SE – Overview of IT Industry

1. What is software? What is software engineering?

Ans. **Software:** Software refers to a set of instructions, data, or programs used to operate computers and execute specific tasks. It encompasses all kinds of programs that run on electronic devices, from operating systems and applications to utilities and middleware.

**Software Engineering:** Software engineering, on the other hand, is a systematic approach to the design, development, testing, and maintenance of software. It involves applying engineering principles and practices to create software that is reliable, efficient, and meets the requirements of its users. Software engineering emphasizes disciplined and structured methodologies to manage the complexity of software systems throughout their lifecycle.

1. Explain types of software?

Ans. **Software:** Software can be categorized into several types based on different criteria such as purpose, usage, and delivery method. Here are some common types of software:

**1. System Software:** This type of software includes operating systems (like Windows, macOS, Linux) and utility software that manage and control computer hardware and provide a platform for other software to run on. Examples include device drivers, system utilities, and compilers.

**2.Application Software:** Application software is designed to perform specific tasks or provide services for end-users. It can be further categorized into:

**- Productivity Software:** Includes tools like word processors (Microsoft Word, Google Docs), spreadsheets (Microsoft Excel, Google Sheets), and presentation software (Microsoft PowerPoint, Google Slides).

- **Graphics Software:** Used for creating and editing images and graphics, such as Adobe Photoshop and CorelDRAW.

-**Media Players and Editors:** Software for playing and editing media files, such as VLC Media Player and Adobe Premiere.

**-Database Software:** Manages data storage and retrieval, like MySQL and Microsoft Access.

**-Games and Entertainment Software:** Includes video games, simulation software, and multimedia applications.

**-Educational Software:** Designed for learning purposes, such as educational games and interactive tutorials.

**3.Embedded Software:** Found within devices and systems other than computers, embedded software controls the functioning of hardware and provides specific functionalities. Examples include software in cars, household appliances, medical devices, and industrial equipment.

**4.Web-based Software:** These are applications accessed via a web browser over the internet. Examples include web-based email clients (Gmail, Outlook), social media platforms (Facebook, Twitter), and online banking systems.

**5.Mobile Apps:** Software designed specifically to run on mobile devices like smartphones and tablets. They can range from games and social networking apps to productivity tools and utilities.

**6.Development Tools:** Software tools used by developers to create, debug, maintain, and support other software. Examples include Integrated Development Environments (IDEs) like Visual Studio, code editors like Sublime Text, and version control systems like Git.

1. What is SDLC? Explain each phase of SDLC?

Ans. SDLC stands for Software Development Life Cycle. It is a structured process used in the development of software products. SDLC aims to produce high-quality software that meets or exceeds customer expectations, is delivered on time and within budget, and is maintainable and scalable. The typical phases of SDLC include:

**1.Requirement Gathering and Analysis:**

**-Purpose:** This phase involves gathering and documenting detailed requirements from stakeholders, including end-users, for the software system.

**-Activities:** Requirements are collected through interviews, workshops, and discussions. They are then analyzed for feasibility, completeness, and consistency.

**-Deliverables:** Requirement specifications document, functional and non-functional requirements.

**2.System Design:**

**-Purpose:** In this phase, the system architecture and design are planned based on the gathered requirements. The goal is to transform requirements into a blueprint for the software to be developed.

**-Activities:** Architectural design specifies the overall structure, components, interfaces, and data for the system. Detailed design focuses on internal logic, data structures, and algorithms.

**-Deliverables:** System architecture diagrams, detailed design documents, database schemas.

3.Implementation (Coding):

**-Purpose:** This phase involves actual coding or programming of the software based on the design specifications.

**-Activities:** Developers write code in accordance with coding standards and guidelines. Unit testing is performed to detect and fix defects early.

**-Deliverables:** Source code, unit test cases, developer documentation.

4. Testing:

**-Purpose:** The software is tested to ensure that it meets the specified requirements and is free of defects.

-**Activities**: Different types of testing are performed, including functional testing, integration testing, system testing, and acceptance testing. Defects found are reported and fixed.

-**Deliverables**: Test plans, test cases, defect reports, test execution reports.

5. Deployment and Integration:

- **Purpose**: The software is deployed to the production environment and integrated with other systems as required.

- **Activities**: Installation, configuration, and deployment of the software. Data migration may be performed if necessary. Integration with existing systems is tested.

- **Deliverables**: Deployed software system, installation guides, migration plans.

6. Maintenance:

-**Purpose**: After deployment, the software enters the maintenance phase where it is maintained and updated to address issues that arise and to accommodate changes in requirements.

- **Activities**: Bug fixes, performance improvements, updates for compliance with new regulations, and enhancements based on user feedback.

- **Deliverables**: Patch releases, updated documentation, support and training materials.

1. What is DFD? Create a DFD diagram on Flipkart?

Ans. Data Flow Diagram (DFD) represents the flow of data within information systems. Data Flow Diagrams (DFD) provide a graphical representation of the data flow of a system that can be understood by both technical and non-technical users. The models enable software engineers, customers, and users to work together effectively during the analysis and specification of requirements.

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

| Order Processing |

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

|

v (Customer Order)

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

| Order Validation | (Process)

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

| (Valid Order)

v

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

| Payment Processing | (Process)

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

| (Payment Successful)

v

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

| Inventory Check | (Process)

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

| (Available Stock)

v

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

| Order Fulfillment | (Process)

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

|

v (Order Confirmation)

Customer (Entity)

|

v (Shipment Information)

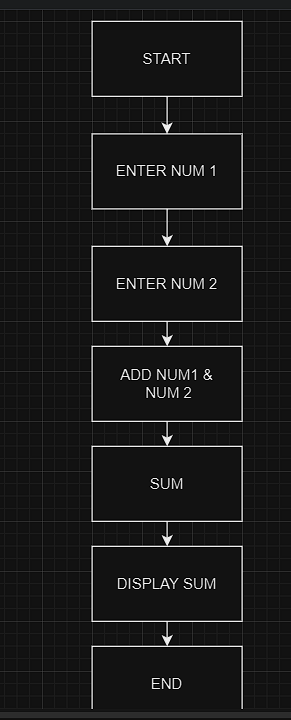
+-------------------+

| Delivery |

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

1. What is Flow chart? Create a flowchart to make addition of two numbers?

Ans. Certainly! A flowchart is a graphical representation of a process, algorithm, or workflow that uses symbols like boxes, diamonds (decision shapes), arrows, and text to depict the steps involved, their order, and any decision points that need to be made. They are widely used in various fields, including computer programming, process documentation, and planning, to visually communicate processes.



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

Ans. A Use Case Diagram is a visual representation of interactions between users (actors) and a system, depicting the functional requirements and scope of the system. It helps in understanding the user's goals and how they interact with the system to achieve those goals

