MM PG COLLEGE FATEHABAD

INTRODUCTION

System Analysis and Design (SAD ☹) B.Sc. 2nd Year/ 3rd Sem.

Department of Computer Science

Information System

- ☑ *Information systems* are **software** applications which manage large amounts of data.
- ☑ Most of the software out there is information systems software,
  - □ written in languages such as Java, C++, .NET and the like.

The Bad News

- 30% of large IT projects are **cancelled** before completion
- 50% of IT projects are **over-budget** by more than 200%
- The majority of completed projects deliver *60% or less of prescribed functionality*
- Many delivered information systems are **under-used** because they don’t meet user needs and/or expectations
- Legacy systems are a serious and growing **bottleneck** to organizational evolution

Software Horror Stories

- **Bank of America** spent $23,000,000 on a 5-year project to develop a new accounting system. Spent over $60,000,000 trying to make new system work, finally abandoned it. Loss of business estimated in excess of $1,000,000,000

- **The B1 Bomber** required an additional $1,000,000,000 to improve its air defense software, but the software still isn't working to specification

- **Ariane 5, Flight 501**
  - The loss of a $500,000,000 spacecraft was ultimately attributed to errors in requirements, specifications and inadequate software reuse practices.

Why is this Course Important?

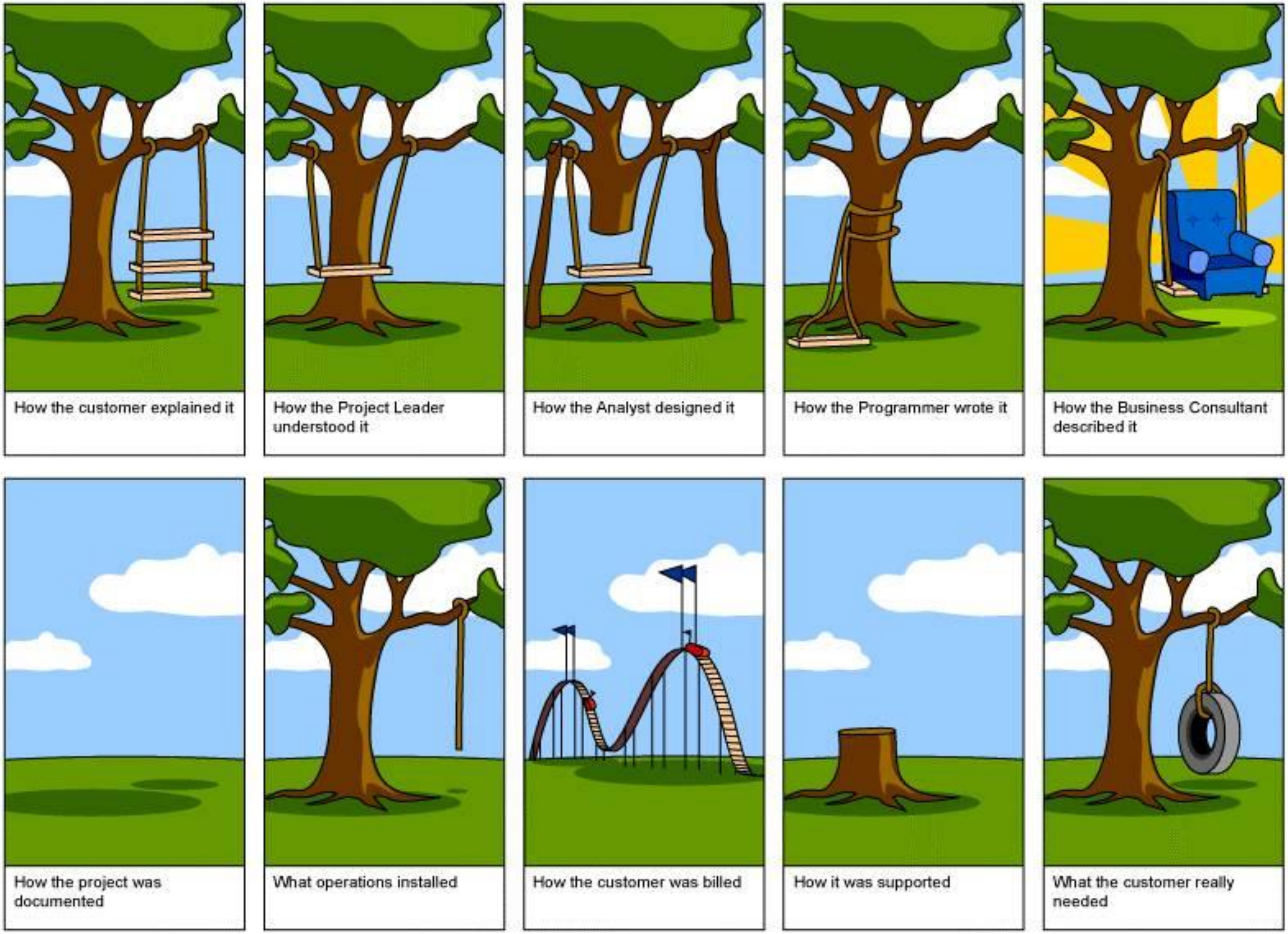
- ***Most errors (54%) are detected after coding and testing.***

- ***Almost half of all errors in software (45%) are in requirements and design.***

- ***Most errors made during requirements analysis are non-clerical (77%)***

- ***Requirements errors can cost up to 100 times more to fix than implementation errors***
  - ***if they are not caught early on.***

Need to do requirements and design right!


