**SE - Overview of IT industry**

Q1) What is Software? What is Software engineering?

=> Software: Software is the collection of data or computer instructions that tell the computer how to work and it provide us with a specific functionality.

Types of Software:

1)System Software

2)Programming Software

3)Application Software

4)Driver Software

5)Middleware

=> Software Engineering: Software Engineering is the systematic application of engineering approaches to the development of the software. It involves the application of principles from design, development, operation, and maintenance of software.

Q2) Explain types of software

-> Types of Software

1) System Software: System software is designed to provide a platform for other software. It manages hardware and software resources and provides common services for application software.

Examples:

- Operating Systems: Windows, macOS, Linux

- Utilities: Antivirus software, Disk management tools

Functions:

- Manages hardware components

- Provides a user interface

- Executes and provides services for applications

2) Programming Software: Programming software provides tools to assist developers in writing programs. These tools include compilers, debuggers, interpreters, and various editors.

Examples:

- Compilers: GCC, Microsoft Visual C++

- Interpreters: Python, Ruby

- Integrated Development Environments (IDEs): Eclipse, Visual Studio

- Text Editors: Notepad++, Sublime Text

Functions:

- Converts source code written in high-level languages into machine code

- Helps in debugging and testing code

- Provides an environment to write and manage code

3) Application Software: Application software allows users to perform specific tasks or activities. It is designed to help users be productive in personal, educational, and business tasks.

Examples:

- Office Software: Microsoft Office

- Web Browsers: Google Chrome, Mozilla Firefox, Microsoft Edge

- Media Players: VLC, Windows Media Player

- Graphic Design Software: Adobe Photoshop

Functions:

- Facilitates user-specific tasks like word processing, web browsing, and media playback

- Enhances productivity and creativity

- Provides specialized functionalities for various user needs

4) Driver Software: Driver software allows higher-level computer programs to interact with a hardware device. It acts as a translator between the hardware and the operating system or application software.

Examples:

- Printer Drivers: HP Printer Drivers, Canon Printer Drivers

- Graphics Card Drivers: NVIDIA GeForce Drivers, AMD Radeon Drivers

- Network Drivers: Realtek Ethernet Drivers, Intel Wireless Drivers

Functions:

- Enables communication between the operating system and hardware devices

- Ensures hardware devices operate correctly

- Translates high-level commands into device-specific actions

5) Middleware

Middleware is software that lies between an operating system and the applications running on it. It provides common services and capabilities to applications outside of what's offered by the operating system.

Examples:

- Database Middleware: ODBC, JDBC

- Application Servers: WebLogic, WebSphere

- Amazon Services: AWS Lambda, Amazon API Gateway

Functions:

- Facilitates communication and data management for distributed applications

- Provides runtime services like transaction management and messaging

- Enables interoperability among different applications and services

Q3) What is SDLC? Explain each phase of SDLC

-> SDLC is a structure imposed on the development of a software product that defines the process for planning, implementation, testing, documentation, deployment, and ongoing maintenance and support. There are a number of different development models.

-> A Software Development Life Cycle is essentially a series of steps, or phases, that provide a model for the development and lifecycle management of an application or piece of software.

Phases of SDLC are:

1)Planning: Defines Scope and purpose of the project.

=> Activities:

- Feasibility Study: Determine if the project is viable technically, economically, and operationally.

- Project Planning: Outline the project timeline, resources, budget, and milestones.

- Risk Management: Identify potential risks and develop strategies to mitigate them.

=> Deliverables:

- Project Plan

- Feasibility Report

- Risk Management Plan

2)Analysis: Gather information and data regarding project and analyse requirements of what software should do.

=> Activities:

- Requirement Gathering: Collect requirements from stakeholders through interviews, surveys, and observations.

- Requirement Analysis: Analyse and prioritize requirements.

- Requirement Specification: Document the requirements in a detailed and clear manner.

=> Deliverables:

- Requirement Specification Document (SRS)

- Use Case Diagrams

- Data Flow Diagrams (DFD)

3)Designing: Create blueprint for the system architecture and its component.

=> Activities:

- High-Level Design (HLD): Define the system architecture, including modules, data flow, and interfaces.

- Low-Level Design (LLD): Design detailed logic for each module, including algorithms, database schema, and user interfaces.

- Prototyping: Develop prototypes for better visualization and validation of the design.

=> Deliverables:

- High-Level Design Document

- Low-Level Design Document

- Prototype

4)Implementation: Write the actual code of the software.

=> Activities:

- Coding: Developers write code based on the design specifications.

- Code Review: Peer review of code to ensure adherence to standards and best practices.

- Version Control: Use version control systems to manage code changes and collaboration.

=> Deliverables:

- Source Code

- Code Documentation

5)Testing and Integration: The testers and analysers would check the software for any bugs and make sure that all the required requirements are met correctly.

=> Activities:

- Unit Testing: Test individual components for correctness.

- Integration Testing: Test combined components to ensure they work together.

- System Testing: Test the entire system for compliance with requirements.

- User Acceptance Testing (UAT): Validate the software with the end-users to ensure it meets their needs.

=> Deliverables:

- Test Plans and Test Cases

- Test Reports

- Bug Reports

6)Maintenance: Ensure software remains up-to-date and functional after the deployment.

=> Activities:

- Corrective Maintenance: Fix bugs and issues reported by users.

- Adaptive Maintenance: Modify the software to accommodate changes in the environment or requirements.

- Perfective Maintenance: Improve the performance or enhance features of the software.

- Preventive Maintenance: Make changes to prevent potential future issues.

=> Deliverables:

- Maintenance Reports

- Updated Documentation

- New Releases and Patches

Q4) What is DFD? Create a DFD diagram on Flipkart.

=> Data Flow Diagram(DFD) is a graphical representation of the flow of data during the execution of the program which can be understood by both technical and non-technical users. The DFD model helps in the analysis of the software or application.

Q5) What is Flowchart? Create a Flowchart to make addition of 2 numbers?

-> A flow chart is a graphical or symbolic representation of a process. Each step in the process is represented by a different symbol and contains a short description of the process step.

Q6) What is Use Case Diagram? Create a use-case on bill payment on paytm.

-> A Use Case Diagram is a visual representation of the interactions between users (actors) and a system. It illustrates the various use cases (functionalities or services) that the system provides and how different actors interact with these use cases. Use Case Diagrams are part of Unified Modeling Language (UML) and are commonly used in the requirements analysis phase of software development.