COMPUTER AIDED SOFTWARE ENGINEERING (CASE)

Dr. Ebadati

Kharazmi University

PLANNING

Chapter 1

THE SYSTEMS DEVELOPMENT LIFE CYCLE (SDLC)

THE SYSTEMS DEVELOPMENT LIFE CYCLE (SDLC)

- The **SDLC** is composed of **four** fundamental **phases**:
  - **Planning**
  - **Analysis**
  - **Design**
  - **Implementation**

SDLC

- **Each of the phases** include a **set of steps**, which **rely on techniques** that produce specific document files that provide **understanding** about the **project**.

SDLC

- To Understand the SDLC:
  - Each phase consists of steps that lead to specific deliverables
  - The **system** evolves through **gradual refinement**

STANDARD AND EVOLUTIONARY VIEWS OF SDLC

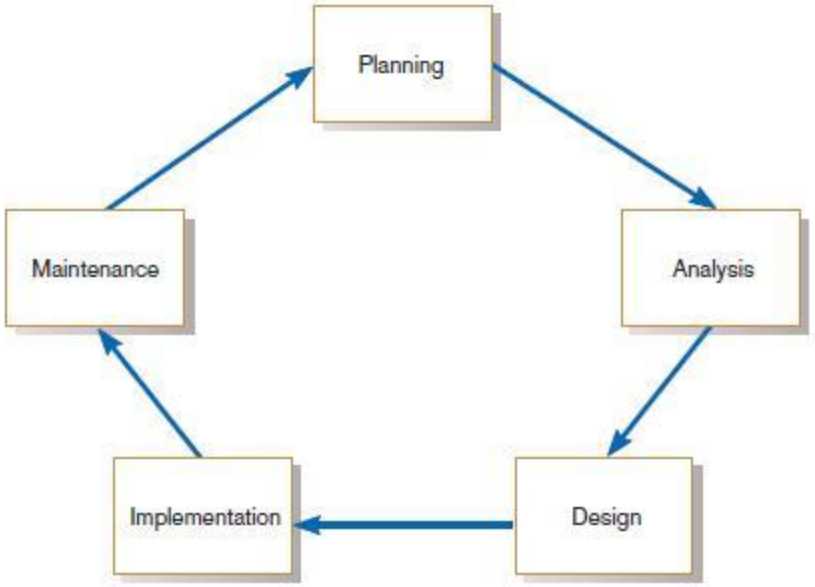


FIGURE 1-2
The systems development life cycle

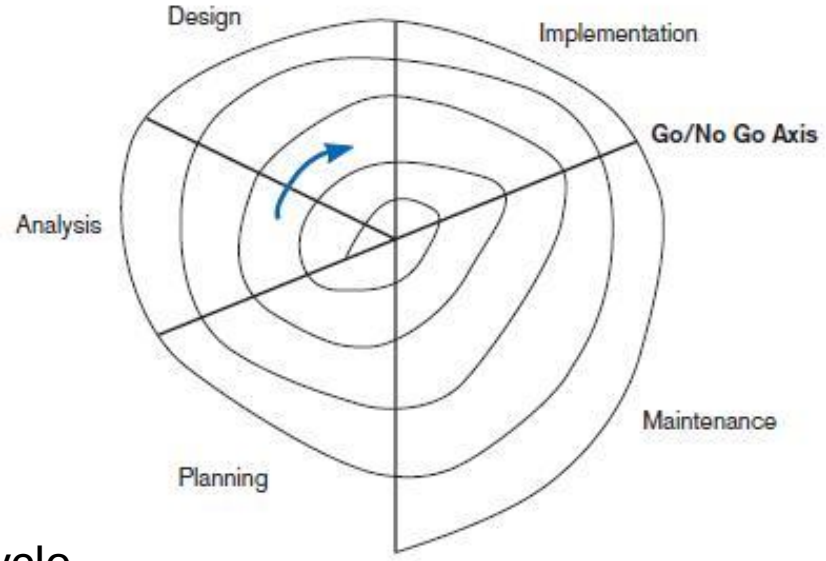


FIGURE 1-3 Evolutionary model

Phase I: Planning

PLANNING

- This phase is the **fundamental** process of **understanding why** an information system should be **built**.
- The Planning phase will also determine **how the project** team will go about **building** the information system.
- The Planning phase is composed of **two** planning **steps**.

