B, D, G, H, or I							

How to Expedite: The Analysis

- Rule #1: Only expedite critical-path activities
- Rule #2: Pick the cheapest critical-path activity per time saved
- Rule #3: Never expedite an activity below its minimum (crash) duration

Eligible Activities	Activity Chosen	Time for Each Path (22)	19	19	16	Cost	Cu. Cost
A, B, D, or I							

How to Expedite: The Analysis

- Rule #1: Only expedite critical-path activities
- Rule #2: Pick the cheapest critical-path activity per time saved
- Rule #3: Never expedite an activity below its minimum (crash) duration

Eligible Activities	Activity Chosen	Time for Each Path				Cost	Cu. Cost
		(22)	19	19	16		
A, B, D, or I A, B, D, or I	B	(21)	18	19	16	$500	$500

How to Expedite: The Analysis

- Rule #1: Only expedite critical-path activities
- Rule #2: Pick the cheapest critical-path activity per time saved
- Rule #3: Never expedite an activity below its minimum (crash) duration

Eligible Activities	Activity Chosen	Time for Each Path				Cost	Cu. Cost
		(22)	19	19	16		
A, B, D, or I	B	(21)	18	19	16	$500	$500
A, B, D, or I	B	(20)	17	19	16	500	1,000
A, B, D, or I							

How to Expedite: The Analysis

- Rule #1: Only expedite critical-path activities
- Rule #2: Pick the cheapest critical-path activity per time saved
- Rule #3: Never expedite an activity below its minimum (crash) duration

Eligible Activities	Activity Chosen	Time for Each Path				Cost	Cu. Cost
		(22)	19	19	16		
A, B, D, or I	B	(21)	18	19	16	$500	$500
A, B, D, or I	B	(20)	17	19	16	500	1,000
A, D, or I							

How to Expedite: The Analysis

- Rule #1: Only expedite critical-path activities
- Rule #2: Pick the cheapest critical-path activity per time saved
- Rule #3: Never expedite an activity below its minimum (crash) duration

Eligible Activities	Activity Chosen	Time for Each Path				Cost	Cu. Cost
		(22)	19	19	16		
A, B, D, or I	B	(21)	18	19	16	$500	$500
A, B, D, or I	B	(20)	17	19	16	500	1,000
A, D, or I	I	(19)	17	18	15	600	1,600
A, D, or I							

How to Expedite: The Analysis

- Rule #1: Only expedite critical-path activities
- Rule #2: Pick the cheapest critical-path activity per time saved
- Rule #3: Never expedite an activity below its minimum (crash) duration

Eligible Activities	Activity Chosen	Time for Each Path				Cost	Cu. Cost
		(22)	19	19	16		
A, B, D, or I	B	(21)	18	19	16	$500	$500
A, B, D, or I	B	(20)	17	19	16	500	1,000
A, D, or I	I	(19)	17	18	15	600	1,600
A or D							

How to Expedite: The Analysis

- Rule #1: Only expedite critical-path activities
- Rule #2: Pick the cheapest critical-path activity per time saved
- Rule #3: Never expedite an activity below its minimum (crash) duration

Eligible Activities	Activity Chosen	Time for Each Path				Cost	Cu. Cost
		(22)	19	19	16		
A, B, D, or I	B	(21)	18	19	16	$500	$500
A, B, D, or I	B	(20)	17	19	16	500	1,000
A, D, or I	I	(19)	17	18	15	600	1,600
A or D	A	(18)	16	(18)	15	800	2,400
A and C, or D	D	(17)	16	17	15	1,000	3,400
A and C, or D	D	(16)	(16)	(16)	15	1,000	4,400
A and C, or D							

How to Expedite: The Analysis

- Rule #1: Only expedite critical-path activities
- Rule #2: Pick the cheapest critical-path activity per time saved
- Rule #3: Never expedite an activity below its minimum (crash) duration

Eligible Activities	Activity Chosen	Time for Each Path				Cost	Cu. Cost
		(22)	19	19	16		
A, B, D, or I	B	(21)	18	19	16	$500	$500
A, B, D, or I	B	(20)	17	19	16	500	1,000
A, D, or I	I	(19)	17	18	15	600	1,600
A or D	A	(18)	16	(18)	15	800	2,400
A and C, or D	D	(17)	16	17	15	1,000	3,400
A and C, or D	D	(16)	(16)	(16)	15	1,000	4,400
A and C							

Comprehensive Control: Earned Value Management (EVM)

- Purpose: Integrates project **scope** (work done), **schedule** (time), and **cost** ($$$) into a unified framework to measure performance and predict outcomes

EVM in Action: Website Example

Project Budget: $18,000

At the End of	Stage	Estimated Cost	Cu. Estimate	Estimated Duration	Stage Completed	Actual Cost of Stage to Date	Actual Cost of Project to Date
Month 1	1	$6,000	$6,000	1 month	100%	$6,000	$6,000
Month 2	2	3,000	9,000	1 month	100%	3,000	9,000
Month 3	3	3,000	12,000	1 month	100%	3,000	12,000
Month 4	4	3,000	15,000	1 month	50%	5,000	17,000
Month 5	5	3,000	18,000	1 month	0%	Not yet begun	Not yet begun

- EVM at End of Month 4
  - $BAC = \$18,000$
  - $PV = \$15,000$ (Cumulative Estimated Cost)
  - $AC = \$17,000$
  - EV =? (Needs to be calculated)

EVM Calculations: Where Do We Stand?

