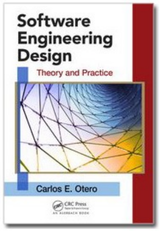
CHAPTER 10: MANAGEMENT, LEADERSHIP, AND ETHICS

-- Dr. L. Daniel Otero, Florida Institute of Technology --

SESSION I: DESIGN MANAGEMENT FRAMEWORK

Software Engineering Design: Theory and Practice
by Carlos E. Otero



SESSION'S AGENDA

- Software Design Management
  - ✓ What is software design management?
  - ✓ Why study software design management?

- A Design Management Framework
  - ✓ Quality in software design
  - ✓ Typical project lifecycle
  - ✓ Typical performance measures in project phases
  - ✓ Components of the design management framework

- What's next…

SOFTWARE DESIGN MANAGEMENT

- What is software design management?
  - Software design is the process of transforming functional and non-functional requirements into models that describe a technical solution.
    - Can be viewed as a complex decision problem since
      - there are many ways to design a software system, and
      - software engineers must make design decisions to achieve the required goals of the software system in the most effective manner
  - Management is a *set of activities* directed at an organization's *resources* to achieve organizational goals in an efficient and effective manner.
    - Activities: planning and decision making, organizing, leading, and controlling
    - Resources: human, financial, physical, and information
  - In the design phase, management refers to activities required to efficiently create *quality* design artifacts, within schedule and budget constraints.

Software Design Management

- Why study software design management?
  - ✓ In large-scale software projects, software design management is essential to plan, organize, staff, track, and lead the activities required to carry out successfully the software architecture and detailed design steps.
  - ✓ To understand relevant managerial techniques and concepts
    - This will help to effectively use resources to achieve the various goals for each main component in the design phase (i.e., software architecture, detailed design, and documentation).
    - Effective use of resources is key to the success of software design management.
  - ✓ In the general sense, software design management is needed to:
    - Control/monitor design processes
    - Manage the resources necessary for completing quality design artifacts according to organizational goals

The Concept of Quality

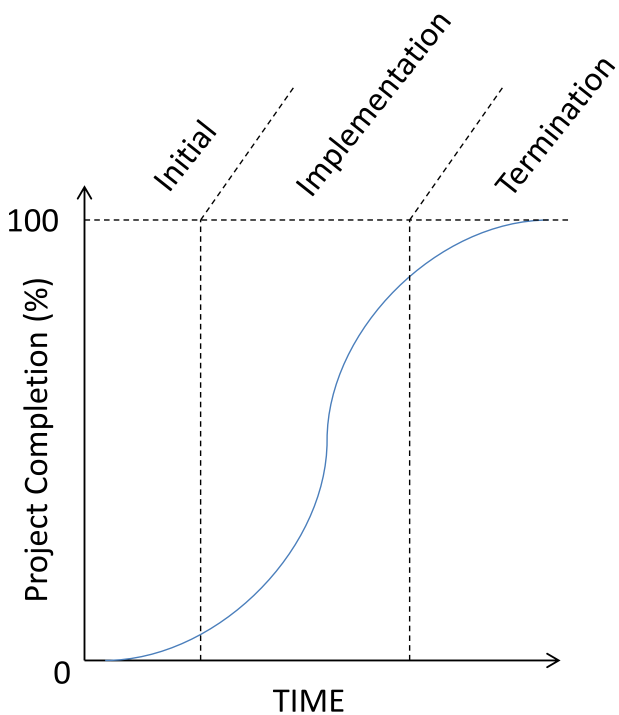
- Quality is a performance measure for a service provided or a product produced

- Quality is *relative* to a particular stakeholder. For example:
  - ✓ A personal website can be classified as high-quality by the programmer that developed it, and as low quality by user clients. Why?
    - Because both stakeholders have different parameters by which they measure quality
      - The programmer may perceive quality in terms of being able to use appropriate colors and organize the website to run on a particular web browser.
      - A client using a different web browser may not see the website's organization and colors as intended; thus perceiving the website's quality as poor.

- Although the quality of a software design can be assessed in various ways, from a management's perspective, quality of software design tends to be evaluated in terms of *cost* and *scheduling*.
  - ✓ Thus, it is important to understand project management techniques that can be used to keep design artifacts within cost and schedule thresholds.

Design Management Framework

- It is beneficial to view the design phase as a project. Why?
  - ✓ Project management techniques can be directly applied to the design phase
    - This will help to efficiently and effectively manage the software design activities
    - E.g., the life cycle of the design phase can be represented with three stages as in:



- This life cycle represents a typical project pattern
  - ✓ Contributions to a project's completion are:
    - Relatively small during the initial and termination phases
    - Significantly high during the implementation phase
- When defining a life cycle structure for a project, we can decompose a project into its appropriate stages, and manage each stage individually.

LIFECYCLE STAGES

- In the initial stage, there are two main objectives:
  - Achieve a clear understanding of the tasks to be accomplished and the resources that are necessary to successfully complete the tasks.
  - Develop a plan to complete the project's tasks within schedule and budget constraints.
- The initial phase is characterized by planning activities.
  - Although some of the artifacts resulting from this phase (e.g., schedule) may be deliverables to clients, this phase is not viewed as contributing significantly to the percentage project completion.
- The implementation stage is where:
  - Most of the effort is employed
  - Most of the design project deliverables are achieved, including completed design documents
  - The percent project completion is the highest

Objectives in the Lifecycle Stages

- In the termination stage,
  - Key activities involve verification that everything is in place for a smooth transition into the code construction phase. Examples of activities are:
    - Making sure that the latest versions of the design documents are securely stored according to configuration management procedures
    - Updating schedule and cost current values
    - Re-evaluating schedule and budget plans based on the resulting performance measures of the design phase
    - Communicating results to upper management.

- The design documents generated in this stage are critical for the robustness and maintainability of the software designed.
  - However, the relative impact of this stage to project completion is not viewed as significant as that of the implementation phase.

Managing Lifecycle Stages

- The lifecycle stages of a design project must be *individually managed* to ensure acceptable levels of quality and performance measures.
  - What does this means?
    - For example, an important outcome of the *planning stage* is a clear understanding of the tasks to be accomplished and the resources that are necessary to successfully complete the tasks.
      - Various project management techniques exist to help achieve this outcome.

- It is important to know how to apply appropriate project management techniques to each of the design project stages
  - Afterwards, *peer reviews* will help ensure that the objectives of the planning stage are achieved.
    - Peer reviews must be conducted throughout these stages to minimize the propagation of errors to subsequent stages and apply corrective actions.

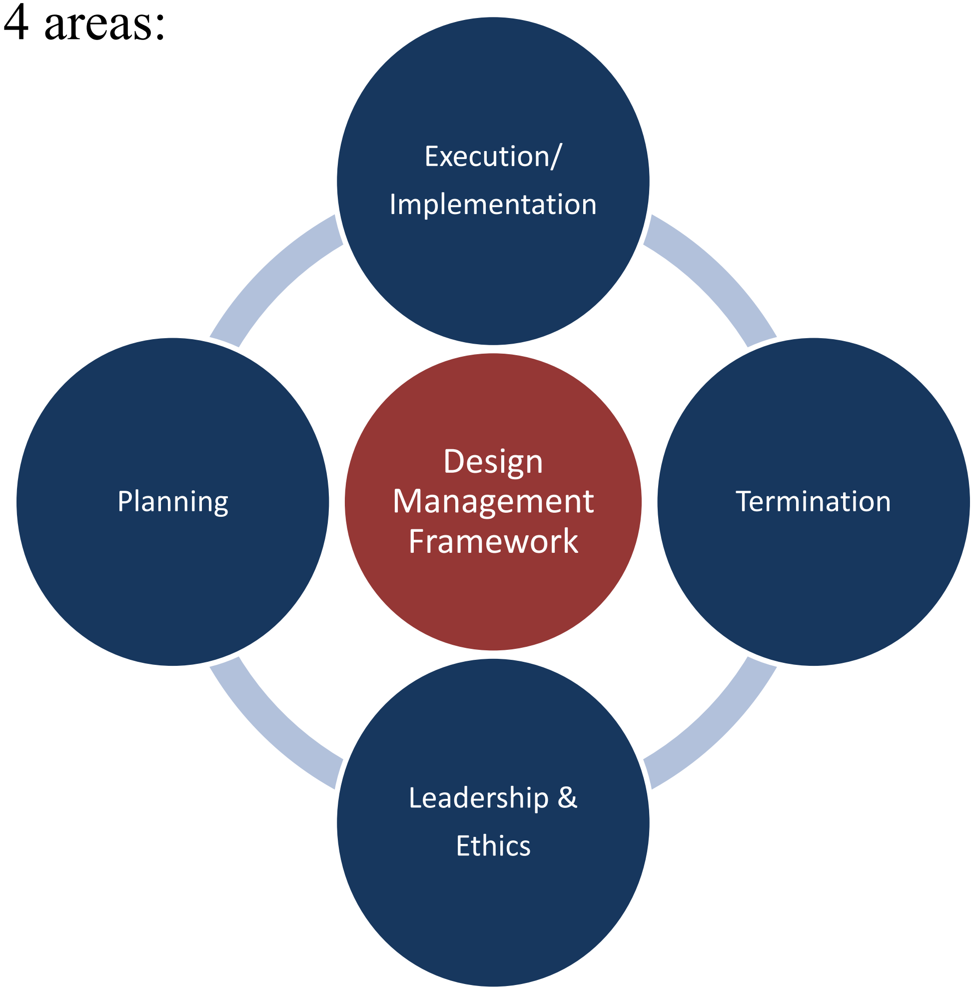
DESIGN MANAGEMENT FRAMEWORK

- Each of the design project lifecycle stages are associated with different performance measures. For example:
  - ✓ In the planning stage, performance measures such as the following are relevant:
    - Thoroughness (everything is considered),
    - Completeness (complete definition of plans),
    - Accuracy of estimates.
  - ✓ In the execution/implementation stage,
    - Quality is a function of a series of technical attributes of software designs.
    - Managing during this stage is mainly a monitoring approach to make sure that the plan is been followed, and take corrective actions if necessary.

