

Systems Analysis and Design

CHAPTER 1



Alan Dennis, Barbara Haley Wixom, and Roberta Roth
John Wiley & Sons, Inc.

Slides by Candace S. Garrod
Red Rocks Community College

Systems Development Methodologies

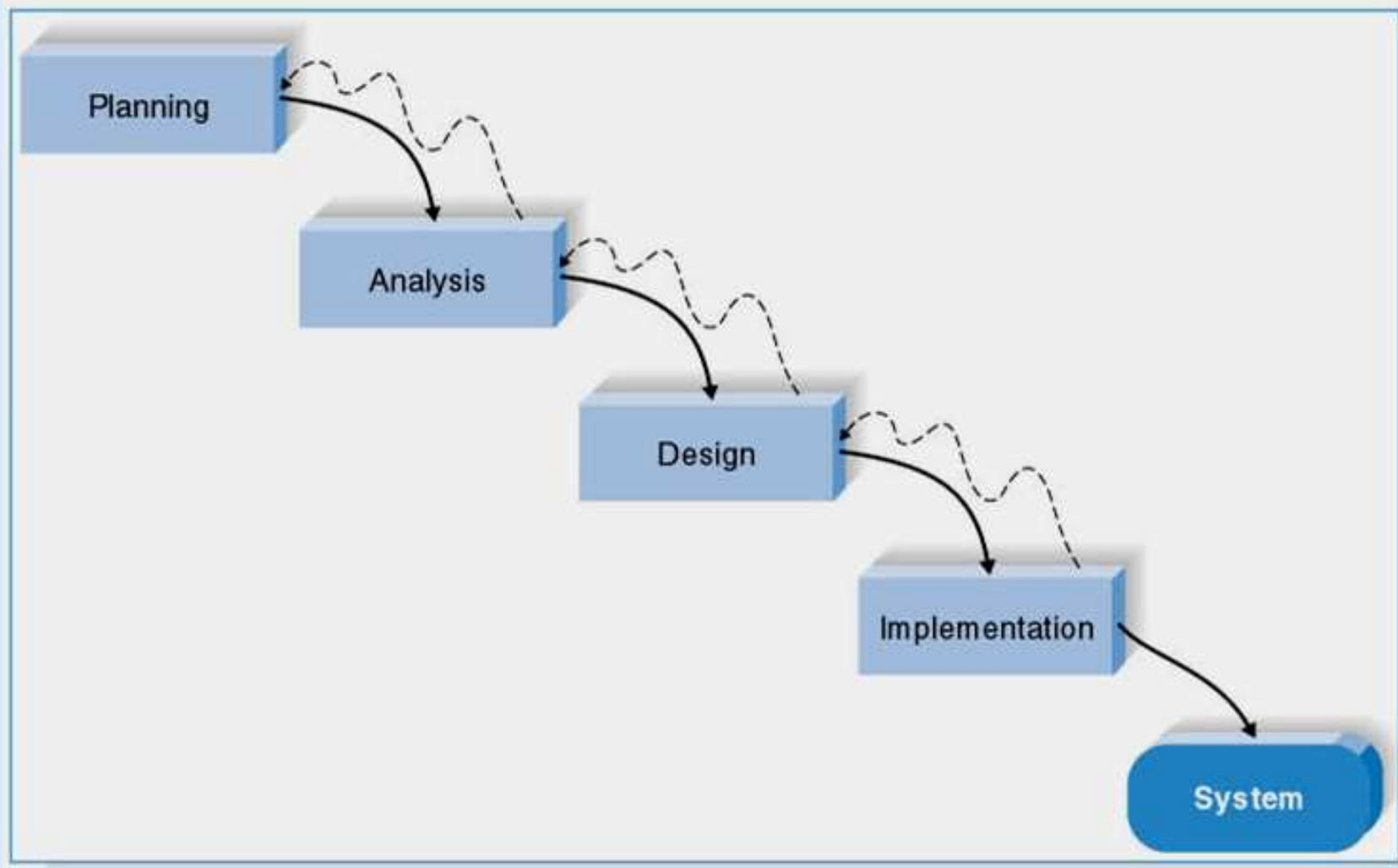


- ☑ A methodology is a formalized approach to implementing the SDLC.
- ☑ The methodology will vary depending on whether the emphasis is on businesses processes or on the data that supports the business.