Two Planning Steps

PLANNING

- **During project initiation**, the system's business value to the organization is identified (**How will it lower costs** or **increase revenues**?)
- During project management, the project manager **creates** a **work plan**, **staffs** the project, and puts **techniques** in place to help the **project team** control and direct the project **through** the entire **SDLC**.

Phase II: Analysis

ANALYSIS

- The analysis phase answers the questions of **who will use the system**, **what the system will do**, and **where and when it will be used**.
- During this phase the project team **investigates any current system**(s), **identifies improvement opportunities**, and **develops a concept** for the new system.
- This phase has **three analysis steps**.

THREE ANALYSIS STEPS

- **Analysis strategy:** This is developed to guide the **projects team's efforts**. This **includes** an **analysis of the current system**.
- **Requirements gathering:** The analysis of this information leads to the **development of a concept for a new system**. This concept is used to build a set of analysis models.
- **System proposal:** The proposal is **presented** to the **project sponsor** and other key individuals who decide **whether** the project should continue to **move forward**.

ANALYSIS

- The **system proposal** is the **initial** deliverable that describes what business **requirements** the **new system** should meet.
- The **deliverable from this phase** is both an **analysis and a high-level initial design for the new system.**

Phase III: Design

DESIGN

- In this phases it is decided **how the system will operate**, in terms of the **hardware**, **software**, and **network infrastructure**; the **user interface, forms**, and **reports** that will be used; and the **specific programs**, **databases**, and **files** that will be **needed**.

FIVE DESIGN STEPS

- **Design Strategy:** This clarifies whether the system will be **developed by the company or outside** the company.
- **Architecture Design:** This describes the **hardware**, **software**, and **network infrastructure** that will be used.
- **Database and File Specifications:** These documents define **what and where the data will be stored**.
- **Program Design:** Defines **what programs need to be written** and **what** they **will do**.

Phase IV: Implementation

IMPLEMENTATION

- During this phase, **the system is either developed or purchased** (in the case of packaged software).
- This phase is usually the **longest and most expensive** part of the process.
- The phase has **three steps**.

Three Implementation Steps

IMPLEMENTATION

- **System Construction:** The system is **built and tested** to make **sure** it **performs** as designed.
- **Installation:** Prepare to **support the installed system**.
- **Support Plan:** Includes a **post-implementation review**.

Systems Development Methodologies

SYSTEM DEVELOPMENT

- 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**.

PROCESS-CENTERED METHODOLOGIES

- With this methodology, the **focus is on defining the activities associated** with the **system**.
- The **concentration is on representing** the **system concept** as a set of **processes** with information flowing into and out of the processes.

Data-centered Methodologies

DATA-CENTERED

- This methodology **focuses** on **defining the content of the data** storage containers and how they are organized.
- Data-centered methodologies utilize **data** models as the **core of the system concept**.

Object-oriented Methodologies

OBJECT ORIENTED

- This methodology attempts to **balance the focus between processes and data**.
- The **Unified Modeling Language** (UML) is used to describe the **system concept** as a **collection of objects** incorporating both **data and processes**.

Category I of the System Development Methodology: Structured Design

1: SYS. DEV., STRUCTURE DESIGN

- 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 and follows a basic approach of two structured design categories.

