Chapter 2. Project Selection and Management

Chapter 2 Outline

- Project selection.
- Creating the project plan.
- Managing and controlling the project.

INTRODUCTION

- CIOs (chief information officers) are challenged to select projects that will provide highest return on the IT investments.
- Project portfolio management has become a critical success factor for IT departments.
- A selected system development project must undergo a thorough process of project management.
- A critical success factor for project management is to start with a realistic assessment of the work and then manage the project according to the plan.

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.

(cont’d)

- An approval committee must be selective about where to allocate resources as most organizations have limited funds.
- If there are several potentially high-payoff projects, and they all have the same risk, then maybe only one of the projects will be selected.

CREATING THE PROJECT PLAN

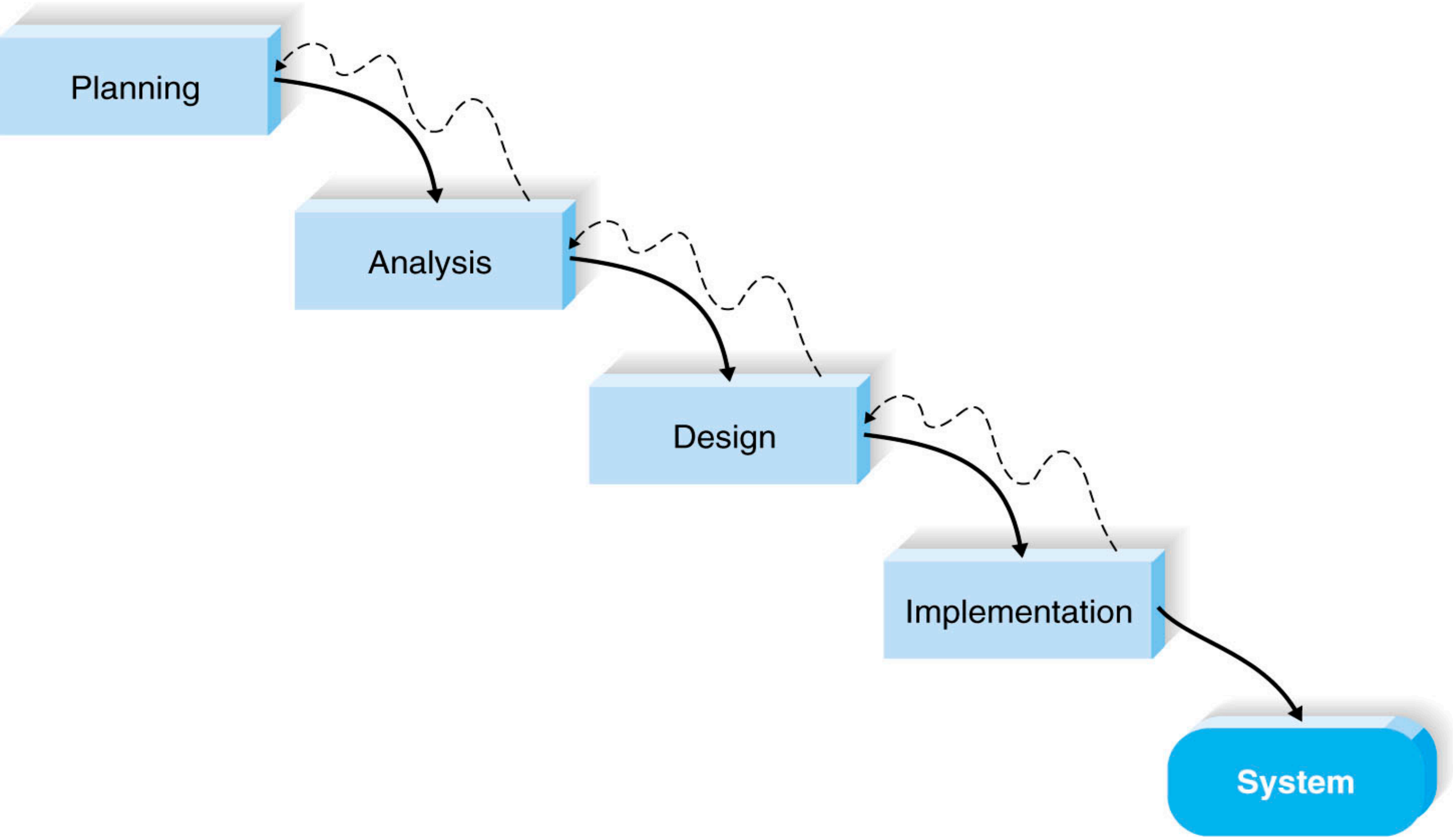
- Project management phases consist of
  - initiation
  - planning
  - execution
  - control, and
  - enclosure.

Project Methodology Options

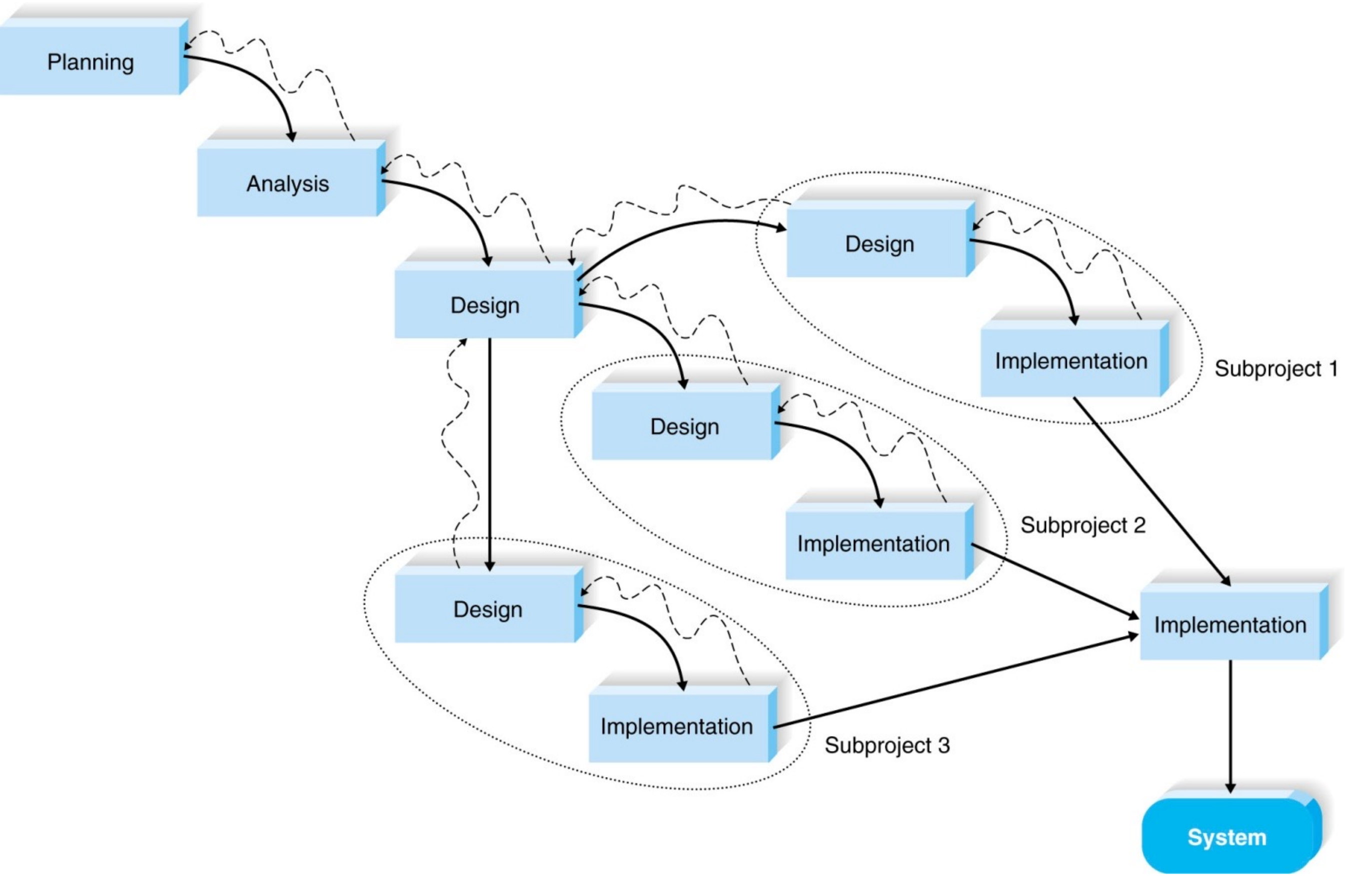
- A methodology is a formalized approach to implementing the SDLC.

