### ****SDLC (Software Development Life Cycle)****

**SDLC** (Software Development Life Cycle) is a process used by software developers to design, create, test, and maintain software applications. It's like a step-by-step guide to making sure the software is built correctly, works well, and is delivered on time.

In simple terms, SDLC is a set of stages or phases that help developers build software in an organized way, ensuring that everything is done properly from start to finish.

### ****Phases of SDLC and Documents Required****

### ****1. Requirement Analysis****

**Objective**: This phase involves understanding and gathering the business requirements, functional and non-functional requirements, and user expectations for the software project.

**Key Activities**:

* Interacting with stakeholders to collect their requirements.
* Analyzing business needs and defining the system requirements.

**Documents Required**:

* **Business Requirements Document (BRD)**: Defines the high-level business goals and objectives of the software project.
* **System Requirements Specification (SRS)**: Detailed description of both functional and non-functional requirements for the system.
* **Test Strategy Document**: Defines the approach, scope, resources, and schedule for testing the software.
* **Requirement Traceability Matrix (RTM)**: Maps each requirement to its corresponding test cases to ensure all requirements are covered during testing.

### ****2. Design****

**Objective**: This phase involves transforming the gathered requirements into a blueprint for building the system. This stage defines how the system will operate.

**Key Activities**:

* Creating the system architecture and design based on requirements.
* Preparing high-level and detailed designs.

**Documents Required**:

* **High-Level Design (HLD)**: Provides an overview of the system architecture and components.
* **Low-Level Design (LLD)**: Describes the internal workings of each component or module, including algorithms, data structures, and APIs.
* **Database Design Document**: Specifies the database schema, tables, relationships, and constraints.
* **Test Plan**: A comprehensive document detailing the testing approach, resources, schedule, and types of testing to be carried out during the project lifecycle.

### ****3. Implementation (Coding)****

**Objective**: This phase involves the actual development or coding of the software. Developers build the system according to the design specifications.

**Key Activities**:

* Writing the code for individual components or modules.
* Ensuring the code is aligned with the requirements and design.

**Documents Required**:

* **Source Code / Codebase**: The actual code written by the developers to implement the system.
* **Unit Test Plan**: Outlines the testing approach for individual components or modules, ensuring that they work as expected.
* **Code Review Document**: Documents the results of peer reviews to ensure code quality and consistency with design.

### ****4. Testing****

**Objective**: The software is tested to identify defects and ensure it meets the requirements. The testing phase ensures that the software is stable, reliable, and meets business objectives.

**Key Activities**:

* Performing various types of testing (e.g., unit testing, integration testing, system testing, acceptance testing).
* Identifying, reporting, and fixing defects.

**Documents Required**:

* **Test Case Document**: A detailed list of test cases to be executed, including input, expected output, and execution conditions.
* **Test Execution Report**: Results of the executed test cases, indicating which tests passed, failed, or were blocked.
* **Defect Report**: A document detailing any defects or issues found during testing, including severity, steps to reproduce, and their resolution status.
* **Test Summary Report**: A comprehensive report summarizing the testing activities, results, and the overall quality of the software.

### ****5. Deployment****

**Objective**: This phase involves deploying the software into a production environment where it will be available for users.

**Key Activities**:

* Deploying the final version of the software into the production environment.
* Ensuring a smooth deployment process with minimal downtime.

**Documents Required**:

* **Deployment Plan**: A document that outlines the steps for deploying the software to production, including timing, rollback procedures, and resource requirements.
* **Release Notes**: A document that details the changes, bug fixes, new features, and installation instructions for the software release.
* **User Manuals**: Documentation that helps end-users understand how to use the software effectively.

### ****6. Maintenance****

**Objective**: After the software is deployed, the maintenance phase ensures it continues to perform as expected and any issues are resolved. This phase also involves making updates, patches, and improvements.

**Key Activities**:

* Addressing any issues that arise after the software is in use (e.g., bug fixes, updates).
* Enhancing the software with new features based on user feedback.

**Documents Required**:

* **Maintenance Report**: Records all updates, bug fixes, and changes made to the software during the maintenance phase.
* **Issue/Defect Logs**: Logs tracking issues encountered in the live environment and their resolution.
* **Patch Notes**: Detailed documentation on any patches or updates applied to the software, including their impact and fixes.

### ****Summary of SDLC Phases and Documents:****

| ****Phase**** | ****Objective**** | ****Documents Required**** |
| --- | --- | --- |
| **1. Requirement Analysis** | Gather and analyze the system requirements from stakeholders. | - Business Requirements Document (BRD)  - System Requirements Specification (SRS)  - Test Strategy Document  - Requirement Traceability Matrix (RTM) |
| **2. Design** | Create the system architecture and detailed design for the software. | - High-Level Design (HLD)  - Low-Level Design (LLD)  - Database Design Document  - Test Plan |
| **3. Implementation (Coding)** | Develop the software according to the design specifications. | - Source Code / Codebase  - Unit Test Plan  - Code Review Document |
| **4. Testing** | Test the software to find defects and ensure it meets requirements. | - Test Case Document  - Test Execution Report  - Defect Report  - Test Summary Report |
| **5. Deployment** | Deploy the software to the production environment. | - Deployment Plan  - Release Notes  - User Manuals |
| **6. Maintenance** | Maintain the software, fixing bugs and making updates as needed. | - Maintenance Report  - Issue/Defect Logs  - Patch Notes |