Waterfall Development

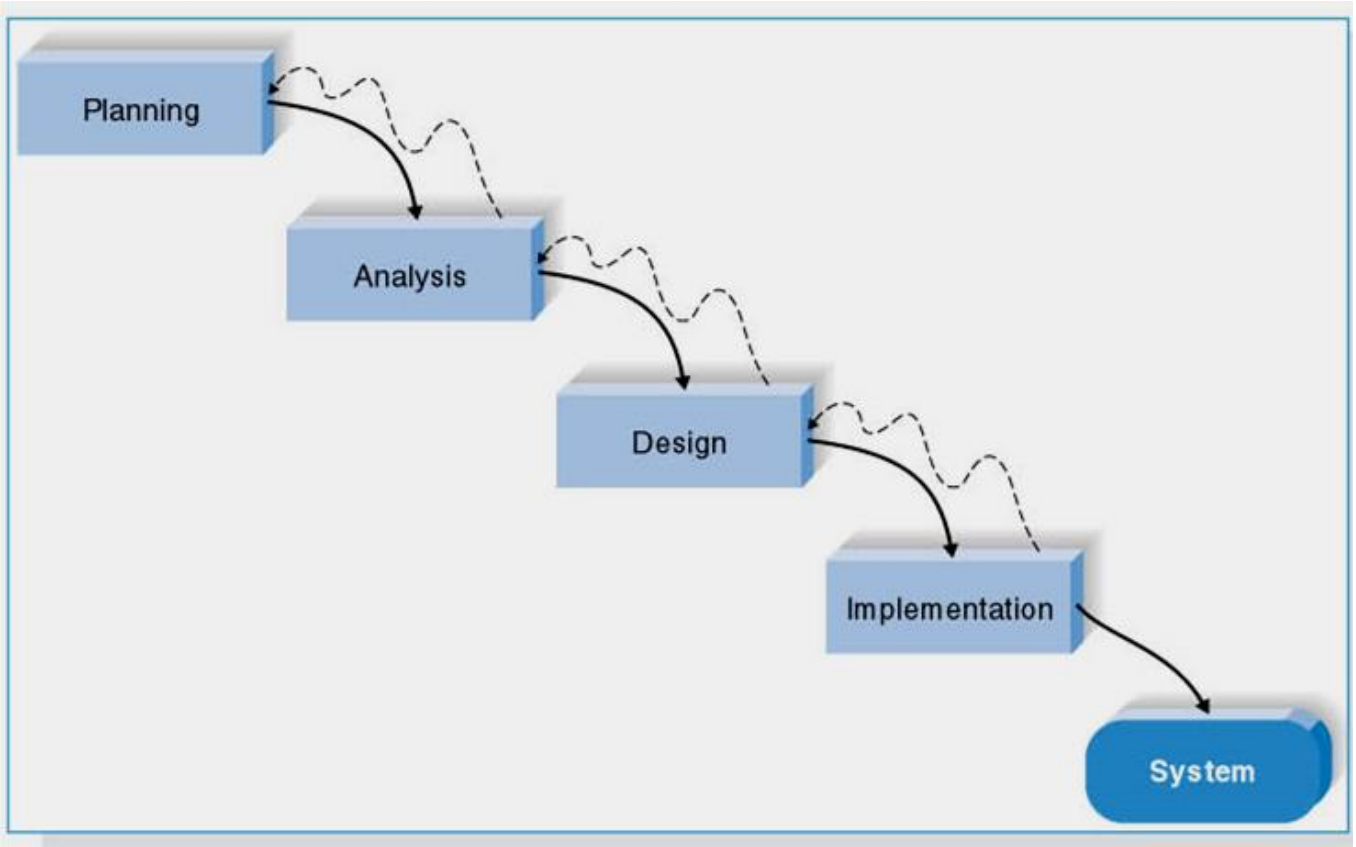
WATERFALL MODEL

- 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.

WATERFALL MODEL

- 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.

