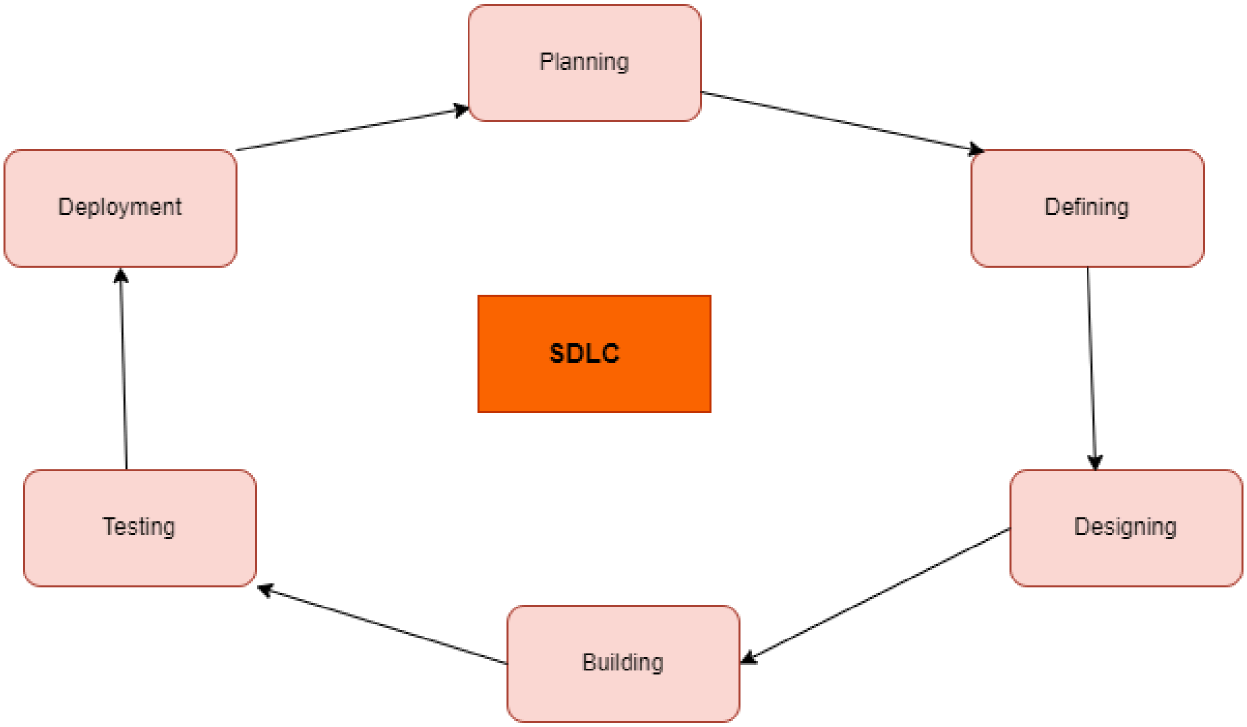
Assignment no.2

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.

ANSWER:-



Importance of each Phase:-

1] Planning:-

- Requirement analysis is the most important and fundamental stage in SDLC.
- It is performed by the senior members of the team with inputs from the customer, the sales department, market surveys and domain experts in the industry.
- This information is then used to plan the basic project approach and to conduct product feasibility study in the economical, operational, and technical areas.

2] Defining:-

- Once the requirement analysis is done, the next stage is to certainly represent and document the software requirements and get them accepted from the project stakeholders.

- This is accomplished through "SRS"- Software Requirement Specification document which contains all the product requirements to be constructed and developed during the project life cycle.

3] Designing:-

- The next phase is about to bring down all the knowledge of requirements, analysis, and design of the software project.
- This phase is the product of the last two, like inputs from the customer and requirement gathering.

4] Building:-

- In this phase,development team build a code for the given product.
- Developers have to follow the coding guidelines defined by their organization and programming tools like compilers, interpreters, debuggers etc are used to generate the code.
- If the design is performed in a detailed and organized manner, code generation can be accomplished without much hassle.
- The programming language is chosen with respect to the type of software being developed.
- Developers analyze the requirements to identify smaller coding tasks they can do daily to achieve the final result.

5] Testing :-

- After the developers build a software,it is deployed to the testing phase.
- This stage refers to the testing only stage of the product where products defects are reported, tracked, fixed and retested, until the product reaches the quality standards defined in the SRS.

6] Deployment :-

- Once the testing is done, and the product is ready for deployment, it is released for customers to use.
- Sometimes product deployment happens in stages as per the organizations' business strategy.
- The product may first be released in a limited segment and tested in the real business environment (UAT- User acceptance testing).
- Then based on the feedback, the product may be released as it is or with suggested enhancements in the targeting market segment.
- After the product is released in the market, its maintenance is done for the existing customer base.

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.