Structured Design Methodology


- ☑ Structured design methodologies adopt a formal step-by-step approach to the SDLC that moves logically from one phase to the next.
- ☑ This design methodology introduces the use of formal modeling or diagramming techniques to describe a system's basic business processes.

Waterfall Development-based Methodology



Waterfall Development

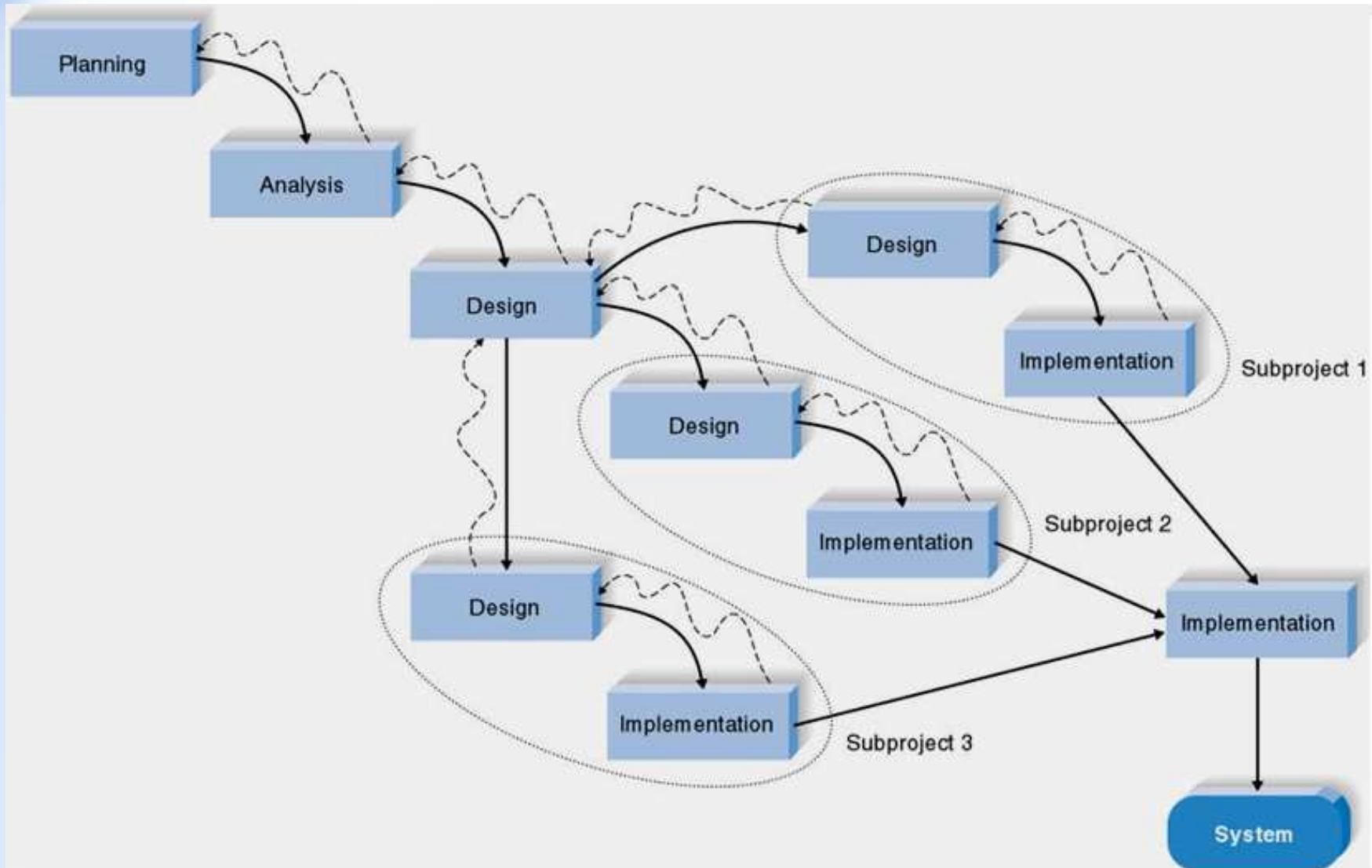
- ❑ With waterfall development- based methodologies, the analysts and users proceed sequentially from one phase to the next.
- ❑ The two key advantages of waterfall development-based methodologies are:
 - The system requirements are identified long before programming begins.
 - Changes to the requirements are minimized as the project proceeds.

