

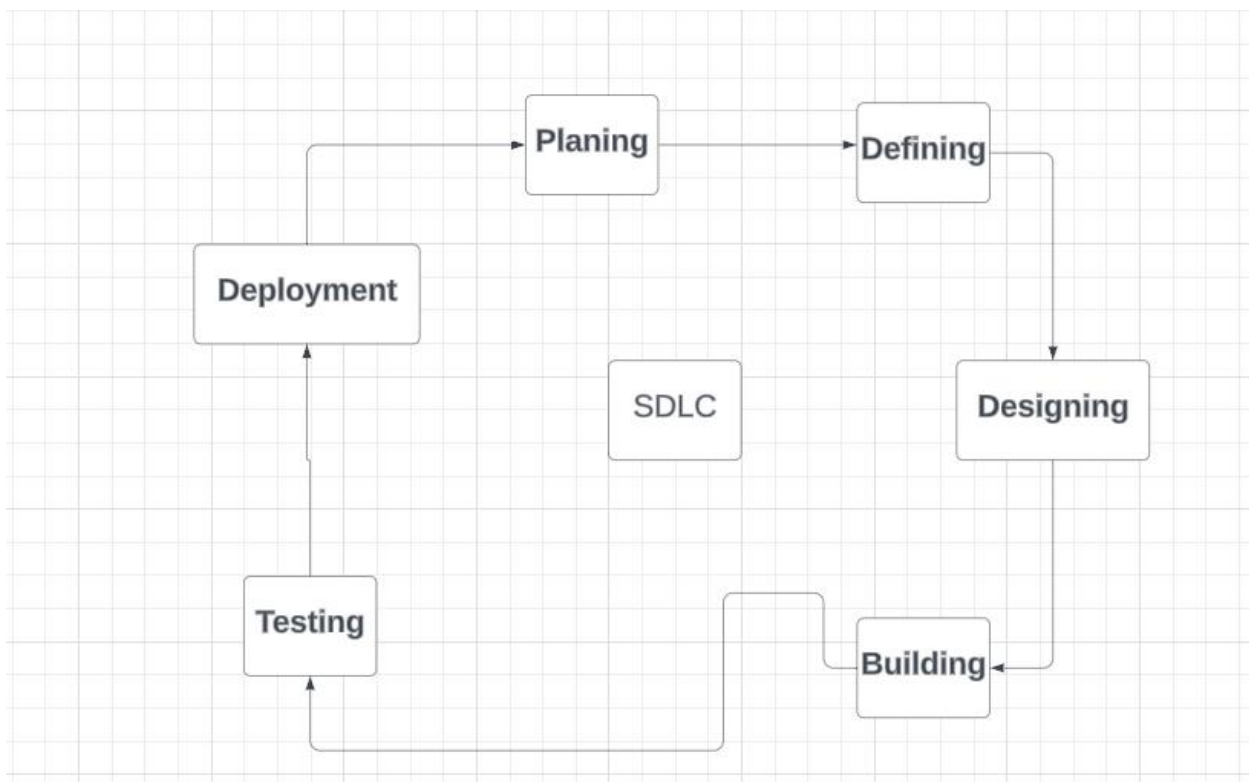
Assignment 1:

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

SOLUTION:

SDLC is a process followed for a software project, within a software organization. It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software. The life cycle defines a methodology for improving the quality of software and the overall development process.

The following figure is a graphical representation of the various stages of a typical SDLC.



A typical Software Development Life Cycle consists of the following stages

Step 1:

Planning:-

Requirement analysis is the most important and fundamental stage in SDLC. It is performed by the senior members of the team with inputs from the customer, the sales department, market surveys and domain experts in the industry. This information is then used to plan the basic project approach and to conduct product feasibility study in the economical, operational and technical areas.

Planning for the quality assurance requirements and identification of the risks associated with the project is also done in the planning stage. The outcome of the technical feasibility study is to define the various technical approaches that can be followed to implement the project successfully with minimum risks.

Step 2:

Defining Requirements:-

Once the requirement analysis is done the next step is to clearly define and document the product requirements and get them approved from the customer or the market analysts. This is done through an **SRS (Software Requirement Specification)** document which consists of all the product requirements to be designed and developed during the project life cycle.

