Day 1 - Name :-Meera J

E id ojmeera

SDLC:

Task1

What is SDLC?

The software development lifecycle (SDLC) is the cost-effective and time-efficient process that development teams use to design and build high-quality software. The goal of SDLC is to minimize project risks through forward planning so that software meets customer expectations during production and beyond. This methodology outlines a series of steps that divide the software development process into tasks you can assign, complete, and measure.

Task 2:

Why is SDLC ?

The Software Development Life Cycle (SDLC) is essential for several key reasons, all of which contribute to the effectiveness, quality, and efficiency of software development. Here’s why it’s needed:

1. Structured Approach

2. Improved Quality

3. Risk Management

4. Reduce project risk

5. provide proper communication and collaboration

Task 3:

What are the stages of SDLC ?

1. Planning
2. Feasibility Study / Requirements Gathering
3. System Design
4. Implementation (Coding)
5. Testing
6. Deployment
7. Maintenance

Task 4
SDLC Models:

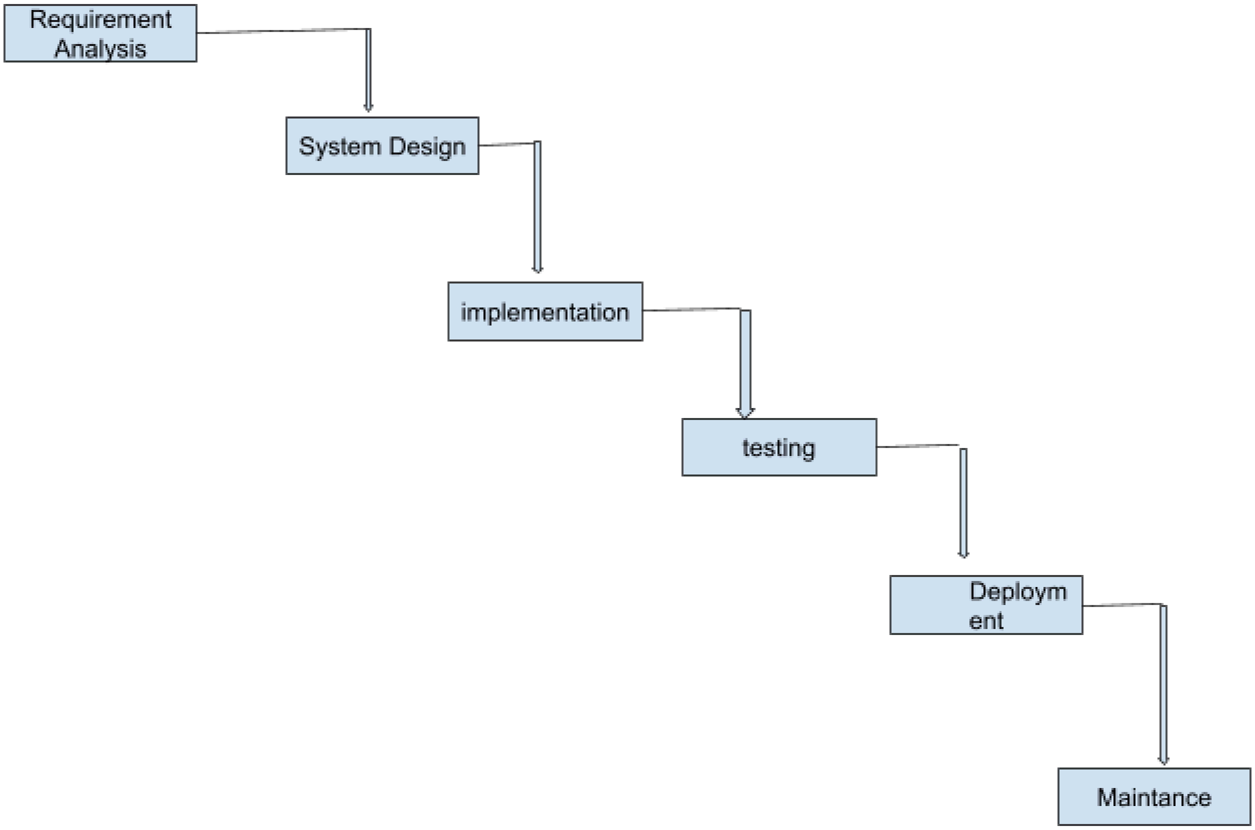
List them , description - 4 lines min and with a image

1. Waterfall Model

Description: The Waterfall model is a linear and sequential approach where each phase must be completed before the next begins. It’s one of the earliest SDLC models, suitable for projects with clear and fixed requirements. The process flows in one direction like a waterfall.

Advantages: Simple to understand and use; clear milestones.

Disadvantages: Inflexible to changes; not ideal for complex or dynamic projects.