After 4 months,

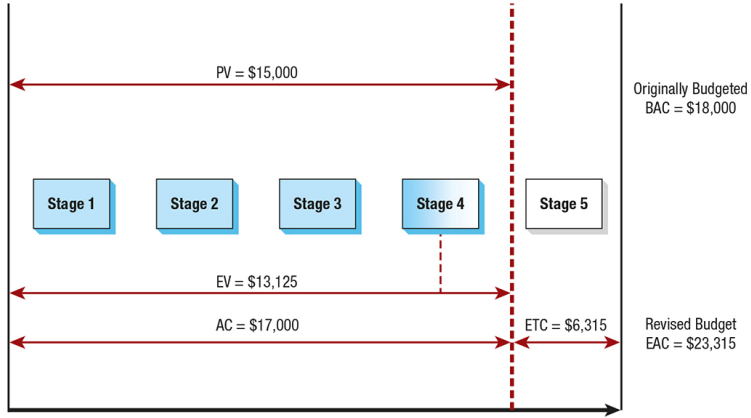
- Calculate % Work done so far (p): $\frac{100\%+100\%+100\%+50\%}{100\%+100\%+100\%+100\%} = \frac{350}{400} = 0.875$

EVM Forecasting: Where Are We Headed?

- Purpose: Use CPI to predict future costs

EVM Visualization: Key Takeaway

- Analyst's Role: Balance Cost, Time, and Scope based on this information



It's About People: Managing the Project Team

- Equally Important: Managing the people doing the work

Who's On the Team? Assembling the Right Mix

- Core Values: Look for shared values (teamwork, quality, on-time/on-budget delivery)

The Right Skills for the Job: Team Composition

- Essential Roles/Skills
  - Business Knowledge: At least one person who deeply understands the business area/domain (e.g., Marketing expert for e-commerce site)

Making the Team Tick: Communication and Dynamics

- Team Personality: Each team develops a unique interaction style

Unwritten Rules: Understanding Team Norms

- What: Collective expectations, values, and standard ways of behaving within a *specific* team (Can be explicit or implicit)

Aiming High: Setting Goals and Motivating the Team

- Setting Productivity Goals
  - Based on team expertise, past performance, project nature
  - Goals should be challenging, but achievable
  - Team participation in goal-setting increases buy-in

Unique Challenges: Managing E-commerce Projects

- Key Differences
  - Scattered Data
    - Cross-department (Marking, Sales, Inventory, Finance)
    - Increases complexity and dept. politics

Getting It In Writing: The Project Charter

- Purpose
  - A written document clarifying project scope, objectives, and expectations
  - Acts as a contract between the team, users, and management

The Grand Finale (of Analysis): The Systems Proposal

- What: A formal written document detailing
  - systems study
  - findings
  - alternatives, and
  - **recommendations**

What Goes Inside? The 10 Proposal Sections

- Standard Structure: Preliminary Materials
  1. Cover Letter
     - Friendly intro
     - Study objectives
     - Team members

What Goes Inside? The 10 Proposal Sections

- Standard Structure: Study and Analysis
  5. Outlines of Systems Study
     - Methods used (interviews, surveys, observation, etc.)
     - Who/what was studied

What Goes Inside? The 10 Proposal Sections

- Standard Structure: Conclusion and Support
  9. Proposal Summary
     - Brief recap mirroring Executive Summary (objectives, recommendation, importance)
     - Positive conclusion

Getting the Word Out: Delivery and Presentation

- Distribution
  - Carefully select recipients (key decision-makers)
  - Hand-deliver copies if possible (increases visibility)

A Picture is Worth... Supporting Your Words with Figures

- Why?
  - People absorb information differently; visuals help
  - Demonstrate responsiveness to audience needs
  - Capture and communicate complex data effectively

Organizing Data: Effective Use of Tables

- Purpose: Present statistical or alphabetical data in an organized, structured way

Visualizing Trends and Comparisons: Effective Graphs

- Purpose: Illustrate comparisons (Line, Column, Bar) or composition (Pie, Area)

References I

[1] D. Cearley and D. Reeves, “Cloud computing innovation key initiative overview,” Gartner, Inc., Stamford, CT, Tech. Rep. 2718918, 2011. [Online]. Available: https://www.gartner.com/en/documents/2718918 (cit. on pp. 102–109).

[2] R. Blair and J. Hewitt, “Market guide for disaster recovery as a service,” Gartner, Inc., Stamford, CT, Tech. Rep. Document No. 4364899, 2024. [Online]. Available: https://www.gartner.com/en/documents/4364899 (cit. on pp. 110–118).

[3] PM Editorial, “Diversity drives better decisions,” (2017), [Online]. Available: https://www.peoplemanagement.co.uk/article/1742040/diversity-drives-better-decisions (cit. on pp. 290–292).

[4] A. Reynolds and D. Lewis, “Teams solve problems faster when they’re more cognitively diverse,” (2017), [Online]. Available: https://hbr.org/2017/03/teams-solve-problems-faster-when-theyre-more-cognitively-diverse (cit. on pp. 290–292).

[5] M. Alexander, “How to build more diverse and inclusive project teams,” (2021), [Online]. Available: https://www.techrepublic.com/article/how-to-build-more-diverse-and-inclusive-project-teams/ (cit. on pp. 290–292).