Step 3:

Designing the Product Architecture:-

SRS is the reference for product architects to come out with the best architecture for the product to be developed. Based on the requirements specified in SRS, usually more than one design

approach for the product architecture is proposed and documented in a DDS - Design Document Specification.

This DDS is reviewed by all the important stakeholders and based on various parameters as risk assessment, product robustness, design modularity, budget and time constraints, the best design approach is selected for the product.

A design approach clearly defines all the architectural modules of the product along with its communication and data flow representation with the external and third party modules (if any). The internal design of all the modules of the proposed architecture should be clearly defined with the minutest of the details in DDS.

Step 4:

Building or Developing the Product:-

In this stage of SDLC the actual development starts and the product is built. The programming code is generated as per DDS during this stage. If the design is performed in a detailed and organized manner, code generation can be accomplished without much hassle.

Developers must follow the coding guidelines defined by their organization and programming tools like compilers, interpreters, debuggers, etc. are used to generate the code. Different high level programming languages such as C, C++, Pascal, Java and PHP are used for coding. The programming language is chosen with respect to the type of software being developed.

Step 5:

Testing the Product:-

This stage is usually a subset of all the stages as in the modern SDLC models, the testing activities are mostly involved in all the stages of SDLC. However, this stage refers to the testing only stage of the product where product defects are reported, tracked, fixed

and retested, until the product reaches the quality standards defined in the SRS.

Stage 6:

Deployment in the Market and Maintenance:-

Once the product is tested and ready to be deployed it is released formally in the appropriate market. Sometimes product deployment happens in stages as per the business strategy of that organization. The product may first be released in a limited segment and tested in the real business environment (UAT- User acceptance testing).

Then based on the feedback, the product may be released as it is or with suggested enhancements in the targeting market segment. After the product is released in the market, its maintenance is done for the existing customer base.

SDLC Models:-

There are various software development life cycle models defined and designed which are followed during the software development process. These models are also referred as Software Development Process Models". Each process model follows a Series of steps

unique to its type to ensure success in the process of software development.

Following are the most important and popular SDLC models followed in the industry –

- Waterfall Model
- Iterative Model
- Spiral Model
- V-Model
- Big Bang Model

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.

BookMyShow Project

Introduction:

BookMyShow is a popular online platform for booking movie tickets, events, and other entertainment activities. The project aims to develop a robust and user-friendly application that allows users to easily browse, select, and book tickets for movies, concerts, plays, and other events. The application will provide a seamless experience for users to discover events, view show times, select seats, make payments, and receive tickets, all from the comfort of their own devices.

1. Planning Phase:

- **Requirement Gathering:** Gather requirements from stakeholders including users, administrators, and management to understand the functionalities and features required in the application.
- **Feasibility Study:** Assess the technical and economic feasibility of the project to determine its viability.
- **Project Scope:** Define the scope of the project, including features, functionalities, timelines, and resources required.
- **Risk Assessment:** Identify potential risks and develop strategies to mitigate them.

2. Analysis Phase:

- **System Analysis:** Analyze the requirements gathered in the planning phase and translate them into detailed system specifications.
- **Prototyping:** Develop prototypes or mockups to visualize the user interface and gather feedback from stakeholders.
- **Data Modeling:** Design the database schema and establish data relationships to support the application's functionalities.

3. Design Phase:

- **Architecture Design:** Define the overall architecture of the application, including the client-server model, database structure, and communication protocols.
- **UI/UX Design:** Design the user interface to ensure a seamless and intuitive user experience.
- **Database Design:** Design the database schema, including tables, indexes, and constraints, based on the data model developed in the analysis phase.
- **Security Design:** Implement security measures to protect user data and prevent unauthorized access.

4. Development Phase:

- **Frontend Development:** Develop the frontend of the application using technologies such as HTML, CSS, and JavaScript to create the user interface.
- **Backend Development:** Develop the backend logic using programming languages like Java, Python, or Node.js to handle user requests, process data, and interact with the database.
- **Database Implementation:** Implement the database using technologies like MySQL, MongoDB, or PostgreSQL, and integrate it with the backend logic.

