## **Day 2 SDLC**

**Assignment 1: SDLC Overview - Create a one-page infographic that outlines the SDLC phases (Requirements, Design, Implementation, Testing, Deployment), highlighting the importance of each phase and how they interconnect.**

**1. Requirements**

* Gather and analyze business requirements.
* Ensures clear understanding of what the clients need.
* Outcome: Requirements Specification Document.
* Connection: Sets the foundation for all subsequent phases.

**2. Design**

* Plan the solution architecture and design.
* Translates requirements into a blueprint for building the software.
* Design Documents (including system architecture, data models, and interfaces).
* Connection: Provides a structured framework for the implementation phase.

**3. Implementation**

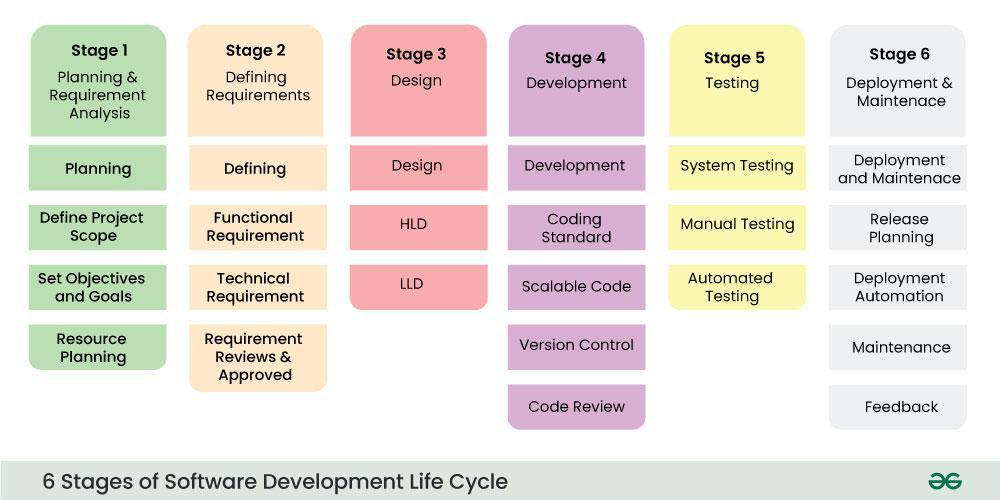
* Actual coding and building of the software.
* Converts design into a functional product.
* Source Code and Executable Software.
* Connection: Follows design specifications to ensure the final product meets requirements.

**4. Testing**

* Verify and validate the software against requirements.
* Identifies defects and ensures quality.
* Test Reports and Defect Logs.
* Connection: Ensures that the implementation phase meets the requirements and design.

**5. Deployment**

* Description: Deliver the software to the production environment.
* Importance: Makes the software available to users.
* Outcome: Deployed Software and Deployment Reports.
* Connection: Marks the transition from development to production use, and initiates maintenance.

****

**Assignment 2 : Explain**

**1.Scrum role:**

* product owner : interact with customer for getting requirements and give to scrum master
* Scrum master : handles the sprint team
* The team : developer and testers

**2.Artefact →**

product backlog:

sprint backlog:When the a sprint is pass the due date of completion

**3. Scrum meeting /ceremony**

### **Sprint Planning**

* Define the sprint goal and select work for the upcoming sprint.

### **Daily Standup**

* Synchronise daily activities in meetings and plan for the next 24 hours.

### **Sprint Review**

* Demonstrate completed work and gather feedback.

### **Sprint Retrospective**

* Reflect on the past sprint and plan improvements.

**Assignment 3:**

**Develop a case study analyzing the implementation of SDLC phases in a real-world engineering project. Evaluate how Requirement Gathering, Design, Implementation, Testing, Deployment, and Maintenance contribute to project outcomes.**

**Real-Life Example: Develop a Song Application**

**Planning and Analysis:** In this phase, business analysts and project managers gather requirements from potential users and stakeholders about how the song application should function and what features it should contain. They create detailed Software Requirement Specification (SRS) documents, which are reviewed and signed off by stakeholders to ensure all needs and expectations are captured accurately.

**Design:** The SRS documents are sent to developers and designers. Developers review these documents to understand the requirements while designers create wireframes and UI mockups. System architects prepare high-level system architecture and database schema, providing a clear blueprint for development.

**Development:** During this phase, developers code the application, creating the web pages and APIs required to implement the desired functionality.

**Testing:** testers conduct thorough end-to-end functionality testing to ensure the application works as intended without any bugs.

**Deployment and Maintenance:** The application is deployed to app stores like Google Play and Apple App Store, with DevOps engineers setting up CI/CD pipelines for seamless updates. Post-deployment, the support team monitors app performance and user feedback, ensuring regular updates and maintenance for continuous improvement and user satisfaction.

**For New Features:** The same process is repeated for developing any new features. Requirements are gathered, designs are updated, new code is developed, tested, and deployed, ensuring the application evolves to meet user needs.

**Assignment 4:**

**Research and compare SDLC models suitable for engineering projects. Present findings on Waterfall, Agile, Spiral, and V-Model approaches, emphasising their advantages, disadvantages, and applicability in different engineering contexts.**

| **Criteria** | **Waterfall** | **Agile** | **Spiral** | **V-model** |
| --- | --- | --- | --- | --- |
| **Advantages** | * Simple and easy to understand. * Structured approach with clear phases. * Good for projects with well-defined requirements. | * Flexible and adaptive to changes. * Continuous feedback and collaboration. * Early and frequent delivery of working software | * Focus on risk assessment and reduction. * Iterative development with repeated cycles. * Allows for customer feedback at early stages. | * Emphasises verification and validation. * Clear, structured phases with corresponding testing. * Easy to manage due to rigidity. |
| **Disadvantages** | * Inflexible to changes. * Late testing phase. * Not suitable for complex and high-risk projects. | * Requires strong team collaboration and communication. * Can be less predictable in timeline and budget. * Not ideal for projects with fixed scope and requirements. | * Can be complex to manage. * Requires expertise in risk assessment. * Can be costly and time-consuming | * Inflexible to changes. * Testing phase occurs late in the process. * Not suitable for projects with evolving requirements. |
| **Applicability** | * Suitable for smaller projects with clear, unchanging requirements. * Ideal for projects where a predictable, linear process is needed. | * Best for complex, dynamic projects. * Suitable for projects needing rapid delivery and iterative improvements. | * used in large-scale and risk management is a significant concern. * Unclear requirements projects | * Best for projects where requirements are well-defined and stable. * Suitable for safety-critical systems requiring thorough testing |