

# INTRODUCTION

ITU 07302

SYSTEM ANALYSIS AND DESIGN

WITH OBJECT ORIENTED PROGRAMMING


Based on Course Textbook:

Systems Analysis and Design With UML 2.0 An Object-Oriented Approach, Second Edition

Alan Dennis, Barbara Wixom, and David Tegarden

© 2005 John Wiley & Sons, Inc.

# Key Ideas



- ❑ Many failed systems were abandoned because analysts tried to build wonderful systems without understanding the organization.
- ❑ The primarily goal is to create value for the organization.
- ❑ Quality is satisfaction of requirements, not 'goodness'

# Key Ideas



- ☑ The **systems analyst** is a key person analyzing the business, identifying opportunities for improvement, and designing information systems to implement these ideas.
- ☑ It is important to understand and develop through practice the skills needed to successfully design and implement new information systems.

# **THE SYSTEMS DEVELOPMENT LIFE CYCLE (SDLC)**



# Major Attributes of the Lifecycle



## ☑ The project

- Moves systematically through phases where each phase has a standard set of outputs
- Produces project deliverables
- Uses deliverables in implementation
- Results in actual information system
- Uses *gradual refinement*

# 4 Main Project Phases



- ☑ **Planning**

- Why build the system?

- ☑ **Analysis**

- What, when, where will the system be?


- ☑ **Design**

- How will the system work?

- ☑ **Implementation**

- System construction & delivery

# Planning



- ❑ Identifying business value (is it worth doing?)
- ❑ Analyze feasibility (is it possible?)
- ❑ Develop work plan (when?)
- ❑ Staff the project (who?)
- ❑ Control and direct project



# Analysis



- ❑ Analysis (what do we want?  
Who will use the system?)
- ❑ Information gathering
- ❑ Process modelling (what  
happens?)
- ❑ Data modelling (... and to  
what?)



# Design



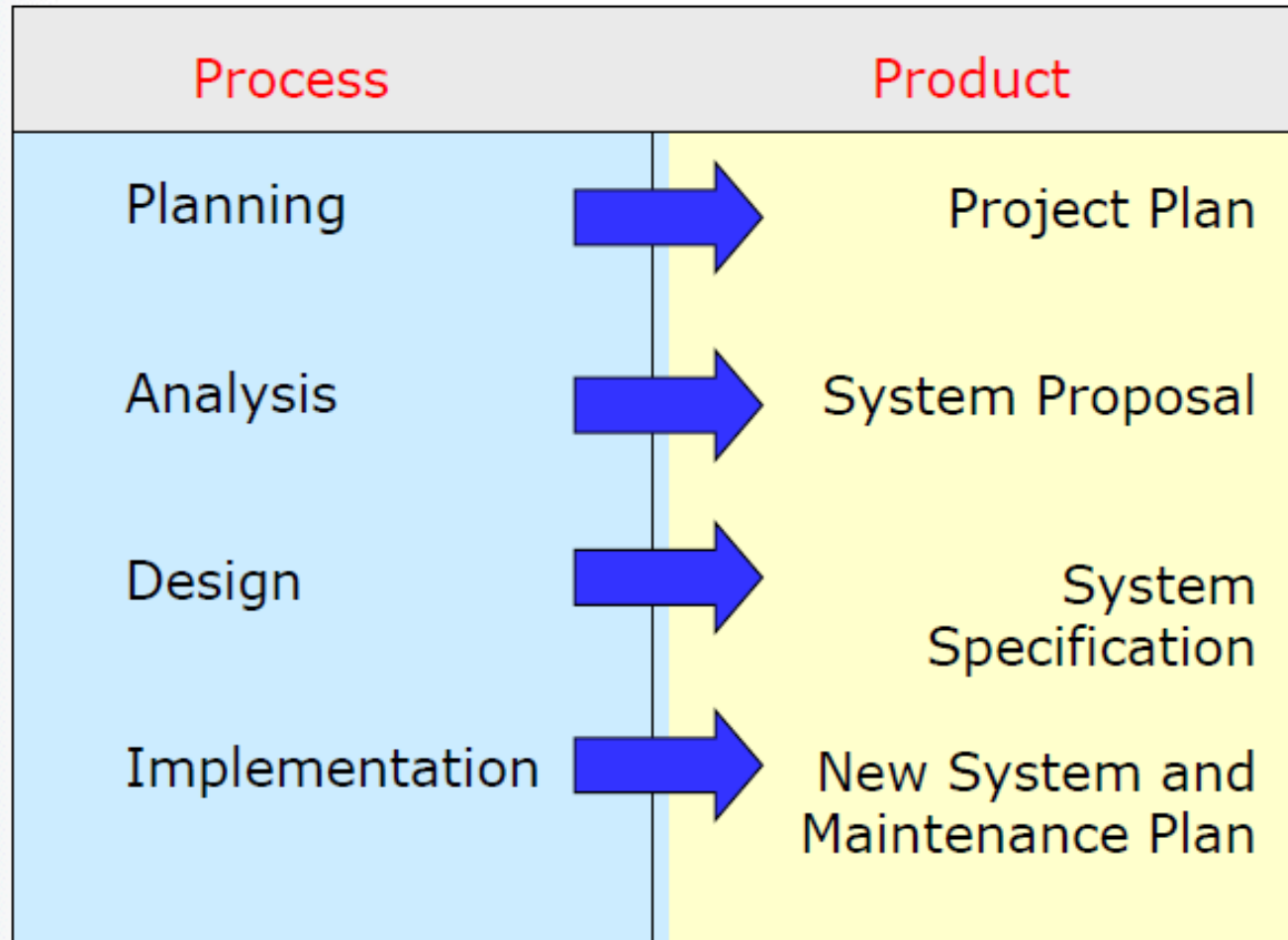
- ☑ Design strategy
- ☑ Architectural design
- ☑ Interface design (HCI)
- ☑ Database and file design
- ☑ Program design (what will the programs do?)