How the customer explained it
How the Project Leader understood it
How the Analyst designed it
How the Programmer wrote it
How the Business Consultant described it
How the project was documented
What operations installed
How the customer was billed
How it was supported
What the customer really needed

Key Ideas

- Many failed systems were abandoned because analysts tried to *build wonderful systems without understanding the organization*.
- The primary goal is to create **value** for the organization.

Key Ideas

- Systems analyst is a key person
  - analyzing the business
  - identifying opportunities for improvement
  - designing information systems to implement these ideas.

What is Systems Analysis?

- The collection of notations, methodologies and tools used to gather details and analyze a problem situation prior to information system design and implementation

- ***Systems analysis*** (or, ***requirements analysis***) must ensure that the proposed information system meets user needs, can be delivered on time, and can be updated inexpensively.

- Problems in "getting the systems analysis right", such as ill-defined situations, ambiguities, inconsistencies, mixing requirements with design

Need for Systems Analysis?

- ***Remember, finding and fixing a fault after software delivery is 100x more expensive than finding and fixing it during systems analysis or early design phases***

THE SYSTEMS DEVELOPMENT LIFE CYCLE

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*

Project Phases

- ☑ **Planning**
  - □ Why build the system?
- ☑ **Analysis**
  - □ Who, what, when, where will the system be?
- ☑ **Design**
  - □ How will the system work?
- ☑ **Implementation**
  - □ System delivery

Planning

- Identifying business value
- Analyze feasibility
- Develop work plan
- Staff the project
- Control and direct project

Analysis

- Analysis strategy
  - Analysis of *current system*
  - Ways to design **new system**
- Requirements gathering
  - Interviews, questionnaires
- Process modeling
- Data modeling

