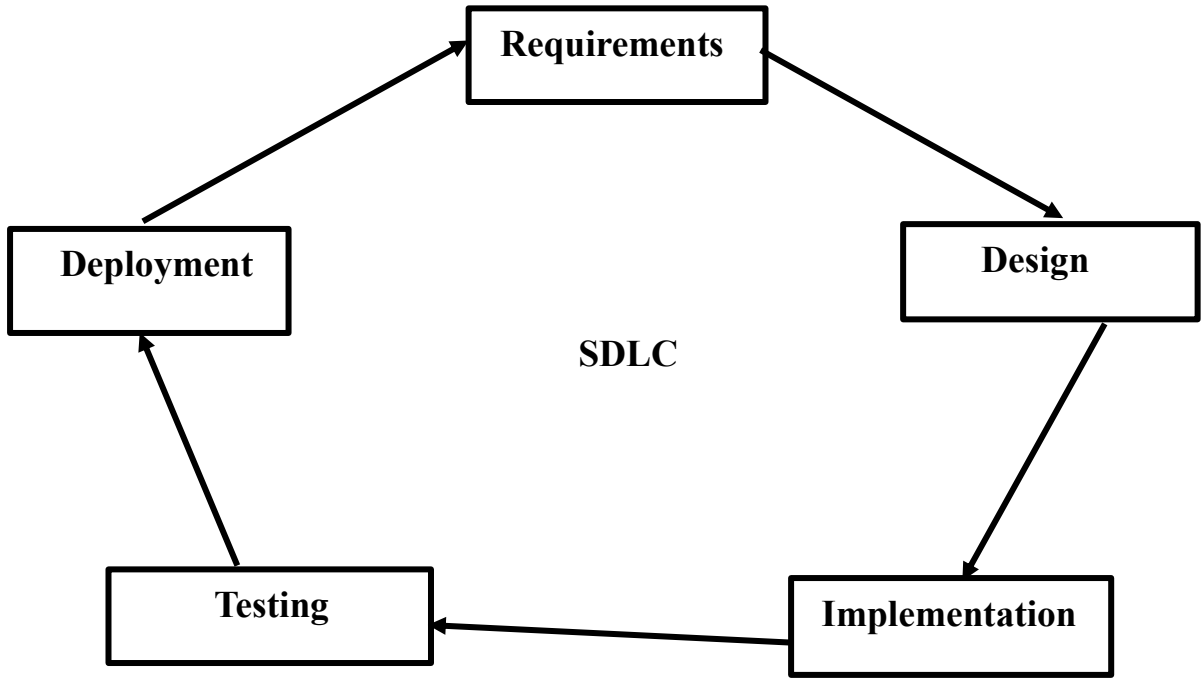
Day 3 -02 May 2024

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.

One of the fundamental procedures of developing software in a step by step manner is by following the Software Development Life Cycle (SDLC). SDLC is a popular practice that is followed by different organizations for designing and developing high-quality software applications. It acts as a framework that holds some specific tasks to be achieved at every phase during the software development progression.

Some specific benefits of using SDLC:

- Increased visibility of the development process for all stakeholders involved
- More efficient estimation, planning, and scheduling
- Improved risk management and cost estimation
- A systematic approach to delivering software that meets customer expectations and improves satisfaction



1. Requirements Phase:

- Understand client needs and project goals.
- Gather requirements through meetings, interviews, and analysis.
- Define project scope, objectives, and deliverables.
- Importance: Foundation of the project, ensures alignment with client expectations.

2. Design Phase:

- Create a blueprint of the system architecture.
- Develop user interface designs and system workflows.
- Define database structure and data flow.
- Importance: Transforms requirements into a structured plan for development.

3. Implementation Phase:

- Write code according to the design specifications.
- Develop modules, features, and functionalities.
- Integrate components to build the complete system.
- Importance: Turning the design into a working software product.

4. Testing Phase:

- Execute various testing methodologies (unit, integration, system, acceptance).
- Identify and fix defects and issues.
- Ensure software quality and compliance with requirements.
- Importance: Validates that the software meets stakeholder expectations.

5. Deployment Phase:

- Roll out the software to production environment.
- Train end-users and provide documentation.
- Monitor performance and address post-deployment issues.
- Importance: Delivers the final product to users and ensures smooth transition.

Interconnection:

- Each phase is interdependent and iterative.
- Feedback loops between phases ensure alignment with requirements.
- Changes in one phase may necessitate revisions in others.
- Continuous communication and collaboration are vital throughout the SDLC.

Assignment 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.

Community Event Organizer App Requirements Brief:

Purpose: Develop an app that simplifies the organization and management of community events.