# Implementation



- Construction (Programming, testing, validation etc)
- Installation (including migration, change management)

# Processes and Deliverables



# **SYSTEM DEVELOPMENT**

## **Methodologies**



# What Is a Methodology?



- ❑ A formalized approach or series of steps
- ❑ *Writing code without a well-thought-out system request may work for small programs, but rarely works for large ones.*

# **System Development Methodologies**



1. Structured Design
2. Rapid Application Development
3. Agile Development

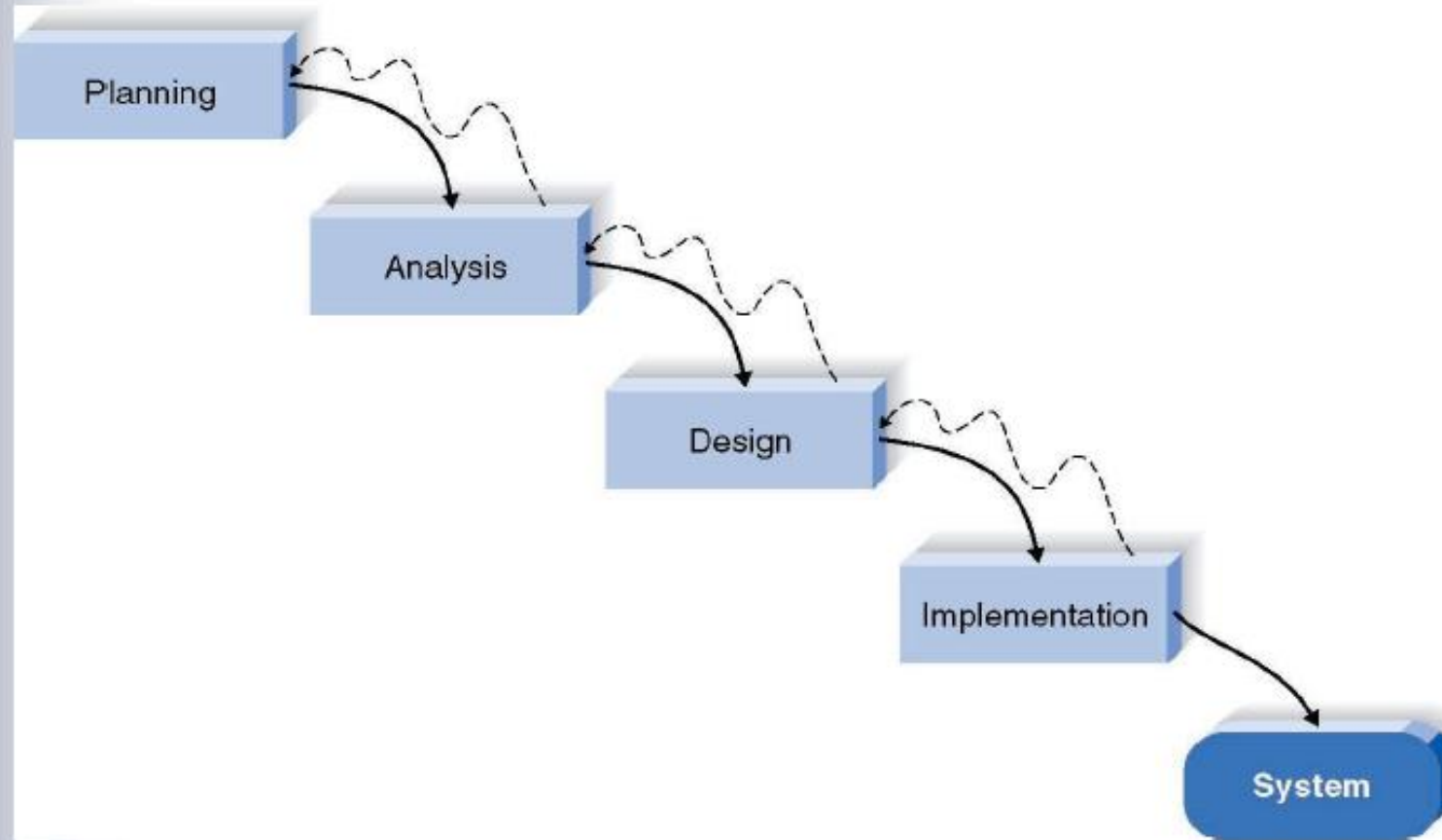


# 1. STRUCTURED DESIGN



- ❑ Projects move methodically from one to the next step
- ❑ Generally, a step is finished before the next one begins

# Waterfall Development Method



# Pros and Cons of the Waterfall Method



Pros	Cons
Identifies systems requirements long before programming begins	Design must be specified on paper before programming begins
	Long time between system proposal and delivery of new system