- 
- The two key disadvantages of waterfall development-based methodologies are:
- The design must be completely specified before programming begins.
 - A long time elapses between the completion of the system proposal in the analysis phase and the delivery of the system.

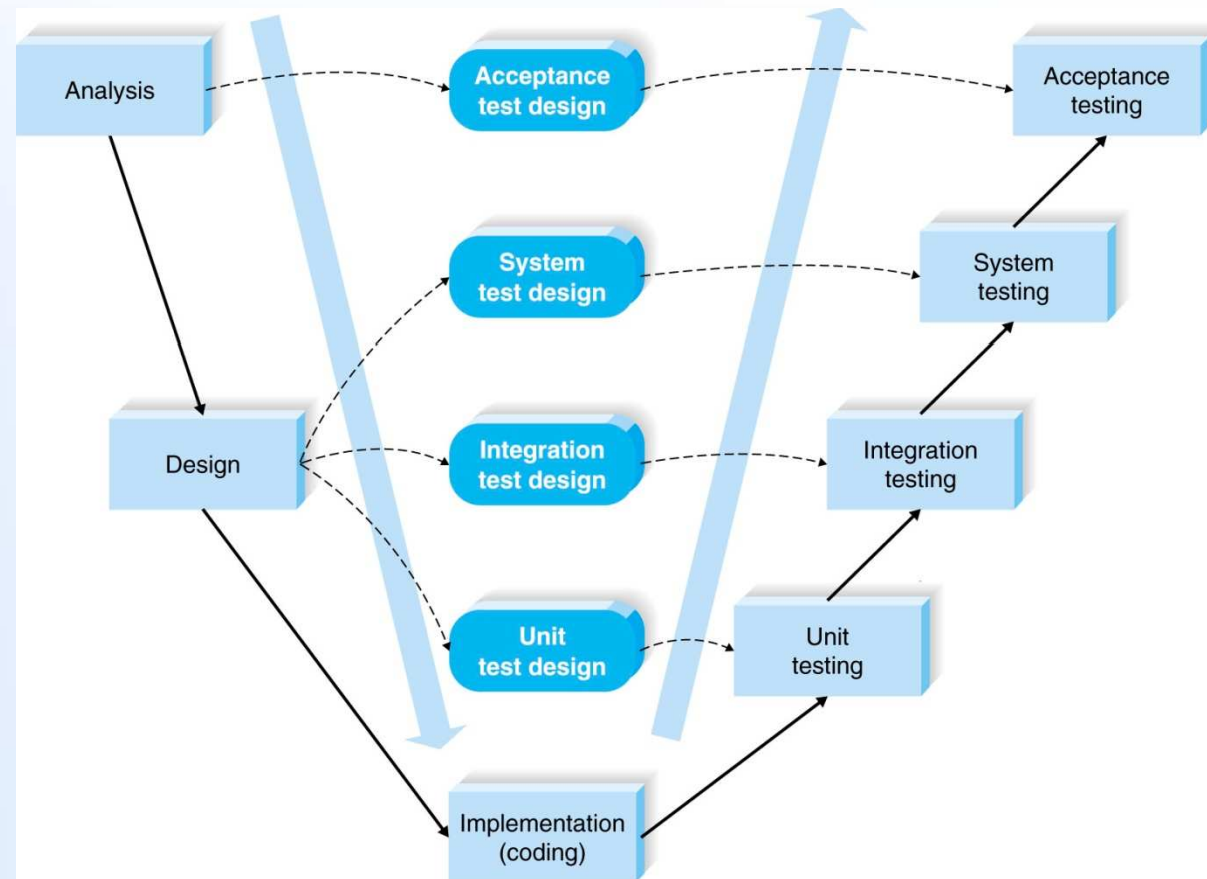
Parallel Development

- ❑ This methodology attempts to address the long time interval between the analysis phase and the delivery of the system.

