Koduru Pushpalatha

**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.**

Software Development Life Cycle(SDLC):

1.Requirement Phase

It defines the project scope,objectives, and functionalities. Basis for all subsequent phases. Clear requirements ensure alignment between stakeholders and developers.

1. Design Phase

Which will translates requirements into a blueprint for development. Design serves as the foundation for inmplementation, ensuring the end product meets specifications.

1. Implementation Phase

Coding and development of the software based on the design. Direct translation of design into executable code, laying the groundwork for testing.

1. Testing Phase

Ensures the software meets quality standards and functions as intended.It Validates that the implementation aligns with requirements and design specifications before deployment.

1. Deployment Phase

Launches the software into production or distribution. Marks the culmination of the SDLC, transitioning the product from development to end-users.

**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.**

*Case study Implementation* of SDLC Phases in a Real-World Engineering Project

Project Overview: Company The spice, a leading software development firm, undertook a project to develop a new customer relationship management (CRM) system for a multinational corporation. The project aimed to streamline customer interactions, improve data management, and enhance overall efficiency.

SDLC Phases Implementation:

1.Requirement Gathering:

Importance: Company XYZ initiated the project by conducting extensive stakeholder interviews, surveys, and market research to gather comprehensive requirements.

Outcome: Clear understanding of client needs, business objectives, and user expectations laid a solid foundation for subsequent phases, ensuring alignment with stakeholders.

2.Design

Importance: Based on gathered requirements, the design phase involved creating wireframes, architecture diagrams, and UI/UX mockups to visualize the CRM system's structure and functionalities.

Outcome: Well-defined design blueprints facilitated effective communication between developers, designers, and clients, guiding the development process with clarity and precision.

1. Implementation

Importance: Skilled development teams utilized modern programming languages, frameworks, and best practices to translate design specifications into functional software components.

Outcome: Timely execution of coding tasks, adherence to coding standards, and effective collaboration among developers ensured the creation of a robust and scalable CRM system.

1. Testing

Importance: Rigorous testing methodologies, including unit testing, integration testing, and user acceptance testing (UAT), were employed to identify and rectify defects, ensuring product quality and reliability.

Outcome: Early detection and resolution of bugs, along with validation of system functionalities against user requirements, minimized post-deployment issues and enhanced end-user satisfaction.

1. Deployment Importance:

Following successful testing, the CRM system was deployed in a phased approach, with thorough deployment plans and rollback strategies in place to mitigate risks.

Outcome: Smooth deployment process with minimal disruptions to business operations, leading to immediate utilization of the CRM system and realization of project objectives.

6.Maintenance

Importance: Post-deployment, Company XYZ established a dedicated maintenance team responsible for monitoring system performance, addressing user feedback, and implementing necessary updates and enhancements.

Outcome: Continuous improvement of the CRM system's features, performance optimization, and timely bug fixes ensured long-term reliability and sustainability, meeting evolving business needs.

Evaluation of project Outcomes:

The systematic implementation of SDLC phases facilitated the successful development and deployment of the CRM system, meeting client requirements and exceeding expectations.

Effective requirement gathering ensured alignment with stakeholder needs, while robust design, implementation, testing, and deployment processes ensured product quality, reliability, and user satisfaction.

Ongoing maintenance and support activities ensured the longevity and adaptability of the CRM system, contributing to enhanced business efficiency and competitive advantage for the client.

In conclusion, the comprehensive integration of Requirement Gathering, Design,

Implementation, Testing, Deployment, and Maintenance phases in the SDLC played a pivotal role in the successful execution and outcome of the engineering project, demonstrating the importance of systematic software development methodologies in real-world scenarios.

**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 context. Comparision of SDLC Models for Engineering Projects**

1.Waterfall Model:

Advantages:

Simple and easy to understand

Well-suited for projects with stable requirements and predefined scope Each phase has specific deliverables and milestones, facilitating process tracking.

Disadvantages:

Lack of flexibility to accommodate changes in requirements.

Limited customer involvement until late stages, potentially leading to misunderstandings or dissatisfaction.

High risk of project failure if initial requirements are inaccurate or incomplete.

Applicability:Waterfall model is suitable for projects with well-defined and stable requirements, such as construction projects or manufacturing processes.

2. Agile Model:

Advantages:

Flexibility to adapt to changing requirements through iterative development and continuous feedback.

Enhanced collaboration between development teams and stakeholders, fostering a customer-centric approach.

Early delivery of working software components allows for rapid validation and course correction.

Disadvantages:

Requires active involvement and commitment from stakeholders throught the project.

May face the challenges in maintaining documentation and ensuring long-term scalability.

Complec projects with strict regulatory requirements may find Agile methodologies challenging to implement.

Applicability: Agile model is ideal for software development projects where requirements are dynamic or not fully known upfront, enabling quick response to market changes or customer feedback.

3. Spiral Model:

Advantages:

Incorporates risk management throughout the development lifecycle, with iterative cycles focusing on risk analysis and mitigation.

Allows for incremental development while accommodating changes in requirements or technology.

