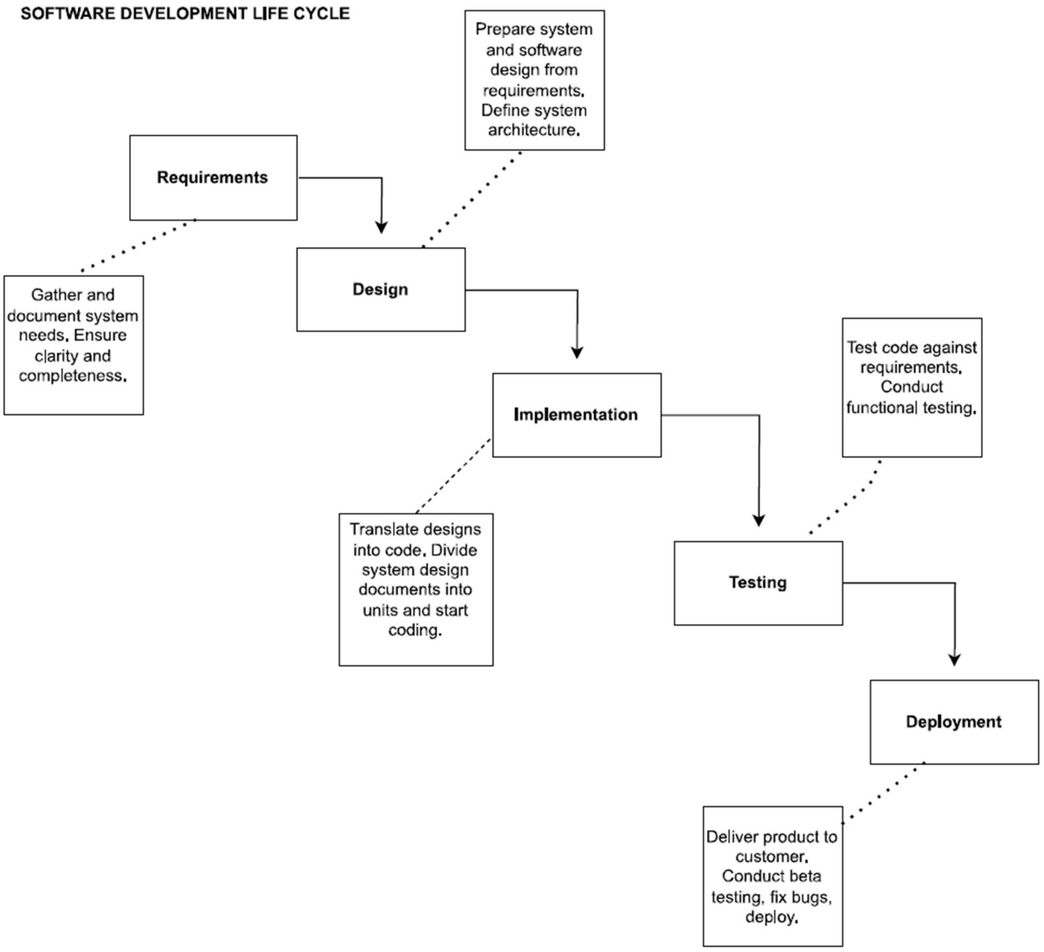
Assignment 1: SDLC Overview - Create a one-page infographic that outlines the SDLC phases (Requirements, Design, Implementation, Testing, Deployment), highlighting the importance of each phase and how they interconnect.


SOFTWARE DEVELOPMENT LIFE CYCLE
Prepare system and software design from requirements. Define system architecture.
Requirements
Gather and document system needs. Ensure clarity and completeness.
Design
Implementation
Test code against requirements. Conduct functional testing.
Translate designs into code. Divide system design documents into units and start coding.
Testing
Deployment
Deliver product to customer. Conduct beta testing, fix bugs, deploy.

Requirements: This is the first phase in the SDLC. It involves gathering and documenting the needs and requirements of the system from the client or end-users. The success of the project largely depends on the clarity and completeness of the requirements.