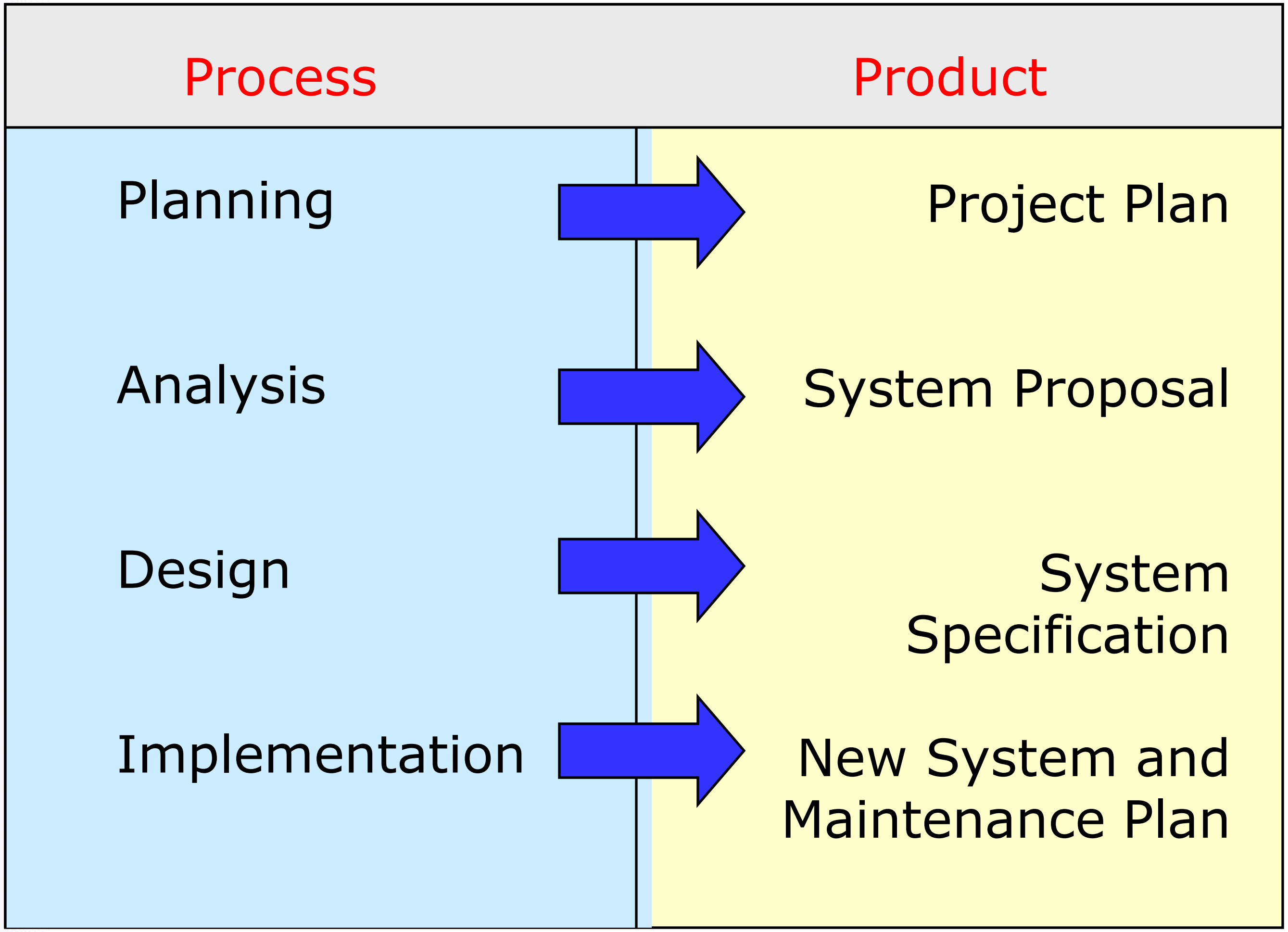
Design

- Architectural design
  - **Hardware**
  - **Software**
  - **Network** infrastructure
- Interface design
- Database and file design
- Program design

Implementation

- **Construction**
  - Writing programs
  - Testing
- **Installation**
  - Replace old with new system
  - Training users
- Support Plan

Processes and Deliverables



Process	Product
Planning	Project Plan
Analysis	System Proposal
Design	System Specification
Implementation	New System and Maintenance Plan

SYSTEM DEVELOPMENT Methodologies

What Is a Methodology?

- **A formalized approach or series of steps to implement SDLC**
- Methodology categories:
  - Process-centered
  - Data-centered
  - Object-oriented

Need for methodology

- ☑ *Writing code without a well-thought-out system request may work for small programs, but rarely works for large ones.*
- ☑ Need to have a good **design** from the requirements before moving on to implementation.

