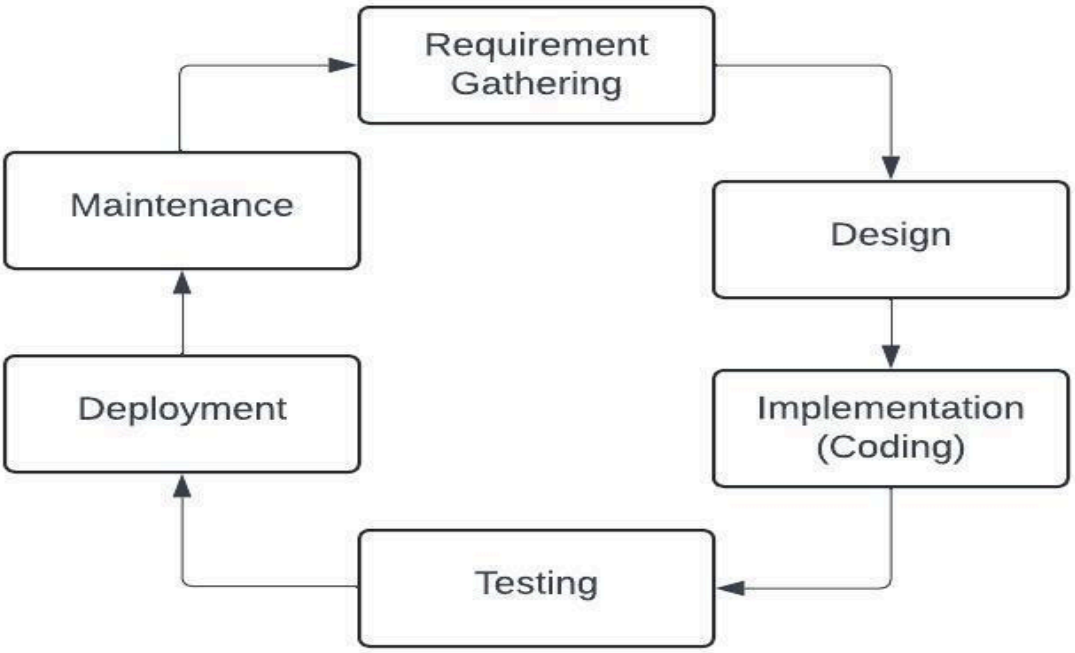
Day 02

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.

Solution:- SDLC stands for software development life cycle.



Key phase of SDLC

1. **Requirement Gathering :-** It is a step for understanding user requirements and functionalities. It create a roadmap, what a software needs to do.
2. **Design :-** Creating a blueprint for a software's architecture. It provide a solid foundation for future maintenance.
3. **Implementation :-** Providing the life for the design by writing the code. Programmers write the code in required programming language.
4. **Testing :-** Used for identifying and fixing bugs to ensure quality, which helps to deliver a reliable software product.
5. **Deployment :-** Process of releasing the software to the user. Making the software available for it's intended purpose.
6. **Maintenance :-** Providing the proper maintenance for the user for a better user friendly experience and resolving the sudden errors. Moreover providing the new updates for future.

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.

Solution :- This case study analyzes the implementation of SDLC phases in the development of a Smart Irrigation System for farms. We will evaluate how each phase contributes to the project's success.

1. **Requirements Gathering:-**
    - **Activities:** Interviews with farmers, agricultural experts, and water management specialists. Analysis of water usage data and soil conditions.
    - **Outcomes:** Identified key requirements like automated scheduling, soil moisture monitoring, remote access, and data analytics for water usage optimization.

2. **Design:-**
    - **Activities:** Designing the system architecture with sensors, controllers, and a user interface (UI) mobile app. Selecting communication protocols and data security measures.
    - **Outcomes:** A well-defined system blueprint that guides development and ensures components work together seamlessly.

3. **Implementation (Coding):-**
    - **Activities:** Developers write code for the sensors, controllers, and mobile app, integrating them with chosen communication protocols and data security features.
    - **Outcomes:** A functional prototype of the SIS that can be tested and refined.

4. **Testing:-**
    - **Activities:** Unit testing of individual components like sensors and the mobile app. Integration testing to ensure all components work together. Field testing with real-world farm environments and user acceptance testing with farmers.
    - **Outcomes:** Identification and fixing of bugs, ensuring system functionality and usability. User feedback allows for improvements before deployment.

5. **Deployment:-**
    - **Activities:** Setting up the SIS at farms, user training on the mobile app, and data analysis dashboards.
    - **Outcomes:** A successfully deployed SIS ready for use by farmers.

6. **Maintenance:-**
    - **Activities:** Monitoring system performance, addressing any technical issues, and gathering user feedback for future improvements. Potential development of additional features based on user needs.
    - **Outcomes:** Ensuring long-term system reliability and user satisfaction. Continuous improvement based on user feedback and data analysis.

By effectively implementing all SDLC phases, the Smart Irrigation System project delivered a valuable tool for farmers. Each phase played a crucial role in ensuring the system addressed real needs, functioned reliably, and ultimately benefited users.

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.

Solution :-

1. **Waterfall Mode :-**

Advantages :-

- Clear plan and structure.
- Defined stages.
- Easy to understand.
- Predictable timeline.

Disadvantages :-

- Inflexible.
- Limited consumer feedback.
- Late defect identification.
- Not ideal for complex projects.

Example :- Bridge construction.

2. **Agile Methodologies :-**

Advantages :-

- Faster feedback and delivery.
- Improved customer satisfaction.
- Reduced risks.
- Increased flexibility.
- Stronger team collaboration.

Disadvantages :-

- Unclear project scope.
- Heavy reliance on individuals.
- Difficulty in managing large projects.
- Potential for scope creep.
- Documentation challenges.

Example :- Renovation of historic building.

3. **Spiral Modal :-**

Advantages :-

- Risk driven approach.
- Iterative development.

- Early user involvement.
- Flexibility.
- Reduced Risk.

Disadvantages :-

- Complexity.
- Cost.
- Project Management challenge.
- Not ideal for small projects.
- Unclear timelines.

Example :- Developing a prototype of next generation vehicle.

4. **V-modal :-**

Advantages :-

- Early defect detection.
- Structured approach.
- Thorough testing.
- Clear documentation.

Disadvantages :-

- Limited Flexibility.
- Upfront requirements clarity.
- Potential delays.
- Less agile.

Example :- Developing a new aircraft autopilot system.