- *Leadership* and *good ethical conduct* are important factors that affect the effectiveness of managing each lifecycle stage.

DESIGN MANAGEMENT FRAMEWORK

- Thus, an appropriate design management framework is one that considers the following 4 areas:



What's Next…

- This session accomplished the following:
  - ✓ Defined software design management
  - ✓ Understood the importance of studying software design management
  - ✓ Discussed the concept of quality in software design management
  - ✓ Described the stages of a typical project lifecycle
  - ✓ Described typical performance measures in project phases
  - ✓ Defined a design management framework composed of four components:
    - Planning
    - Implementation/Execution
    - Termination
    - Leadership & Ethics

- Next session will focus on the planning component of the design management framework,
  - ✓ This session will focus on managerial concepts and techniques that are relevant to the planning phase.

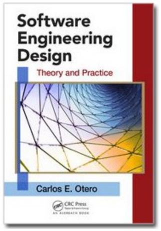
QUESTIONS?

CHAPTER 10: MANAGEMENT, LEADERSHIP, & ETHICS

-- Dr. L. Daniel Otero, Florida Institute of Technology --

SESSION II: PLANNING, IMPLEMENTATION, AND TERMINATION

Software Engineering Design: Theory and Practice
by Carlos E. Otero



SESSION'S AGENDA

- The Planning Stage
  - ✓ Scoping
    - Work breakdown structure
    - Budgeting
  - ✓ Organizing
    - Linear responsibility chart
    - Scheduling
  - ✓ Establish change control policy
- Implementation Stage
  - ✓ Earned value management
- Termination Stage
- What's next…

The Planning Stage
- Scoping -

- The planning stage is critical to the success of any major project.
  - ✓ Planning exists to lay out a strong foundation for a successful project.
  - ✓ Planning establishes the directions to follow to complete the project and improve its probability of success.

- The key functions in the project planning stage are:
  - ✓ *Scoping*
  - ✓ *Organizing*

- Project scoping is the first key function of the planning stage.
  - ✓ It involves two main activities:
    - Identify the tasks
    - Develop the budget

- Two commonly used techniques for project scoping are:
  - ✓ The work breakdown structure (WBS)
  - ✓ Budgeting

The Planning Stage
- Scoping: Work Breakdown Structure -

- The work breakdown structure (WBS) is a technique to represent, either graphically or in list format, a project modularized into task activities.
  - ✓ From the project management point of view, a project begins as a statement of work (SOW), which mainly consists as the set of main objectives to be achieved.
  - ✓ The SOW is decomposed into *tasks* where these tasks are decomposed into *subtasks*, and subtasks into *work packages*.
  - ✓ WBS represents these tasks.

- WBS is a simple but powerful technique that helps to plan, clearly define, and organize the activities related to reach specific milestones and complete a project.
  - ✓ Milestones are defined as specific events to be reached at specific points in time.

- For example, completion of the major design activities, e.g., architectural design, detailed design, and documentation can constitute the main project milestones of a design project.
  - ✓ Note that HCI design can be carried out as part of architectural or detailed design.
  - ✓ Construction design typically occurs during construction.

THE PLANNING STAGE
- SCOPING: WORK BREAKDOWN STRUCTURE -

- A very simple example of a WBS for the design of a software application

Outline #	Task Name
1	Design Project
1.1	Architectural Design
1.1.1	Evaluate Alternative Designs
1.1.1.1	Prioritize Objectives
1.1.1.2	Formal Review of Evaluations
1.1.2	Select Architectural Design
1.1.2.1	Make Initial Selection
1.1.2.2	Formal Review to Select Final Design
1.2	Detailed Design
1.2.1	Evaluate Alternatives
1.2.2	Select Among Alternatives
1.3	Documentation
1.3.1	Write Architectural Design Documentation
1.3.2	Formal Review (Architecture Design)
1.3.3	Final Architectural Design Document
1.3.4	Write Detailed Design Documentation
1.3.5	Formal Review (Detailed Design)
1.3.6	Final Architectural Design Document

THE PLANNING STAGE
- SCOPING: WORK BREAKDOWN STRUCTURE -

- Evaluating the usefulness of a WBS is often more a matter of subjective assessments than not.
  - We can specify a criteria for this evaluation.
    - Scope accuracy
    - Completeness
    - Level of detail

- Scope accuracy
  - We need to agree that the WBS serves its overall objective of being a breakdown of the project into tasks, and nothing else.
  - A WBS does not contain information regarding tasks' relationships such as task precedence or task durations, and it does not include personnel skills required for the completion of tasks.
  - Instead, each element of a WBS represents an activity that consumes resources' time and effort.

- Completeness
  - Ensure that the WBS includes the complete set of tasks necessary to complete the project.

- Level of detail
  - Evaluates if the WBS breaks down the project's tasks into appropriate levels such that it facilitates the estimation of effort for each task.

THE PLANNING STAGE
- SCOPING: BUDGETING -

- Budgeting is the process of estimating the cost of a project.
  - ✓ It is basically a forecasting problem, since cost is a function of various parameters that are uncertain.

- Conceptually, a common strategy to develop a budget is to simply cost each element in the WBS.
  - ✓ To accomplish this, elements are associated with *direct* and *indirect costs*.

- Direct costs are those that can be directly tied to the development of the design.
  - ✓ The most often used direct costs are labor and equipment (i.e., hardware/software equipment).

- Indirect costs, on the other hand, include costs such as fringe benefits and administrative expenses.

The Planning Stage
- Scoping: Budgeting -

- One of the most significant and hard to estimate direct costs for each task is the cost of the staff that will be directly working on the completion of the task.
  - ✓ Given that software projects are typically custom-made solutions to particular problems, tasks are considered non-repetitive and often involve *learning rates*.
    - This is particularly true for tasks that involve new technologies or the use of skills that the staff is unfamiliar with.
    - Therefore, it is important to account for learning rates in the budgeting process.

- It is important to understand that developing accurate budgets for software projects is often very difficult due to the complexity of the projects.
  - ✓ Data from previous similar projects, as well as input from experienced personnel, are used as resources to prepare budgets.

- After the budget for a project is created and approved, it becomes more of a control mechanism for managers because it establishes the cost threshold for the entire project.

THE PLANNING STAGE
- ORGANIZING -

- Organizing is the second key function of the planning stage.
  - It involves two main activities:
    - Responsibility assignments
    - Scheduling

- That is, after identifying and organizing the tasks in the scoping function with the WBS, the organizing function:
  - Conducts task assignments
  - Establishes task durations
    - Task durations are critical because they are integral part of the monitoring and controlling mechanisms in the implementation stage.
  - Identifies any predecessors
    - For each task A, predecessor is a task that must be completed before another task can begin.

- Project management tools that are applicable for organizing include:
  - Linear Responsibility Chart
  - Scheduling with Gantt Charts and Network Diagrams

The Planning Stage
- Linear Responsibility Chart -

- An important outcome of the planning stage is to have a clear understanding of the roles that each staff member plays in the process of completing each element of the WBS.
  - ✓ This can be viewed as a plan for task assignments.
  - ✓ The goal here is to clearly identify the staff members that will be mainly responsible for each task, and those that will serve as reviewers and/or support the task in any way.
  - ✓ A project management tool that allows such outcome is called a *Linear Responsibility Chart (LRC)*.

