Key Ideas

- Determine how projects are selected and understand the various approached to the SDLC that are used to structure a development project.
- Understand how to select a project methodology based on project characteristics and become familiar with project estimation.
- Create a project work plan and understand how to staff a project.
- Important techniques to coordinate and manage a project, and how to manage risk on the project.
-

PROJECT SELECTION

- Systems projects today are evaluated in the context of an entire portfolio of projects.
- Determination of a project's contribution to an entire portfolio of a project reinforces the need for a feasibility study.
- Portfolio management takes into consideration the different of projects that exist in an organization
  - An approval committee must be selective about where to allocate resources as most organizations have limited funds.

If there are three potentially high-payoff projects, and they all have the same risk, then maybe only one of the projects will be selected.

How Do Projects Begin?

- Business needs should drive projects.
- Project sponsor recognizes business need for new system and desires to see it implemented.
- Business needs determine the system's functionality (what it will do).
- The project's business value should be clear.

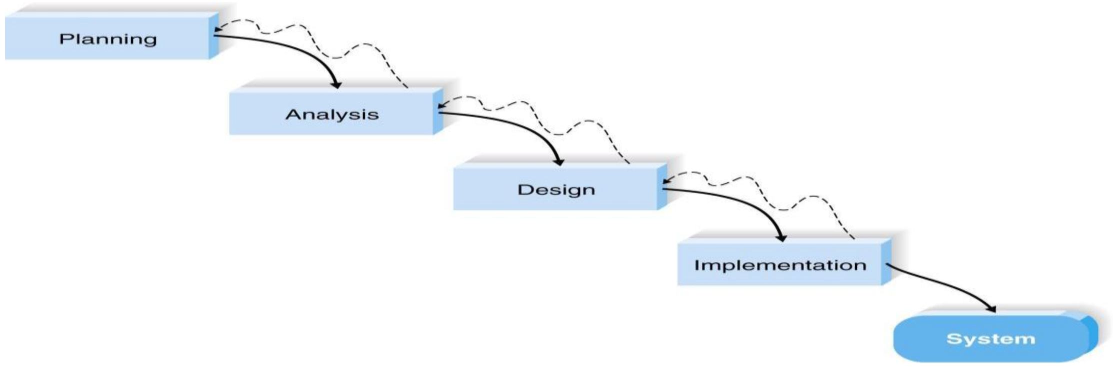
CREATING THE PROJECT PLAN

Project Methodology

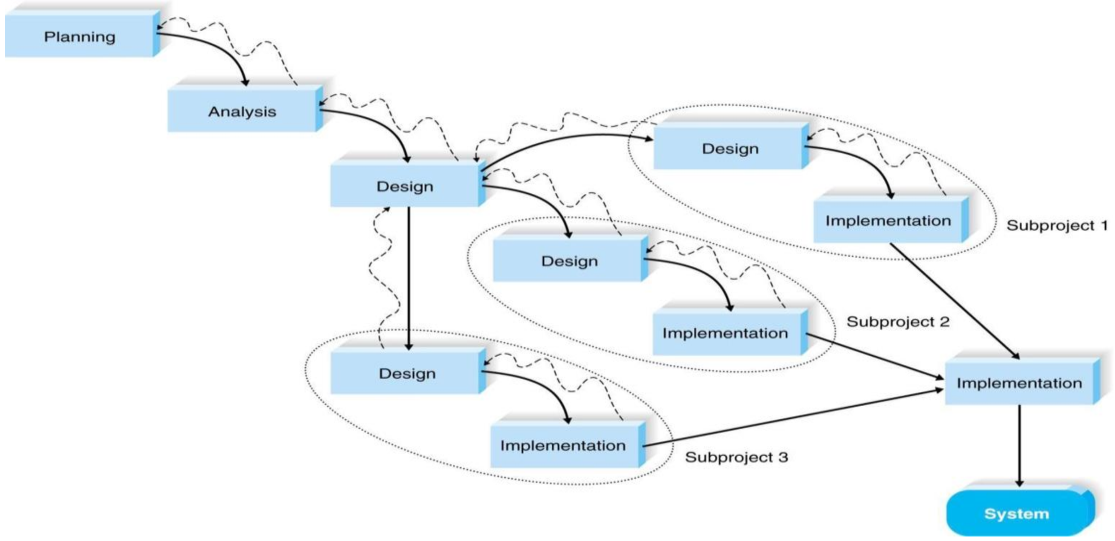
\Options

- Waterfall Development
- Parallel Development
- V-model (variation of the Waterfall Development
- Rapid Application Development (RAD)
- Iterative Development
- Agile Development