A general design for the entire system is performed and then the project is divided into a series of distinct subprojects.



V-model



Rapid Application Development (RAD) Methodology

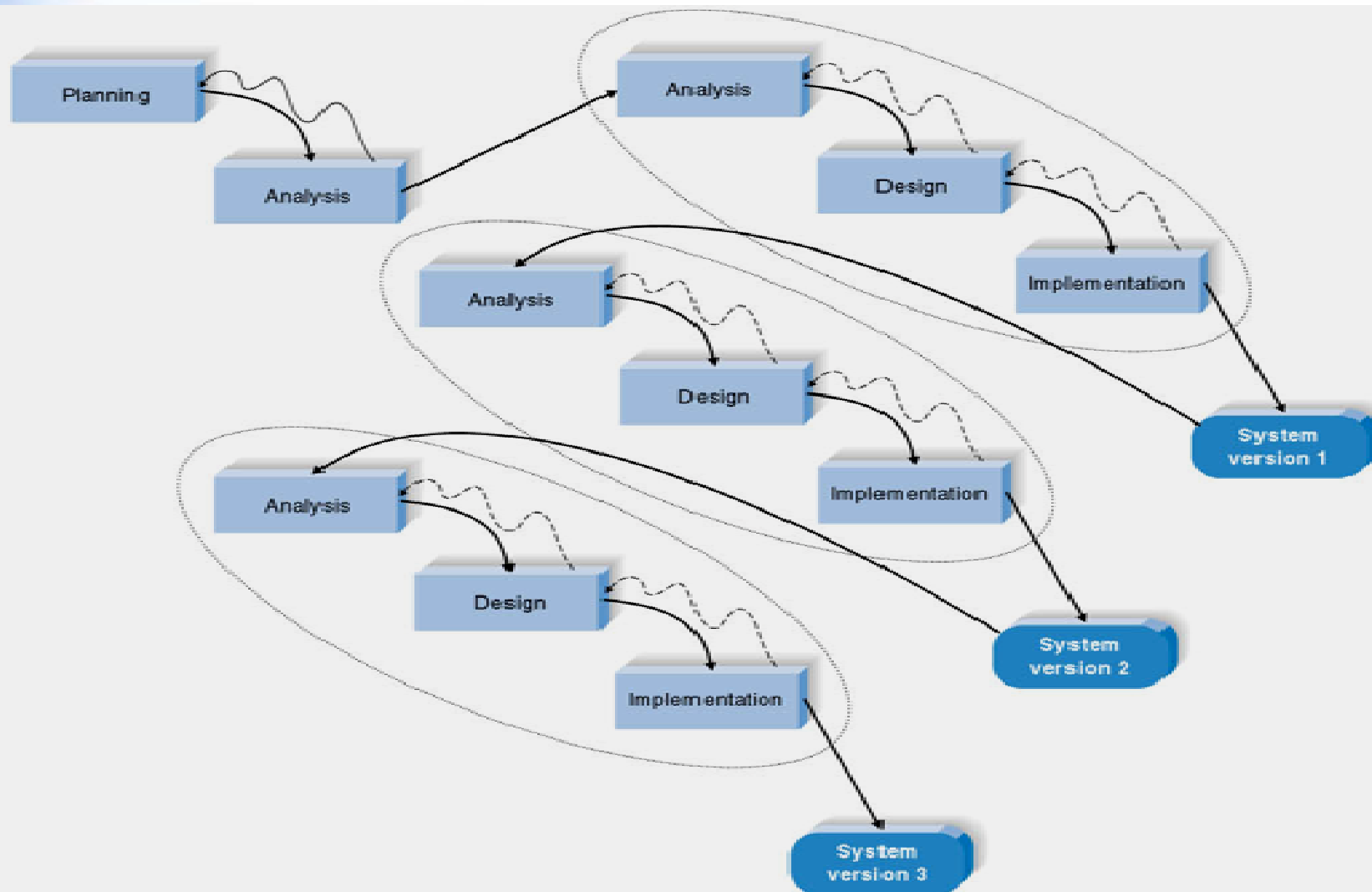
- ❑ RAD-based methodologies adjust the SDLC phases to get some part of system developed quickly and into the hands of the users.
- ❑ Most RAD-based methodologies recommend that analysts use special techniques and computer tools to speed up the analysis, design, and implementation phases, such as CASE (computer-aided software engineering) tools.