- An LRC is a tabular representation of task assignments.
  - ✓ Let’s look at an example in the next slide…

THE PLANNING STAGE
- LINEAR RESPONSIBILITY CHART -

Outline #	Task Name	Staff_1	Staff_2	Staff_3	Staff_4	Staff_5
1	Design Project					
1.1	Architectural Design	1	2			3
1.1.1	Evaluate Alternative Designs	1	2			
1.1.1.1	Prioritize Objectives	1	2			
1.1.1.2	Formal Review of Evaluations			1	3	3
1.1.2	Select Architectural Design	1				
1.1.2.1	Make Initial Selection	1				
1.1.2.2	Formal Review to Select Final Design	1		2	3	3
1.2	Detailed Design				1	3
1.2.1	Evaluate Alternatives			2	1	
1.2.2	Select Among Alternatives			2	1	3
1.3	Documentation					
1.3.1	Write Architectural Design Documentation	1	2			
1.3.2	Formal Review (Architecture Design)	1			3	3
1.3.3	Final Architectural Design Document	1		3		
1.3.4	Write Detailed Design Documentation	2			1	3
1.3.5	Formal Review (Detailed Design)	2			1	3
1.3.6	Final Detailed Design Document			3	1	3
Legend:						
1 = Primary responsibility						
2 = Support development role						
3 = Review role						

THE PLANNING STAGE
- SCHEDULING -

- After task assignments, a key activity in the planning stage is *scheduling*.
  - ✓ Formally defined as the process of establishing task durations in order to convert the WBS into an operating timetable.

- Scheduling helps to identify the critical project activities that cannot be delayed without delaying the overall project duration.
  - ✓ We can also identify those activities that will not affect the overall project duration if delayed.
  - ✓ Scheduling allows tasks to be ordered so that for each task, preceding and subsequent tasks are easily identified.

- There are various project management techniques that are used in scheduling. Two of the most common ones are:
  - ✓ Gantt charts
  - ✓ Network diagrams

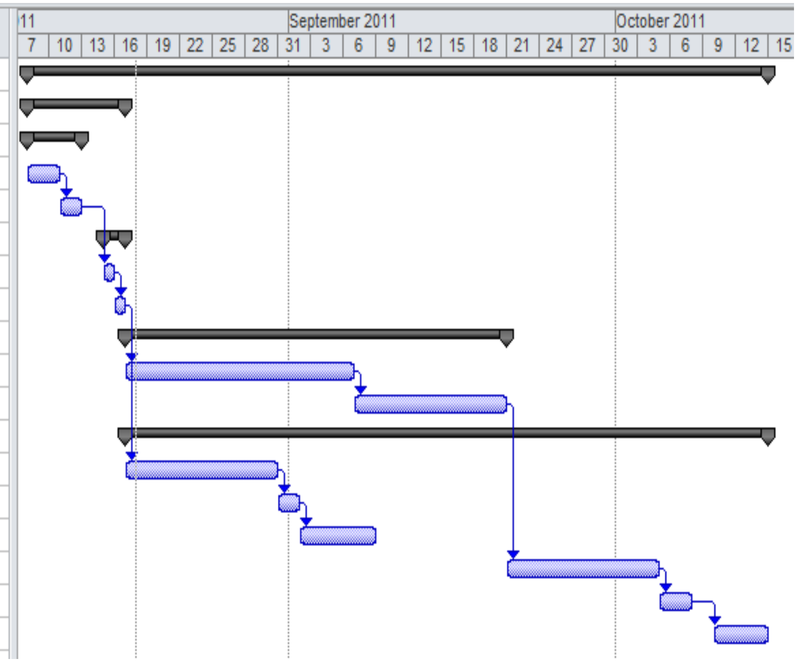
THE PLANNING STAGE
- THE GANTT CHART -

- The Gantt Chart is a graphical representation of the logical flow between tasks (i.e., activities) and their durations.
  - ✓ Very simple to follow.
  - ✓ Often used to show schedule and progress, as well as to serve as a control/monitoring mechanism.
  - ✓ Requires three basic parameters:
    - Tasks
    - Durations
    - Predecessors and successors

- These parameters are used to develop a graph that depicts the durations of tasks and their relationship.
  - ✓ Let’s see an example in the next slide…

THE PLANNING STAGE
- THE GANTT CHART -

	Outline #	Task Name	Duration	Start	Finish	Predecessors
1	**1**	**Design Project**	**50 days?**	**Mon 8/8/11**	**Fri 10/14/11**	
2	**1.1**	**Architectural Design**	**7 days?**	**Mon 8/8/11**	**Tue 8/16/11**	
3	**1.1.1**	**Evaluate Alternative Designs**	**5 days**	**Mon 8/8/11**	**Fri 8/12/11**	
4	1.1.1.1	Prioritize Objectives	3 days	Mon 8/8/11	Wed 8/10/11	
5	1.1.1.2	Formal Review of Evaluations	2 days	Thu 8/11/11	Fri 8/12/11	4
6	**1.1.2**	**Select Architectural Design**	**2 days?**	**Mon 8/15/11**	**Tue 8/16/11**	
7	1.1.2.1	Make Initial Selection	1 day?	Mon 8/15/11	Mon 8/15/11	5
8	1.1.2.2	Formal Review to Select Final Design	1 day?	Tue 8/16/11	Tue 8/16/11	7
9	**1.2**	**Detailed Design**	**25 days**	**Wed 8/17/11**	**Tue 9/20/11**	
10	1.2.1	Evaluate Alternatives	15 days	Wed 8/17/11	Tue 9/6/11	8
11	1.2.2	Select Among Alternatives	10 days	Wed 9/7/11	Tue 9/20/11	10
12	**1.3**	**Documentation**	**43 days**	**Wed 8/17/11**	**Fri 10/14/11**	
13	1.3.1	Write Architectural Design Documentation	10 days	Wed 8/17/11	Tue 8/30/11	8
14	1.3.2	Formal Review (Architecture Design)	2 days	Wed 8/31/11	Thu 9/1/11	13
15	1.3.3	Final Architectural Design Document	5 days	Fri 9/2/11	Thu 9/8/11	14
16	1.3.4	Write Detailed Design Documentation	10 days	Wed 9/21/11	Tue 10/4/11	11
17	1.3.5	Formal Review (Detailed Design)	3 days	Wed 10/5/11	Fri 10/7/11	16
18	1.3.6	Final Architectural Design Document	5 days	Mon 10/10/11	Fri 10/14/11	17



The Planning Stage
- Network Diagrams -

- Similar to Gantt charts, network diagrams are also graphical representations of logical flow between tasks, describe task durations, and need the same three basic parameters:
  - ✓ Tasks,
  - ✓ Tasks durations, and
  - ✓ Predecessors and successors

- Estimating accurate task durations is often a very complex endeavor.
  - ✓ One common approach is to estimate minimum (a), maximum (b), and most likely (m) duration times.
  - ✓ With these three parameters we can calculate the expected time (ET) of each task using a probabilistic approach.
  - ✓ Assuming a beta probability distribution for the time estimates, the expected time for each activity can be approximated using the following:
    - Expected Time = $(a + 4m + b) / 6$

- A key benefit for having three estimates for task durations is that we can use them to develop variance calculations, which can then be used to estimate the probability of completing the project within a specific time.
  - ✓ The variance of the duration of a task is calculated using:
    - Variance = $((b-a)/6)^2$

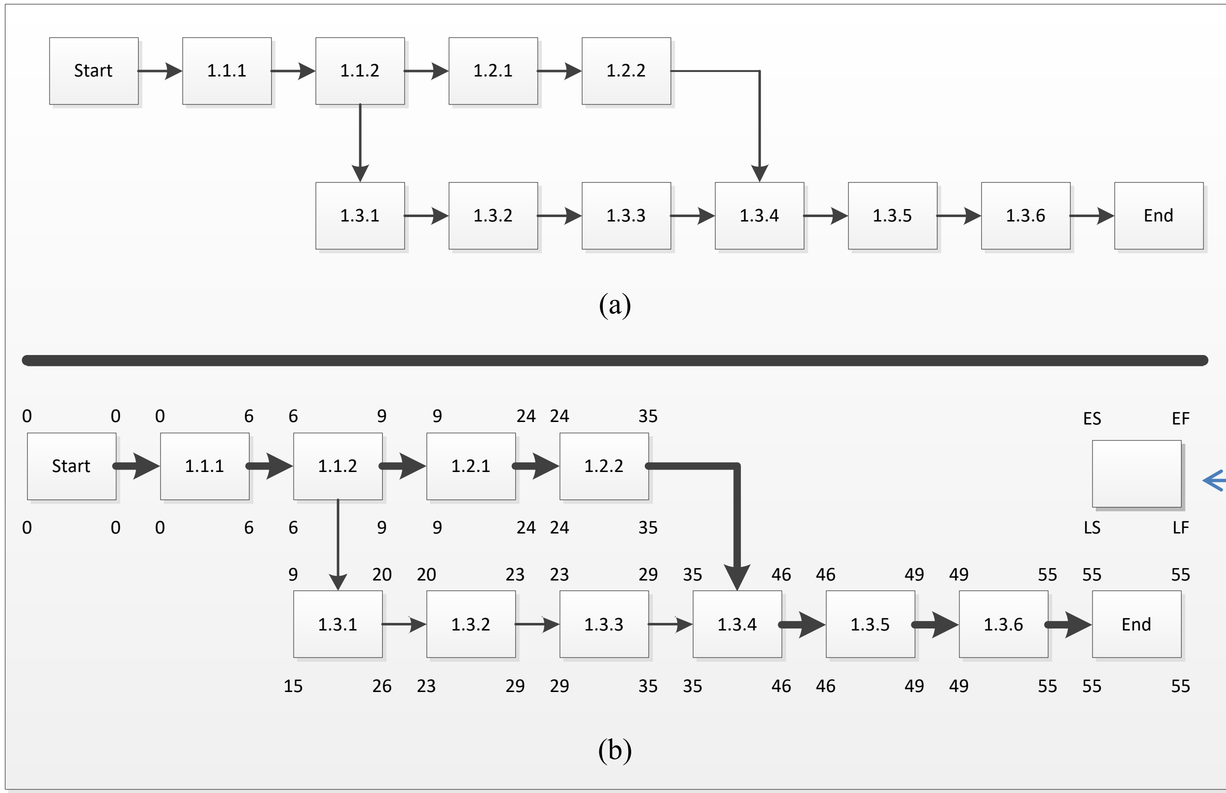
THE PLANNING STAGE
- NETWORK DIAGRAMS -

- Example:
  - ✓ Task durations for the three-level tasks described in the previous WBS