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

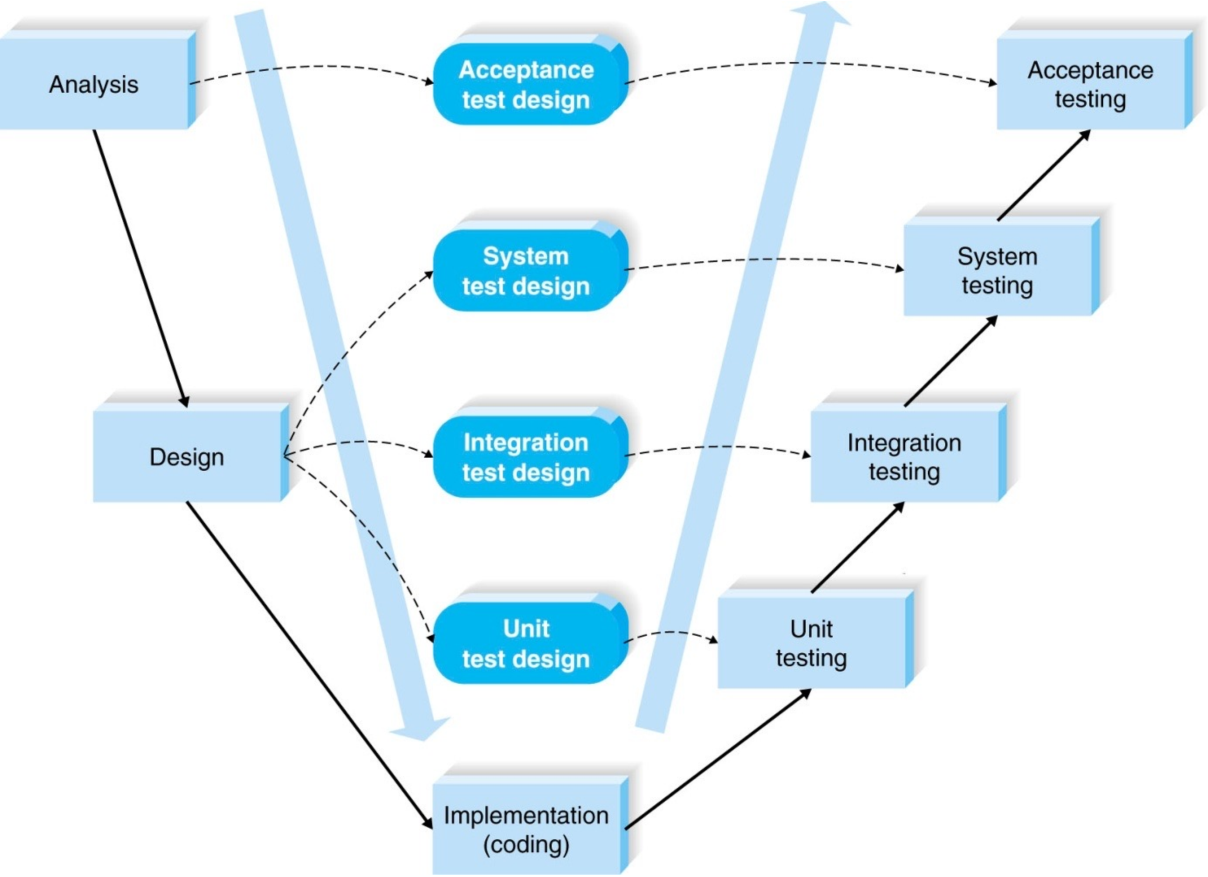
Waterfall Development



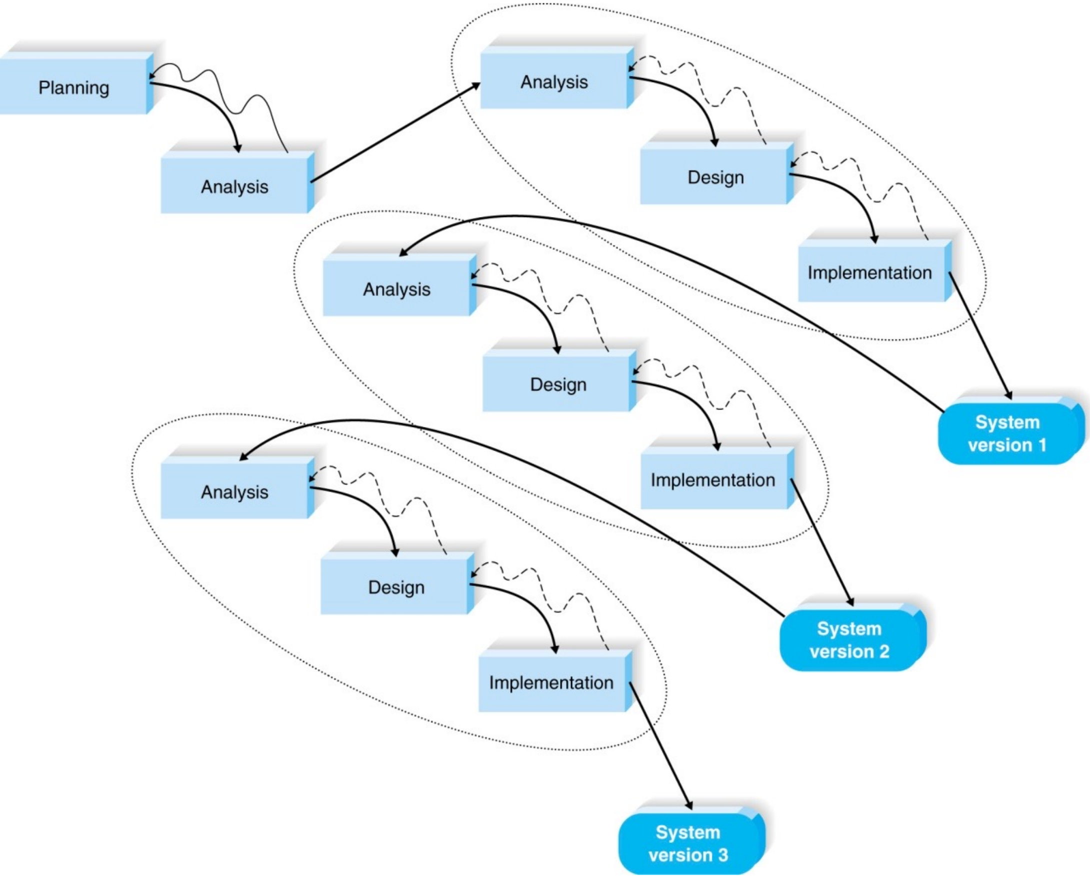
Parallel Development



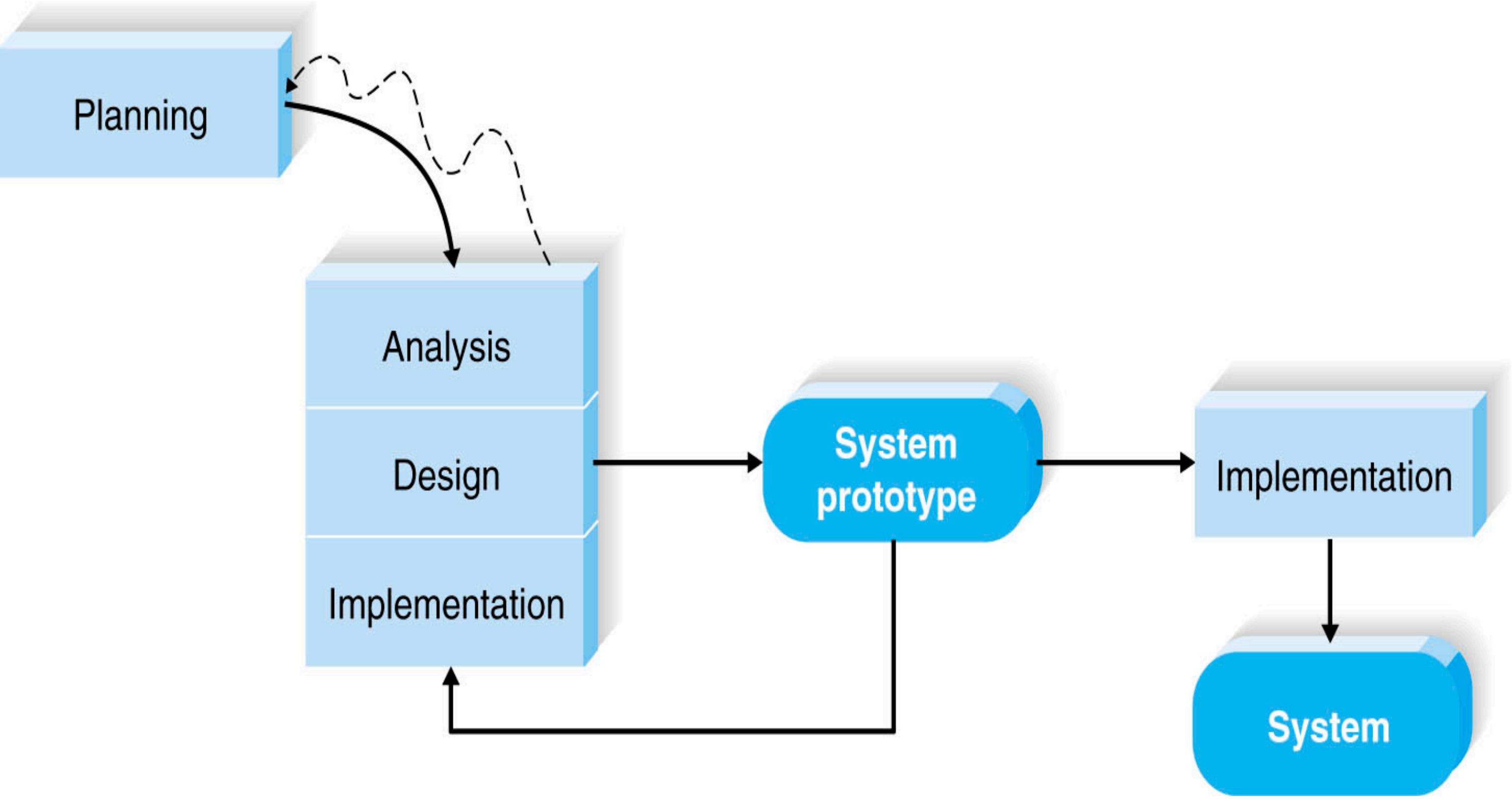
V-model


Analysis
Acceptance test design
Acceptance testing