WATERFALL DEVELOPMENT-BASED METHODOLOGY

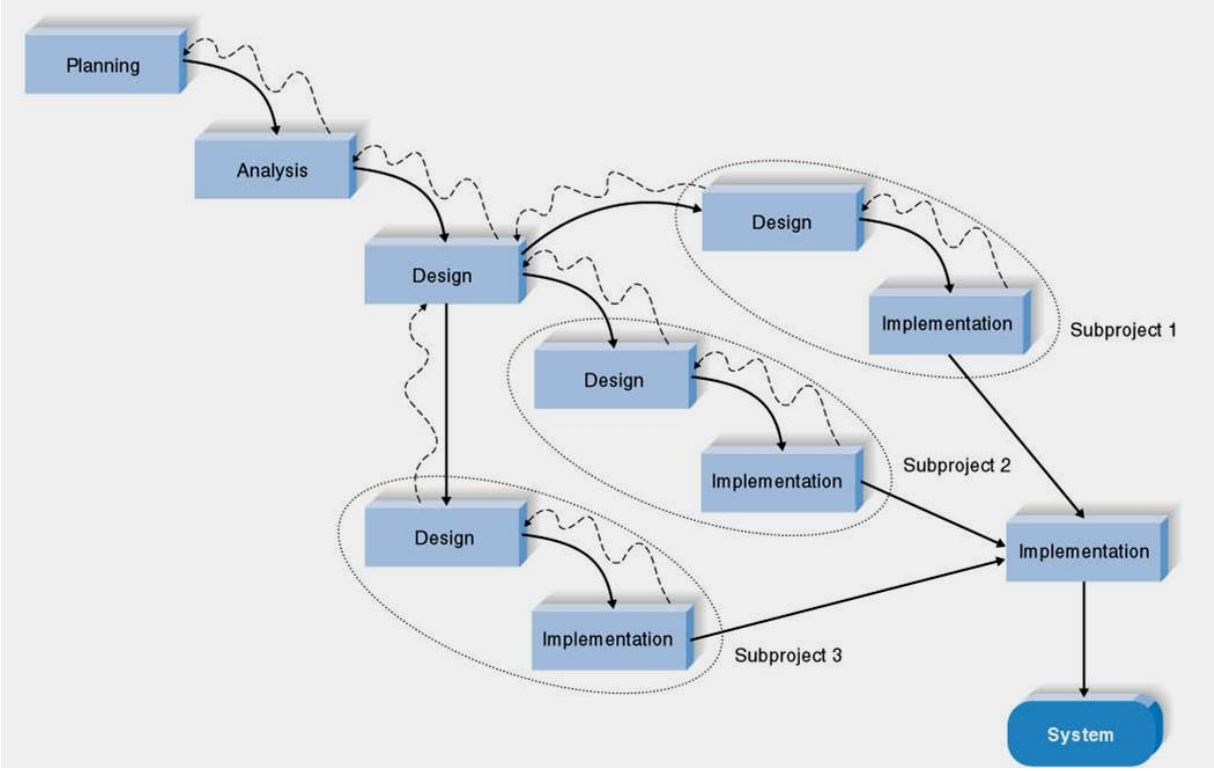


Parallel Development

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.



Category II of the System Development Methodology: Rapid Application Development (RAD)