Outline #	Time (days)					
	Minimum (a)	Most Likely (m)	Maximum (b)	Immediate Predecessors	Expected Time (ET)	Variance (σ^2)
1.1.1	3	5	13	-	6	2.78
1.1.2	1	2	9	1.1.1	3	1.78
1.2.1	8	15	22	1.1.2	15	5.44
1.2.2	5	10	21	1.2.1	11	7.11
1.3.1	5	10	21	1.1.2	11	7.11
1.3.2	1	2	9	1.3.1	3	1.78
1.3.3	3	5	13	1.3.2	6	2.78
1.3.4	5	10	21	1.2.2; 1.3.3	11	7.11
1.3.5	1	2	9	1.3.4	3	1.78
1.3.6	3	5	13	1.3.5	6	2.78

THE PLANNING STAGE
- NETWORK DIAGRAMS -

- Resulting AON network diagram.

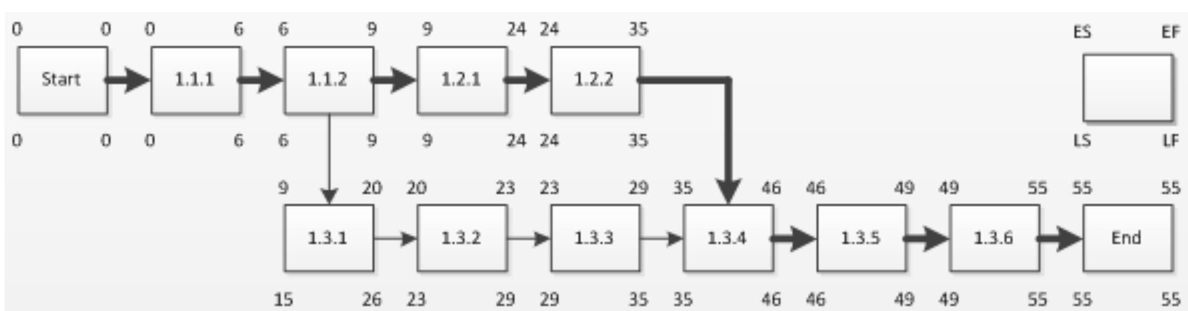


- ES = Earliest Time; EF = Earliest Finish
- LS = Latest Start, LF = Latest Finish

THE PLANNING STAGE
- NETWORK DIAGRAMS -

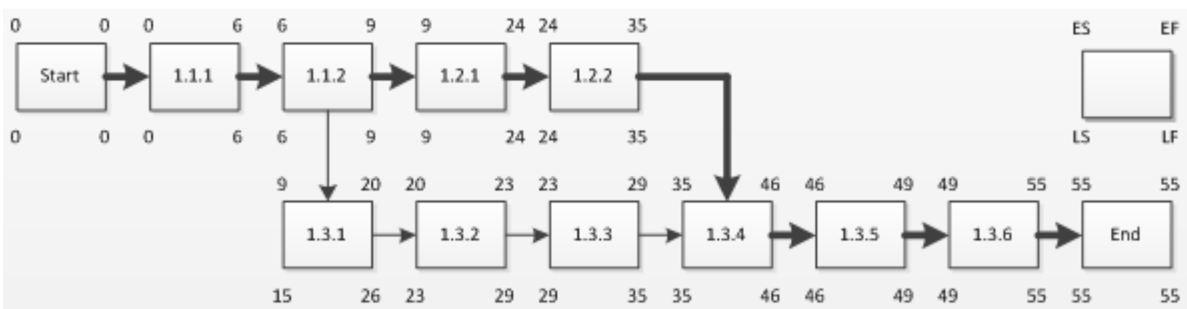
- Network diagrams are used to show additional important information such as:
  - ✓ Critical paths
  - ✓ Earliest and latest completion times
  - ✓ Slack times

- After estimating task durations and variances, the next step is to calculate the critical path.
  - ✓ A path is any combination of nodes from the start node to the last node.
  - ✓ For example, our network diagram shows the following two paths:
    - Path #1: (1.1.1) – (1.1.2) – (1.2.1) – (1.2.2) – (1.3.4) – (1.3.5) – (1.3.6)
    - Path #2: (1.1.1) – (1.1.2) – (1.3.1) – (1.3.2) – (1.3.3) – (1.3.4) – (1.3.5) – (1.3.6)



THE PLANNING STAGE
- NETWORK DIAGRAMS -

- The *critical path* is defined as the path with the longest duration.
  - ✓ Using the expected times computed:
    - The expected duration of Path#1 is 55 days.
    - The expected duration of Path#2 is 49 days.
  - ✓ Therefore, the critical path for this problem is Path#1.
  - ✓ The *critical path duration* is 55 days.



- The activities/tasks along the critical path are called critical activities.
  - ✓ If any of the critical activities gets delayed, the entire project will be delayed!

THE PLANNING STAGE
- NETWORK DIAGRAMS -

- The next step is to calculate earliest and latest times for activities to start and finish.
  - ✓ That is, we need to calculate the earliest time that we can begin each activity, and the earliest time that the activity can be completed.
  - ✓ We denote these terms as *earliest start (ES)* and *earliest finish (EF)* times.

- Similarly, we calculate the latest time that we can begin each activity, and the latest time that the activity can be completed.
  - ✓ We denote these terms as *latest start (LS)* and *latest finish (LF)* times.
  - ✓ The difference between the LS and ES of an activity is called the *slack time* for the activity.
  - ✓ Because activities along the critical path cannot be delayed without delaying the entire project's duration, *only non-critical path activities can have slack times*.

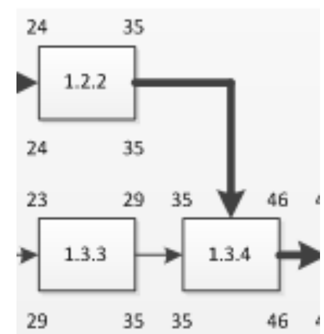
THE PLANNING STAGE
- NETWORK DIAGRAMS -

- To calculate earliest times, move from the start node to the other nodes.
  - ✓ Since activity 1.1.1 begins at time 0,
    - Its ES is 0
    - Its EF time is its expected time duration (i.e., 6 days)
  - ✓ Any immediate preceding activity can start as early as activity 1.1.1 ends.
    - Therefore, the ES of activity 1.1.2 is 6 days, and its EF is 9 days, which is the sum of its ES and expected time duration.
    - Similarly, the ES of activity 1.3.1 is 9 days, and its EF is 20 days.
  - ✓ In cases where an activity has more than one predecessor, the largest EF time of the predecessors become the ES time of the activity.
    - For example, activity 1.3.4 has two predecessors: 1.2.2 (with an EF time of 35 days) and 1.3.3 (with an EF time of 29 days).
      - Since activity 1.3.4 can be started only when both of its predecessors complete, its earliest possible start time is 35 days.

THE PLANNING STAGE
- NETWORK DIAGRAMS -

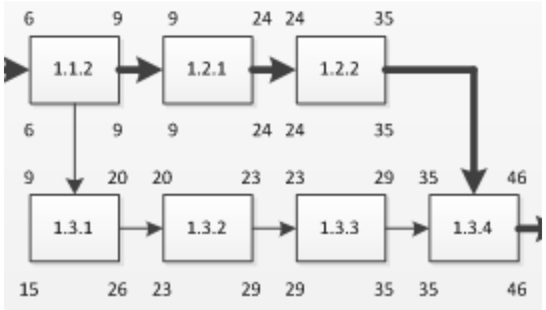
- To calculate latest times, we move from the last node to the start node.
  - For critical activities:
    - LF = EF and
    - LS = ES

  - This is not the case for non-critical activities. For example, consider 1.3.3:
    - If this activity finishes at its EF time of 29 days, activity 1.3.4 (i.e., its successor) will still be unable to start because it needs to wait for activity 1.2.2 to finish at day 35.
      - Thus, activity 1.3.3 can actually finish as late as day 35 and not affect the critical path duration of the project.
      - Consequently, the LF time for activity 1.3.3 is 35 days, and its LS time is 29 days
        - 29 days = its LF minus its expected time duration of 6 days.



    - Activity 1.3.2 can finish as late as the LF time of its successor 1.3.3.
      - Therefore, its LF time is 29 days, and its LS time 26 days.

