**DAY - 2**

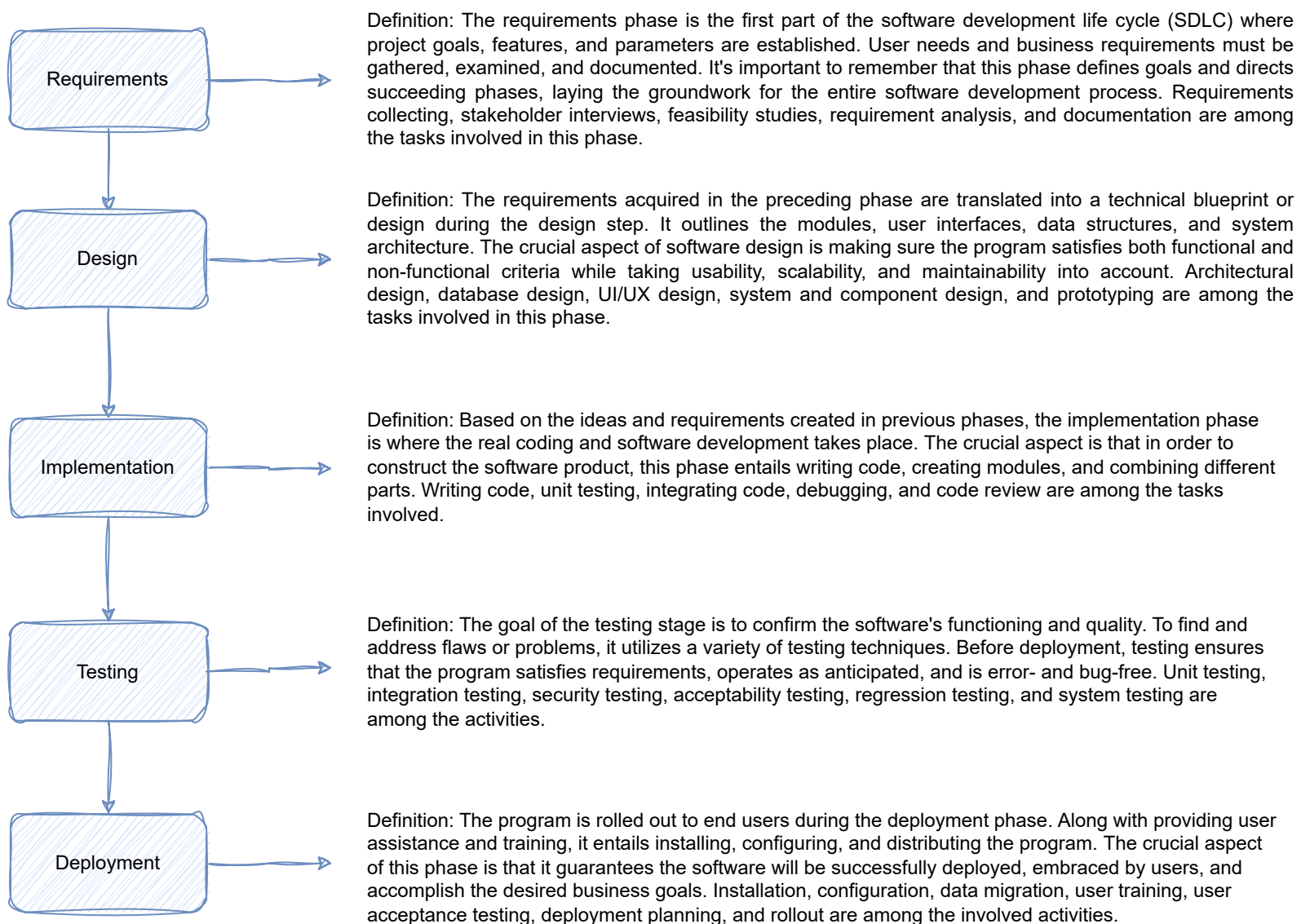**Assignment 1:** SDLC Overview - Create a one-page infographic that outlines the SDLC phases (Requirements, Design, Implementation, Testing, Deployment), highlighting the importance of each phase and how they interconnect.

**Requirements**

Definition: The requirements phase is the first part of the software development life cycle (SDLC) where project goals, features, and parameters are established. User needs and business requirements must be gathered, examined, and documented. It's important to remember that this phase defines goals and directs succeeding phases, laying the groundwork for the entire software development process. Requirements collecting, stakeholder interviews, feasibility studies, requirement analysis, and documentation are among the tasks involved in this phase.