Rapid Application Development

Critical elements

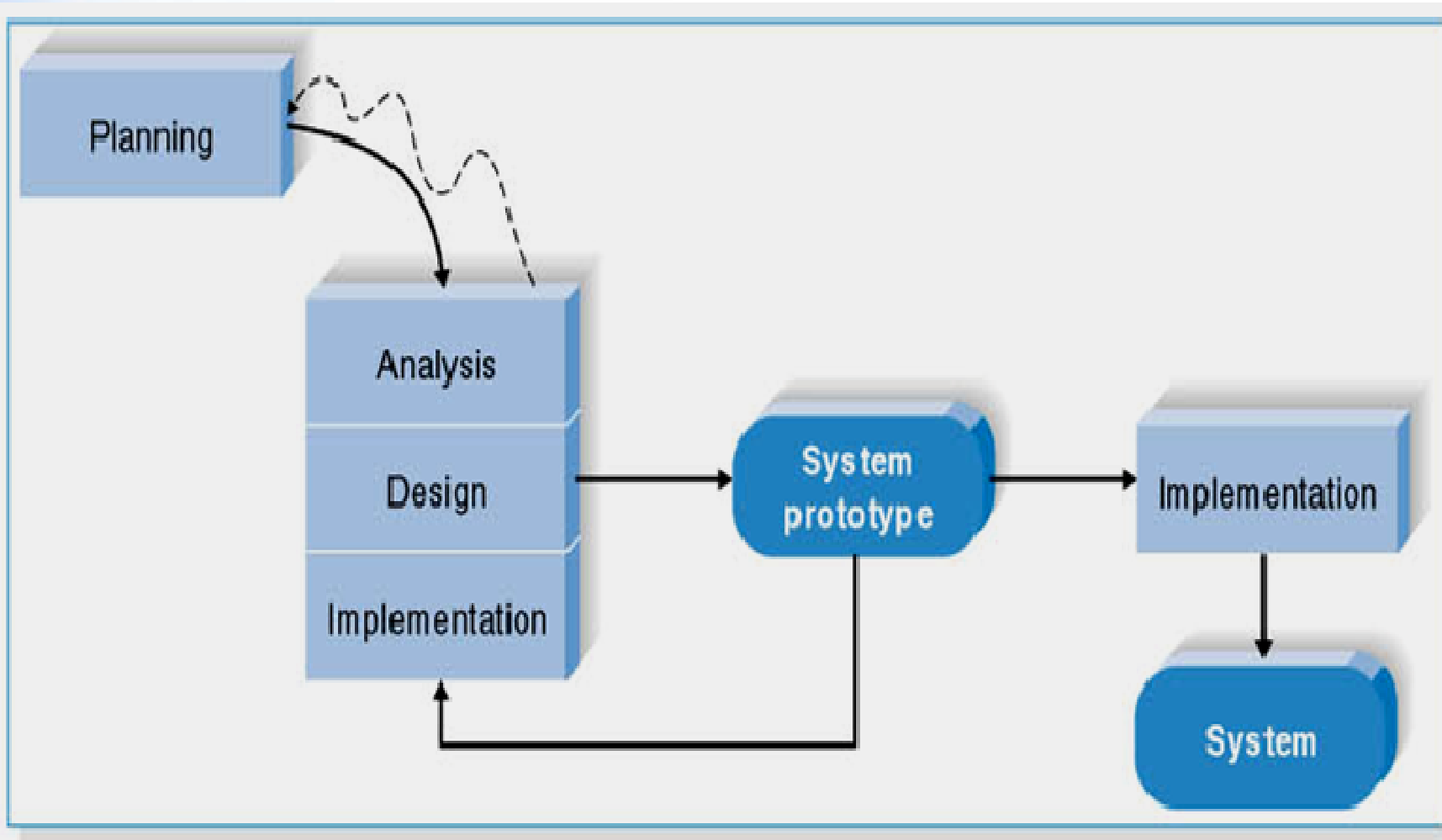
- CASE tools (Computer Aided Software Engineering)
- JAD sessions (Joint Application Development)
- Fourth generation/visualization programming languages
- Code generators

