

Software Development Life Cycle

The Software Development Life Cycle (SDLC) is a comprehensive framework used to guide the creation of software applications. It consists of several stages, each with distinct tasks and goals to ensure the final product meets user needs and expectations.

The SDLC model involves six phases or stages while developing any software and the stages of SDLC are as follows:



Requirements Collection:

In the Requirements Collection phase, the focus is on understanding and documenting what stakeholders need from the software. This involves engaging with users, clients, and other relevant parties to gather their requirements. Once collected, these requirements are meticulously documented to outline both functional aspects (what the software should do) and non-functional aspects (how the software should perform). Prioritization is then carried out to determine which requirements are most critical and impactful for the project's success.

Analysis:

Analysis, involves evaluating the feasibility of the proposed solution. This includes a feasibility study to assess whether the project is technically and economically viable. Risk assessment is also conducted to identify potential challenges and issues that might arise. Use case modeling is used to describe how the system will behave from an end-user's perspective, helping to clarify the software's functionality and interactions.

Designing:

During the Designing phase, the system's overall structure is defined through architectural design. This includes specifying the components, modules, and how they will interact with each other. Detailed design follows, where the specifics of each component are outlined, including diagrams and database schemas. Additionally, user interface (UI) design involves creating blueprint to visualize how users will interact with the system.

Coding/Implementation:

In the Coding/Implementation stage, the design specifications are translated into actual code using programming languages such as Java. This process includes version control to manage and track changes in the codebase, using tools like Git. Code reviews are also conducted to ensure the code adheres to quality standards and is free of errors.

Testing:

Once the coding is complete, the Testing phase begins. This includes unit testing, which checks individual components or functions in isolation to ensure they work correctly. Integration testing follows to verify that different components of the system work together seamlessly. User Acceptance Testing (UAT) is performed to ensure the system meets the requirements and expectations of the end-users.

Maintenance:

Maintenance phase addresses any issues that arise after the software is deployed. This includes fixing bugs reported by users or discovered during testing, enhancing the software by adding new features or improving existing ones based on user feedback, and continuously optimizing performance to ensure the system remains efficient and effective.