8 DEVELOPMENT MODELS

Member

1. Nguyễn Quang Linh
2. Nguyễn Văn Tài
3. Vũ Đình Công
4. Trần Minh Đoan

List of Contents

1.Introduction

Definition: A software development model is a systematic set of processes and principles that guide and manage the entire software life cycle from concept to product completion.

Purpose:

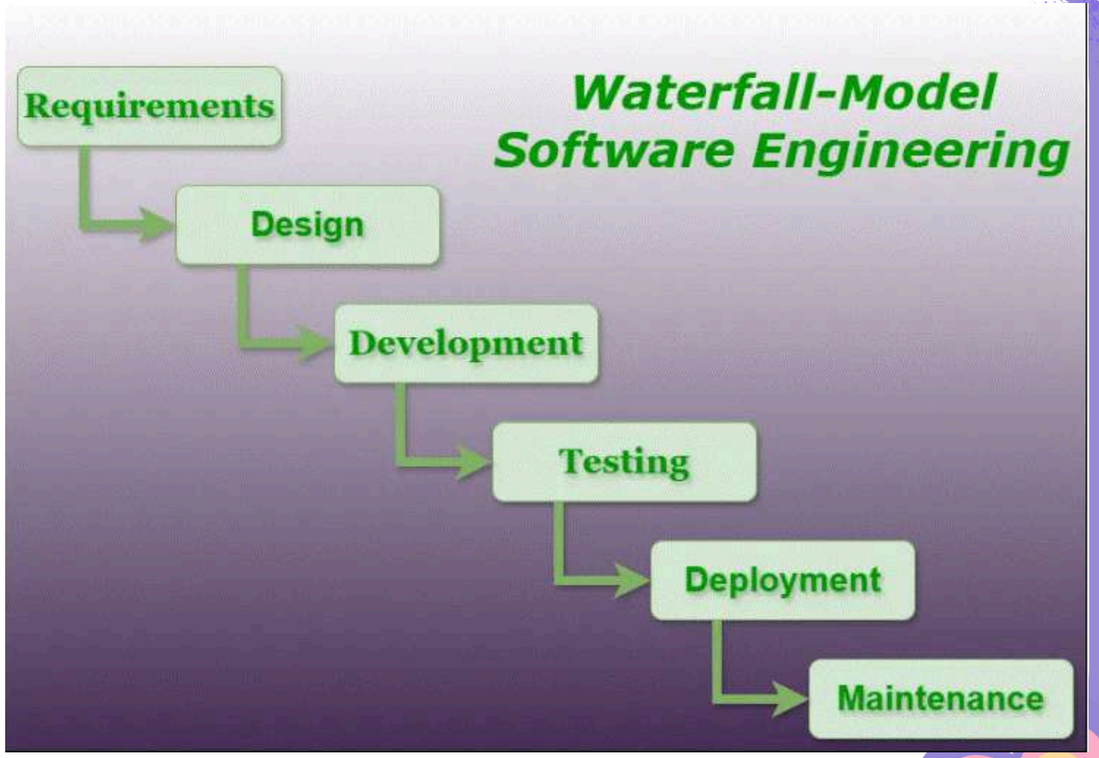
- Helps ensure that the development process is carried out in a planned, scientific and highly effective manner.
- Optimize costs, resources and time.

Importance:

- Effectively manage progress and resources.
- Guide the testing process.
- Reduce the risk of major modifications in later stages.

2. Waterfall Model

Definition: Waterfall is a linear software development model in which each phase must be completely completed before moving on to the next phase. This model is considered a “waterfall” because the work flows from top to bottom, without going back to previous steps.



Advantages:

- Clear, simple and easy to manage.
- Easily applied to small projects with clear requirements.
- Each phase has specific documentation, making it easy to track progress.

Disadvantages:

- Inflexible
- Late error detection
- Not suitable for complex projects or those with unclear requirements from the beginning.

