NAME : RUTVIK B NAKUM
ID : 202312017
SUB : S.E LAB-1

1. Write a summary in your own words for SDLC.

1. Requirements Gathering and Analysis:

The initial phase of the software development life cycle involves extracting requirements from stakeholders and conducting a thorough analysis. This step aims to comprehend the project's scope and objectives.

2. Design Phase:

Following the understanding of requirements, the design phase commences. This stage focuses on creating a comprehensive design document outlining software architecture, user interface, and system components.

3. Implementation and Unit Testing:

During the implementation phase, the software is coded based on the design specifications. Simultaneously, unit testing is performed to verify the functionality of each software component.

4. Integration and System Testing:

The testing phase evaluates the software as a whole to ensure it aligns with requirements and is free from defects. This step is crucial to guarantee the overall integrity and functionality of the system.

5. Deployment:

Once the software has successfully passed testing and received approval, it is deployed to the production environment. This phase marks the transition from development to the operational stage.

6. Maintenance:

The final phase involves ongoing maintenance, addressing any issues that arise post-deployment. This ensures the software continues to meet requirements and remains functional over time. The Waterfall Model emphasizes a sequential flow, with each phase building upon the completion of the preceding one. There is no overlap between phases in this classical model.

2. Identify different tools used in each phase of SDLC. Explore TWO tools for each phase of the SDLC and write a 500 word paragraph on what are your findings about each of that tool.

1. Requirements Gathering:

1.1. Jira:

Jira helps teams manage and prioritize project tasks efficiently. It allows stakeholders to collaborate, ensuring a clear understanding of project goals. Jira's customizable features and real-time collaboration make it a great tool for requirements gathering.

1.2. Confluence:

Confluence, by Atlassian, is a tool for collaborative documentation. It integrates with Jira, providing a space for creating and sharing project documentation. Confluence supports organized documentation, making it a valuable asset during the requirements gathering phase.

2. System Design:

2.1. Lucidchart:

Lucidchart is a diagramming tool that aids in visually representing system architecture. It supports various diagram types, promoting real-time collaboration. Integration with other tools makes Lucidchart a versatile choice for system design.

2.2. Enterprise Architect:

Enterprise Architect by Sparx Systems is a comprehensive modeling tool supporting UML and SysML. It offers advanced features for precise system design, model-driven development, and integration with version control systems.

3. Implementation (Coding):

3.1. Eclipse:

Eclipse is an open-source IDE supporting multiple programming languages. Its features include code completion, debugging, and collaboration tools, making it ideal for team-based coding projects.

3.2 Git:

Git is a version control system crucial for managing code changes. It facilitates collaboration, branching, and merging, ensuring code reliability. Git, along with platforms like GitHub and GitLab, supports efficient code integration.

4. Testing:

4.1 Selenium:

Selenium is an open-source testing framework for web applications. It automates functional and regression testing, contributing to the reliability and quality of web applications.

4.2 Jenkins:

Jenkins is an open-source automation server that automates testing processes. It integrates with testing tools and version control systems, providing a CI/CD solution and facilitating collaboration between development and testing teams.

5. Deployment and Maintenance:

5.1. Docker:

Docker is a containerization platform simplifying application deployment. It packages applications with dependencies into containers, ensuring consistency between development and production environments.

5.2 New Relic:

New Relic is an APM tool offering real-time monitoring and troubleshooting capabilities. It helps maintain application performance, ensuring a positive user experience in production.

3. Prepare a report on software development projects that you have developed till now.

This project is about making a useful computer program to help schools manage things better. We want to make it easier for schools to handle student information, communicate with everyone involved (like teachers, students, and parents), and keep track of important academic stuff

Goals:

- Make it simple for schools to manage student data, attendance, and grades.
- Improve communication between teachers, students, parents, and administrators.
- Create a central place for lesson plans, study materials, and exam details.

Important Features:

- Student Info System (SIS): Keep track of student details, attendance, and grades.
- Communication Portal: Make it easy for everyone to talk and share info.
- Academic Resource Center: Store lesson plans, study materials, and exam details in one place.
- Attendance Tracking: Help teachers easily record and monitor student attendance.
- Grading System: Automate the grading process, making it faster and more accurate.