

Software Engineering

Prepared by: Neha Tripathi

Software Development Life Cycle (SDLC)

Software Development Life Cycle (SDLC)

- **Software Development Life Cycle (SDLC)** is a process used by the software industry to design, develop and test high quality software.
- The SDLC aims to produce a **high-quality software** that meets or exceeds customer expectations, reaches completion **within times and cost estimates**.
- SDLC consists of a **detailed plan** which explains how to plan, build, and maintain specific software.
- **Every phase of the SDLC life cycle has its own process and deliverables that feed into the next phase.**
- SDLC is the acronym of Software Development Life Cycle.
- It is also called as Software Development Process.
- **SDLC is a framework defining tasks performed at each step in the software development process.**

Why SDLC?

- It offers a basis for **project planning, scheduling, and estimating**
- Provides a **framework** for a standard set of activities and deliverables
- It is a **mechanism for project tracking and control**
- **Increases visibility** of project planning to all involved **stakeholders** of the development process
- **Increased and enhance development speed**
- **Improved client relations**
- Helps you to **decrease project risk and project management plan overhead**

SDLC Phases

- **Phase 1: Feasibility study**
- **Phase 2: Requirement Analysis and Specification**
- **Phase 3: Design**
- **Phase 4: Coding**
- **Phase 5: Testing**
- **Phase 6: Installation/Deployment**
- **Phase 7: Maintenance**

Feasibility study:

- Main aim of feasibility study:
 - determine whether developing the product**
 - financially worthwhile
 - technically feasible.
 - i.e., implementable or not.
- **Work out an overall understanding of the problem.**
- **Formulate different solution strategies.**
- **Identify and Examine alternate solution**
- **Determine which solution is the best.**
- **Prepare feasibility report**

Types of Feasibility

- **Economic:** Can we complete the project within the budget or not?
- **Legal:** Can we handle this project as cyber law and other regulatory framework/compliances.
- **Operation feasibility:** Can we create operations which is expected by the client?
- **Technical:** Need to check whether the current computer system can support the software
- **Schedule:** Decide that the project can be completed within the given schedule or not.

Requirement Analysis and Specification

- Aim of this phase:
 - understand the exact requirements of the customer,
 - document them properly.
 - “What to do”
- Consists of two distinct activities:
 - requirements gathering and analysis
 - requirements specification.
- Collect all related data from the customer:
 - analyze the collected data to clearly understand what the customer wants,
 - find out any inconsistencies and incompleteness in the requirements,
 - resolve all inconsistencies and incompleteness.

Steps of Requirement Analysis and Specification

- **Requirements Gathering/Elicitation**-Functional/Non-functional requirements/Constraints
- **Requirements Analysis**-
- **Requirement Specification**-Software Requirement Specification (SRS) document
- **Requirement Validation**
- **Requirement Management**

Design

- **Design phase transforms requirements specification:**
 - into a form suitable for implementation in some programming language.
 - Plan a Solution
 - Diagrammatic representation
 - “How to do it”
 - the system and software design documents are prepared as per the requirement specification document.
- There are two kinds of design,
 - high-level design and**
 - low-level design.**
- According to their definitions, a high-level design (HLD) is the overall plan of the system, while a low-level design (LLD) is a design of its components.

Coding

- In this phase, developers start build the entire system by **writing code using the chosen programming language**.
- In the coding phase, tasks are divided into units or modules and assigned to the various developers.
- It is the **longest phase** of the Software Development Life Cycle process.
- In this phase, Developer needs to follow certain predefined **coding standards and guidelines**.
- They also need to use **programming tools** like compiler, interpreters, debugger to generate and implement the code.

Testing

- Once the software is complete, and it is deployed in the testing environment.
- The testing team starts testing the **functionality** of the entire system. This is done to verify that the entire application works according to the customer requirement.
- They also test all the **non functional requirement** to check the performance of the system.
- They also check whether system adhere to **the quality standards** or not. It is impossible to deliver quality software without testing.
- **During this phase, QA and testing team may find some bugs/defects which they communicate to developers. The development team fixes the bug and send back to QA for a re-test. This process continues until the software is bug-free, stable, and working according to the business needs of that system.**
- **Unit testing ,Integration testing, System testing etc.**

Installation/Deployment

- Once the software testing phase is over and no bugs or errors left in the system then the **final deployment** process starts.
- At this stage, **the goal is to deploy the software to the production environment so users can start using the product.**
- Based on the feedback given by the project manager, the final software is **released and checked for deployment issues** if any.

Maintenance

- Once the system is deployed, and customers start using the developed system, following 3 activities occurs due to which system requires a **change**:
 - Bug fixing** - bugs are reported because of some scenarios which are not tested at all
 - Upgrade** - Upgrading the application to the newer versions of the Software
 - Enhancement** - Adding some new features into the existing software
- The main focus of this SDLC phase is to ensure that needs continue to be met and that the system continues to perform as per the specification mentioned in the first phase.
- **Maintenance of any software product:**
 - requires much more effort than the effort to develop the product itself.
 - development effort to maintenance effort is typically 40:60.

Types of Maintenance

- **Corrective maintenance:**
 - Correct errors which were not discovered during the product development phases.
- **Perfective maintenance:**
 - Improve implementation of the system
 - enhance functionalities of the system.
- **Adaptive maintenance:**
 - Port software to a new environment,
 - e.g. to a new computer or to a new operating system.
- **Preventive maintenance:**
 - to prevent system getting obsolete.

Thank You!