The Planning Stage
- Network Diagrams -



- ✓ When the earliest and latest times for each activity are completed, it can be easily seen that only the non-critical activities (i.e., 1.3.1, 1.3.2, and 1.3.3) have slack times.
  - The total allowable slack time for these activities is 6 days.
    - E.g., if a slack time of 5 days is used in activity 1.3.1, then only one of the remaining non-critical activities can be delayed by 1 day.
    - E.g., if a slack time of 6 days is used in activity 1.3.1, then the network will have two critical paths.
      - » This means that in this case, non-critical activities 1.3.2 and 1.3.3 will become critical activities (i.e., no slack times).

The Planning Stage
- Network Diagrams -

- Probability of Time to Completion
  - ✓ In the previous example, expected task durations were calculated based on estimates of minimum, average, and maximum durations using the beta distribution.
  - ✓ The other obvious option is to overlook these three parameters and estimate a single average task duration instead.
    - The drawback of this option is that it excludes a range of possibilities (e.g., max/min values) and accurate estimates of task times with a single parameter seems unlikely.
  - ✓ Benefits of using min, max and avg. parameters
    - Setting values for these three parameters –instead of one estimate—alleviates the estimation process for decision makers.
    - Estimating these three parameters allows us to develop variance calculations, and then use the variances to estimate the probability of completing the project within a specific time.

The Planning Stage
- Network Diagrams -

- ✓ Estimating the probability of completing a project by a particular time period requires using the *cumulative standard normal distribution* to obtain Z-values.
  - The standard normal distribution is a normal distribution with a mean equal to zero and a standard deviation equal to one.
  - A *Z-value* represents the number of standard deviations from the mean (either to the right or to the left of the mean) of the standard normal distribution.

- ✓ Use the expected completion times of the critical activities and their variances to determine the probability of completing the critical path activities by a deadline.
  - First, transform (a.k.a. "normalize") the estimated project duration time into a Z-value using the following equation:

$$Z = \frac{D - EPT}{\sqrt{\sigma^2_{EPT}}}$$

    - Z = the number of standard deviations of a standard normal distribution
    - D = the desired completion time (usually in days) of a project
    - EPT =expected project completion time (this is the critical path duration)
    - σ^2_{EPT} = the variance of the critical path

THE PLANNING STAGE
- NETWORK DIAGRAMS -

- Example of the cumulative normal distribution table

Z	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
0.0	0.5000	0.5040	0.5080	0.5120	0.5160	0.5199	0.5239	0.5279	0.5319	0.5359
0.1	0.5398	0.5438	0.5478	0.5517	0.5557	0.5596	0.5636	0.5675	0.5714	0.5754
0.2	0.5793	0.5832	0.5871	0.5910	0.5948	0.5987	0.6026	0.6064	0.6103	0.6141
0.3	0.6179	0.6217	0.6255	0.6293	0.6331	0.6368	0.6406	0.6443	0.6480	0.6517
0.4	0.6554	0.6591	0.6628	0.6664	0.6700	0.6736	0.6772	0.6808	0.6844	0.6879
0.5	0.6915	0.6950	0.6985	0.7019	0.7054	0.7088	0.7123	0.7157	0.7190	0.7224
0.6	0.7258	0.7291	0.7324	0.7357	0.7389	0.7422	0.7454	0.7486	0.7518	0.7549
0.7	0.7580	0.7612	0.7642	0.7673	0.7704	0.7734	0.7764	0.7794	0.7823	0.7852
0.8	0.7881	0.7910	0.7939	0.7967	0.7996	0.8023	0.8051	0.8079	0.8106	0.8133
0.9	0.8159	0.8186	0.8212	0.8238	0.8264	0.8289	0.8315	0.8340	0.8365	0.8389
1.0	0.8413	0.8438	0.8461	0.8485	0.8508	0.8531	0.8554	0.8577	0.8599	0.8621
1.1	0.8643	0.8665	0.8686	0.8708	0.8729	0.8749	0.8770	0.8790	0.8810	0.8830
1.2	0.8849	0.8869	0.8888	0.8907	0.8925	0.8944	0.8962	0.8980	0.8997	0.9015
1.3	0.9032	0.9049	0.9066	0.9082	0.9099	0.9115	0.9131	0.9147	0.9162	0.9177
1.4	0.9192	0.9207	0.9222	0.9236	0.9251	0.9265	0.9279	0.9292	0.9306	0.9319
1.5	0.9332	0.9345	0.9357	0.9370	0.9382	0.9394	0.9406	0.9418	0.9430	0.9441
1.6	0.9452	0.9463	0.9474	0.9485	0.9495	0.9505	0.9515	0.9525	0.9535	0.9545
1.7	0.9554	0.9564	0.9573	0.9582	0.9591	0.9599	0.9608	0.9616	0.9625	0.9633
1.8	0.9641	0.9649	0.9656	0.9664	0.9671	0.9678	0.9686	0.9693	0.9700	0.9706
1.9	0.9713	0.9719	0.9726	0.9732	0.9738	0.9744	0.9750	0.9756	0.9762	0.9767
2.0	0.9773	0.9778	0.9783	0.9788	0.9793	0.9798	0.9803	0.9808	0.9812	0.9817
2.1	0.9821	0.9826	0.9830	0.9834	0.9838	0.9842	0.9846	0.9850	0.9854	0.9857
2.2	0.9861	0.9865	0.9868	0.9871	0.9875	0.9878	0.9881	0.9884	0.9887	0.9890
2.3	0.9893	0.9896	0.9898	0.9901	0.9904	0.9906	0.9909	0.9911	0.9913	0.9916
2.4	0.9918	0.9920	0.9922	0.9925	0.9927	0.9929	0.9931	0.9932	0.9934	0.9936
2.5	0.9938	0.9940	0.9941	0.9943	0.9945	0.9946	0.9948	0.9949	0.9951	0.9952
2.6	0.9953	0.9955	0.9956	0.9957	0.9959	0.9960	0.9961	0.9962	0.9963	0.9964
2.7	0.9965	0.9966	0.9967	0.9968	0.9969	0.9970	0.9971	0.9972	0.9973	0.9974
2.8	0.9974	0.9975	0.9976	0.9977	0.9977	0.9978	0.9979	0.9980	0.9980	0.9981
2.9	0.9981	0.9982	0.9983	0.9983	0.9984	0.9984	0.9985	0.9985	0.9986	0.9986
3.0	0.9987	0.9987	0.9987	0.9988	0.9988	0.9989	0.9989	0.9989	0.9990	0.9990
3.1	0.9990	0.9991	0.9991	0.9991	0.9992	0.9992	0.9992	0.9992	0.9993	0.9993
3.2	0.9993	0.9993	0.9994	0.9994	0.9994	0.9994	0.9994	0.9995	0.9995	0.9995
3.3	0.9995	0.9995	0.9996	0.9996	0.9996	0.9996	0.9996	0.9996	0.9996	0.9997
3.4	0.9997	0.9997	0.9997	0.9997	0.9997	0.9997	0.9997	0.9997	0.9998	0.9998
3.5	0.9998	0.9998	0.9998	0.9998	0.9998	0.9998	0.9998	0.9998	0.9998	0.9998

THE PLANNING STAGE
- NETWORK DIAGRAMS -

- ✓ With the Z-value, find the probability of completing the project by a certain time using a cumulative normal distribution table
  - In this table we can find the probability associated with a Z-value.

- ✓ For example, assume that a design project has an expected critical path duration of 75 days, with a critical path variance of 36 days. What is the probability of completing the project in 85 days.
  - Step #1: Calculate a Z-value
    - Z-value = 1.67 using the equation $Z = \frac{D-EPT}{\sqrt{\sigma^2_{EPT}}}$
  - Step #2: Find the probability using the cumulative normal distribution table
    - Probability = 0.9525 (go down the left column until 1.6, then across to the right until column 0.07).
    - This means that there is a 95.25 percent probability of finishing the design project in 85 days, given the variance in the expected project duration time.

THE PLANNING STAGE
-ESTABLISH CHANGE CONTROL POLICY-

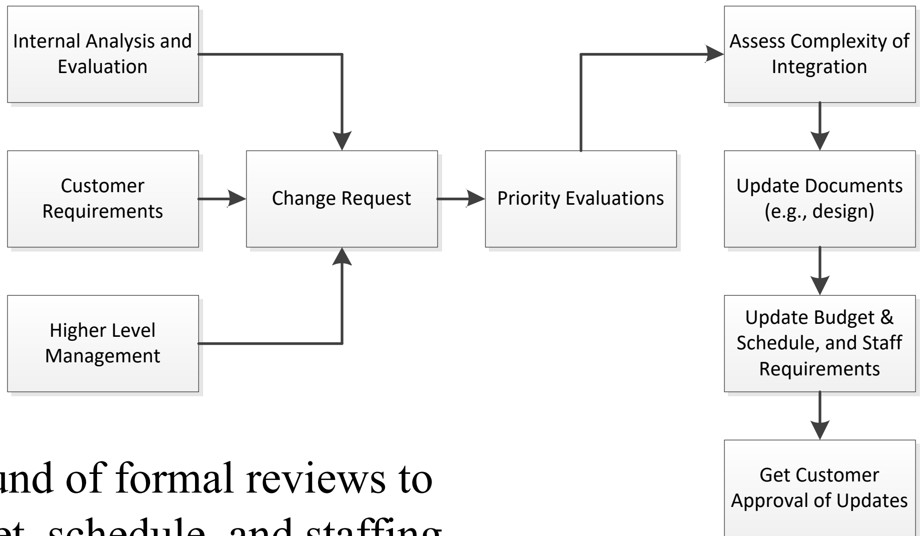
- Managing the design of software systems is often a very complex endeavor, especially when changes to the design occur after the design has successfully gone through peer reviews.
  - ✓ Even a small design change can result in system failures if not managed appropriately.
  - ✓ To avoid major problems later in the project, it is critical that a formal change control process gets adopted early in the planning phase.