# Parallel Development

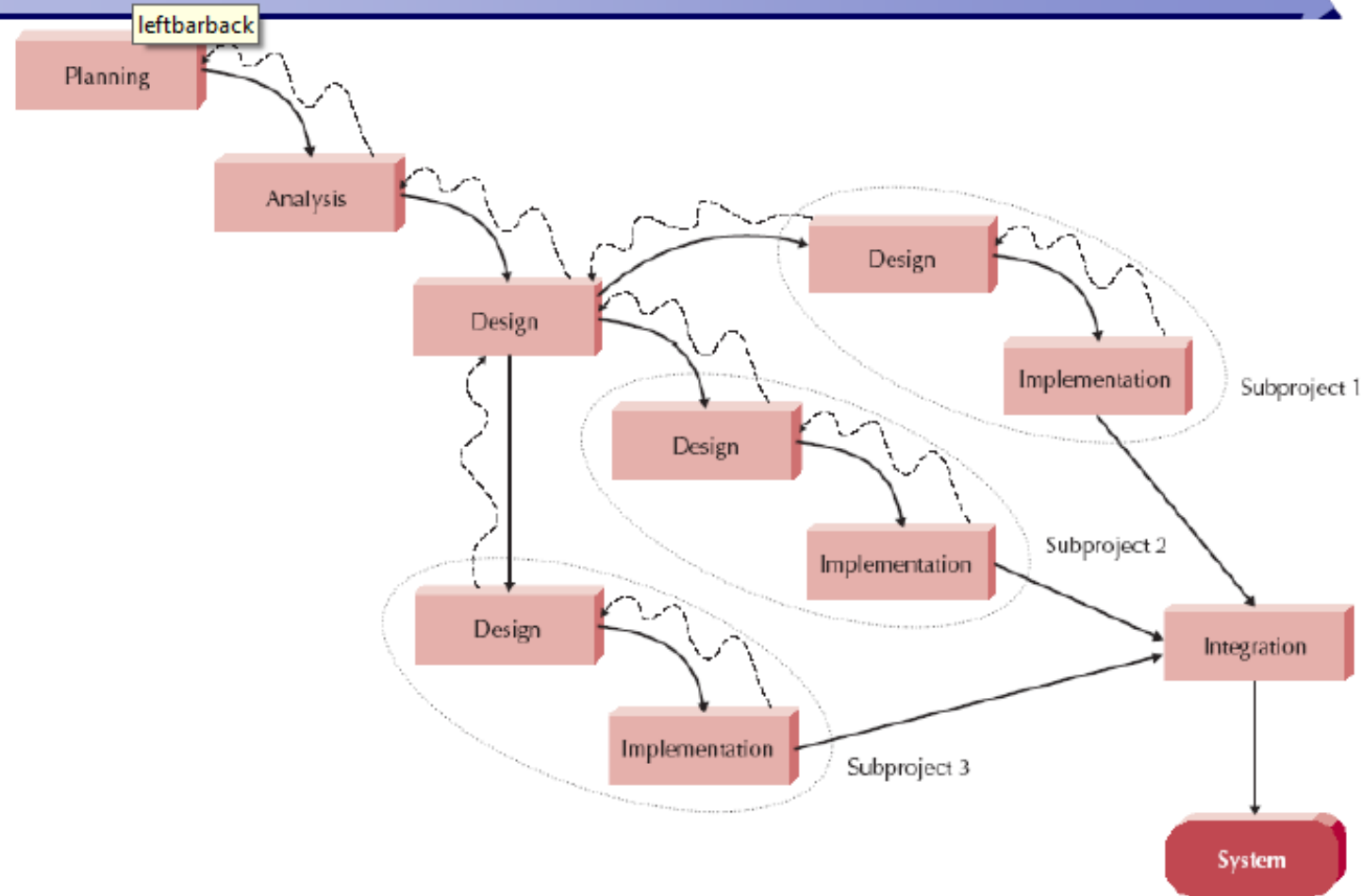


FIGURE 1-3 A Parallel Development-based Methodology

## **2. RAPID APPLICATION DEVELOPMENT (RAD)**



- Critical elements
  - CASE tools
  - JAD sessions
  - Fourth generation/visualization programming languages
  - Code generators

