

Software Development Life Cycle

Software Development Life Cycle (SDLC)

1. Planning Phase

- Define the project scope, requirements, and objectives.
- Conduct feasibility studies to assess the project's viability.
- Create a project plan outlining timelines, resources, and deliverables.

2. Analysis Phase

- Gather and analyze requirements from stakeholders.
- Create use cases, data models, and flowcharts to understand system functionality.
- Document system requirements and obtain approval from stakeholders.

3. Design Phase

- Develop system architecture based on requirements.
- Create detailed technical specifications, including database design and system flow diagrams.
- Design user interfaces and wireframes for visual representation.

4. Development Phase

- Write code according to the design specifications.
- Perform unit testing to ensure individual components work correctly.
- Integrate components and conduct system testing to check for errors and bugs.

5. Testing Phase

- Conduct various types of testing like functional, integration, performance, and user acceptance testing.
- Identify and resolve defects found during testing.
- Verify that the software meets all requirements and is ready for deployment.

6. Deployment Phase

- Release the software to users or customers.
- Provide training and support to end-users.

Software Development Life Cycle

- Monitor performance and gather feedback for future improvements.

7. Maintenance Phase

- Address user feedback and bug reports through updates and patches.
- Implement enhancements and new features based on changing requirements.
- Ensure the software remains functional and up-to-date.

Key Points:

- SDLC is a systematic process for developing software applications.
- It ensures quality, efficiency, and customer satisfaction.
- Each phase has specific activities and deliverables.
- Iterative and agile approaches can be used within the SDLC framework.
- Continuous improvement and feedback are essential throughout the lifecycle.