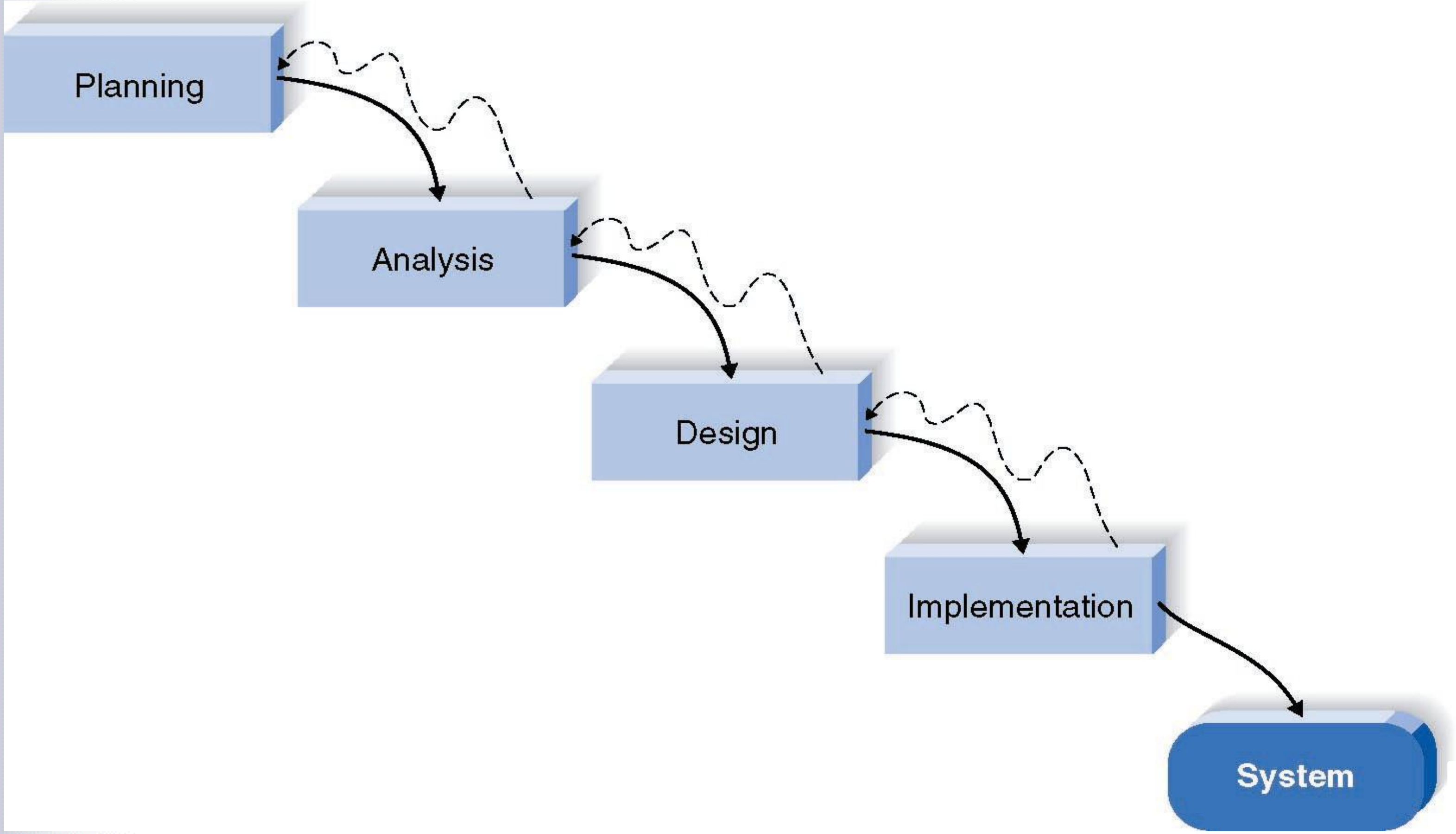
Systems development methodologies

- **Structured Design**
  - Waterfall Development
  - Parallel Development
- **Rapid Application Development (RAD)**
  - Phased Development
  - Prototyping
- **Agile Development**
  - Extreme Programming

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

