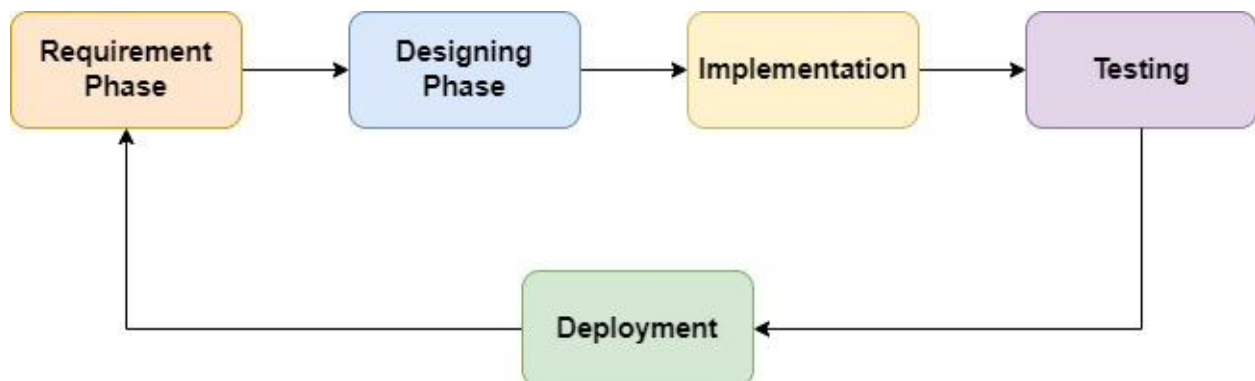


Day 2 Assignment

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.

- **Requirement Phase** : At requirement phase, all the stakeholder that includes client, managers and senior members of the team decide to share inputs to create a vision of the project. It is important to conduct this phase thoroughly such that no confusion or ambiguity is created. Most of the discussion revolves around taking input from clients and deciding the best course of action, resource, time and money required for the project.
- **Designing Phase** : This phase is mostly about giving instruction to the team who are working for the project such as Object Relation, Entity Relations, Pseudocodes. Algorithms etc. Its high level overview of the project phase which will break down to the lowest level with each separate team.
- **Implementation** : This phase is all about writing the code and preparing documentation based on the design discussed earlier. All the code must be written in such a way that it does not confuse while debugging phase and can be read in during the lifetime of the project
- **Testing** : At this phase source code is tested and any defects in the occurred will be reverted back to the developer team. It is important to cover every edge case so the product does not fail after deployment.
- **Deployment** : At this stage the final project is given to the client and it's deployed in the real world. At this stage the product is being used by the client and all the instructions should be met.



Created by draw.io

2. **Develop a case study analyzing the implementation of SDLC phases in real-world engineering projects. Evaluate how requirement gathering, designing,**

implementation, testing, deployment and maintenance contribute to project outcomes.

Case Study : Development of an E-Library Platform

In this case study, a real-world project like E-Library is being studied. The study is about the implementation of SDLC in the project.

- **Requirement Gathering** : The project will begin with a meeting with all the stakeholder which will include the researcher, market research team, investors etc. Then have their views documented. Identifying potential key features required for the project, recommendation, cost analysis, market input, books digital rights, resource and some future competency.
- **Designing** : Based on the feature requirement. Designing a high level user flow diagram to understand the interaction between user and the library. To understand more object-entity relationship diagrams will be needed. Then a thorough discussion about the high level design and low level design for each individual team. High level discussion will be about the type of microservices, cloud system, etc. Low level design will include database design for user, library storage. A demo of a wireframe of features to see how they will interact. Each team will be given a task according to the plan.
- **Implementation** : There will be a feedback loop after each version of feature implementation to make sure it meets the goal of the stakeholders. Team will use web technologies like HTML5, CSS3, Javascript for frontend and backend team will utilize server-side technologies like Node.js. They also integrated with third-party APIs for content aggregation, user authentication, and payment processing. Continuous integration and deployment (CI/CD) pipelines were set up to automate testing and deployment processes, ensuring code quality and reliability. Also with each revision a proper briefed documentation will be prepared for better debugging and fixes.
- **Testing** : Testing team will make sure all the features work well and every possible edge cases are covered. Features like user registration, content browsing, search functionality, reading experience and payment system. Tests will be like performance, security breach, server load etc. If any issue is identified then a proper ticket will be raised and the feature will go back to the implementation stage.
- **Deployment** : After the completion of the final deployable version, the whole system will be hosted on cloud services like Azure to ensure scalability, reliability, and cost effectiveness. A phased rollout will be done to make sure the system does not overload too quickly and seamless onboarding will occur.
- **Maintenance** : A dedicated support team was responsible for addressing user inquiries, troubleshooting technical issues, and monitoring platform performance. Regular software updates and security patches were released to address emerging threats and improve platform stability.

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 the different engineering contexts.

- **Waterfall Model** : It is one of the classic and oldest methods. It includes a linear sequential way of doing a job. The processes involved are Requirement Gathering, Designing, Implementation, Testing, Deployment and Maintenance. All these processes occur one after another.
 - Advantage :
 - Simple method.
 - Easy to follow
 - Each phase is sequential and does not require rigorous review.
 - Disadvantage :
 - Not flexible.
 - Features once implemented cannot be changed in later stages.
 - Mismatched during deployment will stop the entire life cycle.
 - Applicability :
 - Projects where vision is clear.
 - Simple projects
 - Less maintainability or easy to maintain projects
 - Sequence of operation is required
- **Spiral Model** : It is derived from Waterfall and Iterative models. Includes the same strategy like the waterfall model but each stage requires prototyping. If any stage fails then it rolls back to its previous stage until the prototype comes to a satisfactory level.
 - Advantage :
 - Feature can be seen early
 - Feature can be added if needed
 - Stakeholder is involved at each stage
 - Disadvantage :
 - Too much prototyping
 - Complexity increases with each edition
 - Cost increases with each iteration
 - Applicability:
 - Large Project
 - Feature Loaded
 - Requires more stakeholder input
- **V-Model** : It is also derived from waterfall mode but each phase has its own testing and verification stage. After each stage a rigorous testing is done to ensure the quality of the project. It does not have a roll back like the spiral model.
 - Advantage :
 - Early Detection of defects
 - Structured and sequential like a waterfall model.
 - High quality product
 - Disadvantage :

- Testing at each stage is resource intensive
 - Small project does not fit well
 - High Cost
- Applicability :
 - High Risk Project
 - Safety is important
 - Testing required at each stage
- **Agile Model** : Agile model derived from incremental and iterative models. It has its short development phase and stages can be designed according to the need of the project.
 - Advantage :
 - Very Flexible
 - Fast Delivery
 - Feature can be added after cycle
 - Disadvantage :
 - Less vision in the beginning
 - Fast Pace makes it difficult to deliver
 - Management Issues
 - Applicability :
 - Customer Oriented
 - Cycles Changes with each features