

SOFTWARE ENGINEERING

UNIT – 1

TOPIC – 5

PROCESS MODELS – SDLC, WATERFALL MODEL, INCREMENTAL PROCESS MODEL

1. Introduction to Software Process Models

- **Software Process:**

A software process refers to the set of activities, methods, practices, and tools used to develop and maintain software. It involves all the necessary steps to create a software product, including planning, designing, coding, testing, and maintenance.

Example: When building a mobile banking app, the process begins with understanding user needs, designing the app, writing the code, testing the app, and finally launching it.

- **Software Process Model:**

A software process model is a structured approach or plan that outlines how the software development process will be carried out. It helps organize and manage the steps needed to ensure that all important activities are completed in an orderly manner.

Example: For the banking app, a process model helps decide whether to use a method like Waterfall or Agile, depending on the project's requirements.

2. Evolution of Software Development: From Frameworks to Process Models

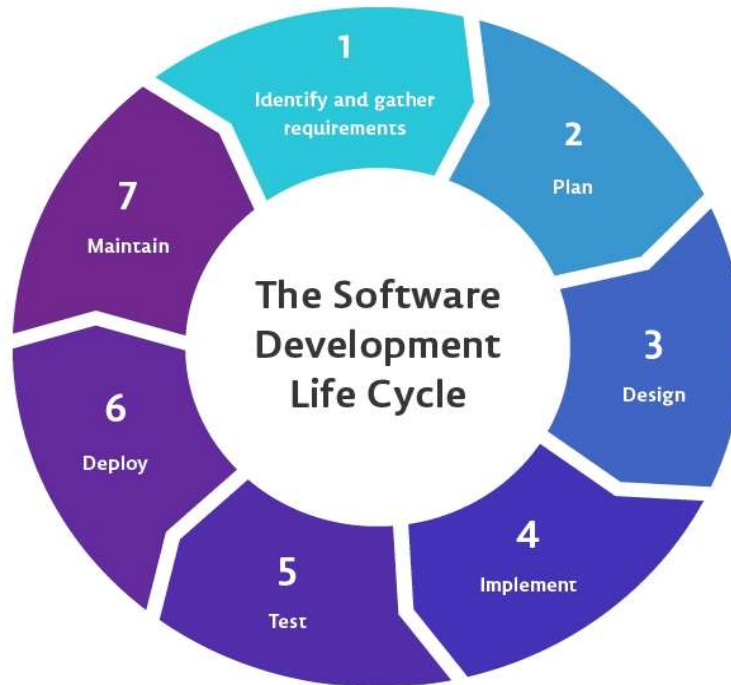
- **Basic Framework:**

The starting point of any software project involves a basic framework that includes essential steps such as talking to users, planning, designing, building, testing, and releasing the software.

Example: In the banking app, the framework involves gathering user needs, designing the app, coding, testing, and finally launching the app.

- **Software Development Life Cycle (SDLC):**

SDLC is an evolved form of the basic framework that introduces specific stages with a clear order for execution, making the software development process more organized.



Stages of SDLC:

- **Requirement Gathering:** Understanding what users need.
- **Design:** Planning how the software will function.
- **Implementation:** Writing the code for the software.
- **Testing:** Ensuring the software works correctly.
- **Deployment:** Releasing the software to users.
- **Maintenance:** Updating and fixing the software after release.

Example: The SDLC stages ensure that the banking app is developed in a structured and organized way, from gathering requirements to maintaining the software post-launch.

3. Overview of Process Models

- **Process Models:**

Process models are like roadmaps that guide the software development process step-by-step. They evolved from the basic SDLC framework to address specific needs of software projects, making the development process easier, more organized, or flexible depending on the project requirements.

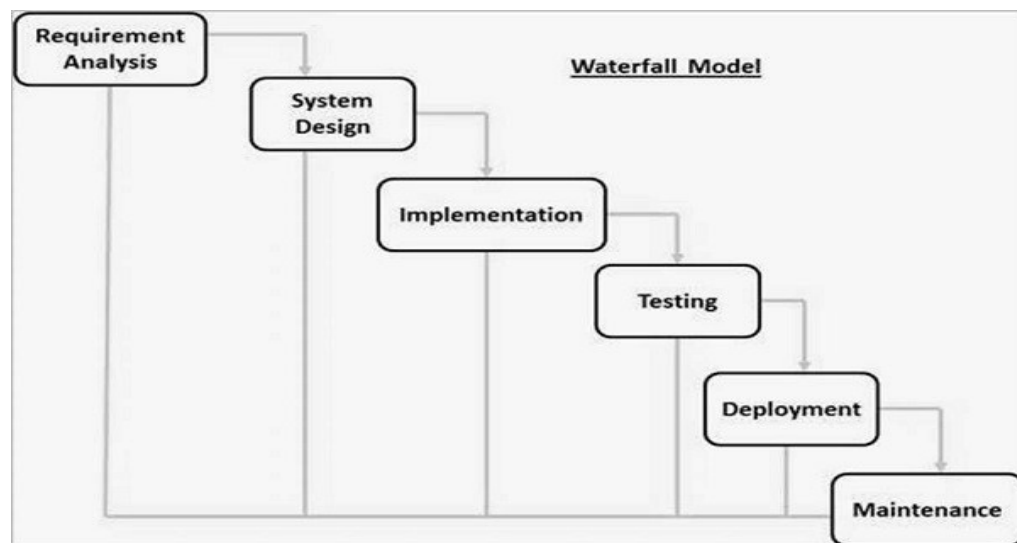
Types of Process Models:

- Waterfall Model
- Incremental Process Model
- Evolutionary Process Models (e.g., Iterative Model, Prototyping Model, Spiral Model, Concurrent Development Model)
- Agile Model

4. Waterfall Model

- **Concept:**

The Waterfall Model is a linear and sequential approach where each phase of the development process must be completed before moving on to the next. It is like following a recipe step-by-step where you cannot skip or go back to previous steps.