System test design
System testing
Design
Integration test design
Integration testing
Unit test design
Unit testing
Implementation (coding)

Rapid Application Development: Iterative Development

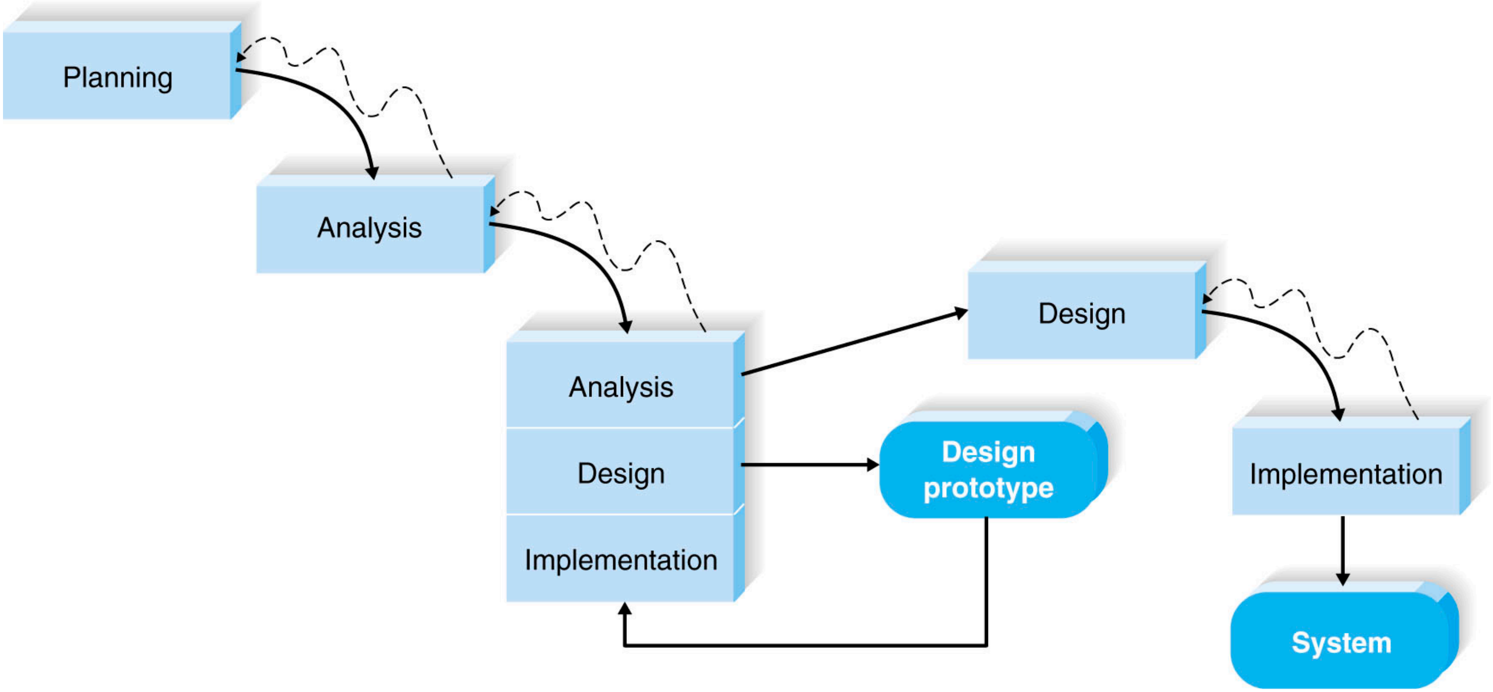


Rapid Application Development: System Prototyping



(cont’d)