When to use:

- Small or medium projects with fixed and clear requirements from the beginning.
- Projects that do not require much interaction with the customer during development.
- When you need to fully document each stage of the process.

3. V-Model

Definition: V-Model is a variation of the Waterfall model, in which each development phase is directly linked to a corresponding testing phase. The model emphasizes the importance of Verification and Validation throughout the software development life cycle.

Phases

Verification Phase:

- Requirements Analysis
- System Design
- Architectural Design
- Module Design
- Coding Phase

Validation Phase:

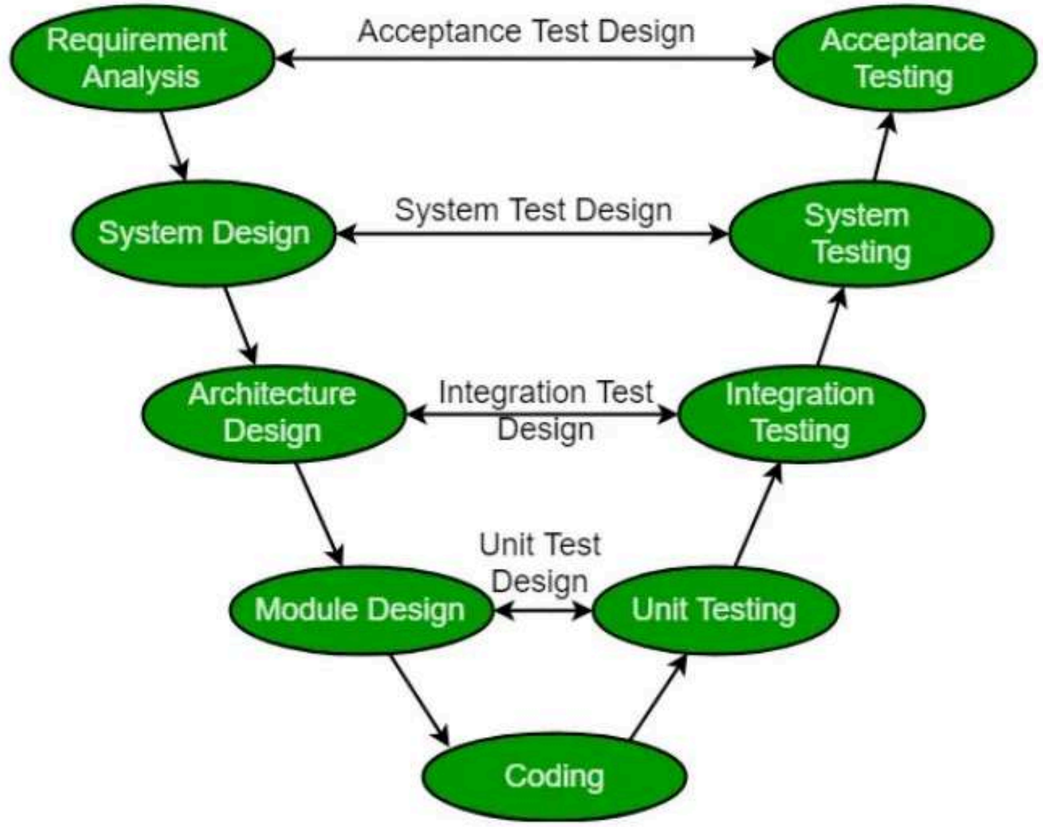
- Unit Testing
- Integration Testing
- System Testing
- User Acceptance Testing (UAT)

Design Phase:

- Requirements Analysis
- System Design
- Architectural Design
- Modular Design

Testing Phases:

- Unit Testing
- Integration Testing
- System Testing
- User Acceptance Testing (UAT)



Advantages:

- Early error detection
- Good control
- Improved product quality

Disadvantages:

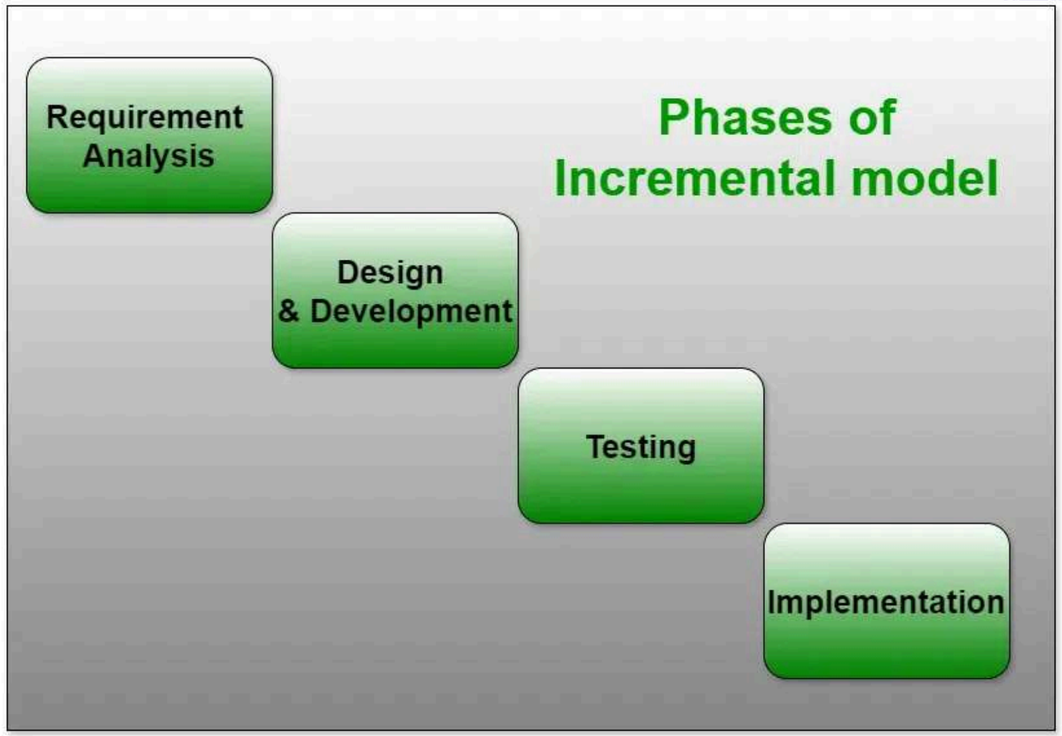
- Lack of flexibility
- High cost
- Depends on initial requirements

When to use:

- Small or medium projects with fixed and clear requirements from the beginning.
- Projects that do not require much interaction with the customer during development.
- When you need to fully document each stage of the process.

4. Incremental Model

Definition: The Incremental Model is a software development model in which the product is built and improved through small versions (increments). Each increment provides a complete piece of functionality, which can operate independently and is gradually added to the final product.



Advantages:

- Early feedback
- Reduced risk
- Flexible
- Easy management

Disadvantages:

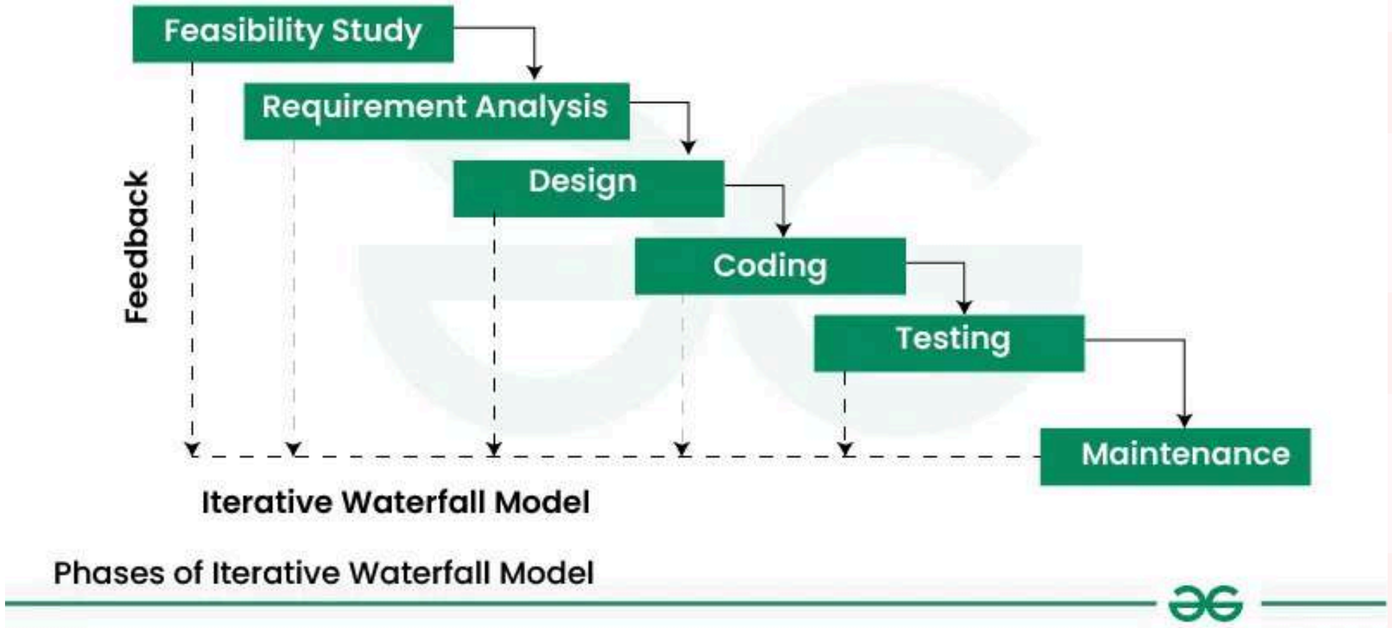
- Time-consuming to manage
- High cost
- Difficulty in defining the entire requirements

When to use:

- When a part of the product needs to be released early to meet urgent requirements.
- Large, complex projects that are likely to have changing requirements over time.
- Projects that require the gradual integration of new features in stages.

5. Iterative Model

Definition: The Iterative Model is an iterative software development model in which a product is built over multiple iterations. Each iteration provides an improvement or additional feature, based on the results of the previous iteration and user feedback.



Phases of Iterative Waterfall Model

Advantages:

- Rapid feedback
- Risk reduction
- Flexibility
- Continuous improvement

Disadvantages:

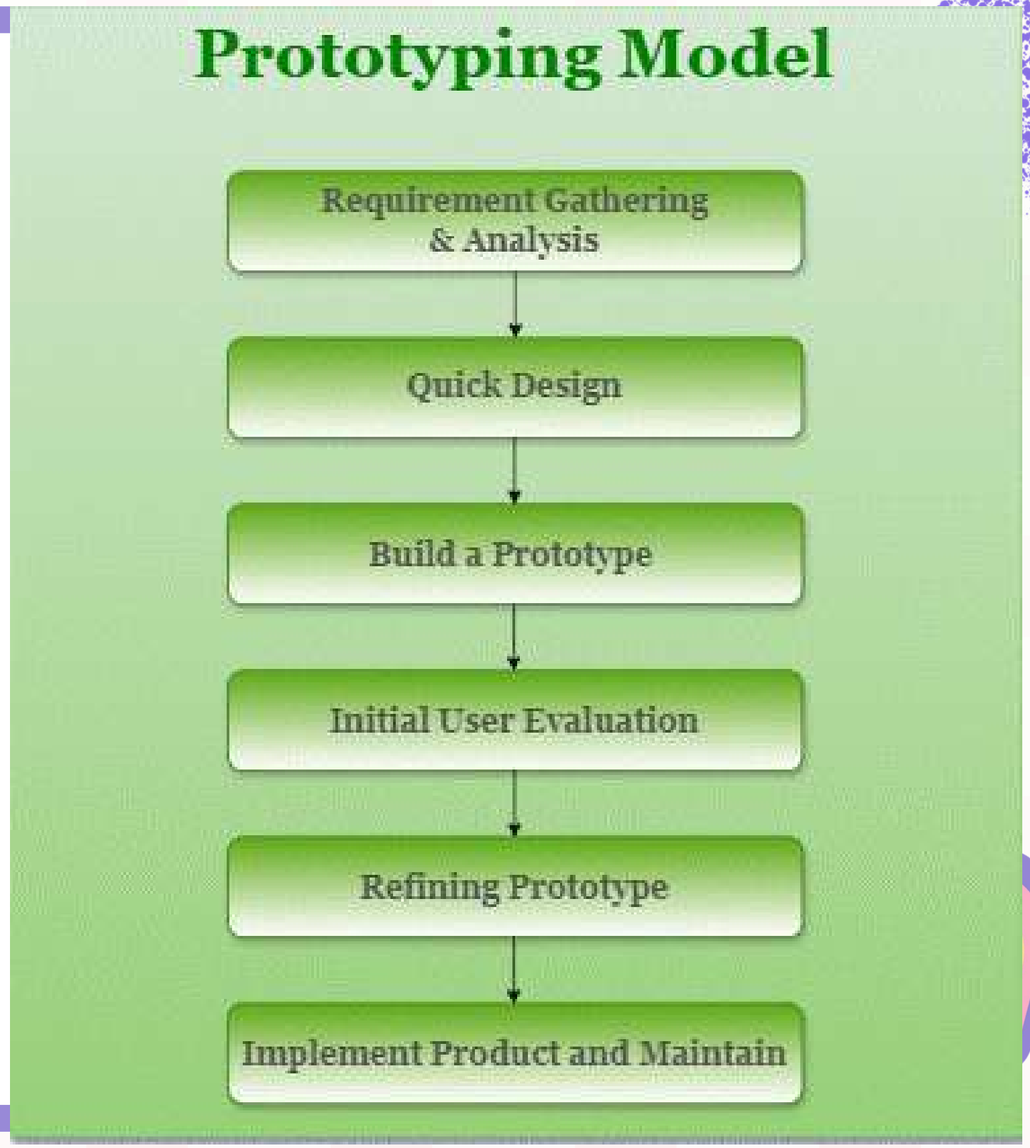
- Dependent on initial requirements
- Expensive resources.
- Complex management

When to use:

- Large and complex projects where it is difficult to fully define requirements at the outset.
- Projects that require continuous improvement based on customer or user feedback.
- When it is necessary to release a product with basic features first and add advanced features later.

6. Prototype Model

Definition: Prototype Model is a software development model in which a prototype of a system is built, tested, and improved based on user feedback before developing the final product. A prototype is a simplified version, focusing on simulating the main functions or interfaces of the system.



Advantages:

- Understand requirements
- Reduce risk
- Increase user engagement

Disadvantages:

- Time-consuming and costly
- Risk of drift
- Lack of focus on performance

When to use:

- When requirements are unclear or difficult to define at the outset.
- When the customer or user is unfamiliar with the technology or the intended interface.
- The project requires flexibility and close collaboration with users during development.

7. Spiral Model

Definition: The Spiral Model is a software development model that focuses on risk management, combining elements of the Waterfall and Iterative models. The model divides the development process into several spiral cycles, where each loop represents a phase of development or improvement.