- A change control process helps to ensure the technical integrity of the design in the presence of changes.

- Let's see an example of a change control process…..

ESTABLISH CHANGE CONTROL POLICY (AN EXAMPLE)

- Steps:
  - ✓ A change request is submitted to the Change Control Authority (i.e., the project manager, the design lead, or the group that makes the final decision on the status and priority of a change.)
  - ✓ Request is evaluated based on technical merit, complexity of integration, & potential side effects.
  - ✓ From the assessments, the request is assigned a change priority.
  - ✓ Developers then take their initial design documents through another round of formal reviews to obtain updated estimates for the budget, schedule, and staffing.
  - ✓ Finally, updated design documents and schedule estimates must then be approved by the customer since they represent changes to the original estimates.



THE IMPLEMENTATION STAGE
- EARNED VALUE MANAGEMENT (EVM)-

- During the implementation stage, main managerial activities deal with *monitoring and controlling* that the design project activities are being accomplished according to the plan (i.e., schedule and budget).
  - ✓ Monitoring and controlling mechanisms exist to:
    - Identify any deviations from the plan, and
    - Ensure that these deviations get corrected in a timely manner in order to avoid major future problems.

- Appropriate project management tools for this phase are:
  - ✓ *Gantt charts* and
  - ✓ *Earned value management*.

- Since we have already covered Gantt charts in the previous section, let's focus now on *earned value management* (EVM).

The Implementation Stage
- Earned Value Management (EVM)-

- EVM is a project management technique to determine the progress of tasks based on the value of the work currently completed versus the work that was expected to be completed at that particular time.

- The term used to define the dollar amount of the work currently completed is called *earned value (EV)*.
  - ✓ EV is calculated by multiplying the percentage work completed times the planned total cost of the work when completed.

    EV = (% work completed) * planned cost for the work

  - ✓ A key factor is to accurately estimate the percentage of work completed,
    - Challenges to estimate % work completed:
      - Complexity of tasks and high degree of subjectivity that is typically involved.
      - Workers tend to inflate the % of work completed to give the perception that work is progressing smoothly for the completion of the task.
    - There are various ways to estimate percentage completion.
      - Let's see some guidelines to estimate % completion of a task…..

THE IMPLEMENTATION STAGE
- EARNED VALUE MANAGEMENT (EVM)-

- Guidelines for estimating % completion for a task [1]

Approach Name	Brief Description
50-50	Assumes 50% task completion as soon as the task begins, and 100% completion when the task is completed
0-100 percent	Assumes 0% until the task gets fully completed
Critical input use	Percent completion is a function of how much of a critical input is used versus its overall expected total to be used
Proportionality	Assumes work completion as the proportion of actual time (or cost) spent over the total time (or cost) planned for the task to be completed

- With EV calculations for each task (i.e., each element in the WBS), we can:
  - ✓ Examine deviations to cost and schedule, and then
  - ✓ Establish a dollar amount to these deviations.

The Implementation Stage
- Earned Value Management (EVM)-

- EVM Definition of Parameters

Parameter	Description
EV	Also called *budgeted cost of work performed (BCWP),* it is the dollar amount of the work currently completed
AC	Also called *actual cost of work performed (ACWP)*, it represents the amount of money that was spent in the work currently completed
PV	Also called *budgeted cost of work scheduled (BCWS)*, it represents the value of the work that was expected to be completed at that point in time of the schedule
CV	Describes the difference between the value of work completed and the actual cost spent to complete it
SV	Describes the difference between the value of work currently completed and the cost of the work that was expected to be completed
CPI	Describes the current cost efficiency of the work (CPI < 1 means over budget; CPI > 1 means under budget; CPI = 1 means right on budget)
SPI	Describes the current schedule efficiency of the work (SPI < 1 means behind schedule; SPI > 1 means ahead of schedule; SPI = 1 means right on schedule)
BAC	Total budget for the project
ETC	Represents the estimated cost remaining to complete the project
EAC	Represents is the total cost of the entire project

- Equations for EVM parameters

Parameter	Equation
EV	EV = (% work completed) * planned cost for the work
CV	CV = EV - AC
SV	SV = EV - PV
CPI	CPI = EV / AC
SPI	SPI = EV / PV
ETC	ETC = (BAC – EV) / CPI
EAC	EAC = ETC + AC

THE IMPLEMENTATION STAGE
- EARNED VALUE MANAGEMENT (EVM)-

- Let's show the implementation of the EVM equations with an example:

Sara is managing a software design project composed of 10 tasks. Each task was estimated to cost $1,000; therefore, BAC = $10,000. Each task was estimated to be completed in a month; therefore, the total planned duration for the project was estimated to be 10 months. Since each task is expected to have the same cost and duration, the completion of an individual task represents 10% completion of the design project. In the current fifth month, only three tasks have been fully completed at a cost of $4,000. Sara needs to provide upper management with current progress status.

- We can now use the EVM equations

Parameter	Calculation
EV	0.3 * $10,000 = $3,000.
CV	$3,000 - $4,000 = -$1,000
SV	$3,000 - $5,000 = -$2,000
CPI	$3,000 / $4,000 = 0.75
SPI	$3,000 / $5,000 = 0.6000
ETC	($10,000 - $3,000) / 0.75 = $9,333
EAC	$9,333 + $7,000 = $16,333

- Let's examine what this means…

THE IMPLEMENTATION STAGE
- EARNED VALUE MANAGEMENT (EVM)-

- EV is calculated to be \$3,000 given that the three tasks completed represent 30% of the project, and the budget for the entire project is \$10,000.

- To calculate SV, we use a PV of \$5,000 because each task was budgeted at \$1,000 and five tasks should have been completed in the fifth month.

- The Table in the previous slide shows the calculations for the rest of the EVM parameters.

- These results clearly show that the project is over both budget and schedule estimates.
  - ✓ The CPI index, for example, shows that only \$0.75 of earned value was received for every dollar actually spent.
  - ✓ The EAC shows an estimated \$6,333 over budget amount.
  - ✓ These alarming values give Sara and upper management the opportunity to implement risk mitigation policies to either get the project to its budgeted path, or at least attempt to minimize losses.

The Termination Stage

- During the termination stage, key activities involve verification that everything is in place for a smooth transition into the code construction phase.

- Some of the activities during the termination stage include:
  - ✓ Making sure that the latest versions of the design documents are securely stored based on configuration management procedures,
  - ✓ Updating schedule and cost current values,
  - ✓ Re-evaluating schedule and budget plans based on the resulting performance measures of the design phase, and
  - ✓ Communicating results to upper management.

WHAT'S NEXT…

- This session accomplished the following:
  - ✓ Defined the *Planning* component of the software design management framework
  - ✓ Defined the *Implementation* component of the software design management framework
  - ✓ Defined the *Termination* component of the software design management framework

- Next session will focus on the *Leadership & Ethics* component of the design management framework,

QUESTIONS?

References

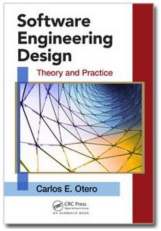
[1] Meredith, Jack, and Samuel Mantel. *Project Management: A Managerial Approach.* 7th. John Wiley & Sons, 2009.

CHAPTER 10: MANAGEMENT, LEADERSHIP, & ETHICS

-- Dr. L. Daniel Otero, Florida Institute of Technology --

SESSION III: LEADERSHIP AND ETHICS

Software Engineering Design: Theory and Practice
by Carlos E. Otero



SESSION'S AGENDA

- Leading the Design Effort
  - ✓ Personality Traits and Leadership
  - ✓ Personality Dimension
  - ✓ Traits of Effective Leaders
  - ✓ Ethical Leadership
  - ✓ Power
- Key Leadership Skills
  - ✓ Communication
  - ✓ Networking
  - ✓ Motivation
  - ✓ Negotiation
- Ethics in Software Design
  - ✓ Public and Product Principles
  - ✓ Judgment Principle
  - ✓ Management Principle

LEADING THE DESIGN EFFORT

- Leadership plays a key role in the success or failure of complex projects.
  - ✓ Therefore, it is necessary to cover important concepts related to leadership in the design effort.
- The field of leadership is one that has been researched by academicians and practitioners for many years.
  - ✓ There is an extensive body of knowledge and literature related to this field.
- This session highlights key leadership concepts that are relevant to our study of software designs.
  - ✓ A significant part of the material covered in this section was gathered from [1]; therefore, the author refers the readers to this work for more detailed explanations of leadership concepts.