Design: In this phase, the system and software design is prepared from the requirement specifications which were studied in the first phase. System Design helps in specifying hardware and system requirements and also helps in defining the overall system architecture.

Implementation: The real code is written here. It is the phase where the designs are translated into code. The system design documents are divided into units and developers start coding according to the agreed design.

Testing: After the code is developed, it is tested against the requirements to make sure that the product is actually solving the needs addressed and gathered during the requirements phase. During this phase, all types of functional testing like unit testing, integration testing, system testing, acceptance testing are done.

Deployment: After successful testing, the product is delivered/deployed to the customer for their use. As soon as the product is given to the customers, they will first do the beta testing. If any changes are required or if any bugs are caught, then they will report it to the engineering team. Once those changes are made or the bugs are fixed then the final deployment will happen.

Assignment 2: Develop a case study analysing 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.

IoT project in the field of Smart Agriculture:

1. Requirement Gathering: The main objective of the project was to increase crop productivity by automating watering and monitoring soil conditions. This phase involved determining the different device categories, data types, and communication standards used.

2.Design: The design phase involved creating a robust online supply chain solution that would encompass the entire logistics process, starting with auto parts and steel suppliers and ending with subcontractors and car dealerships around the world.

3. Implementation: The implementation phase involved developing software systems that efficiently use the data produced by IoT devices while guaranteeing security, dependability, and scalability.

4. Testing: The testing phase ensured that the software systems could handle real-time data processing and reaction without causing delays. It also tested the system's ability to handle the enormous volumes of data produced by IoT devices.

5. Deployment: The deployment phase involved integrating the new system into the existing infrastructure. This included replacing an outdated supply chain management system that involved a large number of spreadsheets that required a lot of manual input.

6. Maintenance: The maintenance phase ensured the continuous functioning of the system. It involved timely improvements to the supply chain, which is crucial for the proper functioning of this huge system.

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 contexts.

Answer:
Comparison of SDLC Models for Engineering Projects

1.Waterfall Model:

Description: A sequential model where each phase (requirements gathering, design, development, testing, deployment, maintenance) is completed in a linear fashion before moving to the next.

Advantages:

- Clear documentation and defined phases
- Easier project management and resource allocation
- Suitable for well-defined projects with clear requirements

Disadvantages:

- Inflexible, difficult to adapt to changing requirements
- High risk of rework due to late detection of errors
- Limited user involvement until late stages

Applicability: Engineering projects with well-defined requirements upfront, such as building a bridge with established specifications.

2.Agile Model:

Description: An iterative and incremental approach where requirements are broken down into user stories, developed in short sprints, and reviewed continuously.

Advantages:

- Adaptable to changing requirements
- Faster feedback loop and early error detection

- Increased user involvement throughout the development process

Disadvantages:

- Requires strong team communication and collaboration
- May not be suitable for projects with strict deadlines or complex dependencies
- More documentation overhead compared to Waterfall

Applicability: Engineering projects with evolving requirements or those requiring user feedback at each stage, such as developing a new robotics controller.

3. Spiral Model

Description: A risk-driven model that combines elements of Waterfall and Agile. Each iteration involves risk identification, mitigation, prototyping, and evaluation before proceeding to the next iteration.

Advantages:

- Explicitly addresses project risks early on
- Iterative approach allows for adjustments based on findings
- Suitable for complex projects with high uncertainty

Disadvantages:

- Can be more complex to manage than Waterfall or Agile Requires strong risk assessment and mitigation skills
- May lead to longer development cycles compared to Agile

Applicability: Engineering projects with high levels of uncertainty or significant risks, such as developing a new medical device.

4. V-Model

Description: An extension of the Waterfall model that emphasizes verification and validation activities alongside each development phase. Verification ensures the system is built according to specifications, while validation ensures it meets user needs.

Advantages:

- Strong emphasis on quality assurance and testing
- Early detection of defects
- Suitable for safety-critical engineering projects

Disadvantages:

- Shares the inflexibility of Waterfall
- May require additional resources for extensive testing
- Less adaptable to changing requirements

Applicability: Engineering projects where safety and rigorous testing are paramount, such as developing an aircraft control system.