**Design**

Definition: The requirements acquired in the preceding phase are translated into a technical blueprint or design during the design step. It outlines the modules, user interfaces, data structures, and system architecture. The crucial aspect of software design is making sure the program satisfies both functional and non-functional criteria while taking usability, scalability, and maintainability into account. Architectural design, database design, UI/UX design, system and component design, and prototyping are among the tasks involved in this phase.

**Implementation**

Definition: Based on the ideas and requirements created in previous phases, the implementation phase is where the real coding and software development takes place. The crucial aspect is that in order to construct the software product, this phase entails writing code, creating modules, and combining different parts. Writing code, unit testing, integrating code, debugging, and code review are among the tasks involved.

**Testing**

Definition: The goal of the testing stage is to confirm the software's functioning and quality. To find and address flaws or problems, it utilizes a variety of testing techniques. Before deployment, testing ensures that the program satisfies requirements, operates as anticipated, and is error- and bug-free. Unit testing, integration testing, security testing, acceptability testing, regression testing, and system testing are among the activities.

**Deployment**

Definition: The program is rolled out to end users during the deployment phase. Along with providing user assistance and training, it entails installing, configuring, and distributing the program. The crucial aspect of this phase is that it guarantees the software will be successfully deployed, embraced by users, and accomplish the desired business goals. Installation, configuration, data migration, user training, user acceptance testing, deployment planning, and rollout are among the involved activities.

# Implementation of SDLC Phases in Meesho E-commerce Online Shopping

**Table of content:**

**Problem Statement:**

Create a productive online shopping platform for e-commerce Meesho. Understanding user needs, creating a scalable system, guaranteeing dependable implementation, thorough testing, smooth deployment, and continuous maintenance for peak efficiency and user happiness are among the challenges.

**Introduction:**

Meesho is a leading e-commerce platform in India that enables individuals to start their own online businesses by reselling products. This case study analyses the implementation of Software Development Life Cycle (SDLC) phases in Meesho's online shopping project and evaluates how each phase contributes to project outcomes.

**Requirement Gathering:**

Meesho carried out in-depth market research during the requirement gathering stage to comprehend user preferences and demands in the online market. They engaged with end users and possible resellers to determine the essential features and capabilities that the platform needed. They also examined rival systems in order to learn about industry best practices. This phase resulted in an extensive needs list that served as the foundation for other phases.

**Design:**

Meesho converted the requirements into a blueprint for the e-commerce platform during the design phase. Scalability, usability, and performance were the main considerations in the design of the system architecture, database structure, and user interface (UI). The development of an intuitive user interface for resellers and customers received particular attention. Wireframes and prototypes were created to show the platform's design and functionality. The groundwork for the implementation step was done during the design phase.

**Implementation:**

Utilizing backend frameworks like Node.js and MongoDB in addition to technologies like HTML, CSS, and JavaScript, they developed the different modules and components of the online retail platform. Practices for version control and continuous integration were used to guarantee code quality and speed up development. Meesho used an agile methodology that permitted regular stakeholder feedback loops and iterative development.

**Testing:**

The software created for Meesho's project received thorough testing during the testing phase to guarantee its dependability and quality. There were other kind of testing carried out, such as system, user acceptability, integration, and unit testing (UAT). To guarantee thorough test coverage and automate repetitive test cases, automated testing tools were used. Feedback from testing was integrated into the development process to fix any flaws or problems found.
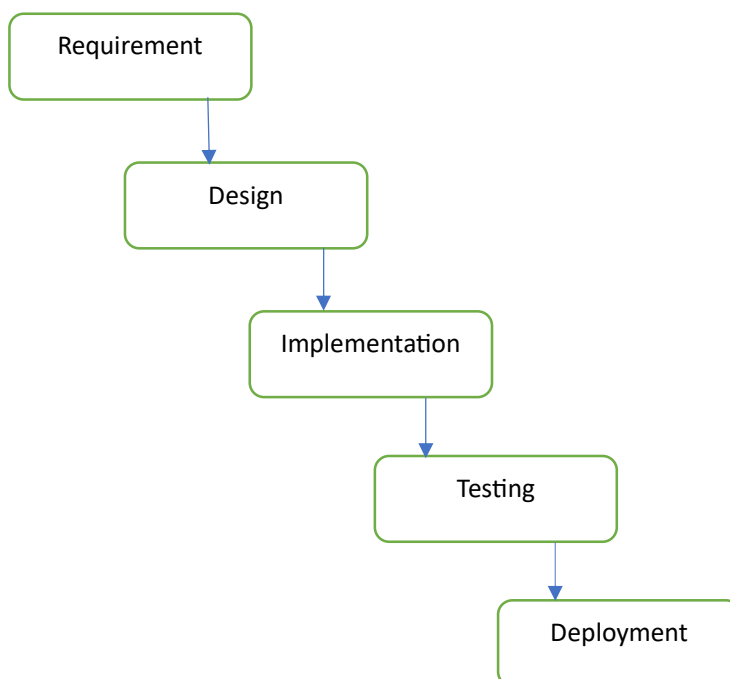
**Deployment:**