LEADING THE DESIGN EFFORT
- PERSONALITY TRAITS AND LEADERSHIP -

- *Traits* are personal characteristics that help to describe a person. For example:
  - ✓ Highly sociable trait: This trait is present in someone that has the ability to communicate well with others and start friendly conversations with unknown people
  - ✓ Achievement trait: This trait is present in someone that works hard and tends to go the extra mile to complete assigned tasks

- The combination of traits that a person has defines that person's *personality*
  - ✓ Personality significantly affects our decisions.
    - For example, a highly sociable person with an achievement trait is expected to make decisions such as volunteering to make presentations to clients.
  - ✓ Given that a good decision-making ability is a major part of being an effective leader, it is important to understand the relationship between traits, personality, decision-making, and leadership.

LEADING THE DESIGN EFFORT
- PERSONALITY DIMENSIONS -

- Researchers have identified five personality dimensions to categorize groups of traits.
  - ✓ The objective is to be able to classify someone's personality into one of these dimensions.
  - ✓ The five dimensions are:
    - *agreeableness*,
    - *surgency*,
    - *adjustment*,
    - *conscientiousness*, and
    - *openness to experience*.

    NOTE: Although called by different names by various researchers, we will refer to [1] and call them the *Big Five Model of Personality*.
  - ✓ Let's describe each of the five personality dimensions…

LEADING THE DESIGN EFFORT
- PERSONALITY DIMENSIONS -

- The *agreeableness dimension* contains traits that are related to getting along with people.
  - ✓ Individuals that are strong in agreeableness are friendly, sociable, and compassionate.

- The *surgency dimension* corresponds to personalities that are dominant (i.e., want to be in control), enjoy competition, and are willing to confront others, among other things.
  - ✓ Individuals who are weak in surgency are typically followers.

- The *adjustment dimension* is related to emotional stability.
  - ✓ Individuals who are strong in this dimension are considered stable people who are in control of themselves, positive minded, and react well under pressure.
  - ✓ Those that are weak in adjustment are considered unstable, and typically are characterized by being negative and perform poorly under pressure.

- The *conscientiousness dimension* contains traits related to achievement.
  - ✓ Individuals that are strong in this dimension are those that are willing to work extra hours and make sacrifices to reach the assigned objectives.

- The *openness to experience dimension* is related to individuals that are flexible to change, open minded, and attempt enjoy trying new things.

LEADING THE DESIGN EFFORT
- PERSONALITY DIMENSIONS -

- Understanding these dimensions is important because researchers have found varying degrees of correlations between leadership and each of the five personality dimensions.

  - ✓ For example, [2] analyzed over 70 prior studies to determine correlations between the five personality dimensions and leadership.

  - ✓ Results showed that the surgency dimension was the highest with a 0.31 correlation.
    - This means that traits corresponding to surgency can be used, to some extent, to describe leadership.
    - The second highest was conscientiousness with a 0.28 correlation, followed by openness to experience with 0.24, and agreeableness with 0.08.

LEADING THE DESIGN EFFORT
- TRAITS OF EFFECTIVE LEADERS -

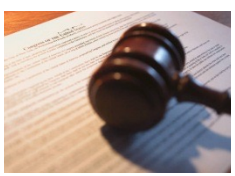
- An approach to studying personality dimensions is to investigate common individual traits that effective leaders possess, without grouping traits into personality dimensions.
- Some of the most important traits are:

Principle	Description (Trait's characteristics in a person)
Dominance	This trait describes a person that wants to be a manager and takes control. Successful leaders with this trait are not overly bossy and avoid the bullying style.
High energy	This trait describes a positive-minded person that works hard to achieve objectives, and is good at taking initiative.
Integrity	This trait describes a person that is highly ethical and trustworthy.
Flexibility	This trait describes a person that can adapt well to new and different situations.
Self-confidence	This trait describes a person who trusts his/her judgment, initiatives, intelligence, and ideas. This is not to be confused with arrogance.
Stability	This trait describes a person that is able to control emotions and act well under pressure.
Intelligence	This trait describes a person that possesses a high cognitive ability to solve problems and make decisions.
Sensitivity to others	This trait describes a person that focuses on the feelings of others and strongly considers them.

LEADING THE DESIGN EFFORT
- ETHICAL LEADERSHIP -

- Ethics can be defined as a set of moral principles that facilitates the process to distinguish between right and wrong behavior.

- Ethical behavior positively correlates with leadership effectiveness [3].
  - ✓ It is well known that one of the best ways to *lead is by example*.
  - ✓ An ethical leader sets the tone for employees to conduct themselves ethically.
  - ✓ Being an ethical leader can be very challenging due to the risk of rejection or loss associated with ethical decisions.
    - To overcome this challenge:
      - Focus on a higher purpose while going through the decision process that involves moral behavior.
      - Find support from other ethical people (e.g., friends, family, or co-workers).



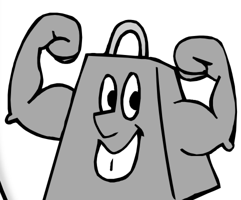
  - ✓ Most companies have their own internal documents that describe expected ethical behavior from employees.
    - It is very important for leaders to conform to these guidelines, and discuss them with employees.

Leading the Design Effort
- Power -

- The potential influence that leaders have over others is called *power*.

- "Potential" means that it is actually the perception of power, and not the power itself, that influences followers.

- The two main sources of power are *position power* and *personal power*.
  - ✓ Position power is related to hierarchical position levels (higher levels = higher power)
    - Example: The president of a software company has more potential power than any other employee in the company because of position power.
  - ✓ Position power allows managers to influence employees in order to reach planned objectives.
    - *Reward Power* is a type of position power.
      - Managers can influence employees with something of value to them (e.g., performance evaluations).
    - *Coercive power* is another type of position power
      - Managers can influence others by the idea of punishment and withholding of rewards.



LEADING THE DESIGN EFFORT
- POWER -

- ✓ *Personal power* relates to the potential influence that a person's behavior has to influence others.
  - Being positive, assertive, and hard worker are some of the characteristics that increase personal power.

- ✓ There are various types of personal power.
  - The *expert power* is based influencing others on a leader's skill and knowledge.
    - For example, being an expert in developing architectural designs will influence others to follow the expert.
    - The expert power of an individual gets stronger as people with similar expertise levels become fewer.

  - The *connection power* refers to the ability of a person to influence others because of the person's relationships with influential people.

Key Leadership Skills - Communication -

- Good communication skills are essential for effective leadership.
  - ✓ A high % of a managers' time is spent in communications.

- Leaders can influence and establish productive relationships with others through communication.

- Truly effective communication occurs when the information passed is equally and fully understood by all parties.
  - ✓ Advise:
    - Plan in advance the message that needs to be conveyed.
    - The goal of the message needs to be clearly understood.
    - Think about the best timing to convey the message, how will the message be delivered (e.g., oral, written), and where (e.g., company's auditorium, employee's office, etc.)

KEY LEADERSHIP SKILLS
- COMMUNICATION -

- Effective *oral communication* can be challenging (and often takes much practice).
  - ✓ Have a process in place so that focus can be placed in the individual elements of the process.
  - ✓ Five-step process for effectively sending oral messages [1]:
    - (1) Develop a rapport.
      - Try establishing a good relationship with the receiving party; begin small conversations related to the message.
    - (2) Clearly state the objective of the message.
    - (3) Actually transmit the message effectively.
    - (4) Perceive the receivers' understanding of the message.
      - It is important that the message was well understood; otherwise its objectives were not achieved.
      - This can be accomplished by asking direct questions, or reading the receiver's expressions.
    - (5) Get a commitment from the receiver.
      - Only if the goal is to influence the receiver to accomplish a task (e.g., design document)
      - The leader must be convinced that the receiver is capable of completing the task by its deadline.

Key Leadership Skills - Communication -

- Another type of communication is *written communication*.
  - ✓ This type of communication is now more important than ever, and it is mainly attributed to the e-mail technology.
- The key elements for effective written communication are content and structure.
  - ✓ The message must be structured in such a way that the information it intends to transmit (i.e., content) is clearly described, and the flow of information is smooth and easy to follow.
- It is critical for the communicator to clearly understand the intended objective of the message before writing it. Some advise:
  - ✓ Make an outline with the main points that need to be transmitted.
  - ✓ Avoid including unnecessary information. Messages must be kept short, simple and to the point.
- Writing is a skill! It takes effort and practice to become good at it.
  - ✓ Have others review your work so that you can learn from their feedback.

KEY LEADERSHIP SKILLS
- NETWORKING -

- Networking can be defined as a skill that focuses on building relationships with others through effective communications.

- Communications and networking skills are closely related to each other.

- Networking is particularly important for leadership because it facilitates the process of meeting objectives.

- As with any skill, networking can be improved with practice.