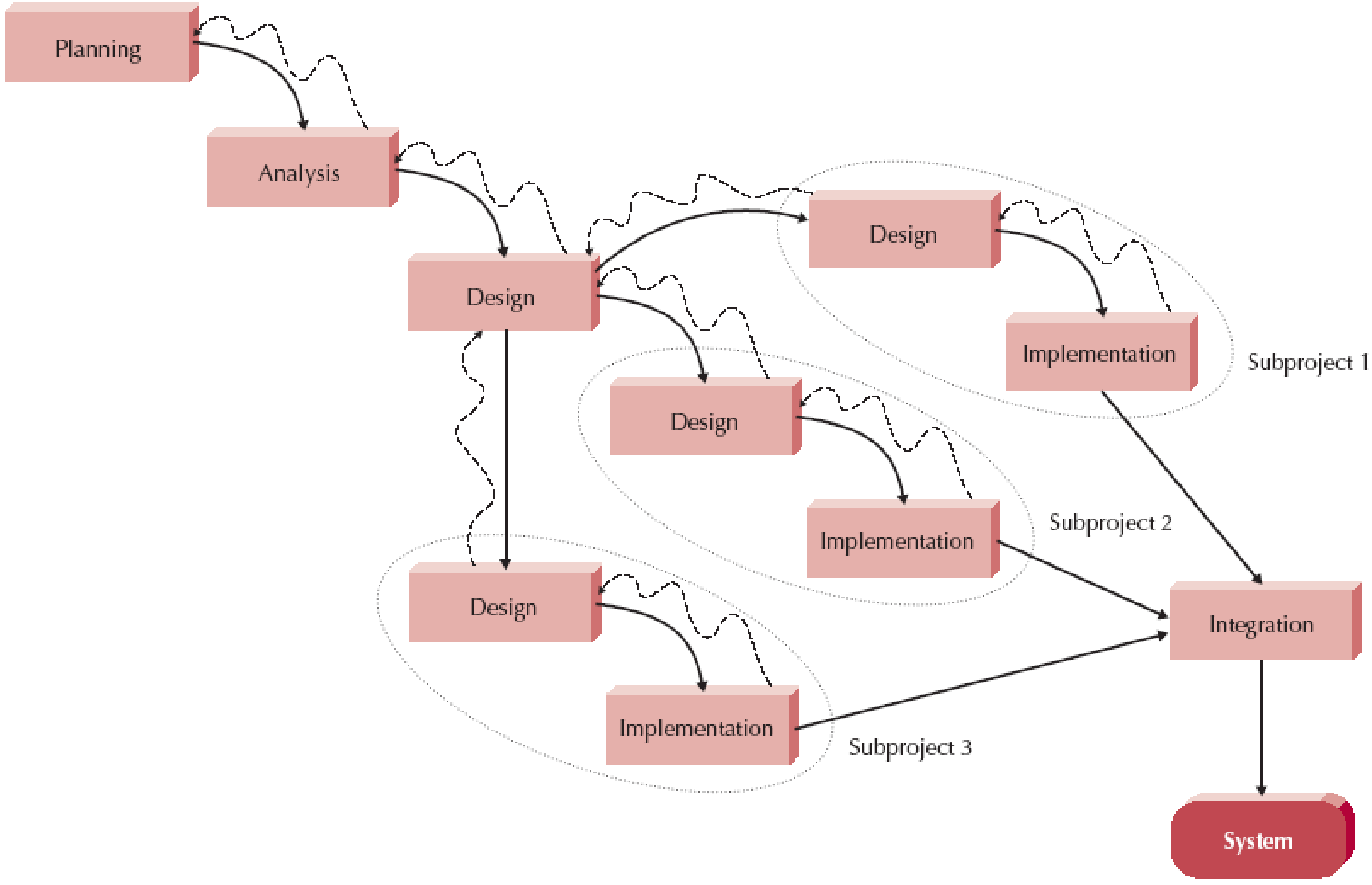


FIGURE 1-3 A Parallel Development-based Methodology

Pros and Cons of the Parallel Method

Pros	Cons
Reduce schedule time to deliver the system	Subprojects may not be completely independent – integration may be complicated
	Paper document outputs still cause problems