5. Testing Phase:

- **Unit Testing:** Test individual components and modules of the application to ensure they function correctly in isolation.
- **Integration Testing:** Test the integration between different components and modules to verify that they work together as expected.
- **System Testing:** Test the entire system to ensure that it meets the specified requirements and performs as intended.
- **User Acceptance Testing (UAT):** Allow end-users to test the application and provide feedback to identify any issues or areas for improvement.

6. Deployment Phase:

- **Deployment Planning:** Plan the deployment strategy, including server setup, configuration, and deployment process.
- **Deployment:** Deploy the application to the production environment and make it available to users.
- **Post-Deployment Testing:** Perform additional testing in the production environment to ensure that the application functions correctly and is accessible to users.

7. Maintenance Phase:

- **Bug Fixes:** Address any bugs or issues identified post-deployment through regular maintenance and updates.
- **Feature Updates:** Incorporate new features and functionalities based on user feedback and changing requirements.
- **Performance Optimization:** Monitor and optimize the performance of the application to ensure scalability and reliability.

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

Individuals and Interactions:

- Regular stand-up meetings and brainstorming sessions foster collaboration and idea-sharing.
- Face-to-face communication promotes a positive team dynamic and ensures everyone's input is valued.
- Emphasis on teamwork leads to innovative solutions and a stronger sense of ownership over the project.

Working Solutions:

- Agile methodology prioritizes building functioning features incrementally, ensuring continuous progress.
- Iterative development allows for quick adaptation to changing requirements and feedback.
- Focus on delivering tangible value to users through frequent releases and updates.

Customer Collaboration:

- Regular engagement with community stakeholders ensures the app meets their specific needs and preferences.
- User feedback drives iterative improvements, leading to a product tailored to the user's requirements.
- Collaborative approach fosters a sense of partnership between developers and users, resulting in a more user-centric product.

Customer Collaboration in Community Event App Development:-

- ❖ **User Feedback Integration:** -Actively solicit feedback from community members through surveys, interviews, and feedback forms.
- ❖ **Feature Prioritization:** - Collaborate with users to prioritize features based on their needs, preferences, and pain points.
- ❖ **Iterative Development:** - Involve users in iterative development cycles, showcasing prototypes and gathering feedback to refine features.
- ❖ **Transparency:** - Maintain transparent communication with users regarding development progress, timelines, and feature releases.
- ❖ **User Testing:** - Conduct user testing sessions to validate assumptions, ensure usability, and gather insights for further improvements.
- ❖ **Continuous Engagement:** - Maintain ongoing communication channels with users to gather feedback, address concerns, and foster a sense of community ownership.
- ❖ **Adaptation to User Needs:** -Continuously adapt development priorities and strategies based on user feedback and evolving community requirements.
- ❖ **Empowerment:** - Empower users to contribute ideas, suggestions, and even participate in co-creation workshops to shape the direction of the app.
- ❖ **Feedback Incorporation:** - Actively incorporate user feedback into development iterations, demonstrating responsiveness to user needs and preferences.
- ❖ **Long-term Relationship Building:** - Foster long-term relationships with users by demonstrating a commitment to collaboration, responsiveness, and ongoing improvement.

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

Scrum is an agile framework for managing and completing complex projects. It emphasizes teamwork, flexibility, and continuous improvement.

Roles:

Scrum Master: Facilitates the team's progress, removes impediments, and ensures Scrum practices are followed.

Product Owner: Prioritizes product features, manages the product backlog, and communicates with stakeholders.

Development Team: Self-organizing, cross-functional group responsible for delivering a working product increment during each Sprint.

Responsibilities:

Scrum Master: Guides the team in Scrum practices, protects the team from distractions, and coaches the organization.

Product Owner: Maximizes the value of the product, manages the product backlog, and makes prioritization decisions.

Development Team: Develops the product, creates a working product increment during each Sprint, and maintains the definition of "Done."

Artifacts:

Product Backlog: A prioritized list of product features, requirements, and bugs.

Sprint Backlog: A set of product backlog items selected for the Sprint, along with a plan for delivering them.

Increment: A usable, potentially releasable product portion at the end of each Sprint.

Ceremonies:

Sprint Planning: The Development Team and Product Owner collaborate to create the Sprint Backlog.