KEY LEADERSHIP SKILLS
- MOTIVATION -

- There exists a positive correlation between employees' motivation levels and their productivity.
  - ✓ A leader must motivate others to work hard to achieve particular project goals.

- The *hierarchy of needs theory* is a motivational theory that describes five types of needs through which employees are motivated.

Type of Need	Activities to Meet the Needs
Physiological	• Adequate salaries • Allowance of breaks • Adequate working conditions
Safety	• Safe working conditions • Salary increases (considering inflation) • Fringe benefits
Social	• Social activities that conform to individual behavior • Team building retreats • Team sports • Lunch gatherings
Esteem	• Raises based on performance • Awards • Public recognition • Participation in decision-making
Self-Actualization	• Skill development activities • Promotions • Increase control of an employee's task

KEY LEADERSHIP SKILLS
- NEGOTIATION -

- Negotiation skills are important part of any management job.

- Research studies have resulted in various sets of guidelines that can help to improve a person's negotiation skills.
  - One such set of guidelines is called *principled negotiation*

- *Principled negotiation* is composed of four guidelines.
  - (1) *Separate the people from the problem*.
    - This principle helps to keep the focus on the problem at hand rather than on interpersonal issues.
  - (2) *Focus on interests rather than positions*.
    - This principle helps to keep the focus of the negotiation on the interests of people rather than their positions.
  - (3) *Generate options before trying to reach an agreement*.
    - This principle promotes creativity and reminds the negotiating parties to brainstorm to find various potential solutions that can be brought to the negotiation.
  - (4) *Insist on using objective criteria*.
    - This principle promotes decision-making based on reasonable standards rather than on subjective ones.

Ethics in Software Design

- Engineers must abide to the highest possible standards when developing software systems
  - ✓ Helps to make the software engineering profession beneficial and highly respected.
  - ✓ Engineers have the responsibility to public welfare, including health and safety.

- Various *principles of ethics* collectively ensure that professionals in the software engineering field adhere to high-levels of ethical conduct.

- The IEEE-CS/ACM organizations developed a *Software Engineering Code of Ethics (SECE)* [4] as the standard set of ethical guidelines that engineers must adhere to.

ETHICS IN SOFTWARE DESIGN

- Principles of Ethics in Software Engineering [4]

Principle	Description
Public	Software engineers shall act consistently with the public interest.
Client and Employer	Software engineers shall act in a manner that is in the best interests of their client and employer consistent with the public interest.
Product	Software engineers shall ensure that their products and related modifications meet the highest professional standards possible.
Judgment	Software engineers shall maintain integrity and independence in their professional judgment.
Management	Software engineering managers and leaders shall subscribe to and promote an ethical approach to the management of software development and maintenance.
Profession	Software engineers shall advance the integrity and reputation of the profession consistent with the public interest.
Colleagues	Software engineers shall be fair to and supportive of their colleagues.
Self	Software engineers shall participate in lifelong learning regarding the practice of their profession and shall promote an ethical approach to the practice of the profession.

Ethics in Software Design
- Public and Product Principles -

- The *public principle* state that software engineering professionals must:
  - ✓ Take responsibility for their own work
  - ✓ Ensure that their work positively affects the public good
  - ✓ Strive for high quality within acceptable cost and schedule thresholds

No.	Guideline Description
1.01	Engineers must accept full responsibility for their own work.
1.02	Moderate the interests of the software engineer, the employer, the client and the users with the public good.
1.03	Approve software only if they have a well-founded belief that it is safe, meets specifications, passes tests, and does not diminish quality of life, diminish privacy or harm the environment. The ultimate effect of the work should be to the public good.
1.04	Disclose to appropriate persons or authorities any actual or potential danger to the user, the public, or the environment, that they reasonably believe to be associated with software or related documents.
1.05	Cooperate in efforts to address matters of grave public concern caused by software, its installation, maintenance, support or documentation.
1.06	Be fair and avoid deception in all statements, particularly public ones, concerning software or related documents, methods and tools.
1.07	Consider issues of physical disabilities, allocation of resources, economic disadvantage and other factors that can diminish access to the benefits of software.
1.08	Be encouraged to volunteer professional skills to good causes and contribute to public education concerning the discipline.

Ethics in Software Design
- Public and Product Principles -

- Guidelines related to the *product principle*

No.	Guideline Description
3.01	Strive for high quality, acceptable cost and a reasonable schedule, ensuring significant tradeoffs are clear to and accepted by the employer and the client, and are available for consideration by the user and the public.
3.02	Ensure proper and achievable goals and objectives for any project on which they work or propose.
3.03	Identify, define and address ethical, economic, cultural, legal and environmental issues related to work projects.
3.04	Ensure that they are qualified for any project on which they work or propose to work by an appropriate combination of education and training, and experience.
3.05	Ensure an appropriate method is used for any project on which they work or propose to work.
3.06	Work to follow professional standards, when available, that are most appropriate for the task at hand, departing from these only when ethically or technically justified.
3.07	Strive to fully understand the specifications for software on which they work.
3.08	Ensure that specifications for software on which they work have been well documented, satisfy the users requirements and have the appropriate approvals.
3.09	Ensure realistic quantitative estimates of cost, scheduling, personnel, quality and outcomes on any project on which they work or propose to work and provide an uncertainty assessment of these estimates.
3.10	Ensure adequate testing, debugging, and review of software and related documents on which they work.
3.11	Ensure adequate documentation, including significant problems discovered and solutions adopted, for any project on which they work.
3.12	Work to develop software and related documents that respect the privacy of those who will be affected by that software.
3.13	Be careful to use only accurate data derived by ethical and lawful means, and use it only in ways properly authorized.
3.14	Maintain the integrity of data, being sensitive to outdated or flawed occurrences.
3.15	Treat all forms of software maintenance with the same professionalism as new development.

ETHICS IN SOFTWARE DESIGN
- JUDGMENT PRINCIPLE -

- The *judgment principle* deals with maintaining integrity in professional judgment.
  - ✓ Integrity and moral values of a person plays a major role when making judgment calls.

- Guidelines related to this principle are:

No.	Guideline Description
4.01	Temper all technical judgments by the need to support and maintain human values.
4.02	Only endorse documents either prepared under their supervision or within their areas of competence and with which they are in agreement.
4.03	Maintain professional objectivity with respect to any software or related documents they are asked to evaluate.
4.04	Not engage in deceptive financial practices such as bribery, double billing, or other improper financial practices.
4.05	Disclose to all concerned parties those conflicts of interest that cannot reasonably be avoided or escaped.
4.06	Refuse to participate, as members or advisors, in a private, governmental or professional body concerned with software related issues, in which they, their employers or their clients have undisclosed potential conflicts of interest.

Ethics in Software Design
- Management Principle -

- Guidelines related to the *management principle*

No.	Guideline Description
5.01	Ensure good management for any project on which they work, including effective procedures for promotion of quality and reduction of risk.
5.02	Ensure that software engineers are informed of standards before being held to them.
5.03	Ensure that software engineers know the employer's policies and procedures for protecting passwords, files and information that is confidential to the employer or confidential to others.
5.04	Assign work only after taking into account appropriate contributions of education and experience tempered with a desire to further that education and experience.
5.05	Ensure realistic quantitative estimates of cost, scheduling, personnel, quality and outcomes on any project on which they work or propose to work, and provide an uncertainty assessment of these estimates.
5.06	Attract potential software engineers only by full and accurate description of the conditions of employment.
5.07	Offer fair and just remuneration.
5.08	Not unjustly prevent someone from taking a position for which that person is suitably qualified.
5.09	Ensure that there is a fair agreement concerning ownership of any software, processes, research, writing, or other intellectual property to which a software engineer has contributed.
5.10	Provide for due process in hearing charges of violation of an employer's policy or of this Code.
5.11	Not ask a software engineer to do anything inconsistent with this Code.
5.12	Not punish anyone for expressing ethical concerns about a project.

SUMMARY

- In this session, we presented:
  - Key concepts to leading the design effort
    - Personality Traits and Leadership
    - Personality Dimension
    - Traits of Effective Leaders
    - Ethical Leadership
    - Power
  - Key leadership skills
    - Communication
    - Networking
    - Motivation
    - Negotiation
  - Key principles of ethics in software design
    - Public and Product Principles
    - Judgment Principle
    - Management Principle

QUESTIONS?

References

[1] Lussier, Robert, and Christopher Achua. *Leadership: Theory, Application, & Skill Development.* 4th. Cengage Learning, 2010.

[2] Judge, T. A., R. Ilies, J. E. Bono, and M. W. Gerhardt. "Personality and Leadership: A Qualitative and Quantitative Review." *Journal of Applied Psychology* 87, no. 4 (2007): 765-768.

[3] Veiga, J. F. "Special Topic Ethical Behavior in Management, Bringing Ethics into the Mainstream: An Introduction to the Special Topic." *Academy of Management Executive* 18, no. 2 (2004): 37-38.

[4] IEEE Computer Society. "Software Engineering Code of Ethics and Professional Practice." 2010. http://www.computer.org/portal/web/certification/resources/code_of_ethics.