Phases

- Objectives Defined
- Risk Analysis
- Engineering
- Evaluation
- Planning

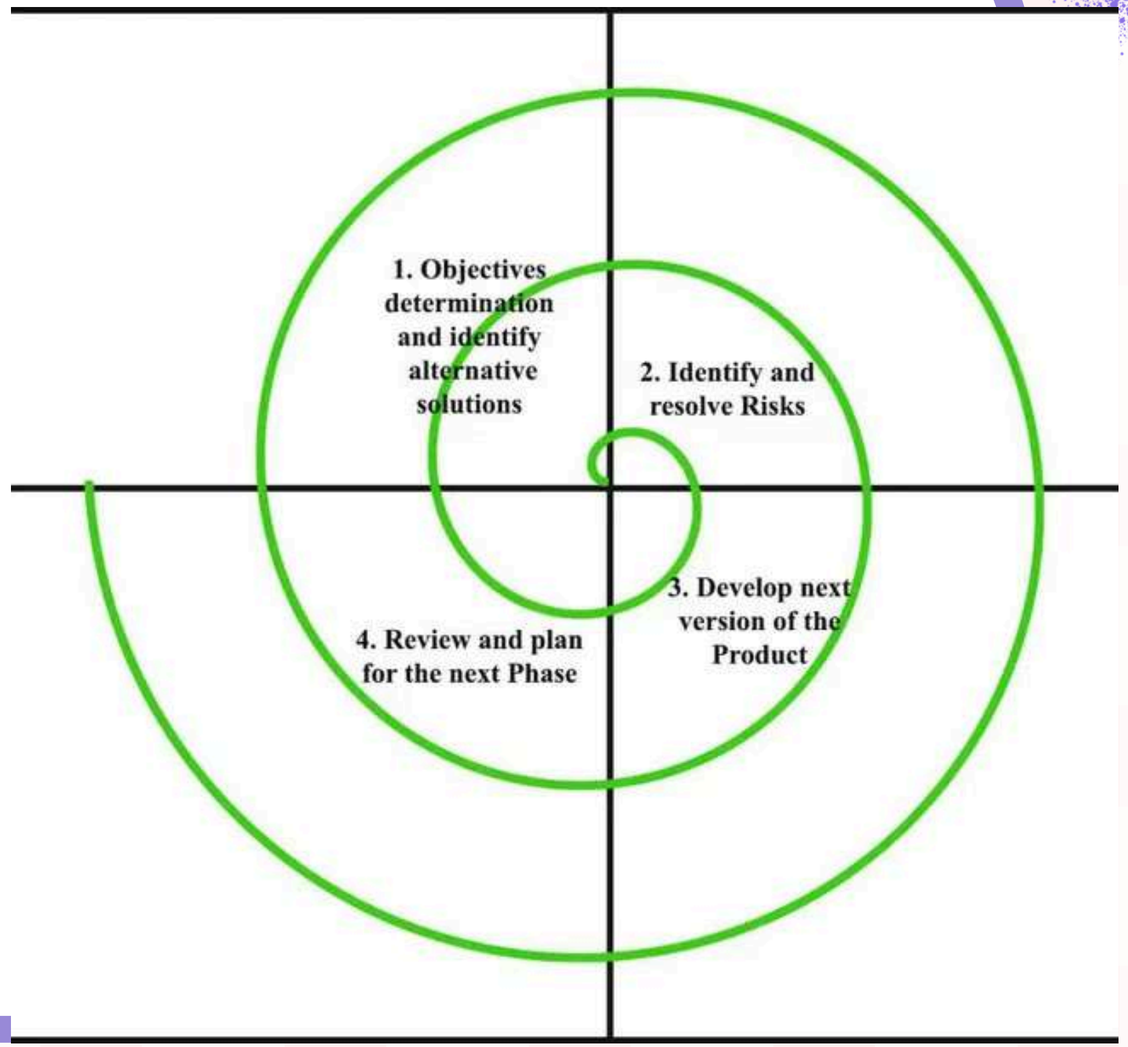
Each phase of the Spiral Model is divided into four quadrants as shown in the figure. The functions of these four quadrants are discussed below:

1. Determine and Identify Alternative Solutions: Gather customer requirements, identify and analyze objectives, and propose alternative solutions for each phase.

2. Identify and Resolve Risks: Evaluate solutions, select the best one, identify associated risks, and resolve them. Build the prototype for the chosen solution.