Well-suited for large-scale projects with high levels of complexity and uncertanity.

Disadvantages:

Resource-intensive due to the emphasis on risk analysis and extensive documantation.

Requires experienced project management and technical expertise to effectively manage risks and iterations.

May result in longer development cycles compared to other models.

Applicability: Spiral model is suitable for projects with evolving requirements significant technical challenges, or high-risk factors, such as software for critical infrastructure or aerospace systems.

4. V-Model:

Advantages:

Emphasizes a structured approach with clear validation and verification processes aligned with each development phase.

Ensures early identification and resolution of defects through parallel testing activities.

Well-defined deliverables and traceability between requirements and test cases enhance quality assurance.

Disadvanatges:

Sequential nature may lead to delays in project timelines, especially if defects are discovered late in the development cycle.

Limited flexibility to accommodate changes in requirements or scope after initial planning. Heavy reliance on documentation and formal processes may hinder agility and innovation.

Applicability: V-Model is suitable for projects with well-defined requirements and strict regulatory compliance, such as medical device development or government contracts.

Conclusion: Each SDLC model offers unique advantages and challenges, making them suitable for different engineering contexts depending on project requirements, complexity, and risk tolerance. Understanding the characteristics and trade-offs of each model is essential for selecting the most appropriate approach to ensure project success.

**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.**

Software Development Life Cycle(SDLC):

1.Requirement Phase

It defines the project scope,objectives, and functionalities. Basis for all subsequent phases. Clear requirements ensure alignment between stakeholders and developers.

1. Design Phase

Which will translates requirements into a blueprint for development. Design serves as the foundation for inmplementation, ensuring the end product meets specifications.

1. Implementation Phase

Coding and development of the software based on the design. Direct translation of design into executable code, laying the groundwork for testing.

1. Testing Phase

Ensures the software meets quality standards and functions as intended.It Validates that the implementation aligns with requirements and design specifications before deployment.

1. Deployment Phase

Launches the software into production or distribution. Marks the culmination of the SDLC, transitioning the product from development to end-users.

**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.**

*Case study Implementation* of SDLC Phases in a Real-World Engineering Project

Project Overview: Company The spice, a leading software development firm, undertook a project to develop a new customer relationship management (CRM) system for a multinational corporation. The project aimed to streamline customer interactions, improve data management, and enhance overall efficiency.

SDLC Phases Implementation:

1.Requirement Gathering:

Importance: Company XYZ initiated the project by conducting extensive stakeholder interviews, surveys, and market research to gather comprehensive requirements.

Outcome: Clear understanding of client needs, business objectives, and user expectations laid a solid foundation for subsequent phases, ensuring alignment with stakeholders.

2.Design

Importance: Based on gathered requirements, the design phase involved creating wireframes, architecture diagrams, and UI/UX mockups to visualize the CRM system's structure and functionalities.

Outcome: Well-defined design blueprints facilitated effective communication between developers, designers, and clients, guiding the development process with clarity and precision.

1. Implementation

Importance: Skilled development teams utilized modern programming languages, frameworks, and best practices to translate design specifications into functional software components.

Outcome: Timely execution of coding tasks, adherence to coding standards, and effective collaboration among developers ensured the creation of a robust and scalable CRM system.

1. Testing

Importance: Rigorous testing methodologies, including unit testing, integration testing, and user acceptance testing (UAT), were employed to identify and rectify defects, ensuring product quality and reliability.

Outcome: Early detection and resolution of bugs, along with validation of system functionalities against user requirements, minimized post-deployment issues and enhanced end-user satisfaction.

1. Deployment Importance:

Following successful testing, the CRM system was deployed in a phased approach, with thorough deployment plans and rollback strategies in place to mitigate risks.

Outcome: Smooth deployment process with minimal disruptions to business operations, leading to immediate utilization of the CRM system and realization of project objectives.

6.Maintenance

Importance: Post-deployment, Company XYZ established a dedicated maintenance team responsible for monitoring system performance, addressing user feedback, and implementing necessary updates and enhancements.

Outcome: Continuous improvement of the CRM system's features, performance optimization, and timely bug fixes ensured long-term reliability and sustainability, meeting evolving business needs.

Evaluation of project Outcomes:

The systematic implementation of SDLC phases facilitated the successful development and deployment of the CRM system, meeting client requirements and exceeding expectations.

Effective requirement gathering ensured alignment with stakeholder needs, while robust design, implementation, testing, and deployment processes ensured product quality, reliability, and user satisfaction.

Ongoing maintenance and support activities ensured the longevity and adaptability of the CRM system, contributing to enhanced business efficiency and competitive advantage for the client.

In conclusion, the comprehensive integration of Requirement Gathering, Design,

Implementation, Testing, Deployment, and Maintenance phases in the SDLC played a pivotal role in the successful execution and outcome of the engineering project, demonstrating the importance of systematic software development methodologies in real-world scenarios.

