Day 2 Assignments

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.

Stages of the Software Development Life Cycle
SDLC specifies the task(s) to be performed at various stages by a software engineer or developer. It ensures that the end product is able to meet the customer's expectations and fits within the overall budget. Hence, it's vital for a software developer to have prior knowledge of this software development process.

Stage-1: Planning and Requirement Analysis
Planning is a crucial step in everything, just as in software development In this same stage, it is also performed by the developers of the organization. This is attained from customer inputs, and sales department/market surveys.

The information from this analysis forms the building blocks of a basic project. The quality of the project is a result of planning. Thus, in this stage, the basic project is designed with all the available information

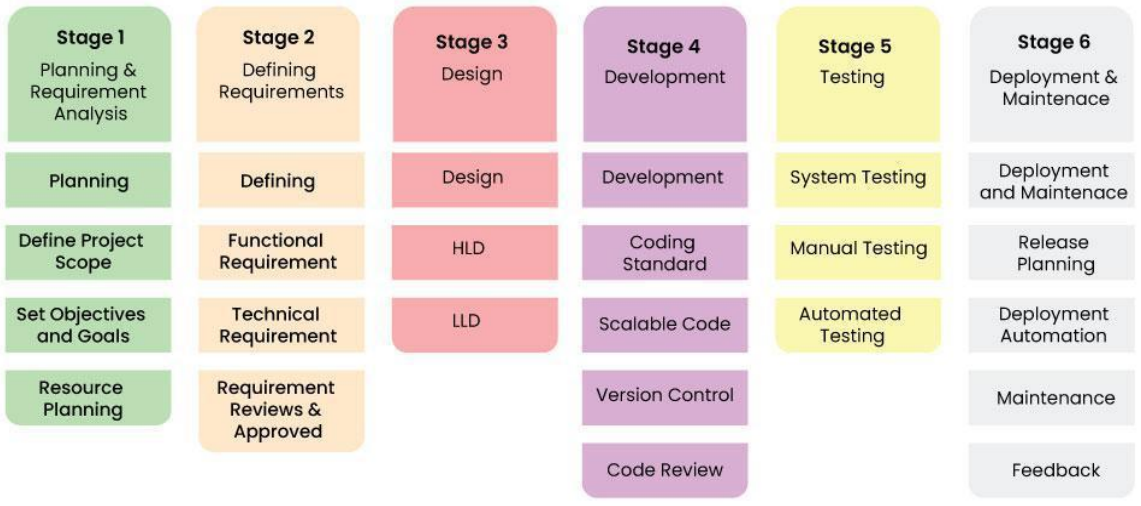
Stage-2: Defining Requirements
In this stage, all the requirements for the target software are specified. These requirements get approval from customers, market analysts, and stakeholders.
This is fulfilled by utilizing SRS (Software Requirement Specification). This is a sort of document that specifies all those things that need to be defined and created during the entire project cycle.

Stage-4: Developing Product
At this stage, the fundamental development of the product starts. For this, developers use a specific programming code as per the design in the DDS. Hence, it is important for the coders to follow the protocols set by the association. Conventional programming tools like compilers, interpreters, debuggers, etc. are also put into use at this stage. Some popular languages like C/C++, Python, Java, etc. are put into use as per the software regulations.

Stage-5: Product Testing and Integration
After the development of the product, testing of the software is necessary to ensure its smooth execution. Although, minimal testing is conducted at every stage of SDLC. Therefore, at this stage, all the probable flaws are tracked, fixed, and retested. This ensures that the product confronts the quality requirements of SRS

Stage-6: Deployment and Maintenance of Products

After detailed testing, the conclusive product is released in phases as per the organization's strategy. Then it is tested in a real industrial environment. It is important to ensure its smooth performance. If it performs well, the organization sends out the product as a whole. After retrieving beneficial feedback, the company releases it as it is or with auxiliary improvements to make it further helpful for the customers. However, this alone is not enough. Therefore, along with the deployment, the product's supervision.



Stage 1 Planning & Requirement Analysis	Stage 2 Defining Requirements	Stage 3 Design	Stage 4 Development	Stage 5 Testing	Stage 6 Deployment & Maintenace
Planning	Defining	Design	Development	System Testing	Deployment and Maintenace
Define Project Scope	Functional Requirement	HLD	Coding Standard	Manual Testing	Release Planning
Set Objectives and Goals	Technical Requirement	LLD	Scalable Code	Automated Testing	Deployment Automation
Resource Planning	Requirement Reviews & Approved		Version Control		Maintenance
			Code Review		Feedback

6 Stages of Software Development Life Cycle

2: Develop a case study analysing 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.

