Software Engineering Assignment

MODULE: 1

SE – Overview of IT Industry

1)What is software? What is software engineering?

ANS:-

Software is a program or set of programs containing instructions which provide desired functionality.

Software engineering is the art of developing quality software on time and within budget.And it is also a systematic approach to the design, development, operation, and maintenance of a software system.

2)Explain types of software?

ANS:-

There are three groups of softwares
1) System Software
2) Application Software
3) Programming Software

1) System Software:- It provides the basic functions for computer usage and helps to run the computer hardware and system. This software are used by the computer to translate inputs from various language which a machine can understand to complete task.
eg:- Linux, window, macOS, Android, iOS

2) Application Software**:-** It is the general designation of computer programs for performing user tasks. Types of application software are:

i) Mobile app:- It is an application that runs on mobile
eg:- Instagram, facebook, etc
ii) Desktop app:- This apps runs in a desktop or laptop computer.
eg:- Microsoft office suite which includes Word, Excel and PowerPoint.
eg:- Outlook for email, and firefox, Google Chrome, Mozilla are the web browser.
iii) Web app:- This runs on a web browser.
eg:- google.com, facebook.com, etc

3) Programming Softwar**e:-** It is the process of designing, writing, testing, debugging, and maintaining of source code of computer programs. The purpose of programming is to create a program that exhibits a certain desired behavior.
eg:- c++, html, java, Simlab, php, Python and Visual basic.

3)What is SDLC? Explain each phase of SDLC

ANS:-

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.
The methodology within the SDLC process can vary across industries and organizations, but standards such as ISO/IEC 12207 represent processes that establish a lifecycle for software, and provide a mode for the development, acquisition, and configuration of software systems.
There are six types of phases in SDLC:-

1)Requirement Gathering

- Features
- Usage scenarios
- Although requirements may be documented in written form, they may be incomplete, unambiguous, or even incorrect.
- Requirements will Change!
- Inadequately captured or expressed in the first place
- User and business needs change during the project
- Validation is needed throughout the software lifecycle, not only when the ―final system‖ is delivered.
- Build constant feedback into the project plan
- Plan for change
- Early prototyping [e.g., UI] can help clarify the requirements
- Functional and Non-Functional
- Requirements definitions usually consist of natural language, supplemented by (e.g., UML) diagrams and tables.

- Three types of problems can arise:
- Lack of clarity: It is hard to write documents that are both precise and easy-to- read.
- Requirements confusion:Functional and Non-functional requirements tend to be intertwined.
- Requirements Amalgamation: Several different requirements may be expressed together.

- Types of Requirements:
- Functional Requirements: describe system services or functions.
- Compute sales tax on a purchase
- Update the database on the server

 Non-Functional Requirements: are constraints on the system or the development process. Non-functional requirements may be more critical than functional requirements.

If these are not met, the system is useless!

2)Analysis Phase

 The analysis phase defines the requirements of the system, independent of how these Requirements will be accomplished.

 This phase defines the problem that the customer is trying to solve.

 The deliverable result at the end of this phase is a requirement document.

 Ideally, this document states in a clear and precise fashion what is to be built.

 This analysis represents the ―what" phase.

 The requirement documentaries to capture the requirements from the customer's perspective by defining goals.

 This phase starts with the requirement document delivered by the requirement phase and mapsthe requirements into architecture.

 The architecture defines the components, their interfaces and behaviors.

 The deliverable design document is the architecture.

 This phase represents the ―how" phase.

 Details on computer programming languages and environments, machines, packages, application architecture, distributed architecture layering, memory size, platform, algorithms, data structures, global type definitions, interfaces, and many other engineering details are established.

 The design may include the usage of existing components.

3)**Design Phase**

 Design Architecture Document

 Implementation Plan

 Critical Priority Analysis

 Performance Analysis

 Test Plan

 The Design team can now expand upon the information established in the requirement Document.

 The requirement document must guide this decision process.

 Analyzing the trade-offs of necessary complexity allows for many things to remain simple which, in turn, will eventually lead to a higher quality product. The architecture team also converts the typical scenarios into a test plan.

4)Implementation Phase

 In the implementation phase, the team builds the components either from scratch or by composition.

 Given the architecture document from the design phase and the requirement document from

the analysis phase, the team should build exactly what has been requested, though there isstill room for innovation and flexibility.
 For example, a component may be narrowly designed for this particular system, or the Component may be made more general to satisfy a reusability guideline.
 Implementation - Code
 Critical Error Removal
 The implementation phase deals with issues of quality, performance, baselines, libraries, and debugging.