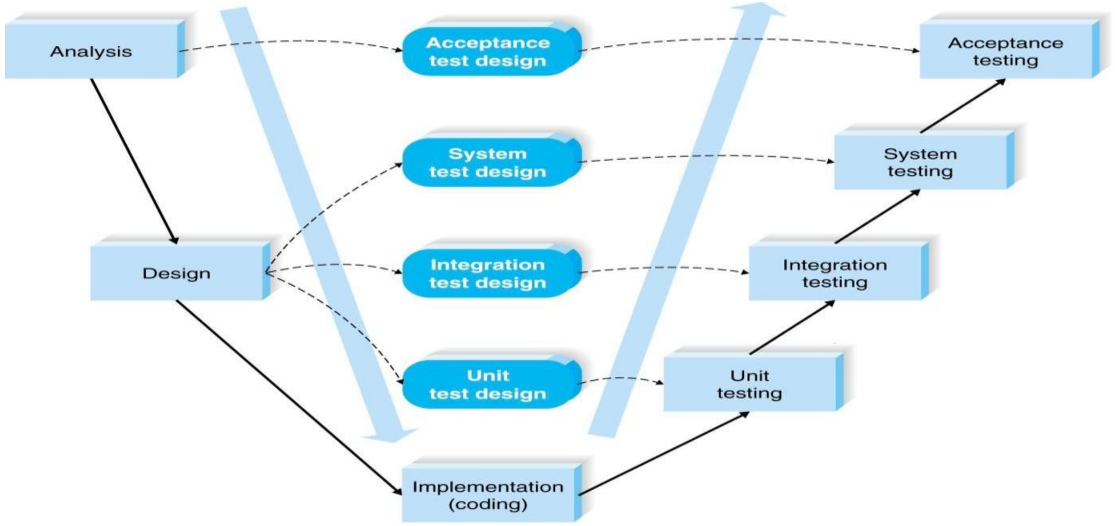
Waterfall Development



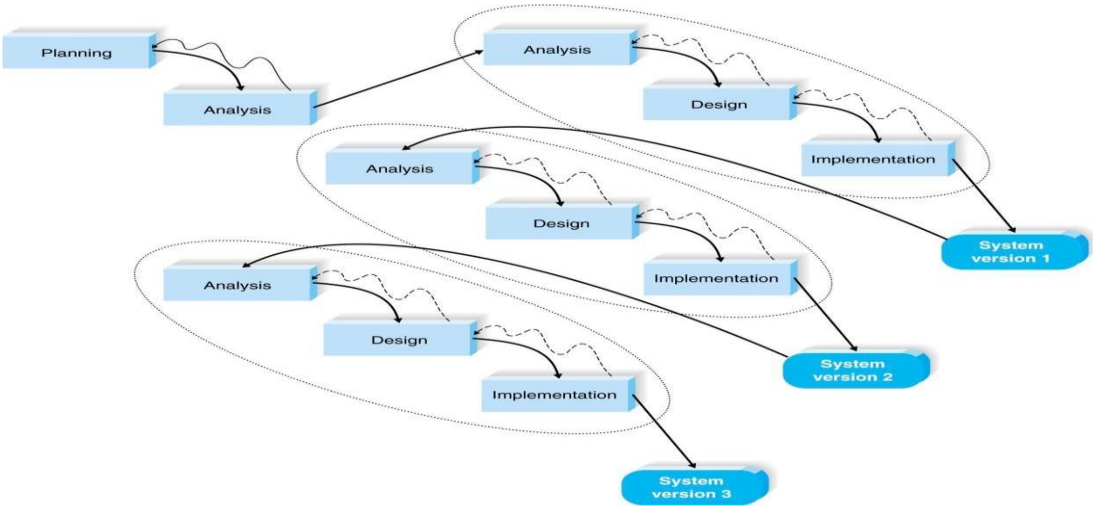
Parallel Development



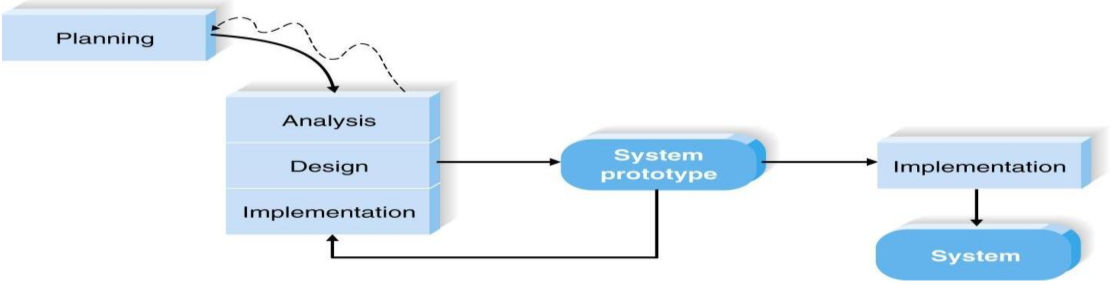
V-model



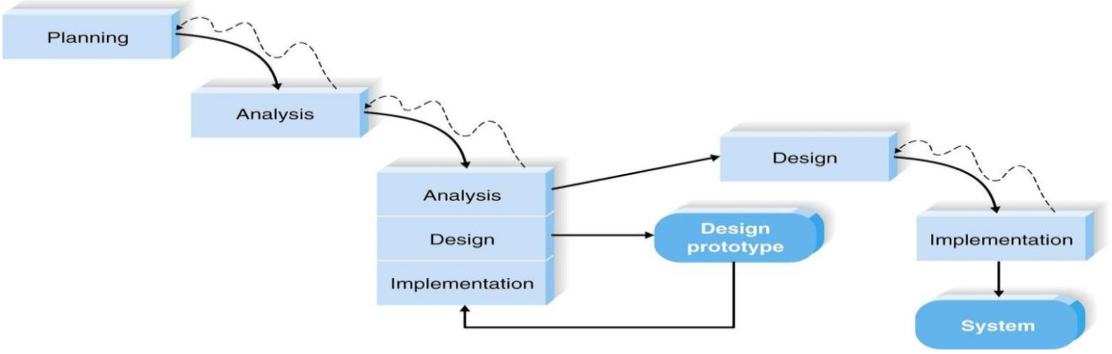
Iterative Development



Example of System Prototyping



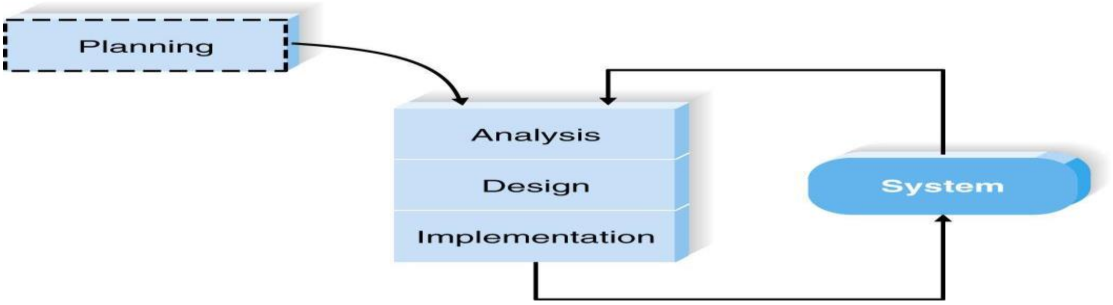
Example of Throwaway Prototyping



Agile Development

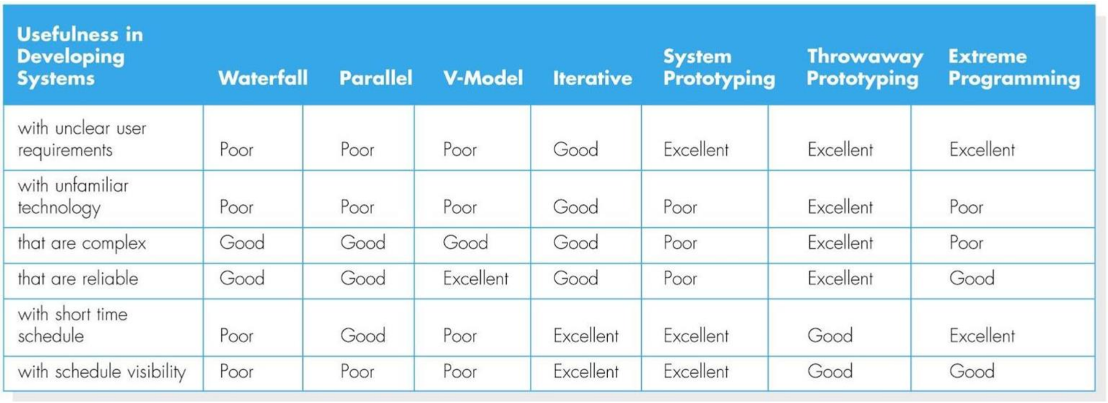
- A group of programming-centric methodologies that focus on streamlining the SDLC.
- Includes face-to-face communication
- *Extreme programming* – emphasizes customer satisfaction and teamwork.

Example of Extreme Programming