RAPID APPLICATION DEVELOPMENT

- 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

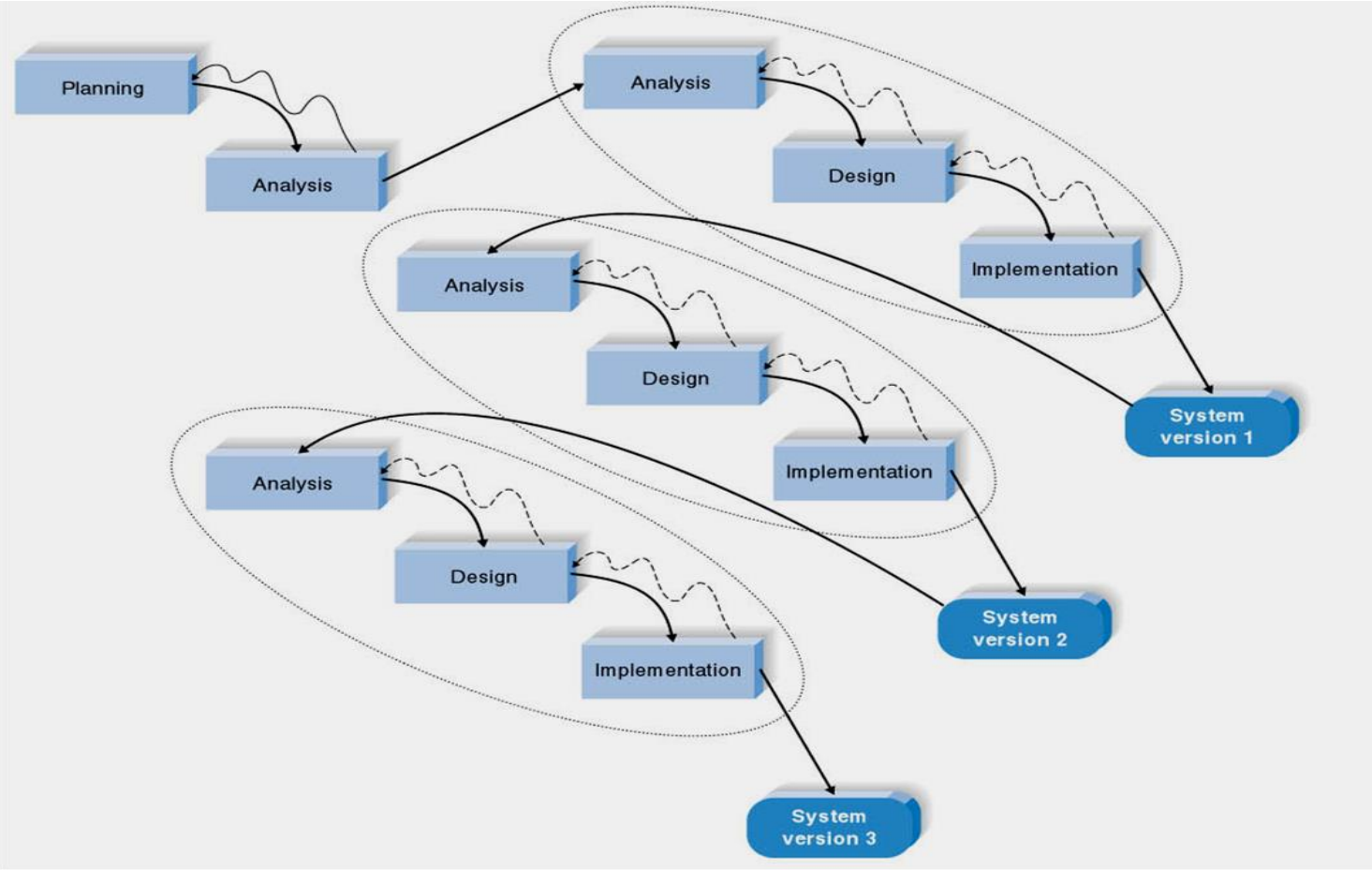
- One possible subtle **problem** with **RAD**-based methodologies is **managing user expectations**.

Phased Development

PHASED 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**.