Iterative Development



- ❑ This methodology breaks the overall system into a series of versions that are developed sequentially.
- ❑ The team categorizes the requirements into a series of versions, then the most important and fundamental requirements are bundled into the first version of the system.
- ❑ The analysis phase then leads into design and implementation; however, only with the set of requirements identified for version 1.
- ❑ As each version is completed, the team begins work on a new version.

Prototyping-based Methodology

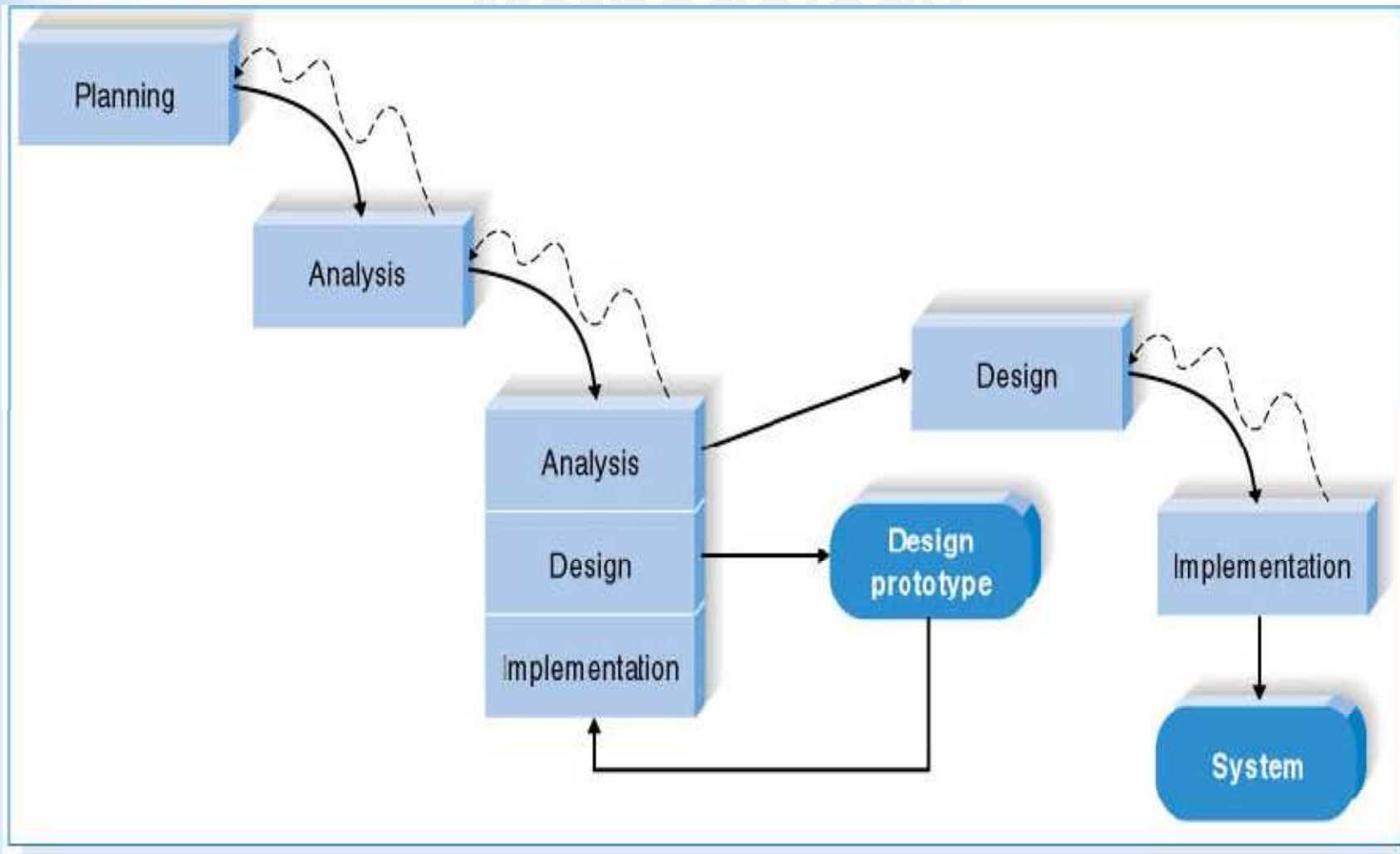


Prototyping

- ❑ Prototyping-based methodologies perform the analysis, design and implementation phases concurrently.
- ❑ All three phases are performed repeatedly in a cycle until the system is completed.
- ❑ A prototype is a smaller version of the system with a minimal amount of features.

- ☑ Advantage: Provides a system for the users to interact with, even if it is not initially ready for use.
- ☑ Disadvantage: Often the prototype undergoes such significant changes that many initial design decisions prove to be poor ones.