**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 context. Comparision of SDLC Models for Engineering Projects**

1.Waterfall Model:

Advantages:

Simple and easy to understand

Well-suited for projects with stable requirements and predefined scope Each phase has specific deliverables and milestones, facilitating process tracking.

Disadvantages:

Lack flexibility to accommodate changes in requirements.

Limited customer involvement until late stages, potentially leading to misunderstandings or dissatisfaction.

High risk of project failure if initial requirements are inaccurate or incomplete.

Applicability:Water model is suitable for projects with well defined and stable requirements, such as construction projects or manufacturing processes.

2. Agile Model:

Advantages:

Flexibility to adapt to changing requirements through iterative development and continuous feedback.

Enhanced collaboration between development teams and stakeholders, fostering a customer-centric approach.

Early delivery of working software components allows for rapid validation and course correction.

Disadvantages:

Requires active involvement and commitment from stakeholders throught the project.

May face the challenges in maintaining documentation and ensuring long-term scalability.

Complec projects with strict regulatory requirements may find Agile methodologies challenging to implement.

Applicability: Agile model is ideal for software development projects where requirements are dynamic or not fully known upfront, enabling quick response to market changes or customer feedback.

3. Spiral Model:

Advantages:

Incorporates risk management throughout the development lifecycle, with iterative cycles focusing on risk analysis and mitigation.

Allows for incremental development while accommodating changes in requirements or technology.

Well-suited for large-scale projects with high levels of complexity and uncertanity.

Disadvantages:

Resource-intensive due to the emphasis on risk analysis and extensive documantation.

Requires experienced project management and technical expertise to effectively manage risks and iterations.

May result in longer development cycles compared to other models.

Applicability: Spiral model is suitable for projects with evolving requirements significant technical challenges, or high-risk factors, such as software for critical infrastructure or aerospace systems.

4. V-Model:

Advantages:

Emphasizes a structured approach with clear validation and verification processes aligned with each development phase.

Ensures early identification and resolution of defects through parallel testing activities.

Well-defined deliverables and traceability between requirements and test cases enhance quality assurance.

Disadvanatges:

Sequential nature may lead to delays in project timelines, especially if defects are discovered late in the development cycle.

Limited flexibility to accommodate changes in requirements or scope after initial planning. Heavy reliance on documentation and formal processes may hinder agility and innovation.

Applicability: V-Model is suitable for projects with well-defined requirements and strict regulatory compliance, such as medical device development or government contracts.

Conclusion: Each SDLC model offers unique advantages and challenges, making them suitable for different engineering contexts depending on project requirements, complexity, and risk tolerance. Understanding the characteristics and trade-offs of each model is essential for selecting the most appropriate approach to ensure project success.

**Assignment 1: 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.**

**Example:**

**Project Name:** Chatbot featureEnhancement

Product Owner:Pushpa Royal

Scrum Master:Hillers Paul

Development Team Manager: Chayan Adhikari

Backlog Items:

1.User story: As a user, I want to upload the images to resolve my query quickly.

Acceptance Criteria: The image should be in with the extension of .jpeg, .jpg, .png and the size of that image should accept upto 4kb.

Story Points:5

2. User story: As a user, I want to receive real-time notifications for new messages.

Acceptance Criteria: Users have to allow third party cookies to get notifications

Story Points:4

3.User story: As a user, I want to upload videos over chatbot

Acceptance Criteria: The video shoud not be extend size of 4kb.

Story Points:7

Prioritized User Stories:

1.User Story:Upload images

2. User Story: Allow notifications

3. User Story: Upload videos

Sprint Planning:

Sprint Planning: 2 Weeks

Sprint Goal: Implement basic functionalities

Sprint Review:

Demo the implemented user stories

Gather feedback from stakeholders

Discuss any adjustments to backlog priorities.

Sprint Retrospective:

Identify what went well and what could be improved in the sprint

Discuss any implements or blockers faced by the team.

Plan actionable items for improvement in the next sprint

By using the above one page planning , the team can be easliy develop the applications with team collaboration.

**Assignment 2:** Daily Standup Simulation - Write a script for a Daily Standup 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.

Daily stand up meeting script:

Scrum Master:Hey everyone, we are good to start with today updates, please go ahead

Dev Team member1: Hi everyone, I am denver today I worked on the ticket 9809 ticket where I have to implement the allow notifications on user screen, I have figured out the way to do that and I started working on that today and I will continue work on that tomorrow also. Thank you.

Team mem 2:Hey everyone, I am Siree, today I have started working on implementing the feature of uploading the videos and images where I am facing some issue while integrating the API with backend and very soon I will resolve that issue, that’s all for today . Thank you.

Team mem3: Hi all,I am tracy, I almost finished with the uploading the videos feature today , I will wrap up that issuse by tommorow EOD and I just need a confirmation from Chayan, once I get that I can proceed further.

Chayan : Sure Tracy, we will connect after this call.

Scrum master: We have time of 2 weeks of time to finish this project, so make sure everybody done with their stuff we need to wrap this by the end of the sprint. If nothing is there, we can wrap up for today’s call. Thank you.

.