Daily Scrum (Stand up): A 15-minute daily meeting for the Development Team to discuss progress and plan for the day.

Sprint Review: A meeting to inspect the Increment and demonstrate its value. The Product Owner reviews the Product Backlog for future adjustments.

Sprint Retrospective: The team reflects on the Sprint process, identifies improvements, and plans for the next Sprint.

Example Sprint Task List for App Project:

- ❖ User Interface Design (Designer)
- ❖ Database Setup and Integration (Developer)
- ❖ Login and Registration Functionality (Developer)
- ❖ User Profile Creation and Management (Developer)
- ❖ Information Retrieval Feature Implementation (Developer)
- ❖ Task Delegation and Reminder System Development (Developer)
- ❖ AI Model Training for Information Retrieval (Developer)
- ❖ Test Cases Creation for Login and Registration

Assignment 5:

Agile Project Planning - Create a one-page project plan for a new software feature using Agile planning techniques. Include backlog items with estimated story points and a prioritized list of user stories.

Project Name: Customer Feedback Module 2.0

Objective: Improve customer satisfaction by adding a feedback feature.

Team: Product Owner, Scrum Master, Developers

Backlog:

1. Easy feedback submission (5 StoryPoints)
2. View feedback history (8 SP)
3. Feedback rating (3 SP)
4. Admin moderation (8 SP)
5. Feedback status updates (5 SP)

Sprints:

- Sprint 1: Submission & basic viewing (1 week)
- Sprint 2: Rating & admin moderation (1 week)
- Sprint 3: Advanced viewing & status updates (3 days)

Release Date: June 21, 2024

Minimum Viable Product : Easy submission & basic viewing

Stretch Goals: Rating, admin features, status updates

Notes:

- Daily meetings at 9
- AM.
- Review & feedback after each sprint.

Assignment 6:

Daily Stand-up Simulation - Write a script for a Daily Stand-up meeting for a development team working on the software feature from Assignment 1. Address a common challenge and incorporate a solution into the communication flow.

Title: Daily Stand-up Meeting - Software Feature Development Team

Time: 9:00 AM

Location: Virtual Meeting Room

Participants:

John (Scrum Master)

Alice (Developer)

Bob (Tester)

Sarah (Designer)

John:

- Good morning, everyone.
- Let's start today's Daily Stand-up meeting. Please introduce yourself, share your progress since yesterday, and mention any challenges you're facing.
- We'll also discuss a common challenge and brainstorm a solution today. Let's begin.

Alice:

- Hi John, Bob, and Sarah.
- I'm Alice, a developer working on AI integration.
- Yesterday, I completed the first phase of the AI model training. Today, I'll be working on integrating it with the user interface.

Bob:

- Good morning, team.
- I'm Bob, the tester. I've been working on creating test cases for the user

authentication feature.

- I've almost finished drafting them, and I'll start executing them today.

Sarah: • Hello, everyone. I'm Sarah, the designer.

- Yesterday, I finalized the wire-frames for the task delegation and reminder system.
- Today, I'll be working on the visual design for the information retrieval feature.

John:

- Thank you for your updates, everyone. Now, let's discuss a common challenge we often face during the development process – communication gaps between team members.
- This can lead to misunderstandings, delays, and rework. To address this, we can implement a more structured communication process.

Alice:

- That's a great idea, John. We can use a shared online document or a dedicated communication channel where we can update our progress and ask questions throughout the day. This way, everyone stays informed, and we can quickly address any issues.

Bob:

- I agree, Alice. Additionally, we can also schedule quick check-ins with team members working on related tasks to ensure we're on the same page and avoid miscommunication.

Sarah:

- I like the suggestions, and I'd also add that we should encourage each other to ask questions or seek clarification whenever needed. This will help us maintain a collaborative environment and improve our overall efficiency.

John:

- Excellent points, everyone. Let's adopt these practices in our daily work and see how they improve our communication and collaboration. Now, if there are no further updates or concerns, the Daily Stand-up meeting is adjourned.

- Thank you all.

In this Daily Stand-up meeting script, the team discusses a common challenge – communication gaps – and proposes solutions to address it. By implementing a structured communication process and encouraging open dialogue, the team