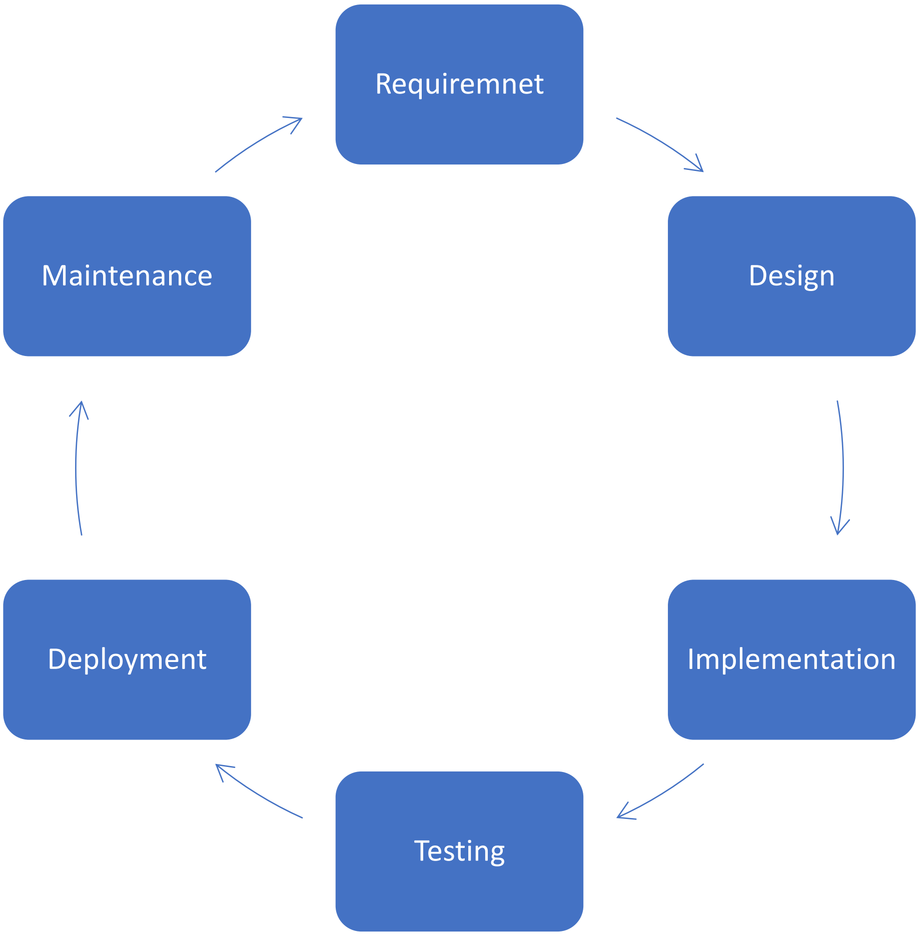
1. SDLC Overview - Create a one-page infographic that outlines the SDLC phases (Requirements, Design, Implementation, Testing, Deployment), highlighting the importance of each phase and how they interconnect.

Ans.



Software Development Life Cycle

1. Requirements

Purpose: Gather and document the needs and objectives of stakeholders to define the project's scope.

Importance: Defines what the system should do, ensuring alignment with business goals and reducing rework.

2. Design

Purpose: Create a blueprint for the system, including architecture, user interfaces, and data structures.

Importance: Establishes a clear roadmap for developers and ensures the system is robust and scalable.

3. Implementation

Purpose: Translate designs into functional code.

Importance: The core phase where the system is built to meet design specifications.

4. Testing

Purpose: Ensure the system functions as intended and meets quality standards.

Importance: Identifies and resolves defects, ensuring a stable and reliable product.

5. Deployment

Purpose: Deliver the completed system to end-users and operational environments.

Importance: Marks the transition from development to production, ensuring the system is accessible and functional.

2. Requirements Gathering - Conduct a 30-minute mock interview to gather requirements for a fictional app that helps organize community events. Summarize the requirements and how you would document and trace them in a one-page brief.

Ans.

Requirements for an app that organize community events.

1.**Event Creation & Management:**

- Create events with date, time, location, description, and event type.
- Manage event details (edit, delete).
- Integration with maps for location display (REQ-001-A).