Target Users:

- Community leaders
- Event organizers
- Local residents

Core Features:

1. **Event Creation:** Users can create event listings with details such as title, description, date, time, location, and images.
2. **RSVP Management:** Attendees can confirm their attendance, and organizers can track participant numbers.
3. **Calendar Integration:** Events can be added to personal calendars with reminders.
4. **Collaborative Planning:** A platform for organizers to collaborate on event planning tasks.
5. **Resource Allocation:** A tool to manage and assign resources like venues, equipment, and volunteers.
6. **Communication Hub:** In-app messaging and announcements for updates and coordination.
7. **Feedback Collection:** Post-event surveys to gather attendee feedback for future improvements.

Non-Functional Requirements:

- **Usability:** Intuitive UI/UX for ease of use.
- **Accessibility:** Compliance with accessibility standards to cater to all users.
- **Scalability:** Ability to handle a growing number of users and events.
- **Security:** Protection of user data and secure payment gateways for donations.

Documentation and Traceability:

- **Requirement Documentation:** Each requirement will be documented in a shared online platform with version control.
- **User Stories:** Requirements will be translated into user stories to guide development.
- **Traceability Matrix:** A traceability matrix will be created to ensure each requirement is accounted for in the design and development phases.
- **Change Log:** Any changes to requirements will be recorded in a change log with justifications.

Stakeholder Engagement:

- Regular meetings with community leaders and potential users to validate and refine requirements.
- Prototyping sessions to gather early feedback on app features and usability.

Testing and Validation:

- Continuous testing throughout development to ensure requirements are met.
- User acceptance testing with a select group of community members before launch.

Conclusion: This app aims to streamline the organization of community events, fostering greater participation and collaboration within communities. The documentation and traceability processes will ensure a clear roadmap from requirements gathering to the final product.

This brief outlines the foundational requirements and the approach to managing them throughout the development lifecycle of the app. The focus is on creating a user-friendly, efficient, and secure platform that meets the needs of community event organizers and participants.

Assignment 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.

In the development of the community event app, the Agile value of "individuals and interactions over processes and tools" underscores the importance of fostering effective communication and collaboration among team members. By prioritizing direct and open communication channels, such as daily stand-up meetings and regular team check-ins, developers, designers, and stakeholders can quickly address challenges, share ideas, and align their efforts towards achieving the project's goals. This emphasis on interpersonal dynamics not only enhances team morale and cohesion but also leads to more innovative solutions and a deeper understanding of user needs.

Moreover, the Agile principle of "working solutions over comprehensive documentation" is particularly relevant to the iterative nature of app development. Rather than aiming for perfection in the initial release, Agile advocates for delivering functional increments of the app that provide tangible value to users. This approach allows for continuous feedback from stakeholders and end-users, enabling the team to adapt and refine the app in response to changing requirements and market dynamics. By prioritizing the delivery of working features and prototypes, the development process becomes more responsive, adaptive, and ultimately more likely to meet the evolving needs of the community event organizers and attendees.

Assignment 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.

Scrum Framework Cheat Sheet

Roles:

- **Product Owner:** Represents stakeholders, prioritizes product backlog, and defines features.
- **Scrum Master:** Facilitates Scrum events, removes impediments, and coaches the team on Agile practices.
- **Development Team:** Self-organizing group responsible for delivering increments of the product.

Responsibilities:

- **Product Owner:** Defines and prioritizes product backlog items, clarifies requirements, and accepts/rejects work results.
- **Scrum Master:** Ensures Scrum framework adherence, removes obstacles, facilitates meetings, and supports the team.
- **Development Team:** Estimates tasks, selects work from the sprint backlog, collaborates to deliver high-quality increments.

Artifacts:

- **Product Backlog:** Prioritized list of features, enhancements, and bug fixes.
- **Sprint Backlog:** Subset of the product backlog items selected for the sprint.
- **Increment:** Potentially shippable product increment at the end of each sprint.

Ceremonies:

- **Sprint Planning:** Plan the work for the sprint.
- **Daily Standup:** Daily check-in to share progress, discuss obstacles, and plan the day's work.
- **Sprint Review:** Review the increment and gather feedback.
- **Sprint Retrospective:** Reflect on the sprint and identify areas for improvement.

Example Sprint Task List for Community Event App:

Sprint Goal: Implement basic event creation and management features.

1. Design user interface for event creation form.
2. Develop backend logic for storing event details in the database.
3. Implement front-end validation for event date and time.
4. Integrate third-party API for location autocomplete in event form.
5. Create user stories for attendee management functionalities.
6. Design database schema for attendee data storage.
7. Develop backend endpoints for attendee registration and RSVP tracking.
8. Implement email notification system for event updates and reminders.
9. Conduct user acceptance testing for event creation and attendee management features.
10. Refactor codebase for improved maintainability and scalability.