

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. Requirements Phase:

In this phase, we define project scope, objectives, and user needs. This is the foundation for all phases. Clear requirements guide design and implementation.

2. Design Phase:

In this phase, we create detailed system architecture, frontend designs. Interconnection: Translates requirements into technical specifications. Guides implementation and testing.

3. Implementation Phase

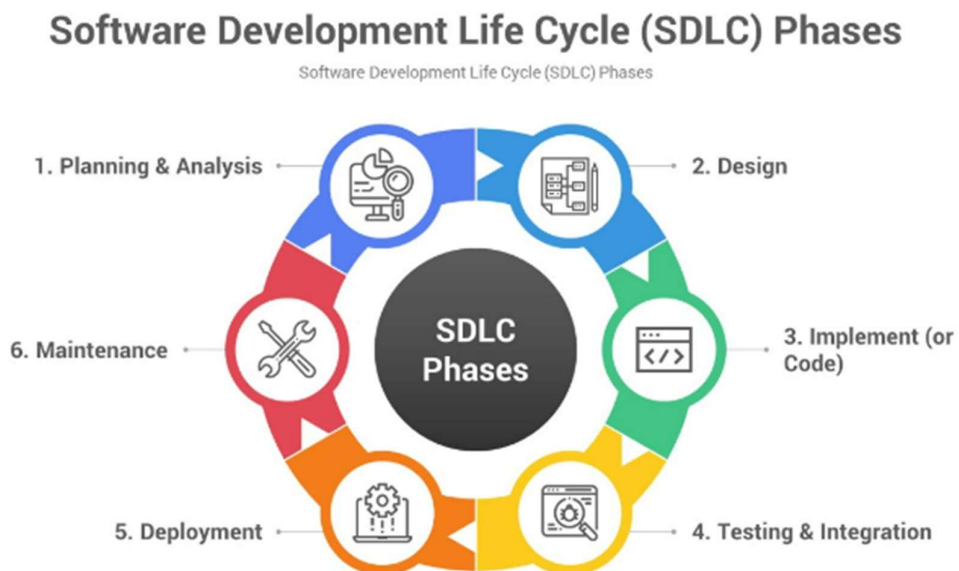
In this phase we write code and build the software based on design specifications. It is the implementation of design phase where concept is developed into product.

4. Testing Phase:

In this phase, we verify and validate software functionality. we check whether requirements and design are achieved.

5. Deployment Phase:

In this phase, we release the software to users or clients.



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.

Case Study: Implementation of SDLC Phases in an E-commerce Website

1. Requirement Phase:

During the requirement gathering phase, the project team conducts market research, analyse competitor websites to understand business goals, user needs, and technical requirements.

Key requirements include user-friendly interface, secure payment gateway, inventory management system, responsive design for mobile and desktop compatibility and efficient order fulfilment processes.

2. Design Phase:

In the design phase, the project team translates the gathered requirements into detailed design specifications for the e-commerce website. They create prototypes to visualize the user interface and user experience.

Navigation flow and product categorization will be designed for usability. Additionally, database schemas, server architecture, and scalability plans will be developed.

3. Implementation Phase:

- Front-end developers focused on coding to build the website's user interface based on the approved designs.
- Back-end developers works on developing server-side logic, database integration, and implementing features such as user authentication, product catalog management, and shopping cart functionality.
- Integration with third-party services, such as payment gateways and shipping providers, was also undertaken during this phase.

4. Testing Phase:

- The testing phase involves testing of the e-commerce website to ensure functionality, performance, security, and compatibility across different devices and browsers.
- Various types of testing were conducted, including unit testing, integration testing, regression testing, performance testing, and security testing.
- Test cases will be developed to validate features such as user registration, product search, checkout process, and payment processing.
- Bugs and issues were identified, prioritized, and resolved iteratively to ensure a high-quality user experience.

5. Deployment Phase:

Upon successful completion of testing, the e-commerce website was ready for deployment to production servers.

6. Maintenance Phase:

Even after deployment, the project entered into the maintenance phase to ensure ongoing support, optimization, and evolution of the e-commerce website.

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.

Waterfall Model:

The waterfall model involves a sequential approach to software development, where each phase of the project is completed before moving on to the next one.

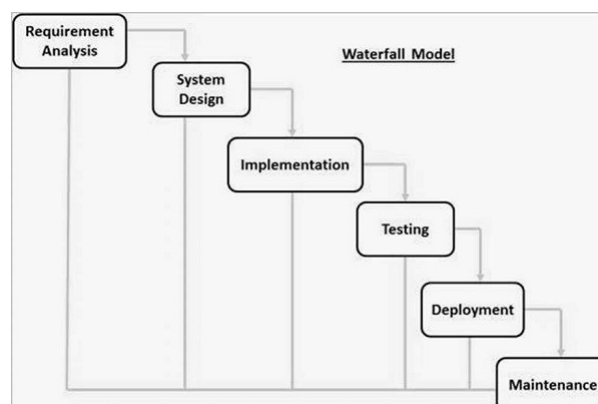
The waterfall model relies heavily on documentation to ensure that the project is well-defined and the project team is working towards a clear set of goals.