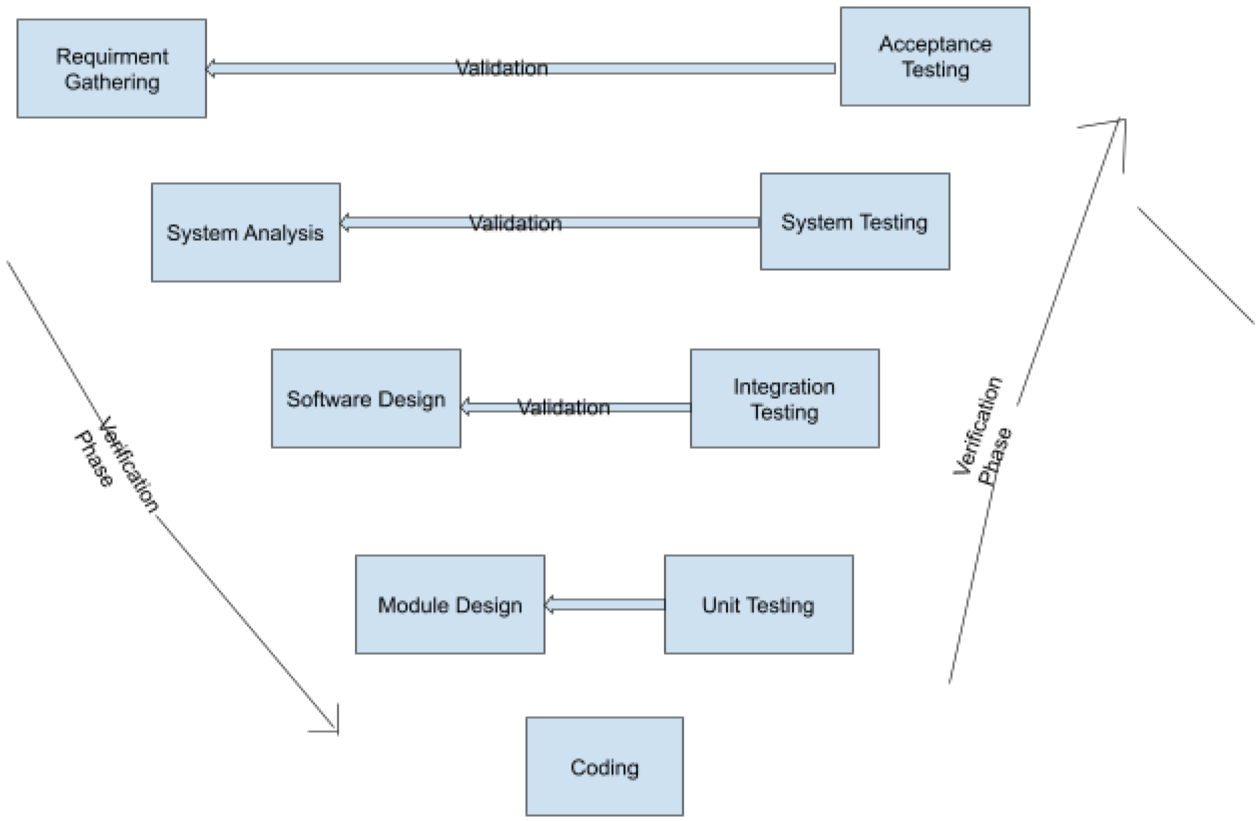


2. V-Model (Verification and Validation)

Description: The V-Model is an extension of the Waterfall model that emphasizes verification and validation. Each development stage is directly associated with a testing phase, ensuring that testing is done in parallel with development.

Advantages: Ensures quality and error-free software by validating at each stage.

Disadvantages: Like Waterfall, it is rigid and doesn't easily accommodate changes.



3. Incremental Model

Description: The Incremental model breaks the project into smaller, manageable units called increments. Each increment is developed, tested, and delivered independently, allowing partial implementations to be deployed early in the development cycle.

Advantages: Allows for partial implementation and faster delivery.

Disadvantages: Needs careful planning and design to ensure integration of increments.

4. Spiral Model

Description: The Spiral model combines the iterative nature of prototyping with the systematic aspects of the Waterfall model. The development is carried out in a series of repetitive cycles (or spirals), each involving planning, design, development, testing, and risk analysis.

Advantages: Flexibility in development and risk management.

Disadvantages: Can be expensive and time-consuming; requires experienced management.

5. Agile Model

Description: The Agile model focuses on iterative development and collaboration between cross-functional teams. Work is divided into small, manageable parts (called sprints), which are completed in short cycles, enabling rapid changes based on feedback.

Advantages: Flexible, quick, and responsive to changes; customer involvement is high.

Disadvantages: Can lead to scope creep and requires constant collaboration.

6. RAD Model (Rapid Application Development)

Description: The RAD model emphasizes rapid development and prototyping over lengthy planning. It uses prototypes and a user feedback loop to quickly develop software, often in smaller phases.

Advantages: Fast development and user feedback integration.

Disadvantages: Requires strong collaboration and can be difficult to manage in large projects.