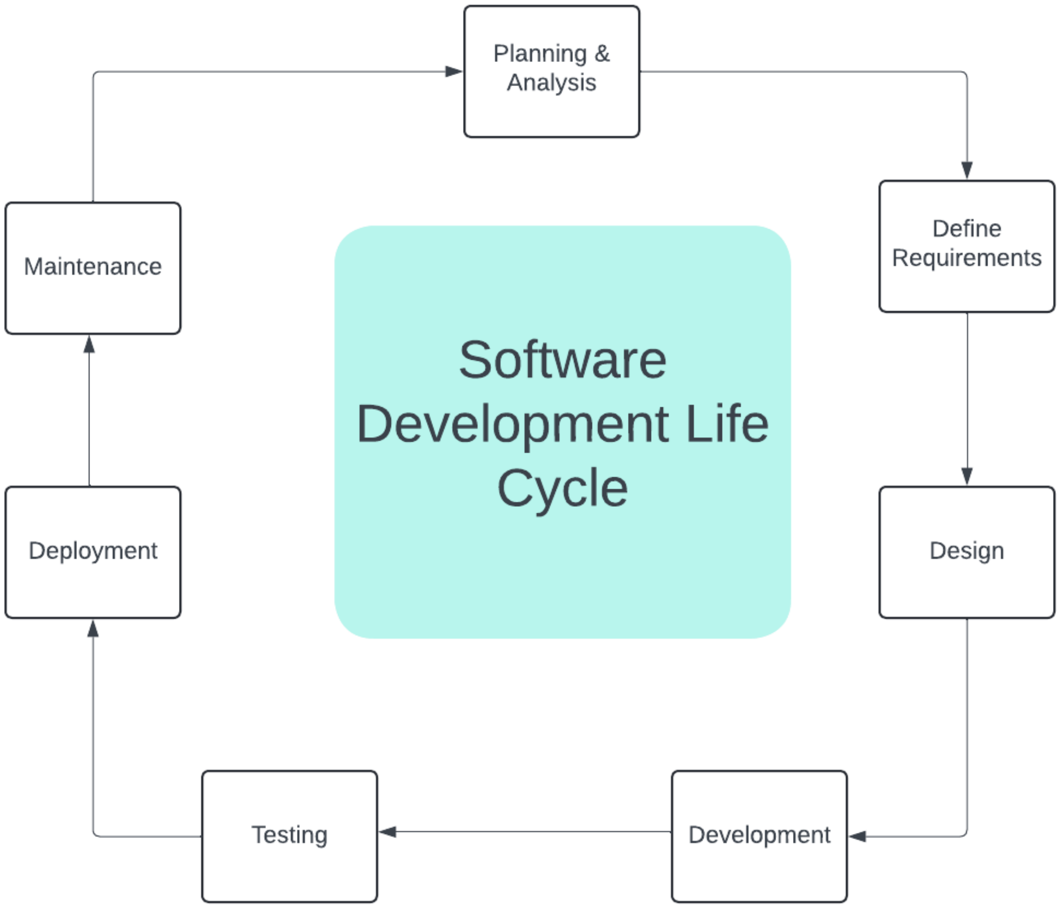
Asssignment1:

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.

Ans:



Software Development Life Cycle (SDLC) is a process where a team of developers, testers, and others make an awesome software with scalability, top-notch quality, less cost and minimize risk having a lot of stages that is given in infographic diagram above. A typical SDLC contains following phase for the development of a software:

1. Planning and analysis
2. Define requirements
3. Design
4. Development
5. Testing
6. Deployment
7. Maintenance

Planning and analysis: This is the first phase of SDLC. In this phase you gather information from clients and stakeholders that what is the requirement of a problem to be solved or product to be launched. Once the requirement is get, we move to the next phase i.e. define requirements.

Define requirements: The requirements that are defined in planning and analysis phase are converted to clear cut requirements to the developers here. This includes development of important product documents, software requirement specifications, product requirements etc.

Design: Now as the name implies, this phase is associated with the design of the product and associated tools and languages, platform required and application security. This is where a flowchart is made to tackle problems and also development of prototype is done in this phase.

Development: This is the actual development phase where development team divide the project into software modules and turn the software requirements into code that makes the project. In this phase it is important to set the timelines and milestones so that the software developers understand the expectations and deliver product efficiently on time.

Testing: Before deploying the actual product it is important to do a quality assurance by QA team members to perform validation test so as to make sure that each function is working correctly properly and to rectify if any bug arise. The types of testing in this phase are:

- Performance testing: It assesses software speed and accuracy.
- Functional testing: It verifies that the software meets the requirements.
- Unit testing: It tests individual function or components of a software.
- Usability testing: It evaluates software's user interface and user experience.
- Acceptance testing: It is also termed end user testing is a test where it is ensured that software meets what is promised

Deployment: During the deployment phase the final software needs to be delivered to the end user but this process needs to be automated. So that we can schedule when a particular feature update or software update needs to be deployed.

