**SOFTWARE TESTING**

**SDLC (SOFTWARE DEVELOPMENT LIFE CYCLE)**

## What is SDLC?

**SDLC** is a systematic process for building software that ensures the quality and correctness of the software built. SDLC process aims to produce high-quality software that meets customer expectations. The system development should be complete in the pre-defined time frame and cost. 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 stands for **Software Development Life Cycle** and is also referred to as the Application Development life cycle.

## What is the need of SDLC?

Here, are prime reasons why SDLC is important for developing a software system.

* 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.

What are the SDLC Phases?

Graphical user interface, application

Description automatically generated

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

### **Phase 1: Requirement collection and analysis:**

The requirement is the first stage in the SDLC process. It is conducted by the senior team members with inputs from all the stakeholders and domain experts in the industry. Planning for the quality assurance requirements and recognition of the risks involved is also done at this stage.

This stage gives a clearer picture of the scope of the entire project and the anticipated issues, opportunities, and directives which triggered the project.

Requirements Gathering stage need teams to get detailed and precise requirements. This helps companies to finalize the necessary timeline to finish the work of that system.

This phase will mainly include BRS (Business Requirement Specification)- BRS will describe brief description about client business needs like users of the application and required services for those users.

### **Phase 2: Analysis/Feasibility study:**

In this phase system analyst (SA) will analyze client requirements in BRS (Business Requirement Specifications) based on that he will prepare detailed requirements document called **FUNCTIONAL REQUIREMENT SPECIFICATION (FRS)/ SYSTEM REQUIREMENT SPECIFICATION (SRS)**

SRS document will describe detailed functionality of the system. In this phase SA will provide information to the client by releasing statement of work (SOW) document.

For successful project development clear FRS document is mandatory. And this SRS document is a technical document not to be given to anyone.

It includes everything which should be designed and developed during the project life cycle. Below are the steps related to Feasibility Study:

**There are mainly five types of feasibilities checks:**

* **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.

### **Phase 3: Design:**

In this third phase, the system and software design documents are prepared as per the requirement specification document. This helps define overall system architecture.

This design phase serves as input for the next phase of the model.

There are two kinds of design documents developed in this phase:

High-Level Design (HLD)

* Brief description and name of each module
* An outline about the functionality of every module
* Interface relationship and dependencies between modules
* Database tables identified along with their key elements
* Complete architecture diagrams along with technology details

Low-Level Design(LLD)

* Functional logic of the modules
* Database tables, which include type and size
* Complete detail of the interface
* Addresses all types of dependency issues
* Listing of error messages
* Complete input and outputs for every module

### **Phase 4: Coding:**

Once the system design phase is over, the next phase is 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 guidelines. They also need to use programming tools like compiler, interpreters, debugger to generate and implement the code.

### **Phase 5: 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.

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.

### **Phase 6: Installation/Deployment:**

Once the software testing phase is over and no bugs or errors left in the system then the final deployment process starts. Based on the feedback given by the project manager, the final software is released and checked for deployment issues if any.

### **Phase 7: Maintenance:**

Once the system is deployed, and customers start using the developed system, following 3 activities occur

* 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.