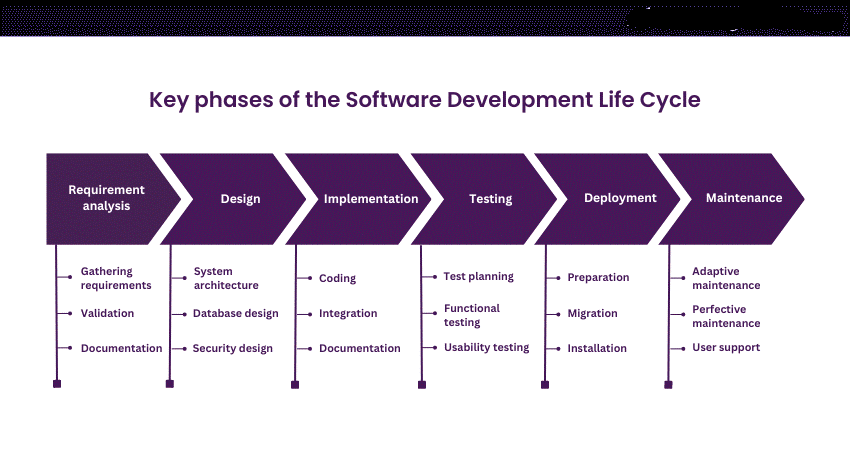
**Assignment 1: SDLC Overview - Create a one-page infographic that outlines the SDLC phases (Requirements, Design, Implementation, Testing, Deployment), highlighting the importance of each phase and how they interconnect.**

****

**Assignment 2: 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**

Company Axis Bank, a leading financial services firm, embarked on a project to develop a loan management automation system for a banking client. The goal is to enhance efficiency, reliability, and compliance in the client's loan processing processes.

1. Requirement Gathering:

Objective: The objective of this phase is to facilitate team collaboration with the bank's stakeholders to gather comprehensive requirements.

Approach: Workshops, interviews, and surveys were conducted with loan officers, compliance officers, and IT personnel to capture user needs effectively.

Documentation was created to ensure a clear understanding of the project .  
  
2. Design:  
  
Collaborate with stakeholders to develop architectural diagrams, database schemas, and UI prototypes. Design technology stack, workflows, and security measures.  
  
3. Implementation:  
  
Approach: Agile development, utilizing tools like Jira and Git. Adhere to coding standards, establish CI/CD pipelines for rapid iteration.  
  
4. Testing:  
  
Validate functionality, reliability, and performance of the system.  
  
Conduct unit, integration, system, Use automated testing frameworks

5. Deployment:  
  
Deploy system to production environment .

6. Maintenance:  
  
Establish dedicated support team, utilize performance monitoring tools. Perform regular maintenance activities.

**Assignment 3: Research and compare SDLC models suitable for engineering projects. Present findings on Waterfall, Agile, Spiral, and V-Model approaches, emphasizing their advantages, disadvantages, and applicability in different engineering contexts.**

**Waterfall Model:**

**Advantages:**

Simple and easy to understand and use.

Well-suited for small projects with clear and well-defined requirements.

**Disadvantages:**

Lack of flexibility where one phase must be completed before moving to the next.

Difficulty in accommodating changes once the development process has started.

Testing occurs only after the development phase, which may lead to issues being discovered late in the cycle.

Applicability: Best suited for projects with stable requirements and where changes are unlikely to occur during the development process.

**Agile Model:**

**Advantages:**

Highly flexible and allowing for changes and iterations throughout the development process.

Encourages customer involvement and feedback, leading to higher customer satisfaction.

Breaks the project into smaller, manageable iterations or sprints, allowing for quicker delivery of working software.

**Disadvantages:**

Requires active involvement and collaboration from all stakeholders throughout the project, which can be challenging.

Lack of documentation may lead to a lack of clarity in requirements and design.

Applicability: Ideal for projects where requirements are likely to change or evolve, and where rapid delivery of working software is essential.

**Spiral Model:**

**Advantages:**

Incorporates elements of both waterfall and iterative approaches, providing flexibility and risk management.

Allows for early identification and mitigation of risks through multiple iterations.

Suitable for large and complex projects where risks need to be managed effectively.

**Disadvantages:**

Can be time-consuming and costly due to the iterative nature of the model.

Requires extensive risk analysis and management expertise.

Applicability: Best suited for projects with high levels of uncertainty and where risk management is crucial.

**V-Model:**

**Advantages:**

Emphasizes testing throughout the development process, ensuring early detection and resolution of defects.

Provides a clear and structured approach, with each stage having corresponding testing activities.

Suitable for projects with strict regulatory or compliance requirements.

**Disadvantages:**

Like the waterfall model, it can be rigid and less flexible to changes.

Testing activities may be time-consuming, leading to longer development cycles.

Applicability: Well-suited for projects where quality and compliance are critical, such as in the development of medical devices or aerospace systems.

In conclusion, the choice of SDLC model depends on various factors such as project size, complexity, requirements volatility, and industry standards. Each model has its strengths and weaknesses, and the selection should be based on careful consideration of these factors to ensure successful project delivery.