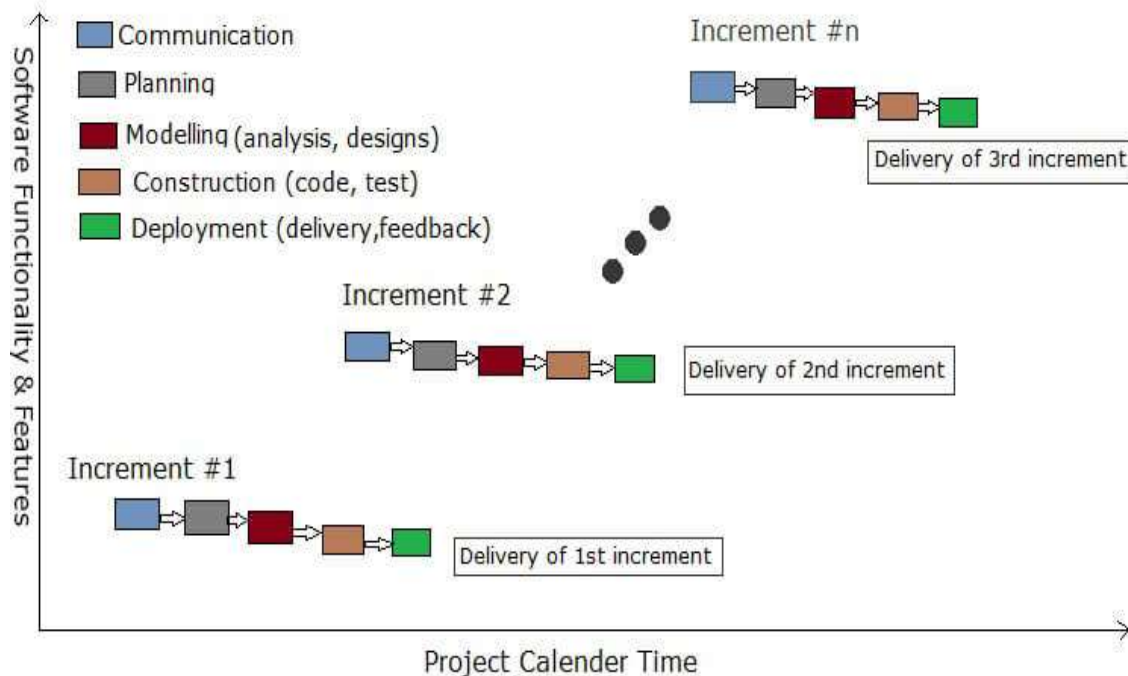
Steps in Waterfall Model:

1. **Requirement & Analysis:** Understanding and gathering all requirements.
 - **Example:** Gathering features like account balance checking or money transfers for the banking app.
 2. **System Design:** Planning how the software will work, including its structure.
 - **Example:** Designing the layout and data storage for the banking app.
 3. **Implementation:** Writing the code to build the software.
 - **Example:** Coding features like login, money transfers, and balance checks in the banking app.
 4. **Testing:** Checking the software to find and fix any issues.
 - **Example:** Testing if users can successfully log in and use all features of the banking app.
 5. **Deployment:** Releasing the software to users.
 - **Example:** Launching the banking app on app stores.
 6. **Maintenance:** Updating and fixing the software post-launch.
 - **Example:** Fixing bugs and releasing updates for the banking app.
- **Advantages:**
 - Simple to understand and easy to manage.
 - Each phase is well-documented and has a clear beginning and end.
 - Works well for smaller projects with clear requirements.
 - **Disadvantages:**
 - Rigid and difficult to go back to previous phases.
 - Late changes can be costly.
 - Not flexible and might miss customer needs until the final product is complete.
 - Not suitable for large or complex projects.

5. Incremental Process Model

- **Concept:**

The Incremental Process Model involves developing software by breaking the project into smaller, manageable parts called increments. Each increment adds a part of the overall system, allowing for step-by-step development and release.



How It Works:

1. **Communication:** Gather and discuss requirements for each increment.
2. **Planning:** Plan the development and integration of each increment.
3. **Modelling:** Design the system components for the current increment.
4. **Construction:** Code and test the increment.
5. **Deployment:** Release the increment to users and collect feedback.

Example: Developing a mobile shopping app:

- **Increment 1:** Release basic features like product browsing and cart functionality.
- **Increment 2:** Add a payment gateway and checkout process.
- **Increment n:** Introduce additional features like order tracking and customer reviews.
- **Advantages:**
 - Early delivery of working software.
 - Easier testing and issue resolution.
 - Flexibility to adjust based on user feedback.
 - Reduced risk as smaller increments minimize the chances of project failure.

- **Disadvantages:**
 - Integration of increments can be complex.
 - Requires detailed planning and coordination for each increment.
 - Initial versions may lack full functionality.
 - Repeated development and testing can increase overall costs.

6. Comparison: Waterfall Model vs. Incremental Process Model

- **Waterfall Model:**
 - Best for projects with a clear, unchanging plan.
 - Follows a strict sequential approach where each phase must be completed before moving to the next.
 - Not flexible and harder to adapt to changes.
- **Incremental Process Model:**
 - Ideal for projects where changes in requirements are expected.
 - Allows for gradual development and improvement of the product.
 - More flexible and better suited for projects that require frequent user feedback and updates.