Advantages:

- Sequential and structured approach, making it easy to understand and manage.
- Clear project milestones and deliverables, facilitates project planning and budgeting.
- Well-suited for projects with stable requirements and predictable outcomes.

Disadvantages:

- Limited flexibility to accommodate changes once the development process has started.
- High risk of project failure if requirements are not accurately defined upfront.
- Long development cycles may lead to delayed feedback and slower time-to-market.
- Limited customer involvement until the end of the development process, potentially resulting in misalignment with user needs.



Agile:

Agile is well-suited for engineering projects with evolving requirements, high uncertainty, and a need for rapid adaptation.

Advantages:

- Iterative and incremental development, allowing for frequent feedback and adaptation.
- Flexibility to accommodate changes in requirements and priorities throughout the project lifecycle.
- Enhanced collaboration between cross-functional teams, fostering creativity and innovation.
- Emphasis on delivering working software early and often, enabling faster time-to-market and customer value realization.

Disadvantages:

- Requires active involvement and commitment from stakeholders throughout the project.
- May lack comprehensive documentation, making it challenging to maintain consistency and traceability.
- Complexity increases with the scale and scope of the project, requiring effective project management and coordination.
- Continuous changes and iterations may lead to scope creep and project scope instability if not managed properly.



Spiral Model:

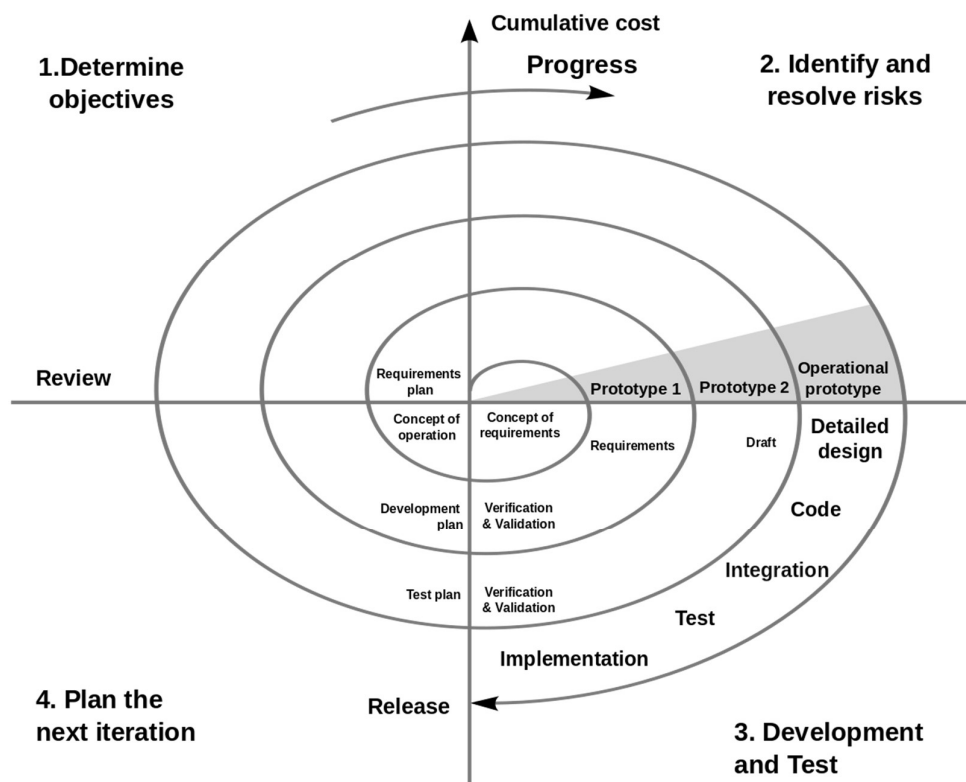
The Spiral model is used for engineering projects with high technical complexity, changing requirements, and a need for rigorous risk management.

Advantages:

- Iterative and risk-driven approach, allowing for early identification and mitigation of project risks.
- Flexibility to incorporate changes and enhancements at each iteration based on stakeholder feedback.
- Emphasis on prototyping and validation of key features, reducing the likelihood of costly rework.
- Well-suited for projects with complex or uncertain requirements, where a phased approach is necessary.

Disadvantages:

- Requires significant effort and resources to manage project risks and iterations effectively.
- Complexity increases with the number of iterations and project size, requiring careful planning and execution.
- May lead to project delays and increased costs if risk management is not prioritized.
- Documentation and traceability may be challenging to maintain, particularly in dynamic environments.



V-Model:

The V-Model is suitable for engineering projects with a focus on quality assurance, validation, and compliance.

Advantages:

- Emphasizes verification and validation activities throughout the project lifecycle, ensuring high quality and reliability.
- Clear correlation between requirements and test cases, facilitating comprehensive test coverage.
- Well-defined milestones and deliverables, enabling effective project planning and monitoring.
- Suitable for projects with strict regulatory requirements or quality standards.

Disadvantages:

- Sequential and rigid structure may limit flexibility and adaptability to changes.
- Requires detailed upfront planning and documentation, which may be time-consuming and costly.
- Complexity increases with the number of verification and validation activities, requiring careful coordination and resource allocation.
- Limited customer involvement until the testing phase, potentially leading to misalignment with user needs.