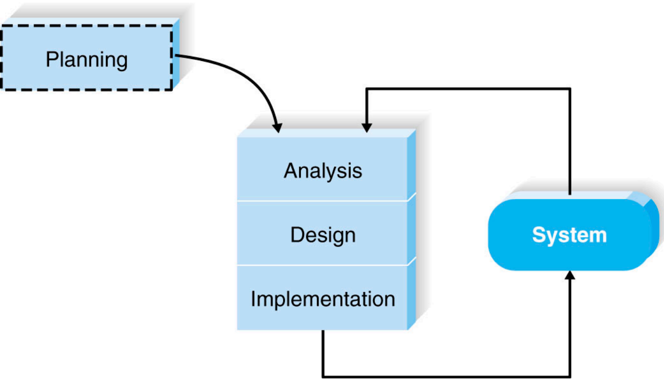
- Throwaway prototyping



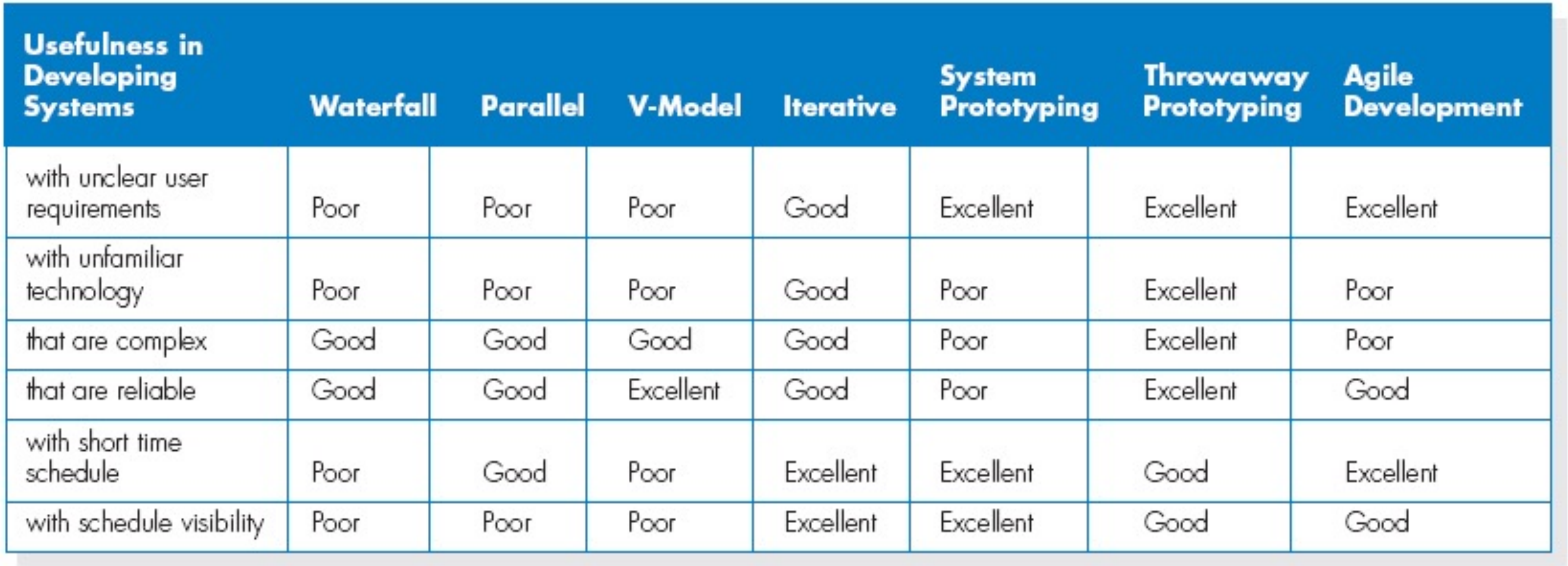
Agile Development

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

Extreme Programming



Selecting the Appropriate Development Methodology



Usefulness in Developing Systems	Waterfall	Parallel	V-Model	Iterative	System Prototyping	Throwaway Prototyping	Agile Development
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

(cont'd)