Answer:-

Developing a Food Delivery App

1. Requirements Gathering Phase:

- During this phase, the project team will gather requirements from stakeholders, including users, restaurant owners, and delivery partners.
- They will identify the functional and non-functional requirements of the Food Delivery App, such as user registration, restaurant listings, menu management, order placement, payment processing, and delivery tracking.Deliverable: Requirements Specification Document

2. System Design Phase:

- Based on the gathered requirements, the project team will design the architecture and user interface of the Food Delivery App.
- They will define the system components, database schema, and navigation flow.
- The design phase will also include prototyping key features to visualize the user experience, including the order flow and checkout process.Deliverable: System Design Document, Prototypes

3. Implementation Phase:

- In this phase, the project team will develop the Food Delivery App according to the specifications outlined in the design document.
- Backend development will focus on implementing functionalities such as user authentication, restaurant data management, order processing, and integration with payment gateways.
- Frontend development will involve creating user-friendly interfaces for browsing restaurants, selecting menu items, and tracking deliveries.Deliverable: Implemented Food Delivery App

4. Testing Phase:

- Once the implementation is complete, the Food Delivery App will undergo comprehensive testing to ensure that it meets the specified requirements.

- Testing will include functional testing, usability testing, performance testing, and security testing.
- Bugs and issues identified during testing will be addressed by the development team.Deliverable: Test Plan, Test Cases, Bug Reports

5. Deployment Phase:

- After successful testing, the Food Delivery App will be deployed to the production environment.
- Deployment activities will involve configuring servers, setting up databases, and releasing the app to app stores (e.g., Apple App Store, Google Play Store).
- User training sessions may be conducted to familiarize stakeholders with the app.Deliverable: Deployed Food Delivery App

6. Maintenance Phase:

- Once the Food Delivery App is deployed, the project team will enter the maintenance phase.
- They will provide ongoing support, address user feedback, and make necessary updates and enhancements to the app.
- Maintenance activities may include bug fixes, performance optimization, and addition of new features based on user requests and market trends.

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:-

1] Waterfall Model

Advantages :-

- Simple and easy to understand and use.
- Easy to manage due to the rigidity of the model – each phase has specific deliverables and a review process.
- Easy to arrange tasks.
- Process and results are well documented.
- Works well for smaller projects where requirements are very well understood.

Disadvantages :-

- High amounts of risk and uncertainty.
- Not a good model for complex and object-oriented projects.
- Poor model for long and ongoing projects.
- It is difficult to measure progress within stages.
- Not suitable for the projects where requirements are at a moderate to high risk of changing. So risk and uncertainty is high with this process model.

Applicability :-

- The Waterfall Model is suitable for projects with well-understood requirements and stable technologies, such as construction projects, where changes are unlikely to occur once the project begins.

2] Agile Model

Advantages :-

- Is a very realistic approach to software development.
- Functionality can be developed rapidly and demonstrated.
- Resource requirements are minimum.
- Suitable for fixed or changing requirements.
- Easy to manage,gives flexibility to developers.

Disadvantages :-

- Not suitable for handling complex dependencies.
- More risk of sustainability, maintainability and extensibility.
- Depends heavily on customer interaction, so if customer is not clear, team can be driven in the wrong direction.
- Transfer of technology to new team members may be quite challenging due to lack of documentation.
- An overall plan, an agile leader and agile PM practice is a must without which it will not work.

Applicability :-

- Agile is suitable for projects where requirements are expected to change or evolve, such as software development, where rapid adaptation to market needs is essential.

3] Spiral Model :-

Advantages :-

- Changing requirements can be accommodated.
- Allows for extensive use of prototypes.
- Requirements can be captured more accurately.
- Development can be divided into smaller parts and more risky parts can be developed earlier which helps better risk management.
- Suitable for large-scale projects with high uncertainty or complexity.

Disadvantages :-

- Management is more complex.
- End of project may not be known early.
- Not suitable for small or low risk projects and could be expensive for small projects.
- Process is complex
- Spiral may go indefinitely.

Applicability :-

- The Spiral model is well-suited for large-scale projects with evolving requirements and high levels of uncertainty, such as the development of complex software systems or aerospace projects.

4] V Model :-

Advantages :-

- This is a highly disciplined model and Phases are completed one at a time.
- Works well for smaller projects where requirements are very well understood.
- Simple and easy to understand and use.
- Easy to manage due to the rigidity of the model – each phase has specific deliverables and a review process.
- Each stage has corresponding testing activities, leading to a more structured approach to quality assurance.

Disadvantages :-

- High risk and uncertainty.
- Not a good model for complex and object-oriented projects.
- Poor model for long and ongoing projects.

- Not suitable for the projects where requirements are at a moderate to high risk of changing.
- Once an application is in the testing stage, it is difficult to go back and change a functionality.

Applicability :-

- The V-Model is best suited for projects with well-defined requirements and where a structured approach to testing and quality assurance is critical, such as in medical device development or safety-critical systems.

To summarize, the suitability and drawbacks of any Software Development Life Cycle (SDLC) model vary based on the specific needs and characteristics of the engineering project. While Agile and Spiral are better suited for projects where requirements are likely to change and flexibility is crucial, Waterfall and V-Model are better suited for projects with stable requirements and a focus on documentation and quality assurance.