2.**Communication Hub:**

- Event calendar view.
- Announcement feature (push notifications and in-app messages).
- Discussion forum for event participants.

3.**Volunteer Management:**

- Sign-up system for volunteer tasks.
- Tracking of volunteer assignments.

4.**Shared Item List:**

- Create a shared list for items (e.g., food, drinks, equipment).
- Participants can sign up to bring items.

5.**Budget Management:**

- Track event expenses and income.

6.**Reminders:**

- Automated reminders for upcoming events and volunteer tasks.

Documentation:

Requirements Document: Detailed description of each requirement, including acceptance criteria.

Design Document: Technical specifications, including architecture, UI/UX design, database schema, and algorithms.

Test Cases: Specific tests to verify that the software meets the requirements.

Traceability Matrix: Links requirements to design documents and test cases to ensure complete coverage.

3. Agile Principles Application - Write a two-paragraph reflection on how the Agile values of individuals and interactions, working solutions, and customer collaboration apply to the development of the community event app.

Ans.

The Agile values of individuals and interactions emphasize the importance of collaboration and communication among team members, which is crucial for the development of the community event app. In this project, stakeholders such as event organizers, vendors, and attendees play a central role in defining the app's requirements. By fostering open communication channels and frequent feedback loops, the development team can ensure that the app addresses the actual needs of its users. This focus on human interactions not only strengthens team dynamics but also helps build a product that resonates with its intended audience, leading to higher adoption rates and satisfaction.

Working solutions and customer collaboration further align with the iterative nature of Agile development. By prioritizing functional prototypes and early delivery of key features, the team can gather real-world feedback from users and refine the app incrementally. For example, releasing an MVP (Minimum Viable Product) with essential features like event creation and ticketing allows organizers to start using the app while providing input for future enhancements. This approach ensures that the app evolves in response to user needs, reducing the risk of developing unnecessary features and fostering a sense of partnership between the development team and the community. Together, these Agile values drive the creation of a practical, user-focused app that effectively supports community event organization.

4. Scrum Framework Overview - Prepare a one-page cheat sheet on the Scrum framework that includes roles, responsibilities, artifacts, and ceremonies. Provide a brief example of a Sprint task list for the earlier mentioned app project.

Ans.

Scrum Framework

1.Roles and Responsibilities

Product Owner (PO):

  - Defines and prioritizes the product backlog.
  - Ensures the development team understands the project goals.
  - Acts as the primary point of contact for stakeholders.

- **Scrum Master:**
  - Facilitates Scrum ceremonies and removes impediments.
  - Ensures the team follows Scrum practices.
  - Shields the team from external distractions.
- **Development Team:**
  - Cross-functional members responsible for delivering increments.
  - Self-organizes to meet sprint goals.
  - Collaborates closely to maintain product quality.

2.Artifacts

- **Product Backlog:**
  - A prioritized list of tasks and features to deliver.
  - Owned and maintained by the Product Owner.
- **Sprint Backlog:**
  - Subset of the product backlog selected for the current sprint.
  - Includes tasks broken down for the sprint.
- **Increment:**
  - A potentially shippable product update delivered at the end of a sprint.

3.Ceremonies

- **Sprint Planning:**
  - Collaborative session to define sprint goals and select backlog items.
- **Daily Scrum:**
  - 15-minute stand-up meeting to discuss progress, plans, and blockers.
- **Sprint Review:**
  - Showcase completed work to stakeholders and gather feedback.
- **Sprint Retrospective:**
  - Reflect on the sprint to improve processes and team dynamics.

Example Sprint Task List for Community Event App Project

Sprint Goal: Enable event organizers to create and publish events.

1. **User Authentication:**
   - Implement login and signup functionality.
   - Add password recovery features.
2. **Event Creation Module:**
   - Design event creation UI/UX.
   - Enable input fields for event details (title, description, date, time, location).

3. **Backend Development:**
   - Set up database for storing event details.
   - Develop APIs for event creation and retrieval.
4. **Testing:**
   - Write and execute unit tests for event creation features.
   - Perform end-to-end testing of the event creation process.
5. **Deployment:**
   - Deploy the MVP to a staging environment.
   - Prepare for user feedback collection.