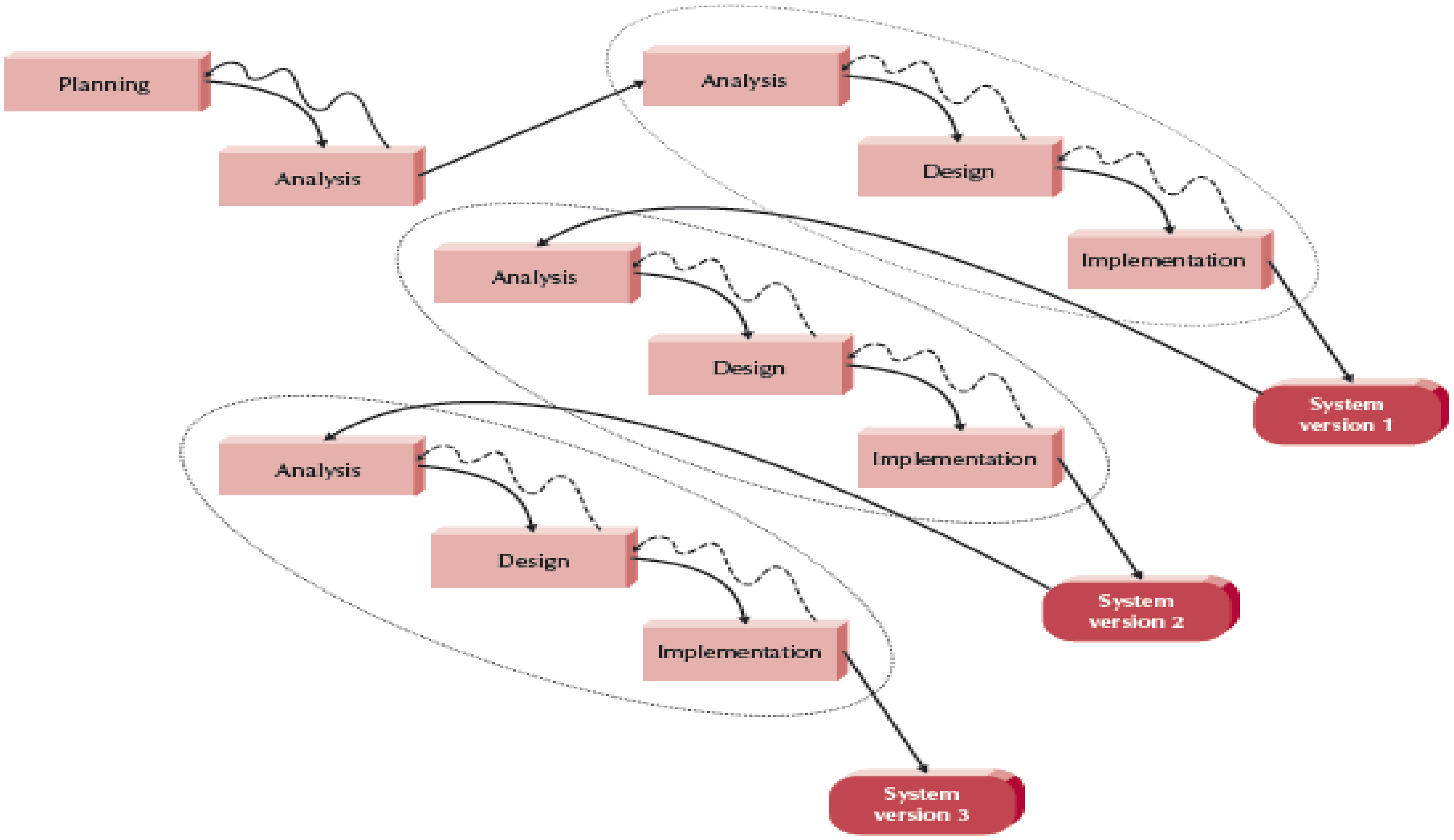
Rapid Application Development

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

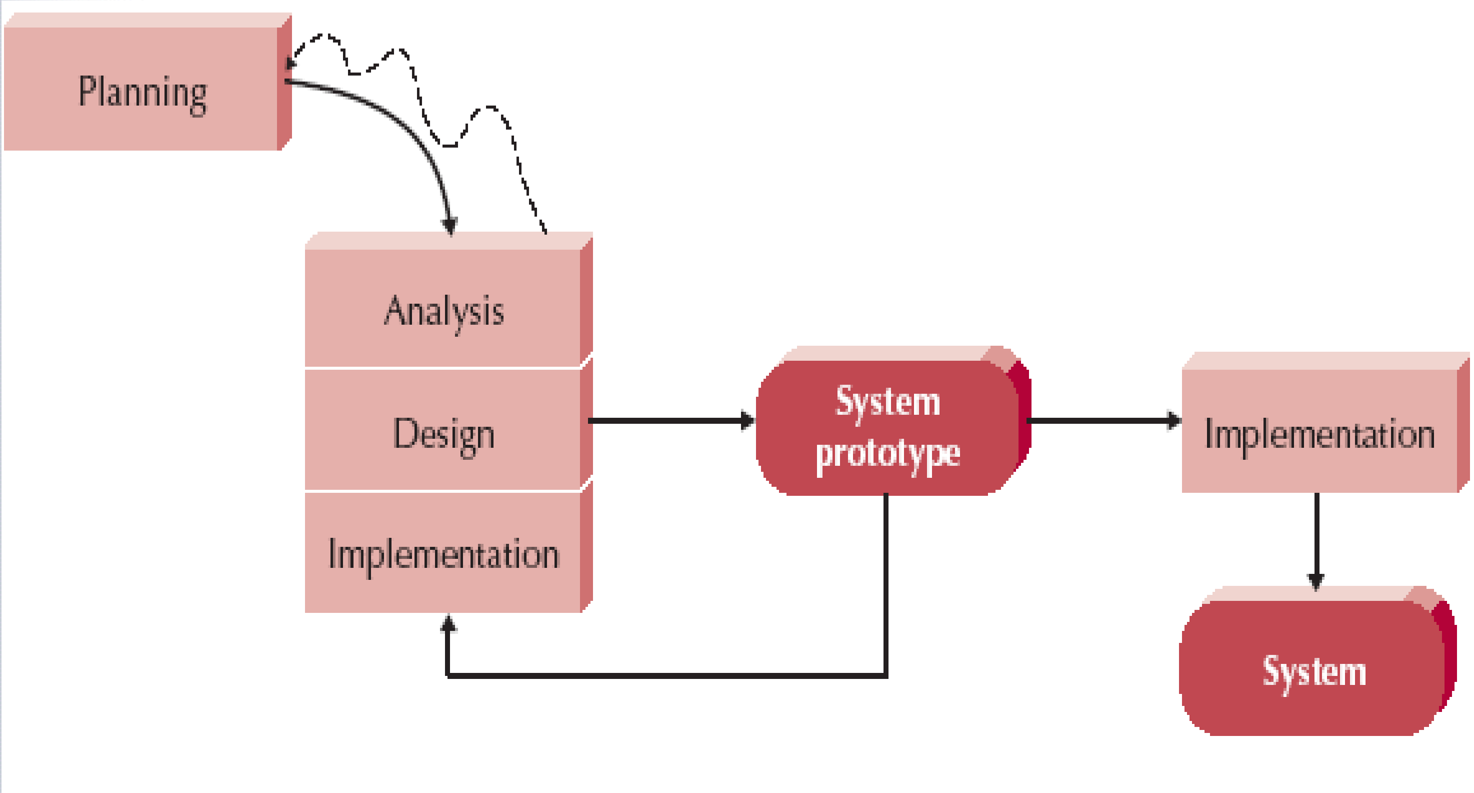
Rapid Application Development Categories

- Phased development
  - A series of versions
- Prototyping
  - System prototyping
- Throw-away prototyping
  - Design prototyping
- Agile Development
- Extreme Development