The online shopping platform was made available to production environments by Meesho. Databases, domain settings, and software installation and configuration on servers were all part of the deployment process. To reduce user inconvenience and guarantee a seamless transition, a staggered deployment strategy was used. The platform's features and capabilities were introduced to resellers through user training sessions. In order to monitor system performance and handle any difficulties that might arise, post-deployment monitoring was put into place.

**Maintenance:**

Meesho entered maintenance mode, where regular upgrades and support were given to keep the platform running. Feature improvements, speed optimization, security patches, and bug fixes were all part of routine maintenance. Meesho gathered feedback from users to identify areas for improvement and prioritize future development activities. The goal of the maintenance phase was to maintain throughout time the platform's security, stability, and user happiness.

**Conclusion:**

The successful development and deployment of Meesho's e-commerce project were largely due to the implementation of SDLC phases, which involved requirement gathering, design, implementation, testing, deployment, and maintenance, resulting in a robust and scalable platform.

**Software development life cycle (SDLC)**

```
┌─────────────────┐
│   Requirement   │
└─────────────────┘
         │
         ▼
    ┌─────────────────┐
    │     Design      │
    └─────────────────┘
             │
             ▼
        ┌─────────────────┐
        │ Implementation  │
        └─────────────────┘
                 │
                 ▼
            ┌─────────────────┐
            │     Testing     │
            └─────────────────┘
                     │
                     ▼
                ┌─────────────────┐
                │   Deployment    │
                └─────────────────┘
```

**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.
- Sequential approach ensures clarity in project phases.
- Well-suited for projects with stable requirements and defined scope.

Disadvantages:
- Limited flexibility for changes after each phase.
- High risk of latest stage requirement changes.
- No working software until late stages, potentially leading to customer dissatisfaction.

*Applicability:*

Best suited for projects with well-defined requirements and minimal changes expected during development, such as construction projects or hardware development.

**Agile Model:**

*Advantages:*
- Highly flexible and adaptable to changing requirements.
- Incremental and iterative development, allowing for early delivery of working software.
- Continuous customer involvement and feedback throughout the project.

*Disadvantages:*
- Requires active customer involvement, which may not be feasible for all projects.
- Lack of emphasis on documentation can lead to potential misunderstandings or issues.
- May require experienced team members to effectively implement Agile practices.

*Applicability:*

Ideal for projects with evolving requirements, dynamic environments, and a need for rapid development cycles, such as software development or mobile app development.

**Spiral Model:**

*Advantages:*
- Iterative approach allows for risk management and mitigation throughout the project lifecycle.
- Flexibility to accommodate changes and address issues early in the development process.
- Emphasizes prototyping and stakeholder involvement, ensuring alignment with user needs.

*Disadvantages:*
- Complex and resource intensive, requiring significant planning and documentation.
- May lead to extended project timelines and increased costs.
- Requires experienced project management to effectively manage risks and iterations.

*Applicability:*

Suitable for projects with high levels of uncertainty, complex requirements, and a need for risk management, such as largescale software systems or defence projects.

**VModel:**

*Advantages:*
- Corresponds testing activities directly with development phases, ensuring comprehensive test coverage.
- Provides clear verification and validation processes, enhancing software quality.
- Emphasizes early identification and resolution of defects.

*Disadvantages:*
- Linear and rigid structure, limiting flexibility for changes during development.
- Requires detailed planning and documentation upfront.
- May lead to increased time and effort spent on testing activities.

*Applicability:*

Suitable for projects with well-defined requirements and a focus on ensuring quality through rigorous testing, such as safety critical systems or regulatory compliance projects.

**Conclusion:**

SDLC models vary based on project requirements, complexity, and stakeholder preferences. Waterfall, Agile, spiral, and VModel are suitable for stable, evolving, high-uncertainty, and rigorous testing projects.

--------------------------------------------------------------|end|--------------------------------------------------------------