**Software Engineering Assignment**

**MODULE: 1**

**SE – Overview of IT Industry**

**Q1. What is software? What is software engineering?**

***Answer***

Software refers to a collection of data or computer instructions that tell the computer how to work. It is a critical component of a computer system that comprises the following:

1. Applications Software
2. System Software
3. Middleware
4. Programming Software

Software engineering is a discipline that involves the application of engineering principles to the design, development, maintenance, testing, and evaluation of software and systems. It encompasses a range of activities and methodologies designed to improve the efficiency, quality, and reliability of software systems.

**Q2. Explain types of software?**

***Answer***

**Applications Software:** These are programs that perform specific tasks for users.

Examples include MS Office, PowerPoint.

**System Software:** This are the software which are been already include in our system.

Examples are Notepads, Cameras.

**Driver Software:** This are type of system software that acts as an intermediary between the operating system and hardware devices.

Example incudes VLC player to watch video.

**Middleware Software:** This software acts as a bridge between systems software and applications software. These lies between operating system and application running on it.

**Programming Software:** These tools assist developers in writing, testing, and maintaining software.

Examples include compilers, debuggers, VS code, etc.

**Q3. What is SDLC? Explain each phase of SDLC?**

***Answer***

The Software Development Life Cycle (SDLC) is a structured process used for developing software applications. It outlines a series of steps and phases that software projects follow to ensure systematic and efficient development, from initial concept through deployment and maintenance. The SDLC provides a framework for planning, creating, testing, and deploying software, ensuring high quality and efficiency throughout the development process.

**Phases of SDLC:**

1. Planning
2. Analysing
3. Designing
4. Implementation (Coding)
5. Testing
6. Deployment
7. Maintenance

**Planning:**

Objective: Define the project’s goals, scope, purpose, and feasibility.

Activities:

Identify the project scope and objectives.

Conduct feasibility studies (economic, technical, operational).

Allocate resources and establish a project timeline.

Develop a project plan and determine milestones.

Outcome: Project Plan, Feasibility Report.

**Analysing:**

Objective: Gather detailed functional and non-functional requirements from stakeholders.

Activities:

Conduct meetings and interviews with stakeholders.

Document user requirements and business needs.

Analyze requirements for completeness, consistency, and feasibility.

Outcome: Requirements Specification Document.

**Designing:**

Objective: Create the architecture and design of the software solution.

Activities:

Design the system architecture.

Develop detailed design specifications for each system component.

Create data models, user interfaces, and system interfaces.

Outcome: System Design Document, Database Schema, Prototypes.

**Implementation (Coding):**

Objective: Transform design specifications into functional code.

Activities:

Write code based on design documents.

Adhere to coding standards and best practices.

Perform unit testing to verify individual components.

Outcome: Source Code, Unit Test Reports.

**Testing:**

Objective: Ensure the software meets requirements and is free of defects.

Activities:

Conduct various levels of testing: unit testing, integration testing, system testing, acceptance testing.

Identify and fix defects.

Validate that the software meets all requirements and performs as expected.

Outcome: Test Plan, Test Cases, Test Reports, Bug Reports.

**Deployment:**

Objective: Deliver the software to the end-users and make it operational.

Activities:

Plan and execute deployment procedures.

Install the software in the production environment.

Provide training and documentation to users.

Perform final verification and validation.

Outcome: Deployed Software, Deployment Plan, User Manuals.

**Maintenance:**

Objective: Ensure the software remains functional, efficient, and up-to-date.

Activities:

Monitor software performance and user feedback.

Perform corrective maintenance (bug fixes).

Implement adaptive maintenance (updating software to accommodate changes in the environment).

Conduct preventive maintenance (enhancing performance and reliability).

Outcome: Updated Software, Maintenance Reports.

**Q.4) What is DFD? Create a DFD diagram on Flipkart**

***Answer***

DFD stands for Data Flow Diagram, which is a graphical representation of how data flows through a system. It's used to show the processes, data stores, data flows, and external entities in a system and how they interact with each other.

+---------------------------+

| Flipkart |

+---------------------------+

|

V

+-------------------------+

| Customer Order |

| Management |

+-------------------------+

| |

V V

+----------------+ +-----------------+

| Inventory | | Payment |

| Management | | Processing |

+----------------+ +-----------------+

| |

V V

+------------------+ +------------------+

| Shipping & | | Customer |

| Logistics | | |

+------------------+ +------------------+

**Q** **5. What is Flow chart? Create a flowchart to make addition of two numbers?**

***Answer***

**flowchart:** A flowchart is a type of diagram that represents a workflow or process. A flowchart can also be defined as a diagrammatic representation of an algorithm.

algorithm: A set of rules that must be followed when solving a particular problem.

**Programme:**

#include<stdio.h>

int main()

{

int no1,no2,no3;

printf("enter the first addition value");

scanf("%d", &no1);

printf("enter the second addition value");

scanf("%d", &no2);

no3=no1+no2;

printf("answer is %d", no3);

return 0;

}

**Algorithm steps:**

1) start

2) declare no1, no2 and no3

3) display message, enter the first addition value

4) read value of no1

5) display message, enter the second addition value

6) read value of no2

7) no3=no1+no2

8) display no3

9) stop



**Q6. What is Use case Diagram? Create a use-case on bill payment on Paytm?**

***Answer***

A Use Case Diagram is a visual representation of the functional requirements of a system, showcasing the interactions between various users (actors) and the system's use cases. It helps in illustrating how different actors interact with the system to achieve specific goals.