Maintenance: The Maintenance phase is the last stage of SDLC. In this stage user may find bugs and errors that were missed in earlier testing phase. These bugs need to be fixed for better user experience and retention.

Assignment2:

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.

Ans:

Case Study: Implementing SDLC Phases in a real world engineering project

Introduction:

ABC Tech is a software development company is given a task to create full stack web development project called EduSchool, an online platform to manage courses students and all the educational related resources. This case study explains how ABC Tech applies SDLC for successful delivery of web application.

1. Planning and analysis phase:
   In this phase ABC Tech defined the project scope, requirement of project to be done, its objectives and who are stakeholders. They need to search for institutions that need online platforms for providing e learning resources and user-friendly interfaces. And also analyzed technician, financial and operational feasibility for that platform. Institutions and students need to align to gather the required information.
2. Requirement Gathering:
   ABC Tech conducted various activities for requirement gathering. They conducted interviews and surveys with stakeholders to get the information. They identified course required, user-authentication, course-management, content delivery and performance tracking.
3. Design:
   Based on requirements ABC Tech designed the architecture and UI of EduSchool. They utilized java framework Spring Boot for backend and angular for frontend. Backend design included RESTful API for data manipulation and frontend focused on intuitive design for easy navigation and more interaction.
4. Development:
   It consists of selecting IDE, JDK versions, and database servers which are part of the development environment. Apart from that front-end and back-end development needs to be done. Writing HTML, CSS and angular codes for frontend, spring MVC, Spring WebFlux to handle HTTP request and spring JPA to handle data in the backend and MySQL server to store and fetch the data for the database purpose.
5. Testing:
   In this phase ABC Tech Testing team did unit testing and integration testing for both frontend and backend components. Also, they perform manual testing to ensure that all the tests are working fine.
6. Deployment:
   Now ABC Tech Team deployed this web application on Heroku server. Also, they set up the configuration environment in case something needs to be changed. And finally, they configure CI/CD (continuous integration and continuous deployment) pipelines for automated deployment.
7. Maintenance:
   After deployment, ABC Tech entered maintenance phase to enhance EduSchool. They monitored the application performance, addressed any reported issues by the end-user, implemented feature updates based on user feedback. Regular security updates were applied to protect against any vulnerabilities and finally documentation and training materials were provided to assist users in utilizing this platform effectively.

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.

Ans:

Software Development Life Cycle (SDLC) models provide approaches for software development in a very well-structured way. Each has its own advantages, disadvantages and sustainability to engineering context. Four important SDLC models are:

1. Waterfall model
2. Agile model
3. Spiral model
4. V-model

Waterfall model:

Advantages:

- It is super easy to understand and thus is good to apply for simple projects with well-defined requirements.
- Its documentation is so clear that it enhances communication between product management and the stakeholders.

Disadvantages:

- Change is difficult because without completing one phase you can't go to the next phase.
- From development to testing a delay may be there result in delay in product delivery and a risk is there as because of delay a very few stakeholders are involved till the last stage resulting in misalignment of application with end-user.

Applicability:

Best suited for projects with stable requirements which means that only simple projects, well defined scopes, and predictable outcomes.

Agile model:

Advantages:

- Here the best advantage is it is open to change which means adaptability and flexibility is there. It allows iterative development and incremental delivery of features.
- It allows better communication between project management and stakeholders resulting in alignment with user needs.

Disadvantages:

- It requires maximum engagement from the stakeholders' side which sometimes is not feasible, and complexities increase in bigger and larger projects which leads to coordination challenges.
- There is a lack of documentation so there is a possibility of miscommunication.

Applicability:

This model is good for projects having changing requirements, unstable scope and need rapid delivery of applications.

Spiral model:

Advantages:

- It shows very high-risk management because there is early identification and resolving of risks.
- It actively involves stakeholder and product management and allows rapid changes in development.

Disadvantages:

- This model is very complex, and a lot of resources is engaging leading to multiple iterations and increased risk.
- It requires experienced managers and technical experts and hence increases costs compared to other models.

Applicability:

This model is good for projects with heavy technical complexity and high risk and changing requirements.

V-model:

Advantages:

- The verification and validation are done throughout the development resulting in the delivery of high-quality applications.
- Provides clear traceability between different phases of development and there is early detection and resolution of any defects resulting in reduced rework and high overall project efficiency.

Disadvantages:

- There may be some delay in feedback because of sequential nature.
- Once the development has started there is limited flexibility for changes, and it relies heavily on documentation.

Applicability:

This model is well suited for projects with strict requirements where validation and verification are critical and rely heavily on documentation.