PHASED DEVELOPMENT-BASED METHODOLOGY

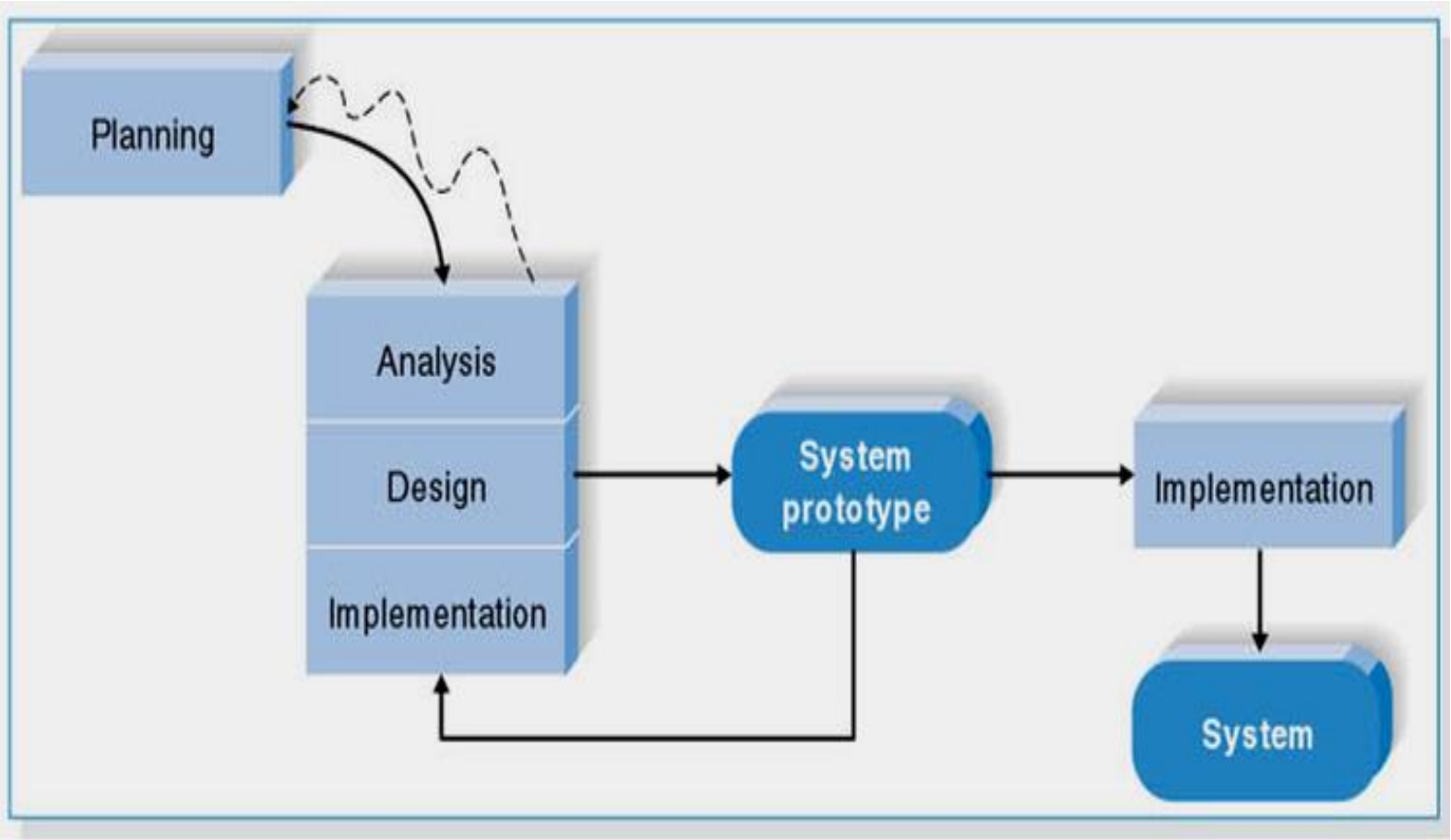


Prototyping

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**.

PROTOTYPING-BASED METHODOLOGY



PROTOTYPING-BASED METHODOLOGY

- **Advantage:** **Provides a system** for the users to interact with, **even if** it is **not** initially **ready** for use.

