Software Development Life Cycle (SDLC)

The Software Development Life Cycle (SDLC) is the step-by-step process of planning, creating, testing, and maintaining software to ensure high quality and efficiency. It defines how software is developed from start to finish.

Here’s a clear explanation of each phase:

1. Requirement Analysis Purpose: Understand what the client or end-user wants. Activities: - Gather requirements from stakeholders. - Analyze feasibility (technical, financial, and operational). Output: Software Requirement Specification (SRS) document. Example: A client wants an e-commerce website; we note down features like login, product listing, cart, and payment gateway.

2. System Design Purpose: Plan how the software will work internally. Activities: - Create High-Level Design (HLD) – architecture, modules. - Create Low-Level Design (LLD) – database design, API design, UI layouts. Output: Design documents and prototypes. Example: Decide tech stack (React + Node.js + MySQL), create ER diagrams, and wireframes.

3. Implementation (Coding/Development) Purpose: Build the actual software according to the design. Activities: - Developers write code in the chosen programming languages. - Follow coding standards and use version control (like Git). Output: Working software modules. Example: Implement login functionality, product API, and frontend pages.

4. Testing Purpose: Ensure the software works correctly and is bug-free. Activities: - Perform Unit Testing (individual modules), - Integration Testing (modules together), - System Testing (complete system), - User Acceptance Testing (UAT). Output: Tested software ready for deployment. Example: Check if the login works with valid credentials, invalid ones, and edge cases.

5. Deployment Purpose: Deliver the software to users. Activities: - Deploy software to a live environment (cloud, server, app store). - Sometimes a pilot launch is done before full release. Output: Software available for end-users. Example: Host the website on AWS or deploy a mobile app to Play Store.

6. Maintenance Purpose: Keep the software running smoothly after release. Activities: - Fix bugs reported by users. - Release updates with new features. - Optimize performance and security patches. Output: Updated and stable software. Example: Add a new payment method or fix a checkout bug after user feedback.

SDLC Models Different companies follow different models to implement SDLC: - Waterfall Model – Sequential, step-by-step (traditional approach). - Agile Model – Iterative and flexible (used in modern startups). - V-Model – Testing happens parallel to development. - Iterative & Spiral Models – Develop in repeated cycles, focus on risk management.

In Short SDLC ensures software is: - Delivered on time - Cost-efficient - High quality and bug-free - Easy to maintain and upgrade