SDLC Process Models

Software Development Life Cycle (SDLC) process models are frameworks or methodologies that guide the development of software applications. These models provide a structured approach to software development, helping teams plan, design, build, test, and maintain software systems. There are several SDLC process models, each with its own set of principles and practices.

Waterfall Model:

- The Waterfall model is a linear and sequential approach to software development.

- It divides the project into distinct phases, such as requirements, design, implementation, testing, deployment, and maintenance.

- Each phase must be completed before the next one begins, and there is minimal room for changes once a phase has started.

Agile Model:

•Agile methodologies, including Scrum, Kanban, and Extreme Programming (XP), emphasize flexibility, collaboration, and iterative development.

•Agile teams work in short iterations, typically 2-4 weeks, and deliver small, incremental releases of the software.

•Customer feedback is incorporated continuously, allowing for changes to be made throughout the project.

Iterative Model:

•The Iterative model involves repeating cycles or iterations of the software development process.

•Each iteration builds on the previous one, adding new features or improvements.

•It allows for flexibility and adaptation to changing requirements.

Spiral Model:

•The Spiral model combines iterative development with risk analysis.

•It divides the project into cycles, each consisting of planning, risk analysis, engineering, and evaluation phases.

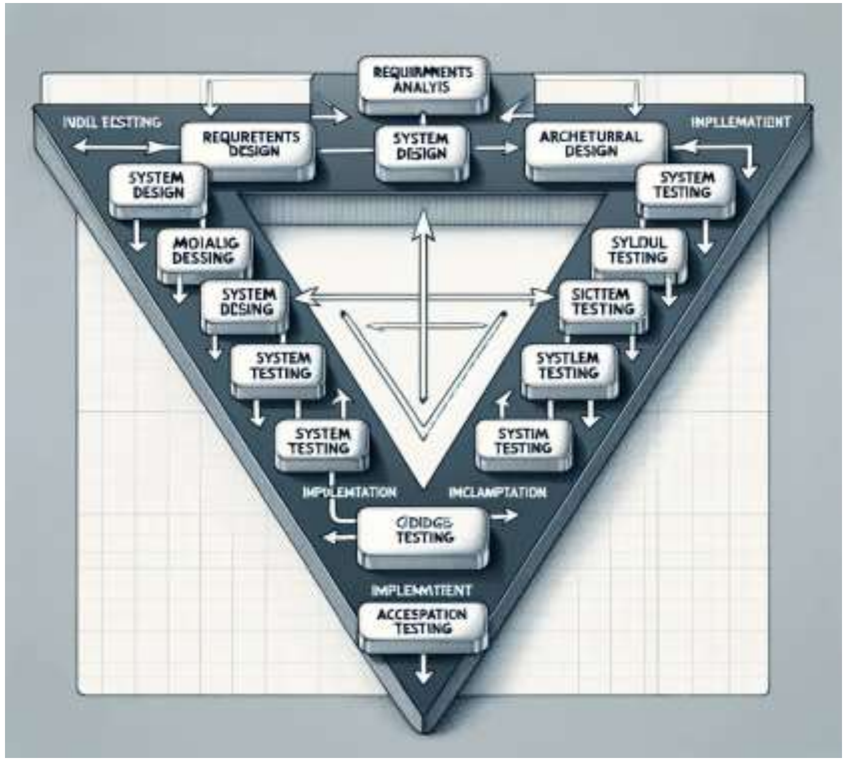
•The process repeats until the software is complete, with each iteration reducing project risks.

V-Model (Validation and Verification Model):

- The V-Model is an extension of the Waterfall model.

- It places a strong emphasis on testing and validation at each stage of development.

- For every development phase, there is a corresponding testing phase, creating a V-shaped diagram.



Big Bang Model:

- The Big Bang model is an informal and unstructured approach to software development.

- Developers start coding without a defined plan or requirements.

- This model is risky and not commonly used in professional software development due to its lack of structure.

Rapid Application Development (RAD):

•RAD is a model that prioritizes rapid prototyping and quick development.

•It focuses on user involvement and feedback to accelerate the development process.

•RAD often relies on pre-built components and tools to speed up development.

Incremental Model:

- The Incremental model divides the project into smaller, manageable parts or increments.

- Each increment is developed separately and can be tested and delivered independently.

- This approach allows for partial deployment and early user feedback.

Waterfall model

The Waterfall Model is a traditional software development methodology that is characterized by a linear and sequential approach to software development.

It was first introduced by Dr. Winston W. Royce in a paper published in 1970.

The Waterfall Model is one of the earliest and most structured approaches to software development and has been widely used in the past, although it has been largely replaced by more flexible and iterative methodologies in recent years.

The Waterfall Model consists of several distinct phases, and each phase must be completed before moving on to the next one:

1.Requirements Gathering and Analysis: In this initial phase, the project team works closely with stakeholders to gather and document all project requirements. The goal is to have a clear and complete understanding of what the software needs to accomplish.

2.System Design: Once the requirements are established, the system design phase begins. During this phase, the system architecture and design specifications are created. It includes designing the software's overall structure and defining data models and algorithms.

3.Implementation (Coding): In this phase, developers write the actual code based on the design specifications. This is where the software is built according to the requirements and design.

4.Testing: After the implementation phase, testing is conducted to identify and fix any defects or issues. This phase ensures that the software meets the specified requirements and functions as expected.

5. **Deployment (Installation):** Once the software has passed testing and is considered complete, it is deployed to the production environment. This phase involves installing the software on users' systems and making it available for use.

6. **Maintenance and Support:** In the final phase, the software is maintained and updated as needed to address issues, add new features, or make improvements. This phase can continue indefinitely throughout the software's lifecycle.

One of the main criticisms of the Waterfall Model is its inflexibility.

It assumes that all requirements can be gathered and defined upfront and that changes are difficult and expensive to make once the project is underway.

This can lead to problems if requirements change or if issues are discovered late in the development process.

Thank You