Selecting the Appropriate Development Methodology

Criteria for Selecting a Methodology



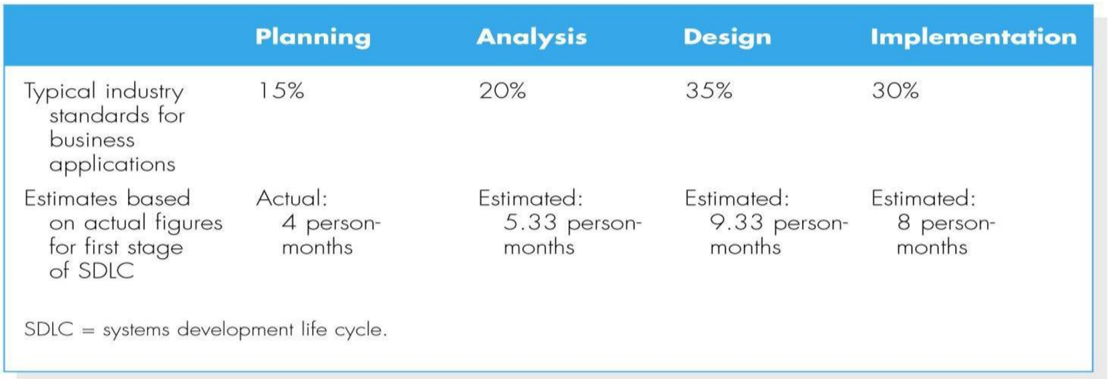
Usefulness in Developing Systems	Waterfall	Parallel	V-Model	Iterative	System Prototyping	Throwaway Prototyping	Extreme Programming
with unclear user requirements	Poor	Poor	Poor	Good	Excellent	Excellent	Excellent
with unfamiliar technology	Poor	Poor	Poor	Good	Poor	Excellent	Poor
that are complex	Good	Good	Good	Good	Poor	Excellent	Poor
that are reliable	Good	Good	Excellent	Good	Poor	Excellent	Good
with short time schedule	Poor	Good	Poor	Excellent	Excellent	Good	Excellent
with schedule visibility	Poor	Poor	Poor	Excellent	Excellent	Good	Good

Important Factors to Consider

- Clarity of User Requirements
- Familiarity with Technology
- System Complexity
- System Reliability
- Short Time Schedules
- Schedule Visibility

Estimating the Project Time Frame

Estimating Project Time Using Industry Standards



	Planning	Analysis	Design	Implementation
Typical industry standards for business applications	15%	20%	35%	30%
Estimates based on actual figures for first stage of SDLC	Actual: 4 person-months	Estimated: 5.33 person-months	Estimated: 9.33 person-months	Estimated: 8 person-months

SDLC = systems development life cycle.