Case Study: Implementation of SDLC Phases in the Development of an E-Commerce Platform

Project Overview

The project under consideration is the development of a comprehensive e-commerce platform for a mid-sized retail company. The platform aims to offer a seamless shopping experience, incorporating features such as product browsing, secure payment processing, customer reviews, and order tracking.

SDLC Phases Analysis

1,Requirement Gathering

Activities:

Stakeholder interviews

Market analysis

Competitor benchmarking

Creation of requirement specification documents

Outcomes:

Clear understanding of customer needs and business objectives

Detailed requirements specification, including functional and non-functional requirements

Prioritized feature list

Impact:

Requirement gathering set the foundation for the project by ensuring that all stakeholders had a common understanding of what the platform needed to achieve. This phase helped prevent scope creep and ensured that the project stayed aligned with business goals.

2.Design

Activities:

System architecture design

Database schema design

User interface (UI) and user experience (UX) design

Preparation of design documents and mockups

Outcomes:

Detailed system architecture blueprint

Wireframes and prototypes for the user interface

Design specifications for development

Impact:

The design phase translated requirements into a structured plan. A well-thought-out design ensured that the system would be scalable, maintainable, and user-friendly. This phase was crucial for identifying potential technical challenges early on.

Implementation

Activities:

Coding and development

Integration of third-party services (e.g., payment gateways, analytics)

Regular code reviews and version control

Outcomes:

Developed front-end and back-end components

Integrated third-party services

Functional modules ready for testing

Impact:

Implementation brought the design to life. Adhering to coding standards and best practices ensured high code quality. Regular integration and continuous delivery practices helped identify and resolve issues promptly.

Testing

Activities:

Unit testing

Integration testing

System testing

User acceptance testing (UAT)

Performance and security testing

Outcomes:

Identification and fixing of bugs

Verification that the system meets requirements

Validation of system performance and security

Impact:

Rigorous testing ensured that the platform was reliable, secure, and performant. User acceptance testing involved stakeholders in validating that the system met their needs, reducing the risk of major issues post-deployment**.**

Deployment

Activities:

Setting up the production environment

Data migration

Deployment of the application

Verification in the live environment

Outcomes:

Successful launch of the platform

Initial monitoring for any issues post-deployment

User onboarding and training

Impact:

Smooth deployment ensured that the platform was available to users as planned. Effective deployment strategies minimized downtime and disruptions. Post-deployment monitoring helped quickly address any immediate issues.

Maintenance

Activities:

Ongoing support and troubleshooting

Performance monitoring and optimization

Regular updates and feature enhancements

Security patches and updates

Outcomes:

Sustained system performance and user satisfaction

Continuous improvement based on user feedback

Secure and up-to-date platform

Impact:

Maintenance ensured the long-term success and relevance of the platform. Regular updates kept the platform competitive and secure, while ongoing support addressed user issues and enhanced user satisfaction.

Assignment 3: 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.

1. Waterfall Model

Description:

A linear and sequential approach where each phase must be completed before the next begins.

Advantages:

Easy to understand and manage.

Well-suited for projects with clear, fixed requirements.

Emphasizes documentation.

Disadvantages:

Inflexible to changes once the project is underway.

High risk and uncertainty, as testing occurs only after development.

Applicability:

Ideal for projects with well-defined requirements and low uncertainty, such as construction or manufacturing projects.

2. Agile Model

Description:

An iterative and incremental approach that focuses on flexibility, customer feedback, and continuous improvement.

Advantages:

Highly adaptable to changes.

Continuous customer involvement ensures product relevance.

Short development cycles allow for frequent reassessment and adjustment.

Disadvantages:

Requires significant customer and team involvement.

Can be challenging to predict time and budget.

Less emphasis on documentation.

Applicability:

Best for projects with high uncertainty and evolving requirements, such as software development and R&D projects.

3. Spiral Model

Description:

Combines iterative development with systematic risk management, emphasizing repeated cycles (spirals) of planning, risk analysis, engineering, and evaluation.

Advantages:

Strong focus on risk assessment and mitigation.

Allows for iterative refinement through each spiral.

Accommodates changes well.

Disadvantages:

Can be complex to manage.

Requires expertise in risk analysis.

Can be costly and time-consuming.

Applicability:

Suitable for large, complex projects with significant risks, such as aerospace and defense systems.

4. V-Model (Verification and Validation)

Description:

An extension of the Waterfall model where each development phase is associated with a corresponding testing phase.

Advantages:

Emphasizes testing at every stage, ensuring validation and verification.

Easy to manage with clear milestones.

Disadvantages:

Inflexible to changes.

Like Waterfall, it assumes requirements are well-understood from the beginning.

Applicability:

Effective for projects where high reliability and safety are critical, such as medical devices and automotive systems.