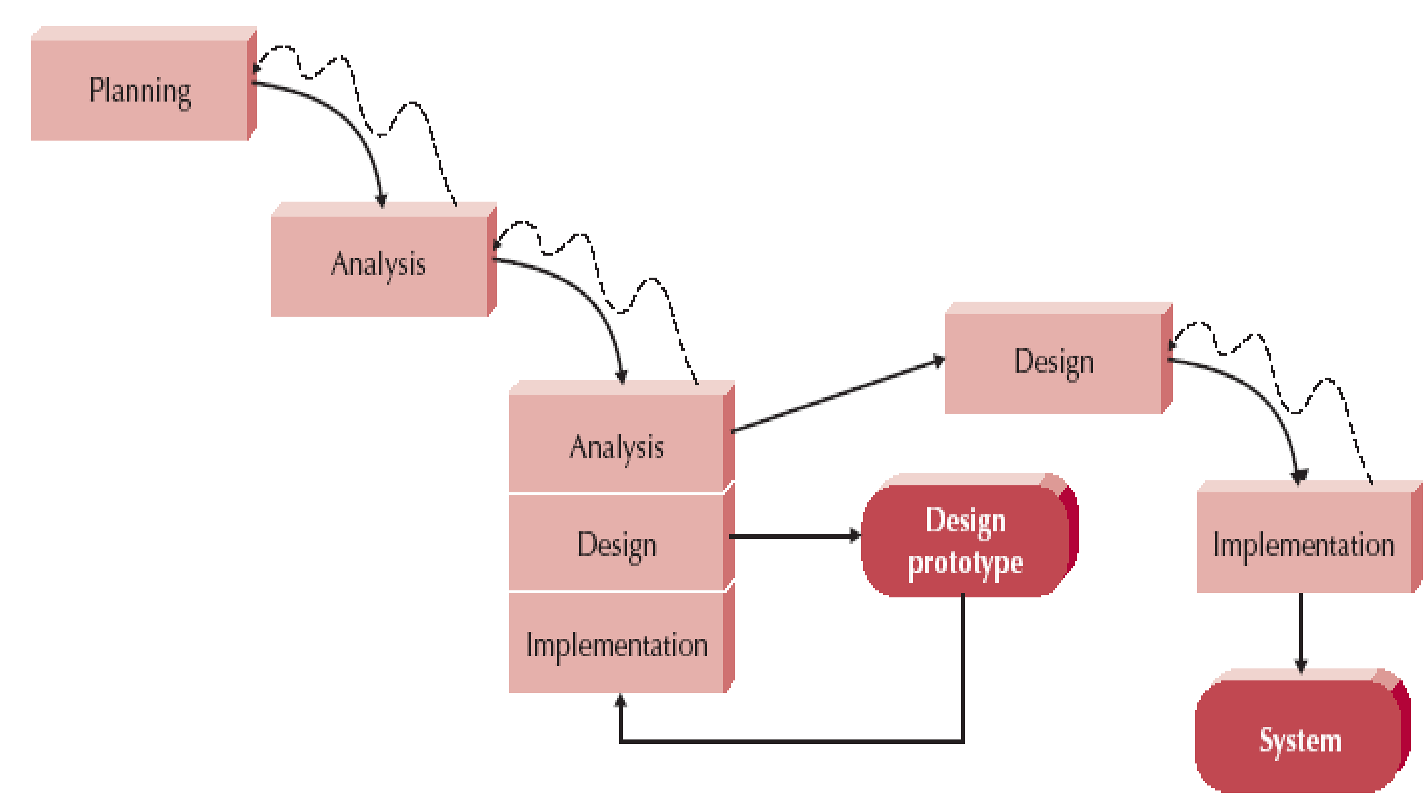
Phased development



How Prototyping Works



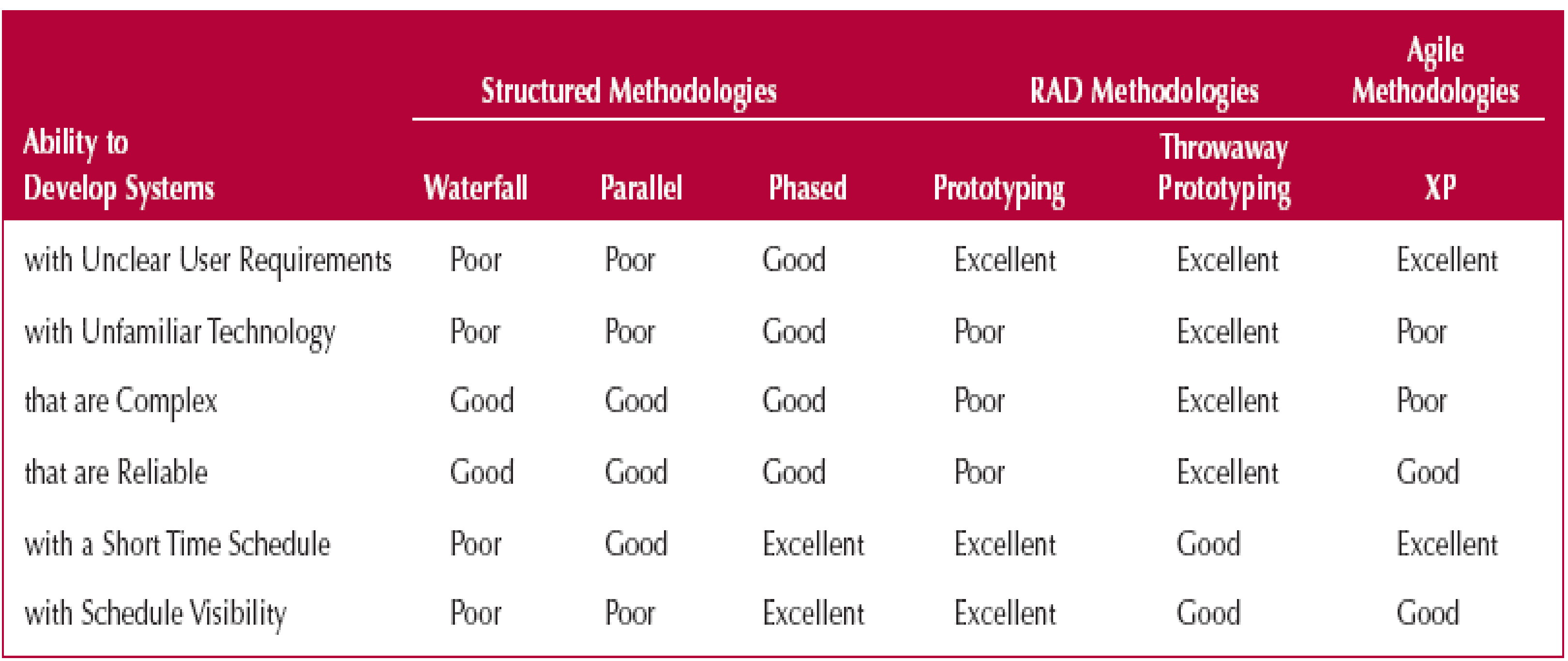
Throwaway Prototyping



Selecting the Appropriate Methodology

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

Criteria for Selecting a Methodology



	Structured Methodologies			RAD Methodologies		Agile Methodologies
Ability to Develop Systems	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