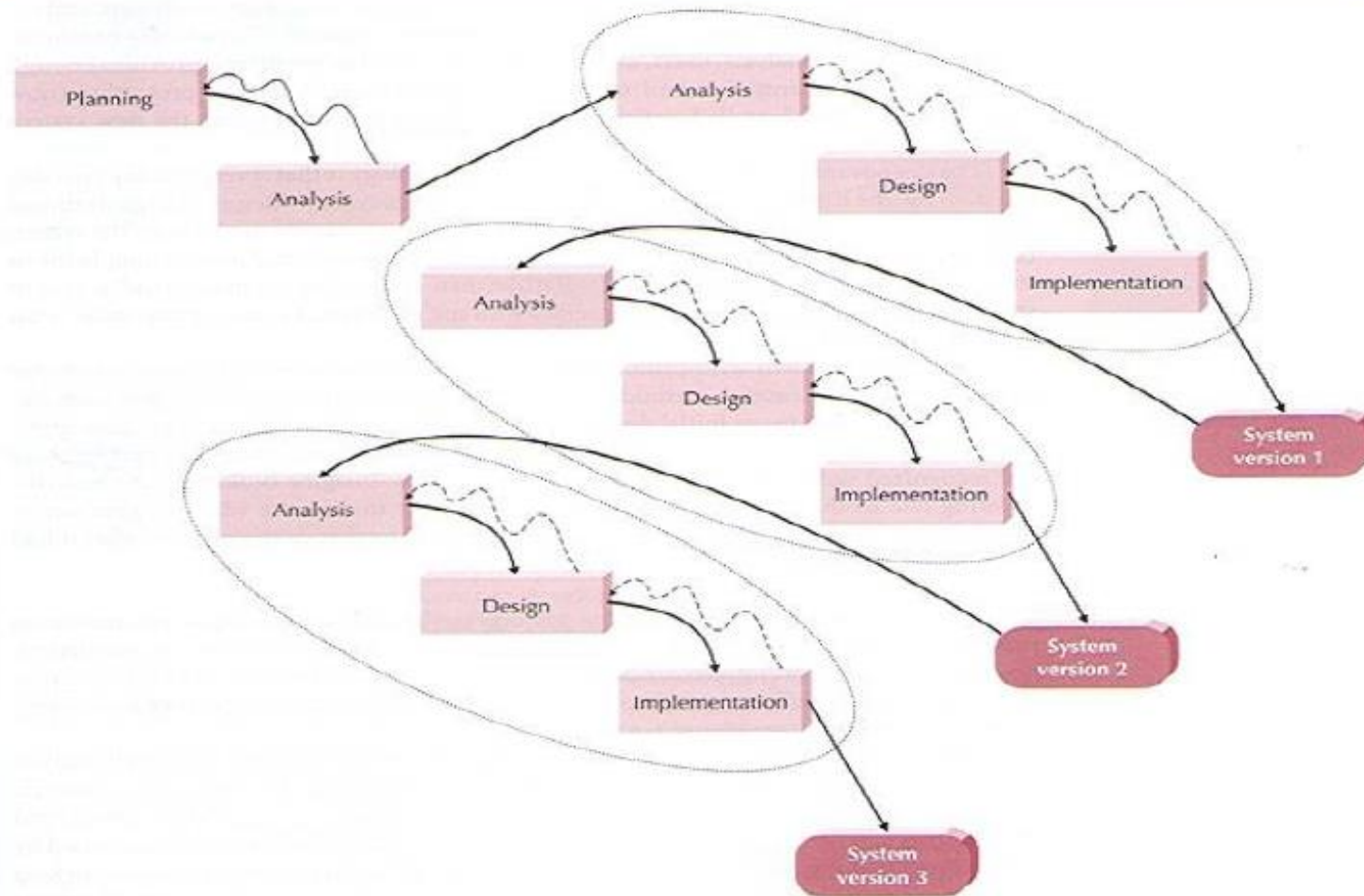
# Rapid Application Development Categories