3. Develop the Next Version of the Product: Develop and test identified features, leading to the next software version.

4. Review and Plan for the Next Phase: Customers evaluate the current version, and planning for the next phase begins.



Advantages:

- Risk Outcome Management
- Flexible
- Continuous Improvement

Disadvantages:

- High cost
- Management complexity
- Difficult to apply to small projects

When to use:

- Large, complex projects with constantly changing requirements.
- When risks are high or requirements are not clearly defined.
- Projects require flexible development and continuous improvement based on feedback.

8. Agile Model

Definition: Agile is a software development methodology that focuses on delivering small pieces of software in short cycles called sprints. Each sprint develops and refines a piece of software, with ongoing customer and user involvement. The model promotes collaboration between development teams, end users, and stakeholders, and is flexible to changing requirements throughout the development process.

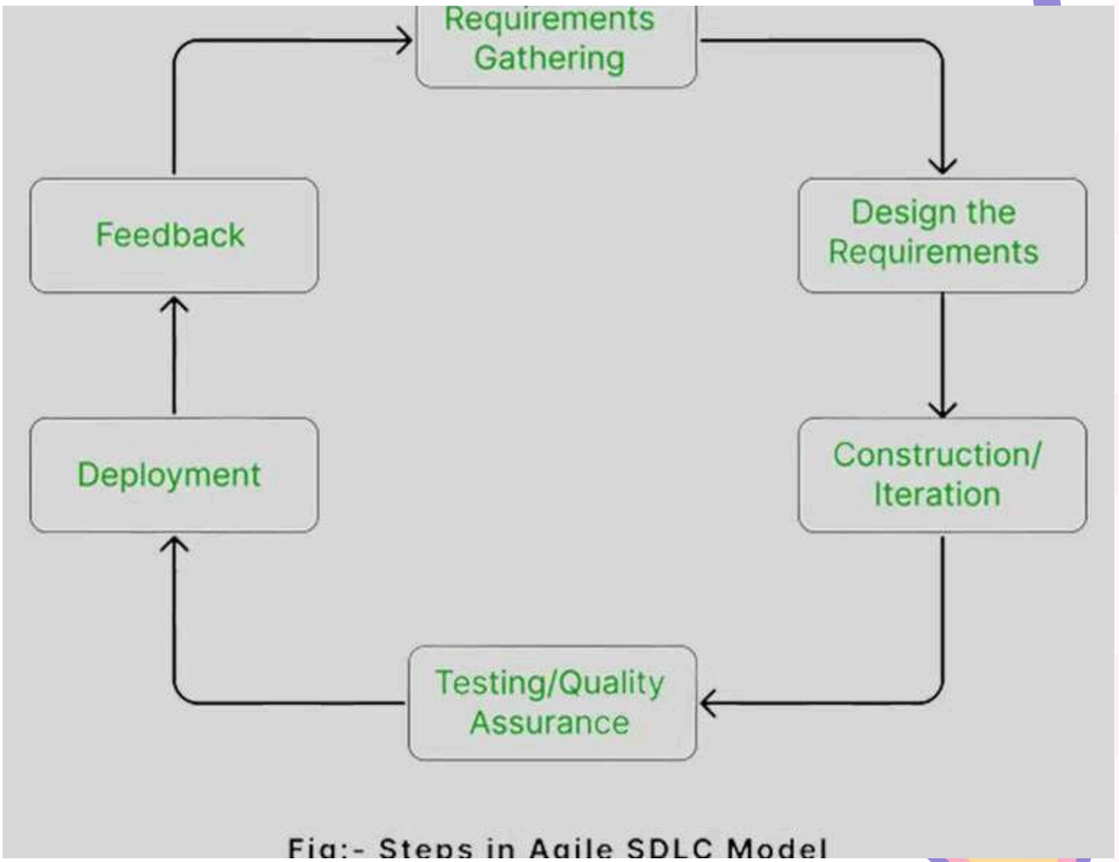


Fig:- Steps in Agile SDLC Model

Advantages:

- Highly flexible
- Continuous improvement
- Rapid feedback
- Increased collaboration

Disadvantages:

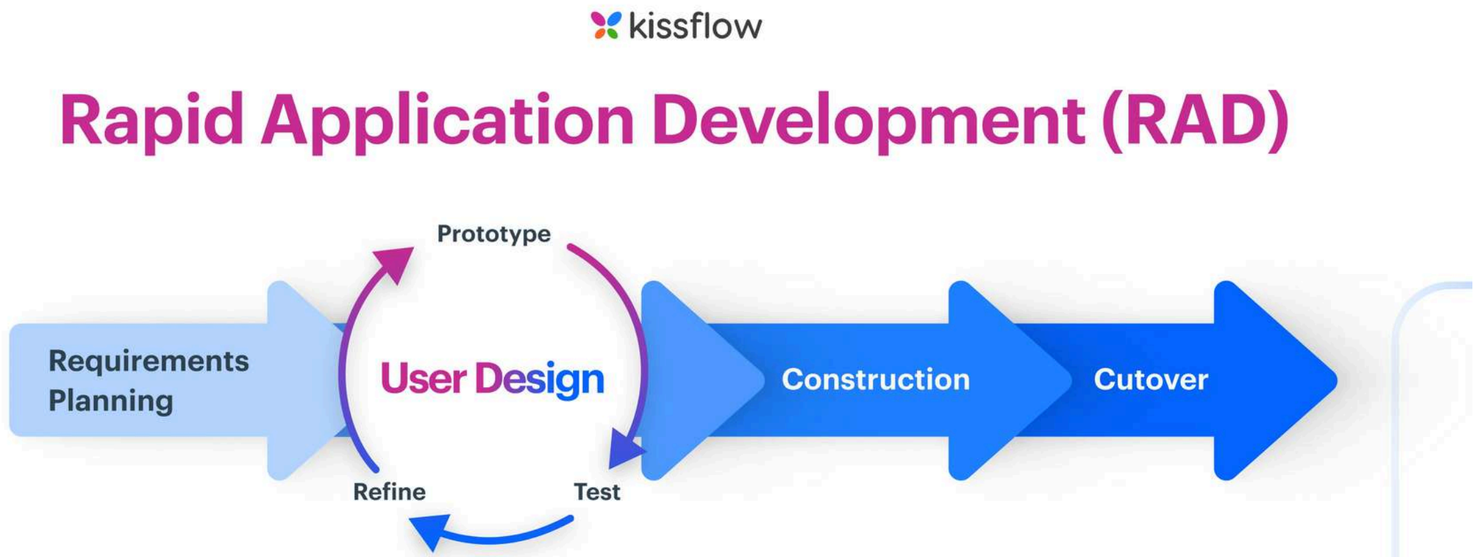
- Difficulty in managing large projects
- Need for good coordination between teams
- Difficulty in pre-determining costs and time

When to use:

- Projects that require constant change or are unclear from the outset.
- Projects that require rapid development and frequent involvement of the customer or end user.
- Projects that are creative, innovative, and require a high degree of flexibility.

9. RAD Model

Definition: Rapid Application Development (RAD) is a software development methodology that focuses on reducing development time and maximizing the use of automation tools to build products. RAD focuses on rapid, agile development through short iterations and collaboration among stakeholders.



Advantages:

- Short development time
- Rapid user feedback
- Agile
- Parallel development

Disadvantages:

- Quality is not always guaranteed
- Need for powerful tools and resources
- Difficulty in scaling projects, especially when a large system and complex integration are required.

When to use:

- When requirements change rapidly and a software product is needed in a short time.
- The project has fairly clear and not too complex requirements and can be developed quickly.
- When there is frequent and continuous participation of customers or end users to provide feedback.
- The project requires rapid software development and needs to apply automation development tools to optimize time.

Choose a software development model: Incremental Model

- Meet the need for flexibility
- Release early and improve gradually
- Manage risks effectively
- Adhere to functional and non-functional requirements
- Support long-term development

Thank You

TA

SNEAKER

TASNEAKER.VN