Developing the Work Plan

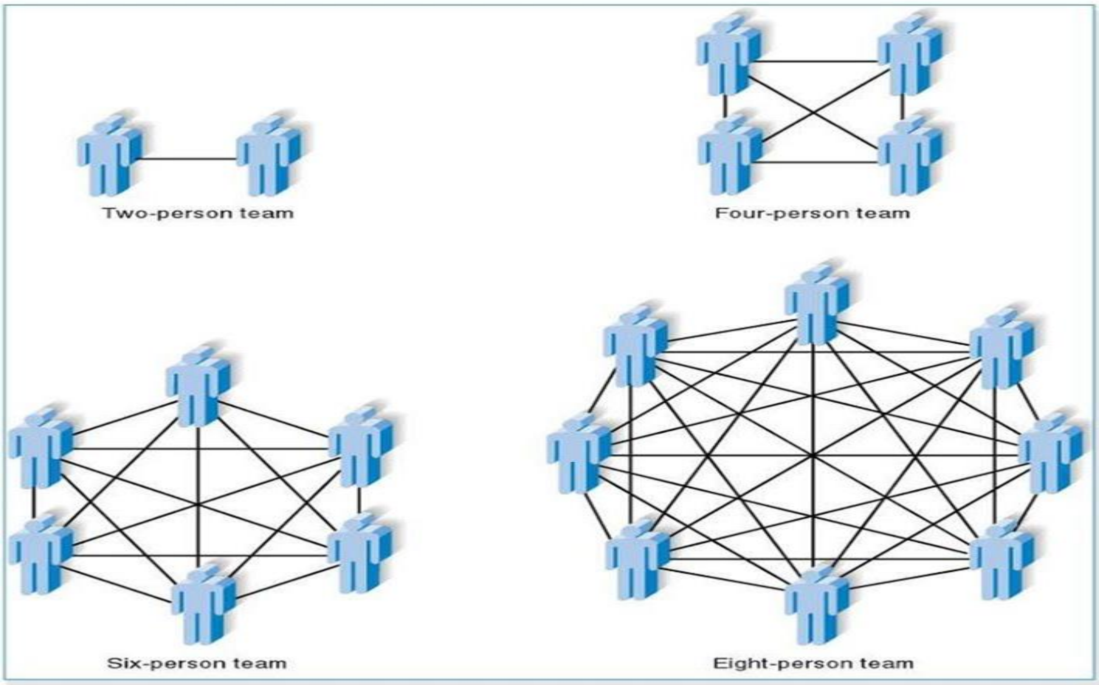
- Identify Tasks
- The Project Work Plan

STAFFING THE PROJECT

Staffing Attributes

- Staffing levels will change over a project's lifetime
- Adding staff may add more overhead than additional labor
- Using teams of 8-10 reporting in a hierarchical structure can reduce complexity

Increasing Complexity with Larger Teams



Key Definitions

- The ***staffing plan*** describes the kinds of people working on the project
- The ***project charter*** describes the project's objectives and rules
- A ***functional lead*** manages a group of analysts
- A ***technical lead*** oversees progress of programmers and technical staff members

Motivation

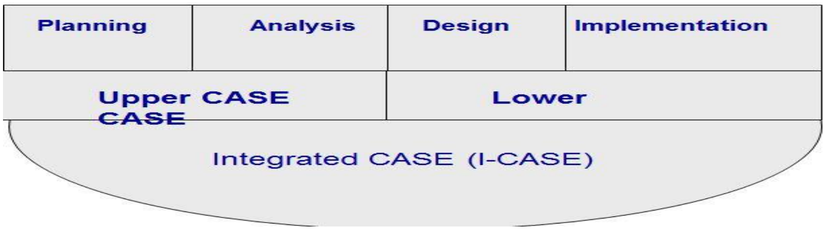
- Use monetary rewards cautiously
- Use intrinsic rewards
  - Recognition
  - Achievement
  - The work itself
  - Responsibility
  - Advancement
  - Chance to learn new skills

Handling Conflict

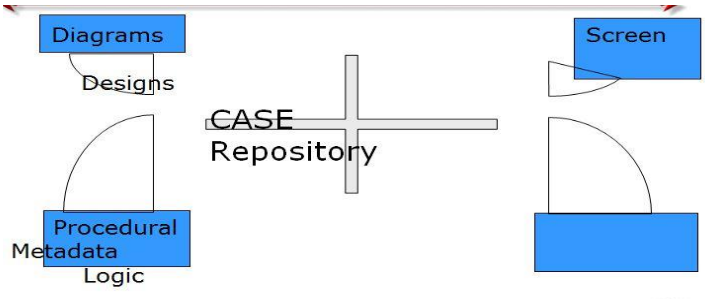
- Clearly define project plans
- Recognize project importance to organization
- Project charter listing norms and groundrules
- Develop schedule commitments ahead of time
- Forecast other priorities and their possible impact on the project