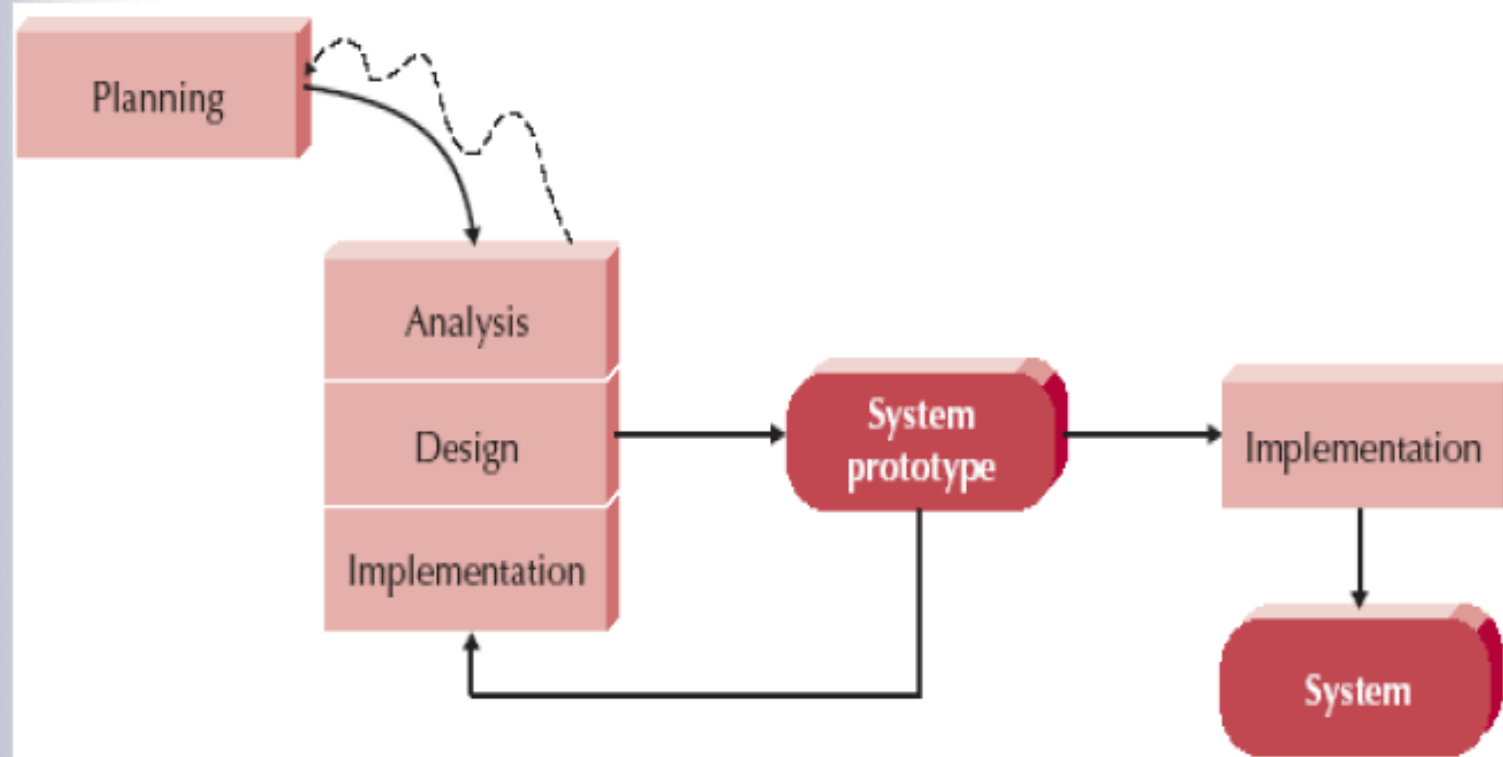
- ❑ Phased development
  - ⊕ a series of versions, later combined
- ❑ Prototyping
  - ⊕ System prototyping
- ❑ Throw-away prototyping
  - ⊕ Design prototyping