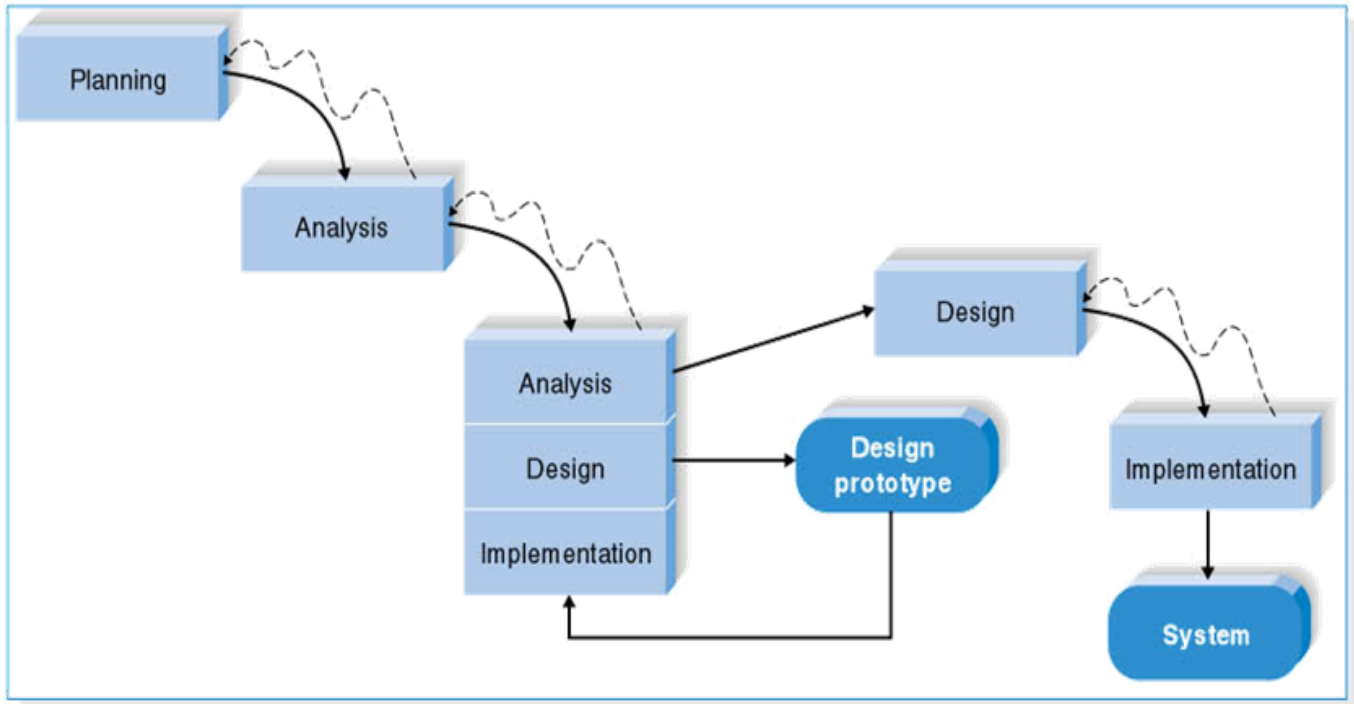
- **Disadvantage:** **Often** the prototype undergoes such significant changes that many **initial design** decisions prove to be **poor ones**.

Throwaway Prototyping

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**.

THROWAWAY PROTOTYPING-BASED METHODOLOGY



Category III of the System Development Methodology: Agile Development

AGILE DEVELOPMENT

- 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.

Extreme Programming (XP)