5)Testing Phase

 Simply stated, quality is very important. Many companies have not learned that quality is important and deliver more claimed functionality but at a lower quality level.
 It is much easier to explain to a customer why there is a missing feature than to explain to a customer why the product lacks quality.
 A customer satisfied with the quality of a product will remain loyal and wait for new Functionality in the next version.
 Quality is a distinguishing attribute of a system indicating the degree of excellence.
 Regression Testing
 Internal Testing
 Unit Testing
 Application Testing
 Stress Testing
 The testing phase is a separate phase which is performed by a different team after the implementation is completed.
 There is merit in this approach; it is hard to see one‘s own mistakes, and a fresh eye can discover obvious errors much faster than the person who has read and re-read the material many times.
 Unfortunately, delegating (alternate) testing to another team leads to as lack (dull) attitude regarding quality by the implementation team.
 If the teams are to be known as craftsmen, then the teams should be responsible for establishing high quality across all phases.
 An attitude change must take place to guarantee quality. Regardless if testing is done after the-fact or continuously, testing is usually based on a regression technique split into severalmajor focuses, namely internal, unit, application, and stress.

6)Maintenance Phase

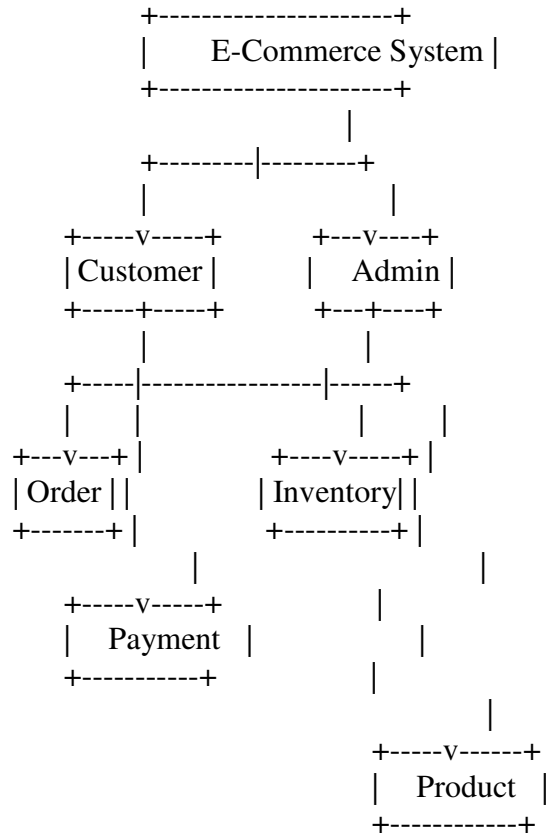
- Software maintenance is one of the activities in software engineering, and is the process of enhancing and optimizing deployed software (software release), as well as fixing defects.
- Software maintenance is also one of the phases in the System Development Life Cycle (SDLC), as it applies to software development. The maintenance phase is the phase which comes after deployment of the software into the field.
- The developing organization or team will have some mechanism to document and track Defects and deficiencies.

- configuration and version management
- reengineering (redesigning and refactoring)
- updating all analysis, design and user documentation
- Repeatable, automated tests enable evolution and refactoring

4)What is DFD? Create a DFD diagram on Flipkart

ANS:-

A Data Flow Diagram (DFD) is a traditional way to visualize the information flows within a system. A neat and clear DFD can depict a good amount of the system requirements graphically. It can be manual, automated, or a combination of both. It shows how information enters and leaves the system, what changes the information and where information is stored. The purpose of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communications tool between a systems analyst and any person who plays a part in the system that acts as the starting point for redesigning a system.