Throwaway Prototyping-based



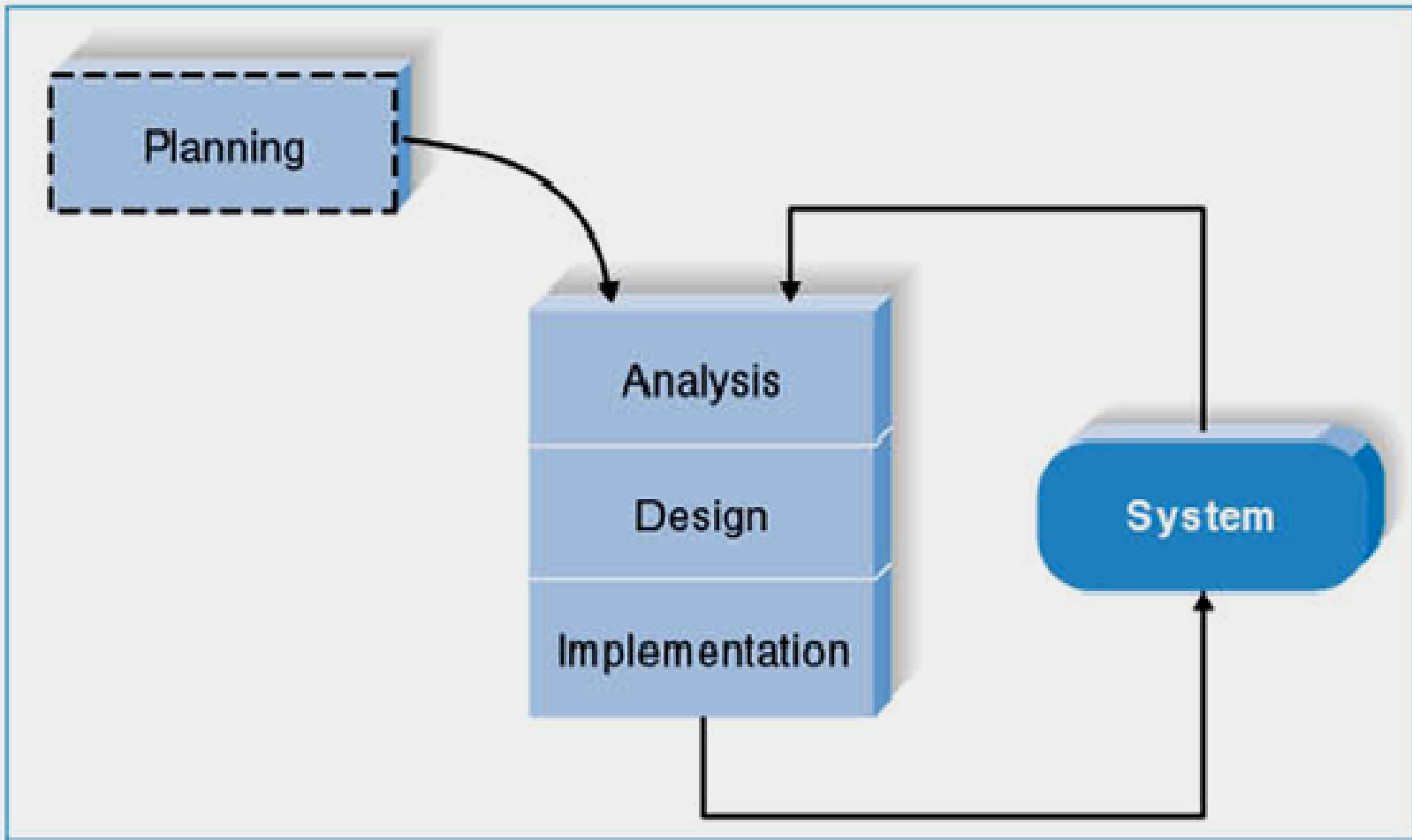
Throwaway Prototyping

- ❑ Throwaway prototyping methodologies are similar to prototyping based methodologies.
- ❑ The main difference is that throwaway prototyping IS completed during a different point in the SDLC.
- ❑ Has relatively thorough analysis phase.

Agile Development Methodology

- ☑ This category focuses on streamlining the SDLC by eliminating much of the modeling and documentation overhead and the time spent on those tasks.
- ☑ Projects emphasize simple, iterative application development.
- ☑ This category uses extreme programming, which is described next.

An Extreme Programming-based Methodology



Extreme Programming (XP)

☑ Extreme Programming (XP) was founded on four core values:

➤ Communication

➤ Simplicity

➤ Feedback

➤ Courage

☑ Key principles of XP include:

- Continuous testing
- Simple coding
- Close interaction with the end users to build systems very quickly

Selecting the Appropriate Development Methodology

- ☑ Selecting a methodology is not simple, as no one methodology is always best.
- ☑ Many organizations have their own standards.
- ☑ The next figure summarizes some important methodology selection criteria.

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 |

Clarity of User Requirements

- ❑ RAD methodologies of prototyping and throwaway prototyping are usually more appropriate when user requirements are unclear as they provide prototypes for users to interact with early in the SDLC.

Familiarity with Technology

- ❑ If the system is designed without some familiarity with the base technology, risks increase because the tools may not be capable of doing what is needed.

System Complexity

- ❑ Complex systems require careful and detailed analysis and design.
- ❑ Project teams who follow iterative development methodologies tend to devote less attention to the analysis of the complete problem domain than they might if they were using other methodologies.

System Reliability

- ❑ System reliability is usually an important factor in system development.
- ❑ Throwaway prototyping-based methodologies are most appropriate when system reliability is a high priority.
- ❑ Prototyping-based methodologies are generally not a good choice as they lack careful analysis and design phases.

Short Time Schedules

- ☑ RAD-based methodologies are well suited for projects with short time schedules as they increase speed.
- ☑ Waterfall-based methodologies are the worst choice when time is essential as they do not allow for easy schedule changes.

Schedule Visibility

- ✓ RAD-based methodologies move many of the critical design decisions earlier in the project; consequently, this helps project managers recognize and address risk factors and keep expectations high.

Summary

- ☑ The Systems Development Lifecycle consists of four stages: Planning, Analysis, Design, and Implementation
- ☑ There are six major development methodologies: the waterfall method, the parallel development method, the phased development method, system prototyping, design prototyping, and agile development.
- ☑ There are five major team roles: business analyst, systems analyst, infrastructure analyst, change management analyst and project manager.