The Software Development Life Cycle (SDLC)

The Software Development Life Cycle (SDLC) is a systematic approach used by software developers and project managers to design, create, test, and deploy software applications. The SDLC provides a framework for arranging the actions required to develop a software system, guaranteeing that the end product fits the user's needs, is of high quality, and is delivered within the time and budget restrictions.

SDLC Phases:

1) Planning

Objective: Determine the scope and objective of the project. Activities include feasibility studies, resource allocation, project scheduling, cost estimation, and risk management.

Output: Project plan and feasibility report.

2)Requirements Gathering and Analysis:

Objective: Determine the user's wants and requirements. Activities include gathering precise needs from stakeholders, assessing those requirements, and documenting them.

Output: A document outlining the requirements.

3) System Design:

Objective: Create a system architecture that meets the provided requirements. Activities include developing system and software architecture, design specifications, database design, and establishing the overall system structure.

Output: Design papers such as high-level design (HLD) and low-level design (LLD).

4) Implementation (code/development):

Convert the design to executable code. Activities include writing code based on design papers, providing functionality, and integrating many modules.

Output: Source code and executable applications.

5)Testing:

Objective: Ensure that the program is functional, dependable, and defect-free. Activities include performing multiple levels of testing (unit testing, integration testing, system testing, and acceptability testing) to identify and resolve defects. Output includes test cases, test results, and bug reports.

5) Deployment:

Objective: Distribute the software to end users or clients. Activities include installing the software on the user's PC, customizing the environment, and completing any necessary setup.

Output: Deployed program and deployment documentation.

6) Maintenance:

Objective: Ensure the software continues to function properly after deployment. Activities include addressing user-reported concerns, creating changes, updating software, and dealing with any post-deployment bugs.

Output includes updated program versions, patches, and maintenance reports.

SDLC Benefits

1. *Structured Approach*
2. *Improved Project Management*
3. *Better Quality Control*
4. *Clear Documentation*
5. *Enhanced Collaboration*
6. *Risk Management*

SDLC MODELS

1. *Waterfall Model*
2. *Agile Model*
3. *V-Model*
4. *Spiral Model*
5. *DevOps Model*

1) Waterfall model

The Waterfall model is a linear and sequential SDLC approach where each phase (requirements, design, implementation, testing, deployment, maintenance) must be completed before the next begins, making it ideal for projects with clear, unchanging requirements.

2) Agile model

The Agile model is an iterative and incremental SDLC approach that emphasizes flexibility, customer collaboration, and frequent releases, allowing for continuous improvement and adaptation to changing requirements.

Agile Frameworks

- **Scrum**: Iterative sprints with defined roles and events.
- **Kanban**: Visual workflow management with continuous delivery.
- **Lean**: Focus on minimizing waste and maximizing value.