- Important factors to consider in selecting the development methodology

  - Clarity of User Requirements

  - Familiarity with Technology

  - System Complexity

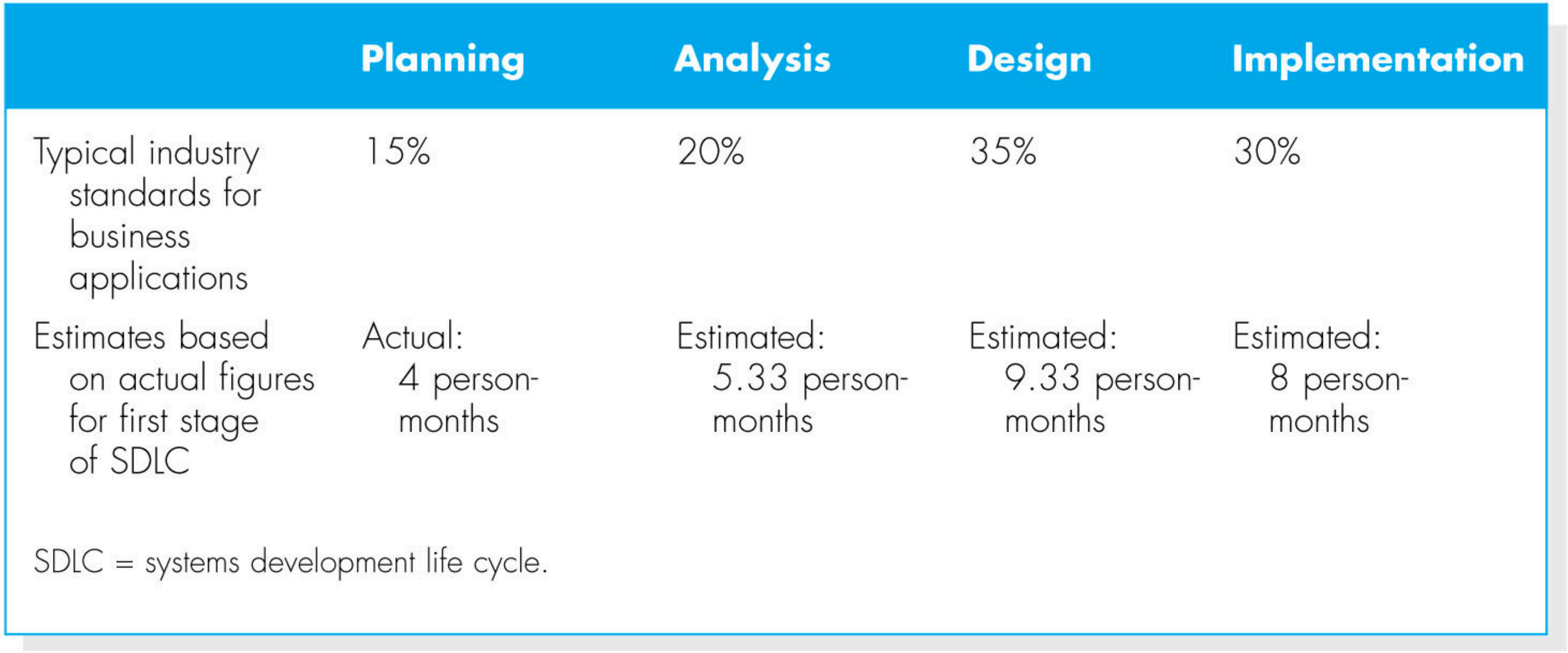
  - System Reliability

  - Short Time Schedules

  - Schedule Visibility

Estimating the Project Time Frame

1. Estimating Project Time Using Industry Standards



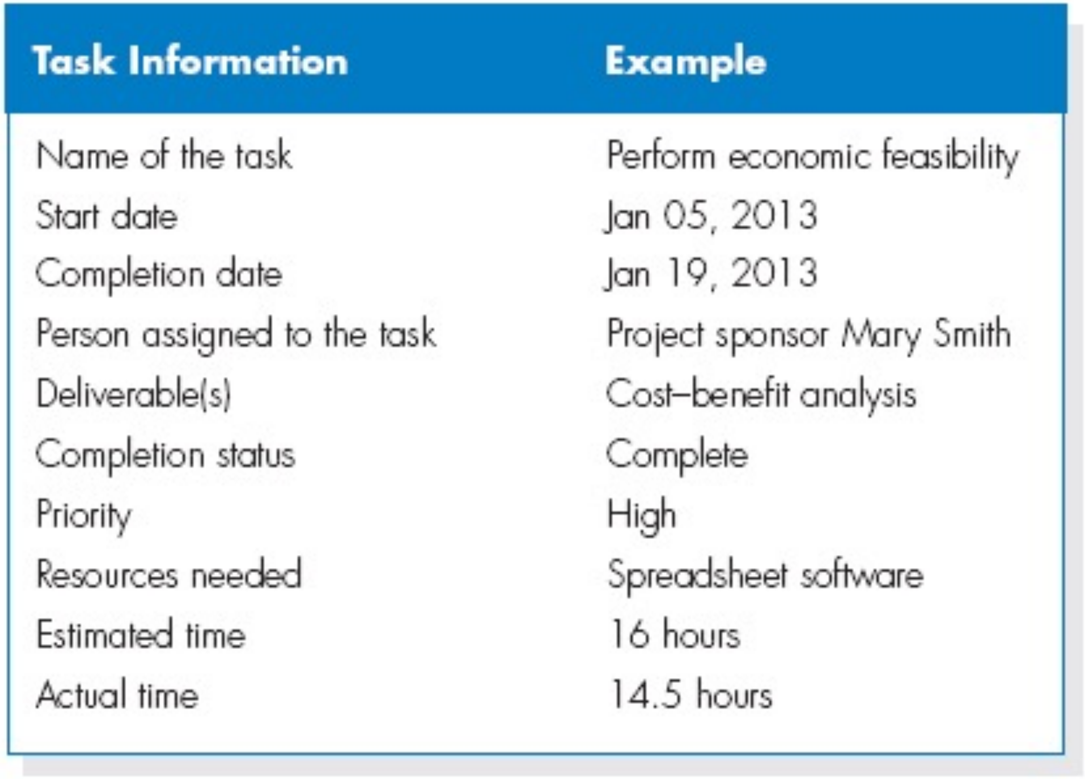
	Planning	Analysis	Design	Implementation
Typical industry standards for business applications	15%	20%	35%	30%
Estimates based on actual figures for first stage of SDLC	Actual: 4 person-months	Estimated: 5.33 person-months	Estimated: 9.33 person-months	Estimated: 8 person-months

SDLC = systems development life cycle.

2. Function point approach (Appendix 2A)

Developing the Work Plan

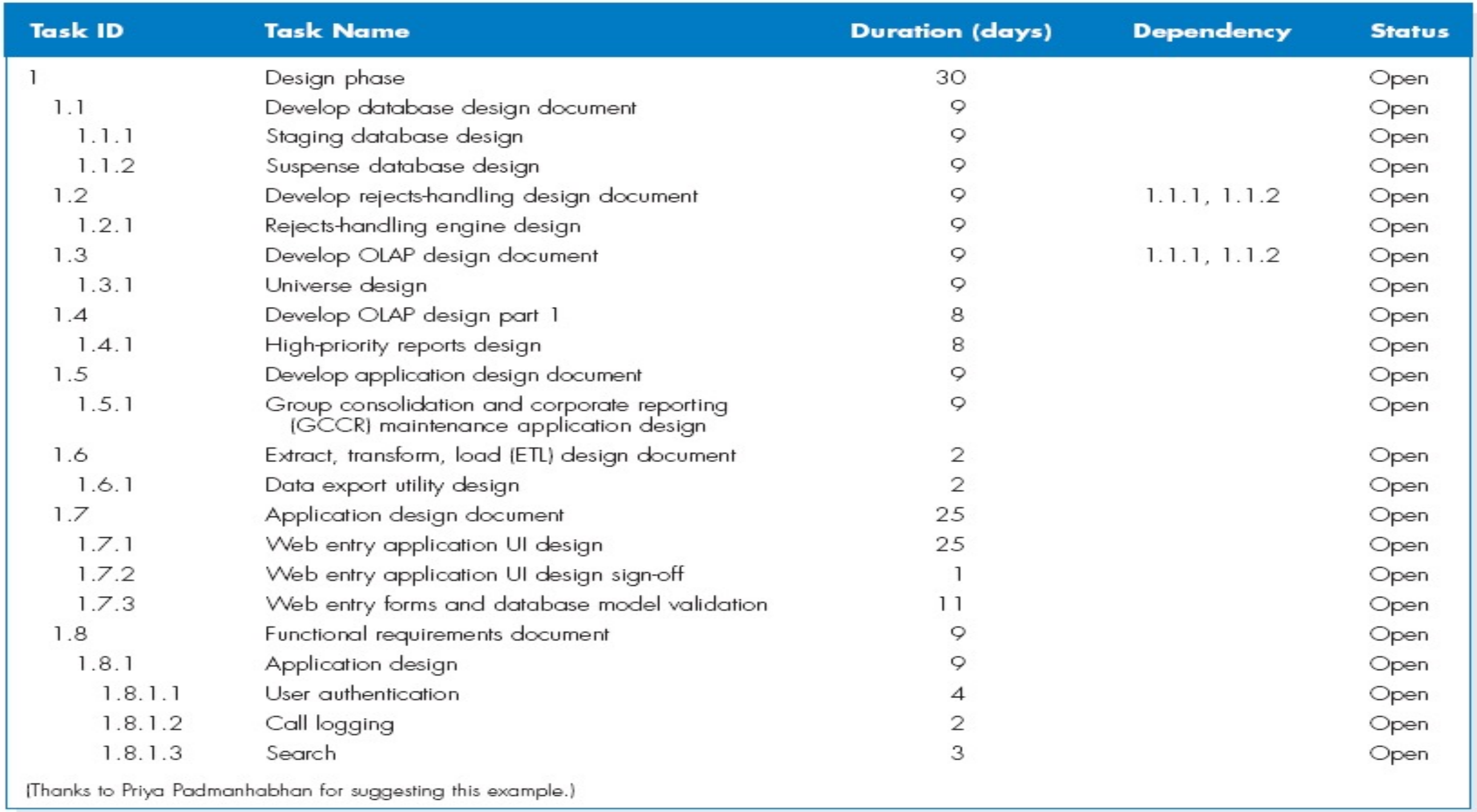
- Identify Tasks



Task Information	Example
Name of the task	Perform economic feasibility
Start date	Jan 05, 2013
Completion date	Jan 19, 2013
Person assigned to the task	Project sponsor Mary Smith
Deliverable(s)	Cost–benefit analysis
Completion status	Complete
Priority	High
Resources needed	Spreadsheet software
Estimated time	16 hours
Actual time	14.5 hours

(cont'd)