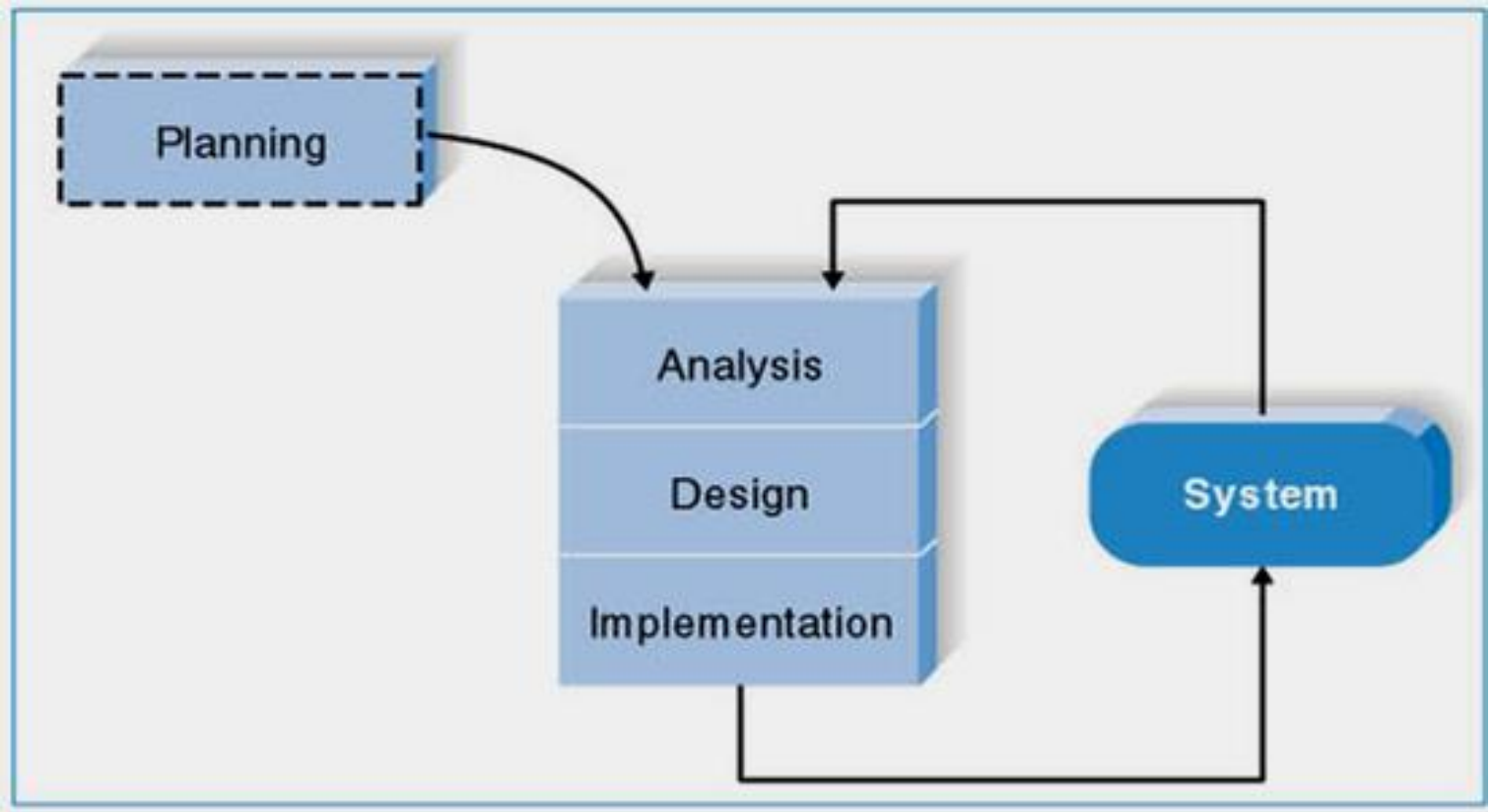
EXTREME PROGRAMMING (XP)

- Extreme Programming (XP) was founded on **four core values**:
  - **Communication**
  - **Simplicity**
  - **Feedback**
  - **Courage**

EXTREME PROGRAMMING (XP)

- **Key principles of XP** include:
  - **Continuous testing**
  - **Simple coding**
  - **Close interaction with the end users** to build systems very quickly

AN EXTREME PROGRAMMING-BASED METHODOLOGY



Selecting the Appropriate Development Methodology

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

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

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 phased** development-based 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.

PROJECT TEAM SKILLS AND ROLES

- **Projects** should consist of a **variety** of **skilled** individuals in order for a system to be successful.
- **Six major skill** sets an analyst should have include:
  - **Technical**
  - **Business**
  - **Analytical**
  - **Interpersonal**
  - **Management**
  - **Ethical**

CATEGORIES OF ANALYSTS

- Business Analyst
- Systems 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

- The **S**ystems **D**evelopment **L**ife **C**ycle 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.**

موفق باشید