**Practical-1**

* **Introduction to Software Engineering**

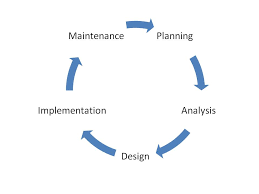
Software engineering is a systematic approach to the development, operation, maintenance, and retirement of software systems. It applies engineering principles to software development, ensuring quality, efficiency, and scalability. The field emphasizes a disciplined approach to problem-solving and project management to meet stakeholder requirements.

* **What is SDLC ?**

Software development life cycle (SDLC) is a structured process that is used to design, develop, and test good-quality software. SDLC, or software development life cycle, is a methodology that defines the entire procedure of software development step-by-step.

The goal of the SDLC life cycle model is to deliver high-quality, maintainable software that meets the user’s requirements.

SDLC in software engineering models outlines the plan for each stage so that each stage of the software development model can perform its task efficiently to deliver the software at a low cost within a given time frame that meets users requirements.



## **The stages of SDLC are as follows:**

The five phases of the Software Development Life Cycle (SDLC) are:

1. **Planning**

In this phase, the team gathers requirements from the customer, analyzes costs, and creates a software requirements specification document.

**Objective** : Define the project's purpose, scope, and goals

1. **Design**

The program developer reviews the software to ensure it meets the end-user's requirements and is feasible for the customer. The design phase often includes the development of a prototype model.

**Objective** : Create a framework that meets the requirement.

1. **Implementation**

The scope of work is defined, such as requirements and design.

**Objective** : Translate the design into code to make it understandable for computers.

1. **Testing**

Quality Assurance (QA) specialists ensure the software is bug-free and can run smoothly.

**Objective** : Ensure a software that is well planned and meets the customer requirement.

1. **Maintenance**

Developers and support teams monitor the software in production, address issues, make updates, and enhance the software.

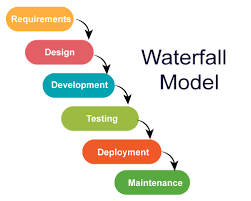
**Objectiv**e : Update the software as needed to meet changing user needs and technology

* **Waterfall Model**

The Waterfall Model was the first Process Model to be introduced. It is also referred to as a **linear-sequential life cycle model**. It is very simple to understand and use. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases.

The Waterfall model is the earliest SDLC approach that was used for software development.

The waterfall Model illustrates the software development process in a linear sequential flow. This means that any phase in the development process begins only if the previous phase is complete. In this waterfall model, the phases do not overlap.



## **The stages of Waterfall Model are as follows:**

## **Stage 1: Planning and requirement analysis:**

* The team brainstorms to understand the project’s goals and feasibility.
* They conduct a feasibility study to evaluate the potential impact and benefits.
* Resources, timeline, and budget estimates are drawn up.
* Potential risks are identified, and mitigation plans are created.

**Stage 2: Defining Requirements:**

* Detailed functional and non-functional requirements are gathered from stakeholders.
* These requirements are documented clearly to serve as a reference for all team members.
* Stakeholders review and approve the requirements document.
* Any ambiguities or changes are resolved and updated in the document.

**Stage3: Designing the Software:**

* The system architecture is designed, outlining the components and their interactions.
* Detailed design specifications, including data flow diagrams and database structures, are created.
* Interface designs and mockups are developed to visualize the user experience.

**Stage 4: Developing the project:**

* Developers start writing the code based on the design documents.
* They follow coding standards and best practices to ensure code quality.
* Regular code reviews and version control practices are maintained.
* Modules are integrated incrementally, ensuring compatibility and functionality.

**Stage 5: Testing**:

* Test cases are created to cover all possible scenarios and functionalities.
* Unit testing is performed to check individual components for defects.
* Integration testing ensures that modules work together as expected.
* Any bugs found are reported, fixed, and retested to ensure the software is error-free.

**Stage 6: Deployment:**

* The software is prepared for release, including packaging and configuration.
* It is installed in the production environment for end-users to access.
* Data migration from old systems, if needed, is performed.
* User training and documentation are provided to help users understand and use the software effectively.

**Stage 7: Maintenance**

* The software is continuously monitored for performance issues and user feedback.
* Regular updates and patches are released to fix bugs and improve functionality.
* New features or enhancements are developed based on changing user needs.
* Any issues that arise are promptly addressed to ensure the software remains reliable and effective.
* **Advantages of Waterfall Method:**

1. There are three main benefits of using the waterfall method. The first is clarity. Because the scope is defined up front, clearly documented, and signed off, the process of running the project is relatively straightforward and measured.
2. It also means that costs are easier to contain, timelines more readily met throughout, and few, if any, unwelcome surprises can derail the project. That leads to the second advantage: predictability. Because you keep changes to a minimum, additional costs incurred to fix and alter designs remain low too.
3. The third advantage is accountability. The structured approach means that everyone involved understands exactly who need to do what and by when. The detailed documentation helps here by providing a useful reference for team members.

* **Disadvantages of Waterfall Method:**

The waterfall model is the oldest paradigm for software engineering. However, over the past three decades, criticism of this process model has caused even ardent supporters to question its efficacy. Among the problems that are sometimes encountered when the waterfall model is applied are:

1. Real projects rarely follow the sequential flow that the model proposes. Although the linear model can accommodate iteration, it does so indirectly. As a result, changes can cause confusion as the project team proceeds.
2. It is often difficult for the customer to state all requirements explicitly. The waterfall model requires this and has difficulty accommodating the natural uncertainty that exists at the beginning of many projects.
3. The customer must have patience. A working version of the program(s) will not be available until late in the project time span. A major blunder, if undetected until the working program is reviewed, can be disastrous.

* **Project Definition:**

I am assuming myself as a Software Analyst/Project developer/Manager for creating a software system for **an online job application portal** that allows users to apply for jobs, take skill tests, attend interviews, and receive job offers.