- Work Breakdown Structure

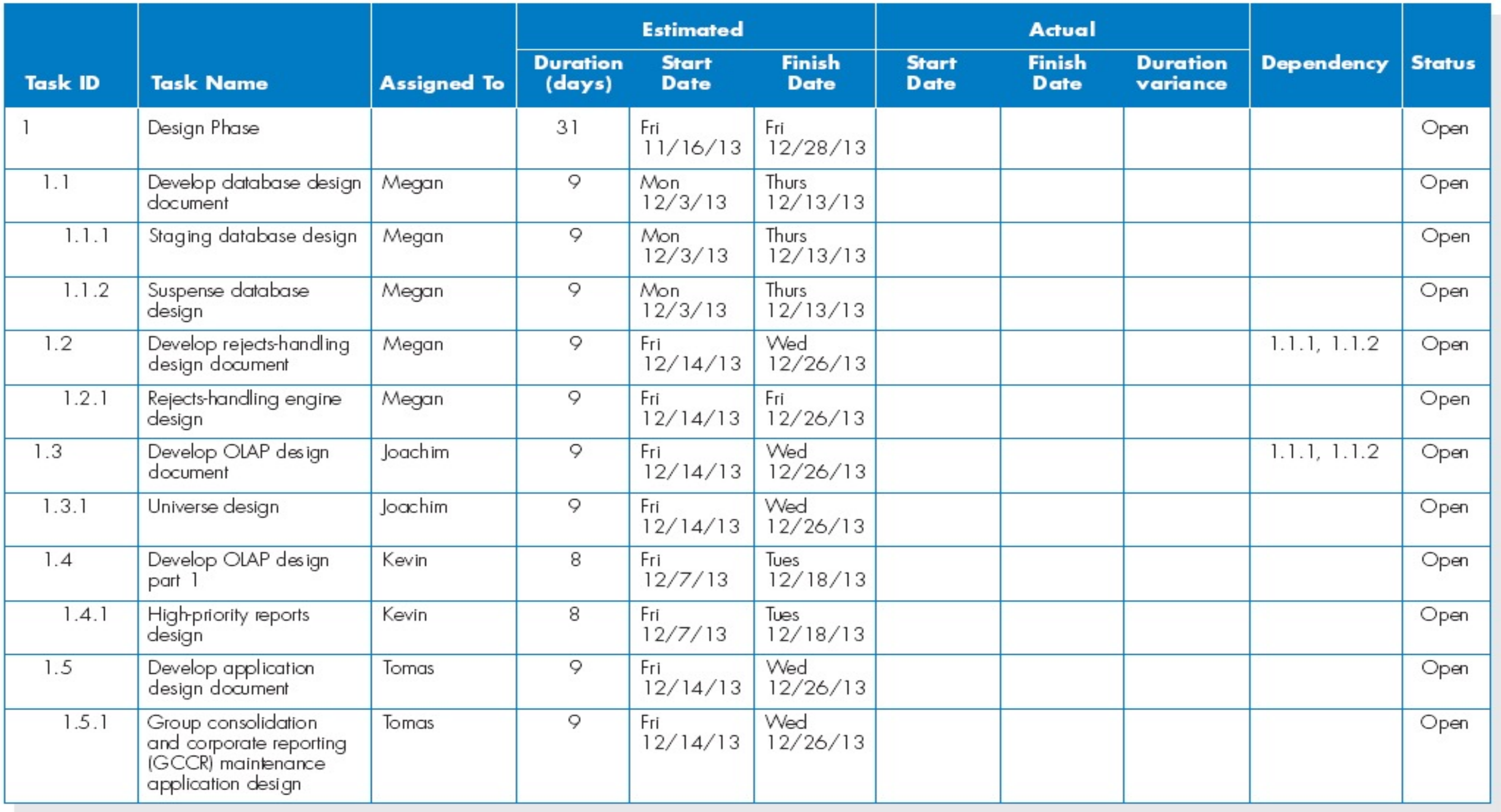


Task ID	Task Name	Duration (days)	Dependency	Status
1	Design phase	30		Open
1.1	Develop database design document	9		Open
1.1.1	Staging database design	9		Open
1.1.2	Suspense database design	9		Open
1.2	Develop rejects-handling design document	9	1.1.1, 1.1.2	Open
1.2.1	Rejects-handling engine design	9		Open
1.3	Develop OLAP design document	9	1.1.1, 1.1.2	Open
1.3.1	Universe design	9		Open
1.4	Develop OLAP design part 1	8		Open
1.4.1	High-priority reports design	8		Open
1.5	Develop application design document	9		Open
1.5.1	Group consolidation and corporate reporting (GCCR) maintenance application design	9		Open
1.6	Extract, transform, load (ETL) design document	2		Open
1.6.1	Data export utility design	2		Open
1.7	Application design document	25		Open
1.7.1	Web entry application UI design	25		Open
1.7.2	Web entry application UI design sign-off	1		Open
1.7.3	Web entry forms and database model validation	11		Open
1.8	Functional requirements document	9		Open
1.8.1	Application design	9		Open
1.8.1.1	User authentication	4		Open
1.8.1.2	Call logging	2		Open
1.8.1.3	Search	3		Open

(Thanks to Priya Padmanhabhan for suggesting this example.)

(cont'd)