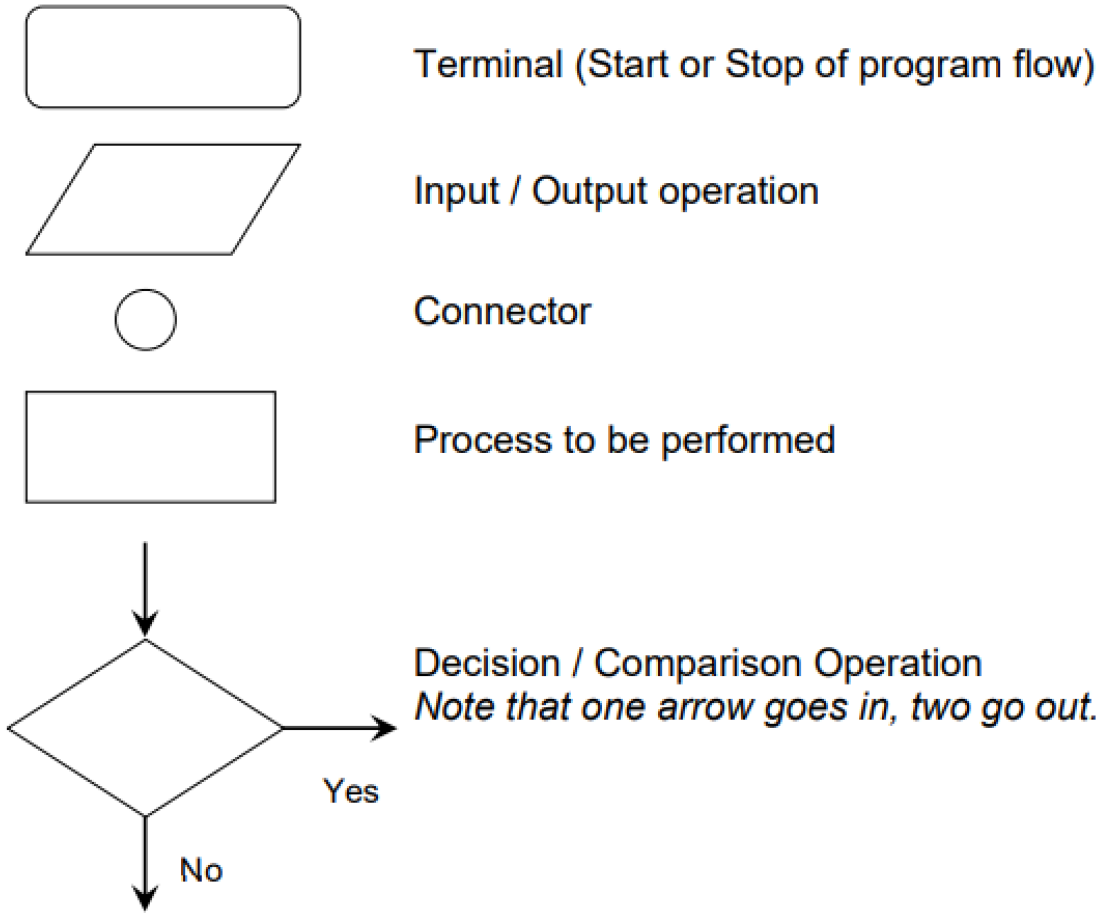
5)What is Flow chart? Create a flowchart to make addition of two numbers

ANS:-

A flowchart is a graphical representation of the operations involved in a data processing system.

- Symbols are used to represent particular operations or data.
- Flow lines indicate the sequence of operations.

Some of flowchart example are:-

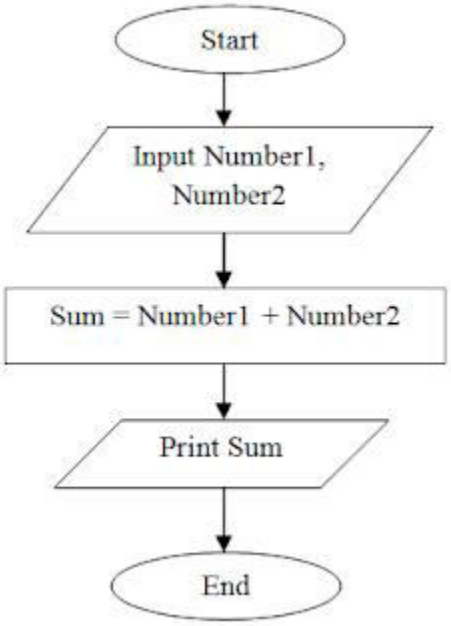


Different flow chart symbols have different meanings. The most common flow chart symbols are:

Terminator: An oval flow chart shape indicating the start or end of the process.

Process: A rectangular flow chart shape indicating a normal process flow step.

Decision: A diamond flow chart shape indication a branch in the process flow.



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

ANS:-

Use-case diagrams which are used in software engineering describe the high-level functions and scope of a system. These diagrams also identify the interactions between the system and its actors. The use cases and actors in use-case diagrams describe what the system does and how the actors use it, but not how the system operates internally.

i)Bill Aggregation:- Opens the Paytm app on smartphone and navigates to the "Bill Payments" section. Then selects the relevant category, such as "Utilities," to view a list of available billers.

ii)Adding Billers:- You have to adds electricity, water, and internet service providers as billers by entering the necessary details, such as account numbers. Paytm securely saves this information, eliminating the need to re-enter it for future transactions.

iii)Bill Notifications:- Paytm sends timely notifications to you, reminding you of upcoming bill due dates. The app consolidates all bill information, providing a clear overview of the amounts due and the due dates.

iv)One-Tap Payments:- On the due date, you will receives a notification with a one-tap payment option. Than opens the Paytm app, reviews the bills, and pays them seamlessly using your linked bank account, credit/debit card, or Paytm wallet.

v)Transaction History:- Paytm maintains a comprehensive transaction history, allowing you to track all your bill payments in one place. The app provides digital receipts and confirmation details for each transaction.

vi)Rewards and Cashback:- As a bonus, Paytm offers rewards and cashback on bill payments, incentivizing you to use the platform regularly. You can use these rewards for future bill payments, adding an extra layer of value to her experience.