COORDINATING PROJECT ACTIVITIES

CASE Tools



CASE Components



Standards

- *Examples*
    - Formal rules for naming files
    - Forms indicating goals reached
    - Programming guidelines

Documentation

- Project binder
- Table of contents
- Continual updating

MANAGING AND CONTROLLING THE PROJECT

Timeboxing

- Fixed deadline
- Reduced functionality, if necessary
- Fewer "finishing touches"

Timeboxing Steps

1. Set delivery date
    - Deadline should not be impossible
    - Should be set by development group
2. Prioritize features by importance
3. Build the system core
4. Postpone unfinished functionality
5. Deliver the system with core functionality
6. Repeat steps 3-5 to add refinements and enhancements

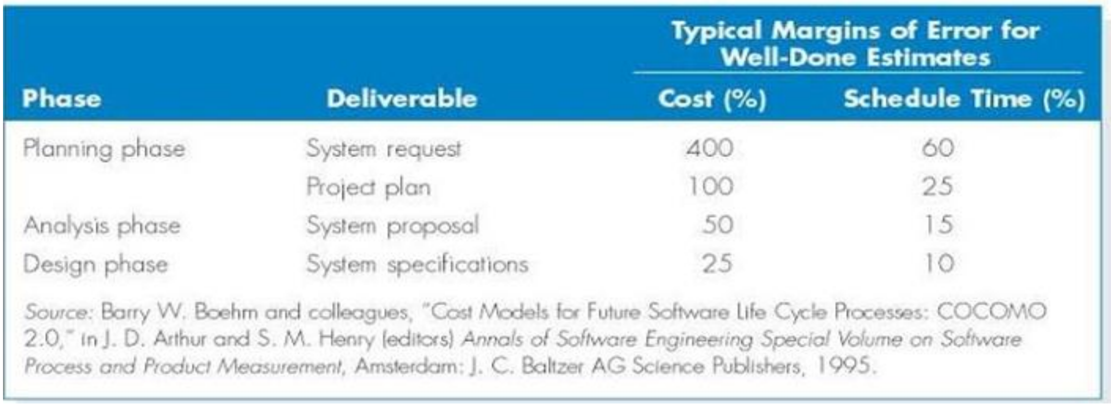
Managing Risk

- Risk assessment
- Actions to reduce risk
- Revised assessment

Classic Mistakes

- Overly optimistic schedule
- Failing to monitor schedule
- Failing to update schedule
- Adding people to a late project