- The project work plan

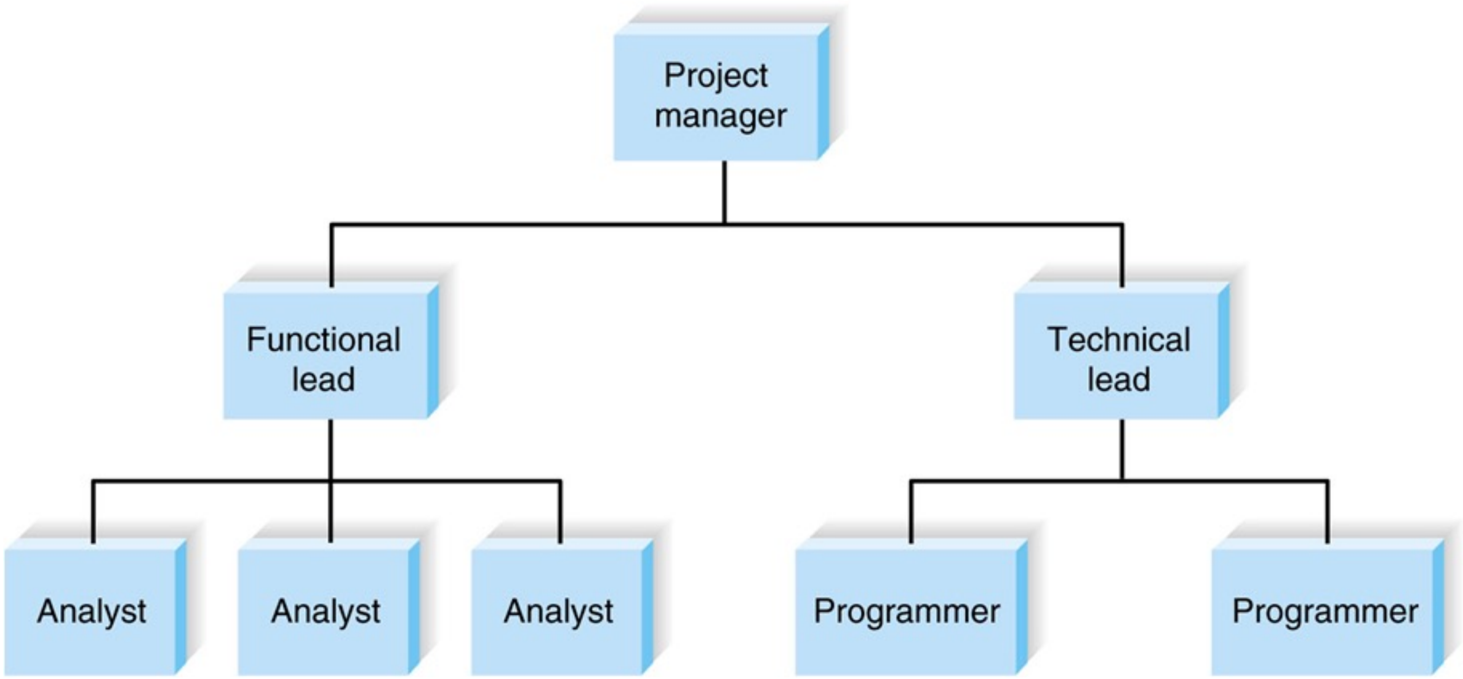


Task ID	Task Name	Assigned To	Estimated			Actual			Dependency	Status
			Duration (days)	Start Date	Finish Date	Start Date	Finish Date	Duration variance		
1	Design Phase		31	Fri 11/16/13	Fri 12/28/13					Open
1.1	Develop database design document	Megan	9	Mon 12/3/13	Thurs 12/13/13					Open
1.1.1	Staging database design	Megan	9	Mon 12/3/13	Thurs 12/13/13					Open
1.1.2	Suspense database design	Megan	9	Mon 12/3/13	Thurs 12/13/13					Open
1.2	Develop rejects-handling design document	Megan	9	Fri 12/14/13	Wed 12/26/13				1.1.1, 1.1.2	Open
1.2.1	Rejects-handling engine design	Megan	9	Fri 12/14/13	Fri 12/26/13					Open
1.3	Develop OLAP design document	Joachim	9	Fri 12/14/13	Wed 12/26/13				1.1.1, 1.1.2	Open
1.3.1	Universe design	Joachim	9	Fri 12/14/13	Wed 12/26/13					Open
1.4	Develop OLAP design part 1	Kevin	8	Fri 12/7/13	Tues 12/18/13					Open
1.4.1	High-priority reports design	Kevin	8	Fri 12/7/13	Tues 12/18/13					Open
1.5	Develop application design document	Tomas	9	Fri 12/14/13	Wed 12/26/13					Open
1.5.1	Group consolidation and corporate reporting (GCCR) maintenance application design	Tomas	9	Fri 12/14/13	Wed 12/26/13					Open

STAFFING THE PROJECT

- Staffing Plan
  - 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

Reporting structure



(cont'd)

- 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 plans for the project.
- Recognize project importance to organization.
- Project charter listing norms and ground rules.
- Develop schedule commitments ahead of time.
- Forecast other priorities and their possible impact on the project.

Coordinating Project Activities

- CASE (computer-aided software engineering) tools – A category of software that automate all or part of the development process.
  - Upper CASE
  - Lower CASE
  - Integrated CASE

(cont'd)

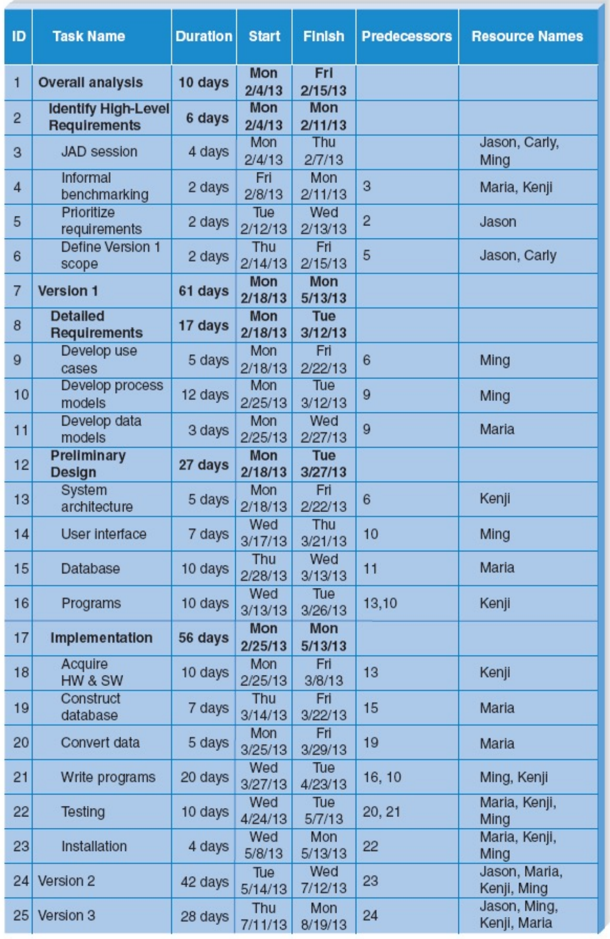
- Standards
  - Formal rules for naming files
  - Forms indicating goals reached
  - Programming guidelines
- Documentation
  - Project binder
  - Table of contents
  - Continual updating

MANAGING AND CONTROLLING THE PROJECT

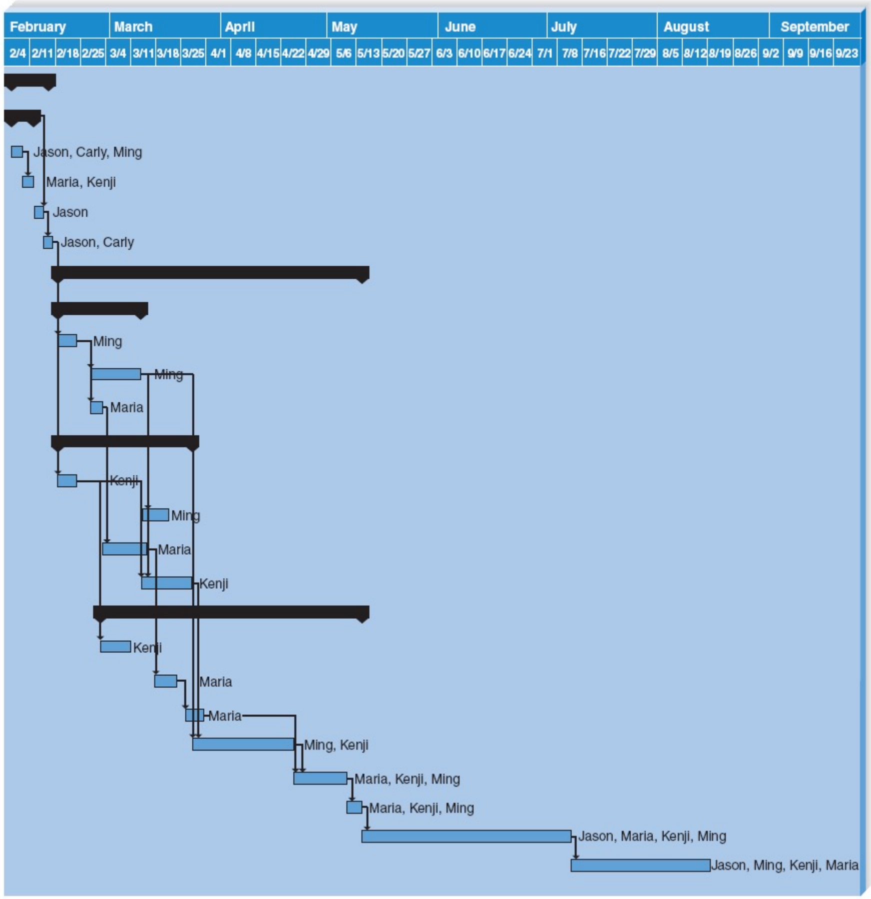
- The science (or art) of project management is in making ***trade-offs*** among three important concepts:
  - the size of the system,
  - the time to complete the project, and
  - the cost of the project.