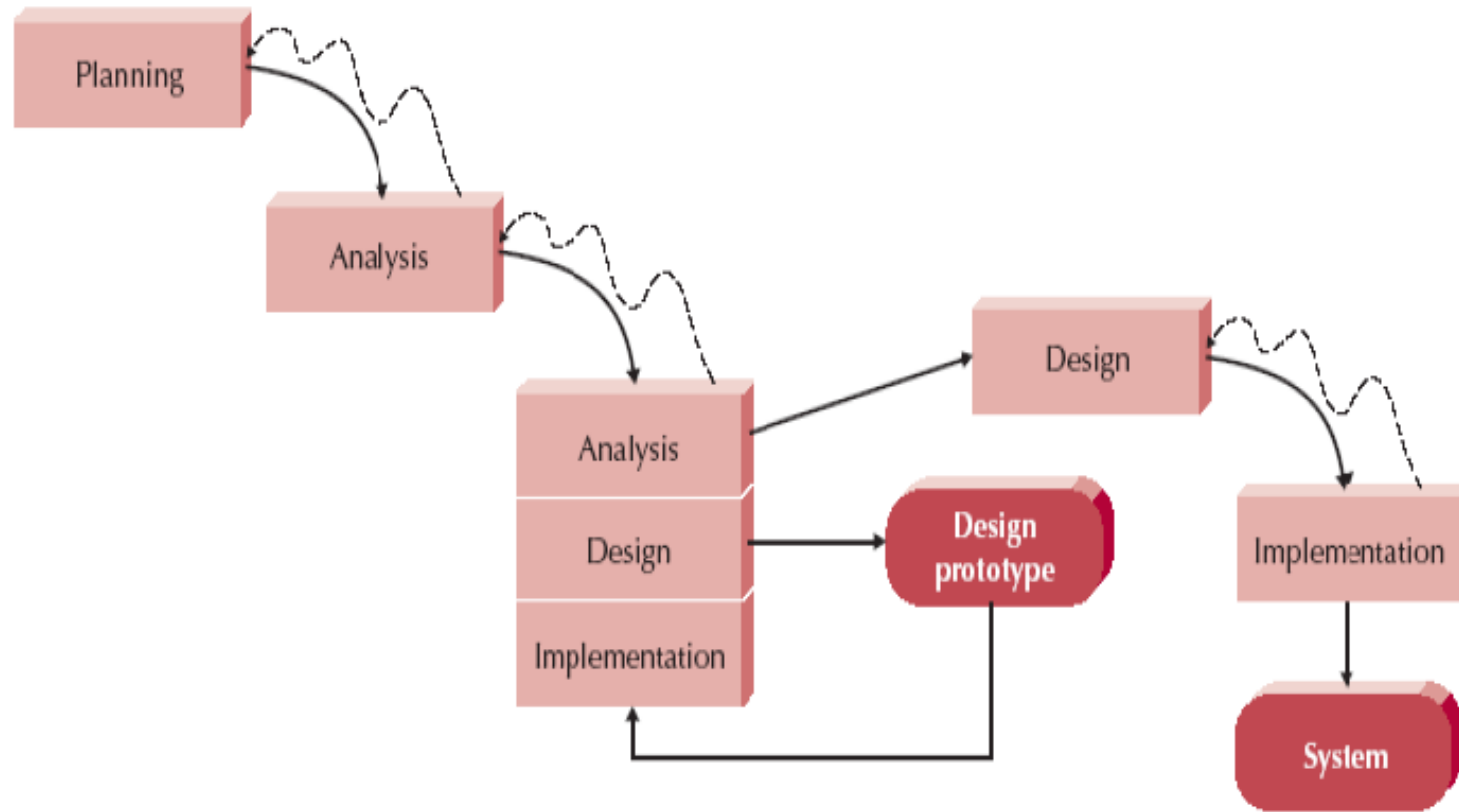
# Phased Development



# How Prototyping Works



# Throwaway Prototyping



### **3. AGILE DEVELOPMENT**



- Simple iterative application development
- Extreme programming (XP)

# Extreme Programming (XP)



- ☒ Key principles

- ✦ Continuous testing
- ✦ Simple coding by pairs of developers
- ✦ Close interactions with end users