Margins of Error in Cost and Time Estimates

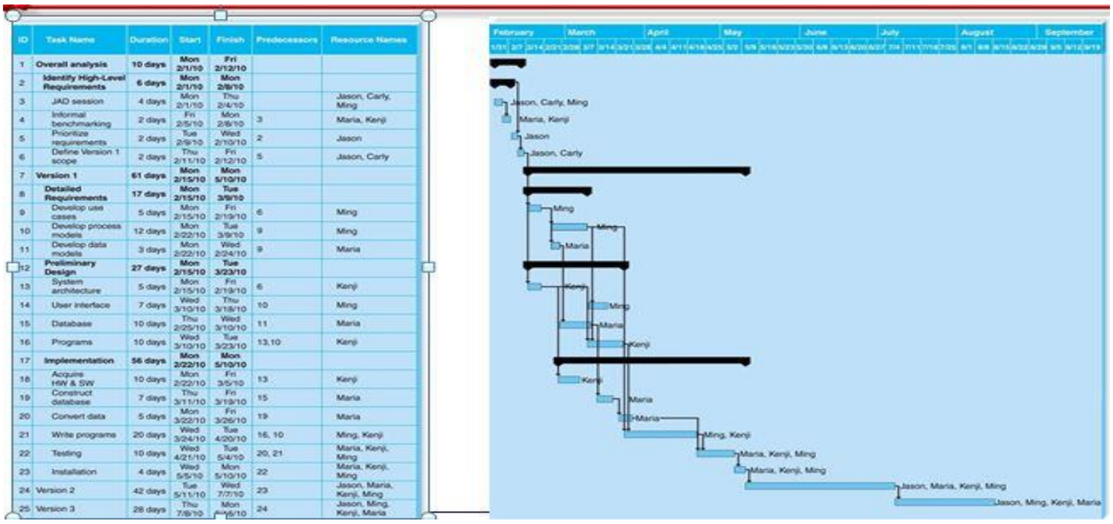


Phase	Deliverable	Typical Margins of Error for Well-Done Estimates	
		Cost (%)	Schedule Time (%)
Planning phase	System request	400	60
	Project plan	100	25
Analysis phase	System proposal	50	15
Design phase	System specifications	25	10

Source: Barry W. Boehm and colleagues, "Cost Models for Future Software Life Cycle Processes: COCOMO 2.0," in J. D. Arthur and S. M. Henry (editors) *Annals of Software Engineering Special Volume on Software Process and Product Measurement,* Amsterdam: J. C. Baltzer AG Science Publishers, 1995.

COORDINATING PROJECT ACTIVITIES

Example of a Gantt Chart



ID	Task Name	Duration	Start	Finish	Predecessors	Resource Names
1	**Overall analysis**	**10 days**	**Mon 2/1/10**	**Fri 2/12/10**		
2	**Identify High-Level Requirements**	**6 days**	**Mon 2/1/10**	**Mon 2/8/10**		
3	JAD session	4 days	Mon 2/1/10	Thu 2/4/10		Jason, Carly, Ming
4	Informal benchmarking	2 days	Fri 2/5/10	Mon 2/8/10	3	Maria, Kenji
5	Prioritize requirements	2 days	Tue 2/9/10	Wed 2/10/10	2	Jason
6	Define Version 1 scope	2 days	Thu 2/11/10	Fri 2/12/10	5	Jason, Carly
7	**Version 1**	**61 days**	**Mon 2/15/10**	**Mon 5/10/10**		
8	**Detailed Requirements**	**17 days**	**Mon 2/15/10**	**Tue 3/9/10**		
9	Develop use cases	5 days	Mon 2/15/10	Fri 2/19/10	6	Ming
10	Develop process models	12 days	Mon 2/22/10	Tue 3/9/10	9	Ming
11	Develop data models	3 days	Mon 2/22/10	Wed 2/24/10	9	Maria
12	**Preliminary Design**	**27 days**	**Mon 2/15/10**	**Tue 3/23/10**		
13	System architecture	5 days	Mon 2/15/10	Fri 2/19/10	6	Kenji
14	User interface	7 days	Wed 3/10/10	Thu 3/18/10	10	Ming
15	Database	10 days	Thu 2/25/10	Wed 3/10/10	11	Maria
16	Programs	10 days	Wed 3/10/10	Tue 3/23/10	13,10	Kenji
17	**Implementation**	**56 days**	**Mon 2/22/10**	**Mon 5/10/10**		
18	Acquire HW & SW	10 days	Mon 2/22/10	Fri 3/5/10	13	Kenji
19	Construct database	7 days	Thu 3/11/10	Fri 3/19/10	15	Maria
20	Convert data	5 days	Mon 3/22/10	Fri 3/26/10	19	Maria
21	Write programs	20 days	Wed 3/24/10	Tue 4/20/10	16, 10	Ming, Kenji
22	Testing	10 days	Wed 4/21/10	Tue 5/4/10	20, 21	Maria, Kenji, Ming
23	Installation	4 days	Wed 5/5/10	Mon 5/10/10	22	Maria, Kenji, Ming
24	Version 2	42 days	Tue 5/11/10	Wed 7/7/10	23	Jason, Maria, Kenji, Ming
25	Version 3	28 days	Thu 7/8/10	Mon 8/16/10	24	Jason, Ming, Kenji, Maria

Summary

- **Project initiation** involves creating and assessing goals and expectations for a new system
- Identifying the **business value** of the new project is a key to success
- **Feasibility study** is concerned with insuring that technical, economic, and organizational benefits outweigh costs and risks
- **Project selection** involves viewing the project within the context of the entire **project portfolio**, and selecting those projects that contribute to **balance** in the portfolio