(cont'd)

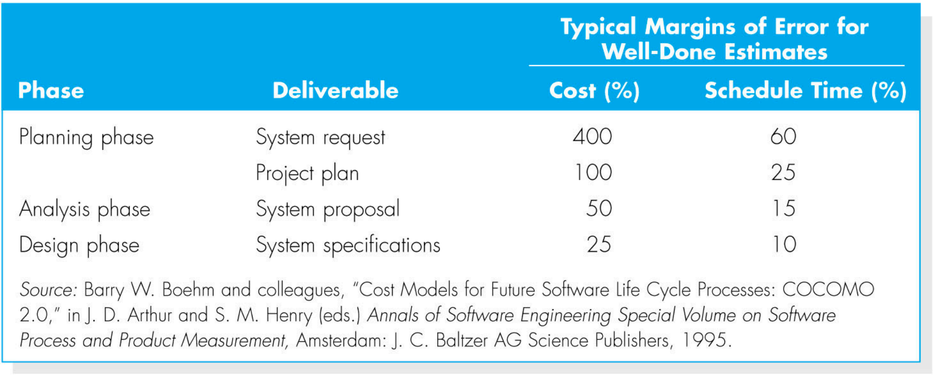
Tools for project management – Example of Gantt Char



ID	Task Name	Duration	Start	Finish	Predecessors	Resource Names
1	**Overall analysis**	**10 days**	**Mon 2/4/13**	**Fri 2/15/13**		
2	**Identify High-Level Requirements**	**6 days**	**Mon 2/4/13**	**Mon 2/11/13**		
3	JAD session	4 days	Mon 2/4/13	Thu 2/7/13		Jason, Carly, Ming
4	Informal benchmarking	2 days	Fri 2/8/13	Mon 2/11/13	3	Maria, Kenji
5	Prioritize requirements	2 days	Tue 2/12/13	Wed 2/13/13	2	Jason
6	Define Version 1 scope	2 days	Thu 2/14/13	Fri 2/15/13	5	Jason, Carly
7	**Version 1**	**61 days**	**Mon 2/18/13**	**Mon 5/13/13**		
8	**Detailed Requirements**	**17 days**	**Mon 2/18/13**	**Tue 3/12/13**		
9	Develop use cases	5 days	Mon 2/18/13	Fri 2/22/13	6	Ming
10	Develop process models	12 days	Mon 2/25/13	Tue 3/12/13	9	Ming
11	Develop data models	3 days	Mon 2/25/13	Wed 2/27/13	9	Maria
12	**Preliminary Design**	**27 days**	**Mon 2/18/13**	**Tue 3/27/13**		
13	System architecture	5 days	Mon 2/18/13	Fri 2/22/13	6	Kenji
14	User interface	7 days	Wed 3/17/13	Thu 3/21/13	10	Ming
15	Database	10 days	Thu 2/28/13	Wed 3/13/13	11	Maria
16	Programs	10 days	Wed 3/13/13	Tue 3/26/13	13,10	Kenji
17	**Implementation**	**56 days**	**Mon 2/25/13**	**Mon 5/13/13**		
18	Acquire HW & SW	10 days	Mon 2/25/13	Fri 3/8/13	13	Kenji
19	Construct database	7 days	Thu 3/14/13	Fri 3/22/13	15	Maria
20	Convert data	5 days	Mon 3/25/13	Fri 3/29/13	19	Maria
21	Write programs	20 days	Wed 3/27/13	Tue 4/23/13	16, 10	Ming, Kenji
22	Testing	10 days	Wed 4/24/13	Tue 5/7/13	20, 21	Maria, Kenji, Ming
23	Installation	4 days	Wed 5/8/13	Mon 5/13/13	22	Maria, Kenji, Ming
24	Version 2	42 days	Tue 5/14/13	Wed 7/12/13	23	Jason, Maria, Kenji, Ming
25	Version 3	28 days	Thu 7/11/13	Mon 8/19/13	24	Jason, Ming, Kenji, Maria



Refining 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 (eds.) *Annals of Software Engineering Special Volume on Software Process and Product Measurement,* Amsterdam: J. C. Baltzer AG Science Publishers, 1995.

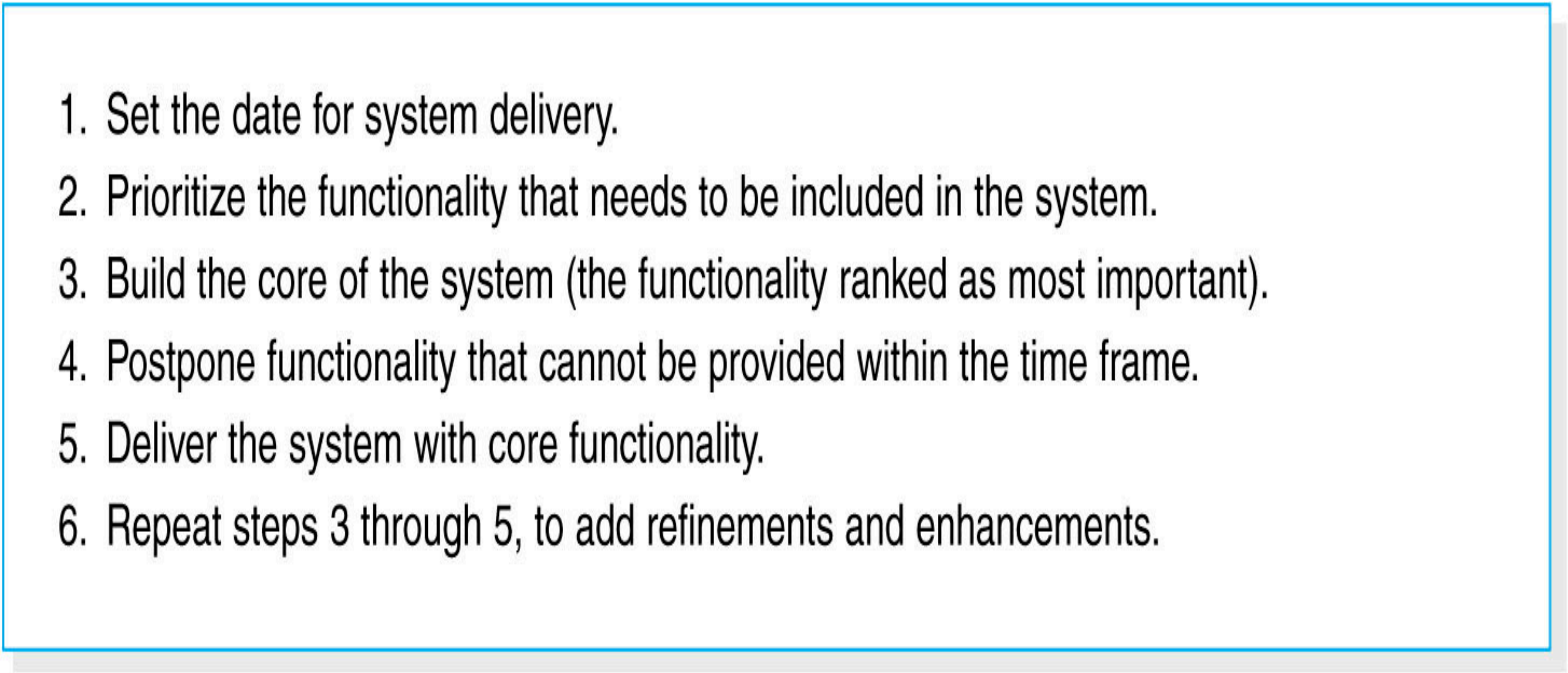
Managing Scope

- ***Scope creep*** – the most common reason for schedule and cost overruns occurs after the project is underway.
- The project manager should allow only absolutely necessary requirements to be added after the project begins.

Timeboxing

- Set a fixed deadline for a project
- Reduce functionality, if necessary
- Don’t get hung up on the final “finishing touches”

Timeboxing steps



1. Set the date for system delivery.
2. Prioritize the functionality that needs to be included in the system.
3. Build the core of the system (the functionality ranked as most important).
4. Postpone functionality that cannot be provided within the time frame.
5. Deliver the system with core functionality.
6. Repeat steps 3 through 5, to add refinements and enhancements.

Managing Risk

- Risk assessment
- Actions to reduce risk
- Revised assessment

SUMMARY

- The **project selection** process takes into account all of the projects in the organization, using project portfolio management.
- **The project plan** defines the tasks, task time estimates, and other information.
- A project requires **staffing** and **coordinating project activities**.
- **Managing and controlling the project** include timeboxing and risk assessment.