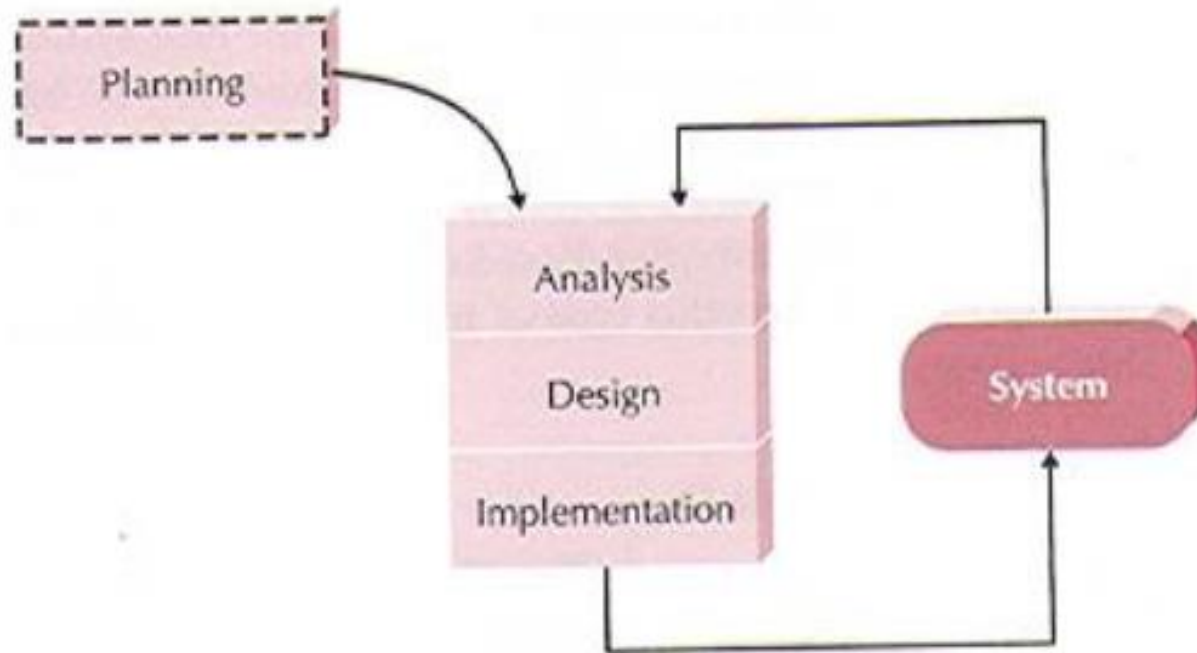
- ☒ Testing & Efficient Coding Practices

- ✦ Integrative testing environment

- ☒ Requires...

- ✦ Stable and experienced teams
- ✦ Small groups of developers ( $\leq 10$ )

# Extreme Programming (XP)






# Selecting the Appropriate Methodology



- ☑ Clarity of User Requirements
- ☑ Familiarity with the Technology
- ☑ System Complexity
- ☑ System Reliability
- ☑ Length of Time Schedules
- ☑ Time Schedule Visibility

# Criteria for Selecting a Methodology




Ability to Develop Systems	Structured Methodologies			RAD Methodologies		Agile Methodologies
	Waterfall	Parallel	Phased	Prototyping	Throwaway Prototyping	XP
with Unclear User Requirements	Poor	Poor	Good	Excellent	Excellent	Excellent
with Unfamiliar Technology	Poor	Poor	Good	Poor	Excellent	Poor
that are Complex	Good	Good	Good	Poor	Excellent	Poor
that are Reliable	Good	Good	Good	Poor	Excellent	Good
with a Short Time Schedule	Poor	Good	Excellent	Excellent	Good	Excellent
with Schedule Visibility	Poor	Poor	Excellent	Excellent	Good	Good

FIGURE 1-8 Criteria for Selecting a Methodology

# **Project Team Roles and Skills**



# Project Team Roles



- Business analyst (business value)
- Systems analyst (IS issues)
- Infrastructure analyst (technical issues – how the system will interact with the organization's hardware, software, networks, databases)
- Change management analyst (people and management issues)
- Project manager (budget, time, planning, managing)

# Summary



- *The Systems Development Life Cycle* (SDLC) consists of four stages: Planning, Analysis, Design, and Implementation
- *The Major Development Methodologies:*
  - ◻ Structured Design
    - ✦ Waterfall Method
    - ✦ Parallel Development
  - ◻ Rapid Application Development (RAD)
    - ✦ Phased Development
    - ✦ Prototyping (system prototyping)
    - ✦ Throwaway Prototyping (design prototyping)
  - ◻ Agile development
    - ✦ eXtreme Programming
- *Project Team Roles*

# Summary -- Part 2



- There are five *major team roles*:
  - business analyst
  - systems analyst
  - infrastructure analyst
  - change management analyst
  - project manager.