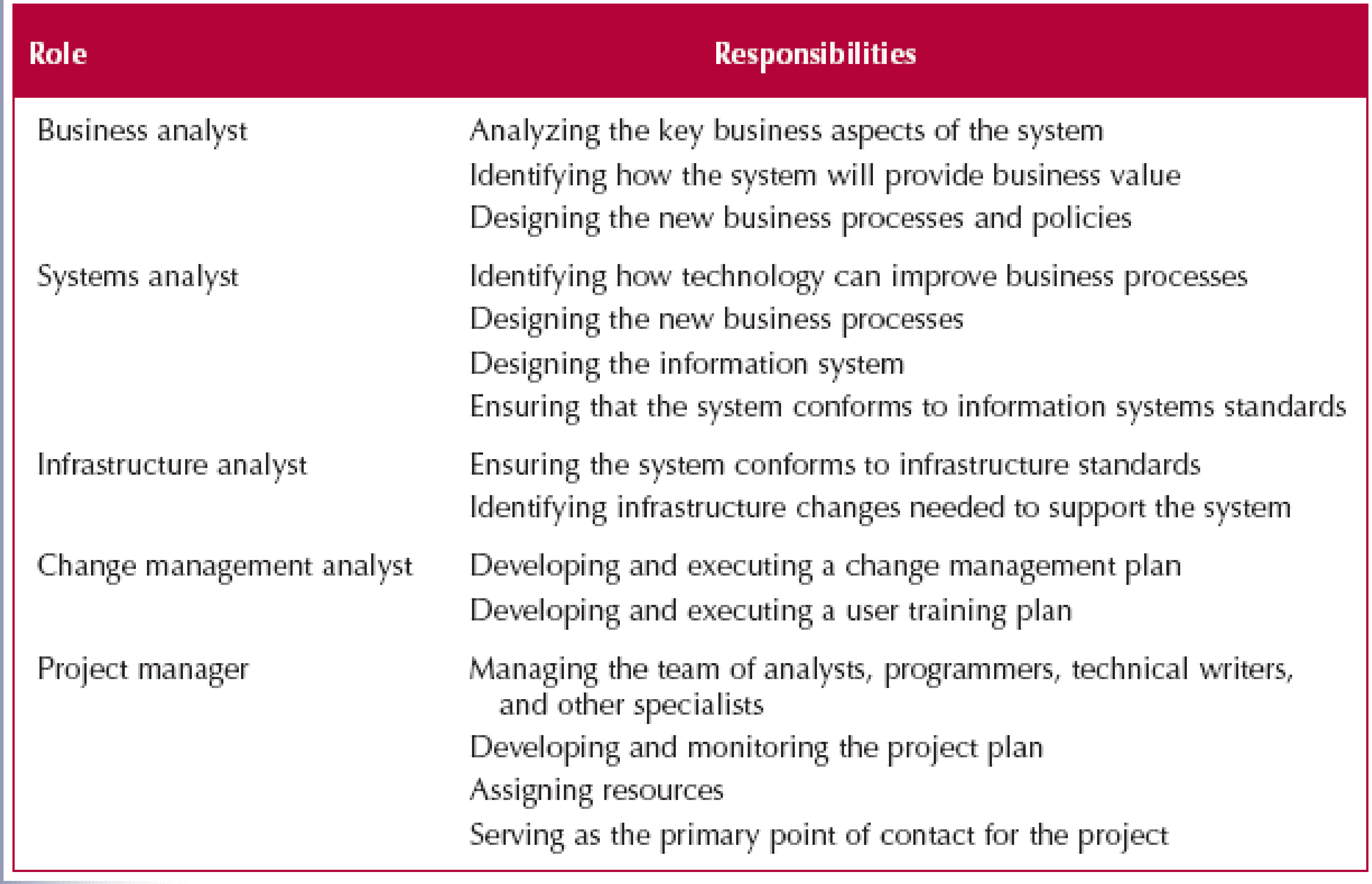
FIGURE 1-8 Criteria for Selecting a Methodology

Project Team Roles and Skills

Information Systems Roles

- ☑ Business analyst
- ☑ System analyst
- ☑ Infrastructure analyst
- ☑ Change management analyst
- ☑ Project manager

Project Team Roles



Role	Responsibilities
Business analyst	Analyzing the key business aspects of the system Identifying how the system will provide business value Designing the new business processes and policies
Systems analyst	Identifying how technology can improve business processes Designing the new business processes Designing the information system Ensuring that the system conforms to information systems standards
Infrastructure analyst	Ensuring the system conforms to infrastructure standards Identifying infrastructure changes needed to support the system
Change management analyst	Developing and executing a change management plan Developing and executing a user training plan
Project manager	Managing the team of analysts, programmers, technical writers, and other specialists Developing and monitoring the project plan Assigning resources Serving as the primary point of contact for the project

Summary -- Part 1

- *The Systems Development Life Cycle* consists of four stages: Planning, Analysis, Design, and Implementation
- The *major development methodologies*:
  - Structured design
    - the waterfall method
    - Parallel development
  - RAD development
    - Prototyping (regular and throwaway)
  - Agile development
    - XP streamline SDLC

Summary -- Part 2

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