***Assignment no.1***

***SE – Overview of IT Industry***

Ques:1:-What is software? What is software engineering?

Ans:- Software is a set of instructions, data or programs used to operate computers and execute specific tasks. It is the opposite of hardware, which describes the physical aspects of a computer. Software is a generic term used to refer to applications, scripts and programs that run on a device. It can be thought of as the variable part of a computer, while hardware is the invariable part.

The two main categories of software are application software and system software. An application is software that fulfills a specific need or performs tasks.

Other types of software include programming software, which provides the programming tools software developers need; middleware, which sits between system software and applications; and driver software, which operates computer devices and peripherals.

----->Software Engineering:-

The term software engineering is the product of two words, software, and engineering.

The software is a collection of integrated programs.

Software subsists of carefully-organized instructions and code written by developers on any of various particular computer languages.

Computer programs and related documentation such as requirements, design models and user manuals.

Engineering is the application of scientific and practical knowledge to invent, design, build, maintain, and improve frameworks, processes, etc.

Software Engineering is an engineering branch related to the evolution of software product using well-defined scientific principles, techniques, and procedures. The result of software engineering is an effective and reliable software product.

----->Why is Software Engineering required?

Software Engineering is required due to the following reasons:

1)To manage Large software.

2)For more Scalability.

3)Cost Management.

4)To manage the dynamic nature of software.

5)For better quality Management.

----->Need of Software Engineering

1)Huge Programming: It is simpler to manufacture a wall than to a house or building, similarly, as the measure of programming become extensive engineering has to step to give it a scientific process.

2)Adaptability: If the software procedure were not based on scientific and engineering ideas, it would be simpler to re-create new software than to scale an existing one.

3)Cost: As the hardware industry has demonstrated its skills and huge manufacturing has let down the cost of computer and electronic hardware.

4)Dynamic Nature: The continually growing and adapting nature of programming hugely depends upon the environment in which the client works.

5)Quality Management: Better procedure of software development provides a better and quality software product.

----->Importance of Software Engineering:-

1)Reduces complexity: Big software is always complicated and challenging to progress. Software engineering has a great solution to reduce the complication of any project.

2)To minimize software cost: Software needs a lot of hardwork and software engineers are highly paid experts.

3)To decrease time: Anything that is not made according to the project always wastes time. And if you are making great software, then you may need to run many codes to get the definitive running code.

4)Handling big projects: Big projects are not done in a couple of days, and they need lots of patience, planning, and management.

5)Reliable software: Software should be secure, means if you have delivered the software, then it should work for at least its given time or subscription.

6)Effectiveness: Effectiveness comes if anything has made according to the standards.

-------------------------------------------------------------------------------

Ques:2:-Explain types of software?

Ans:- Software is a collection of instructions, data, or computer programs that are used to run machines and carry out particular activities. It is the antithesis of hardware, which refers to a computer’s external components. A device’s running programs, scripts, and applications are collectively referred to as “software” in this context.

In a computer system, the software is basically a set of instructions or commands that tell a computer what to do. In other words, the software is a computer program that provides a set of instructions to execute a user’s commands and tell the computer what to do. For example like MS-Word, MS-Excel, PowerPoint, etc.

Types of Software

It is a collection of data that is given to the computer to complete a particular task. The chart below describes the types of software:

System Software

Operating System

Language Processor

Device Driver

Application Software

General Purpose Software

Customize Software

Utility Software

1.System Software->

System software is software that directly operates the computer hardware and provides the basic functionality to the users as well as to the other software to operate smoothly. Or in other words, system software basically controls a computer’s internal functioning and also controls hardware devices such as monitors, printers, and storage devices, etc.

----->Types of System Software

It has two subtypes which are:

1).Operating System: It is the main program of a computer system. When the computer system ON it is the first software that loads into the computer’s memory. Basically, it manages all the resources such as computer memory, CPU, printer, hard disk, etc.

2).Language Processor: As we know that system software converts the human-readable language into a machine language and vice versa. So, the conversion is done by the language processor.

3).Device Driver: A device driver is a program or software that controls a device and helps that device to perform its functions. Every device like a printer, mouse, modem, etc.

----->Features of System Software.

----->Let us discuss some of the features of System Software:

--->System Software is closer to the computer system.

--->System Software is written in a low-level language in general.

--->System software is difficult to design and understand.

--->System software is fast in speed(working speed).

--->System software is less interactive for the users in comparison to application software.

----->Application Software:-

Software that performs special functions or provides functions that are much more than the basic operation of the computer is known as application software. Or in other words, application software is designed to perform a specific task for end-users. It is a product or a program that is designed only to fulfill end-users’ requirements. It includes word processors, spreadsheets, database management, inventory, payroll programs, etc.

----->Types of Application Software:-

There are different types of application software and those are:-

1)General Purpose Software: This type of application software is used for a variety of tasks and it is not limited to performing a specific task only. For example, MS-Word, MS-Excel, PowerPoint, etc.

2)Customized Software: This type of application software is used or designed to perform specific tasks or functions or designed for specific organizations. For example, railway reservation system, airline reservation system, invoice management system, etc.

3)Utility Software: This type of application software is used to support the computer infrastructure. It is designed to analyze, configure, optimize and maintains the system, and take care of its requirements as well. For example, antivirus, disk fragmenter, memory tester, disk repair, disk cleaners, registry cleaners, disk space analyzer, etc.

----->Features of Application Software:-

Let us discuss some of the features of Application Software:-

1)An important feature of application software is it performs more specialized tasks like word processing, spreadsheets, email, etc.

2)Mostly, the size of the software is big, so it requires more storage space.

3)Application software is more interactive for the users, so it is easy to use and design.

4)The application software is easy to design and understand.

5)Application software is written in a high-level language in general.

-------------------------------------------------------------------------------

Ques:3:-What is SDLC? Explain each phase of SDLC?

Ans:-

The software development life cycle is a process that development teams use to create awesome software that's top-notch in terms of quality, cost-effectiveness, and time efficiency. The main goal is to minimize risks and make sure the software meets the customer's expectations both during and after production.

This process is about creating a detailed plan to guide the development of the product and then breaking down the development process into smaller modules that can be assigned, completed, and measured to make the whole thing more manageable.

----->Here are some specific benefits of using SDLC for the product team:-

1)Increased visibility of the development process for all stakeholders involved.

2)More efficient estimation, planning, and scheduling.

3)Improved risk management and cost estimation.

4)A systematic approach to delivering software that meets customer expectations and improves satisfaction.

----->The 6 Phases of the Software Development Life Cycle:-

1)----Requirement gathering:-This phase is critical for converting the information gathered during the planning and analysis phase into clear requirements for the development team. This process guides the development of several important documents: a software requirement specification (SRS), a Use Case document, and a Requirement Traceability Matrix document.

2)----Analysis:- The first phase of the SDLC is the project planning stage where you are gathering business requirements from your client or stakeholders. This phase is when you evaluate the feasibility of creating the product, revenue potential, the cost of production, the needs of the end-users, etc.

To properly decide what to make, what not to make, and what to make first, you can use a feature prioritization framework that takes into account the value of the software/update, the cost, the time it takes to build, and other factors.

Once it is decided that the software project is in line with business and stakeholder goals, feasible to create, and addresses user needs, then you can move on to the next phase.

3)----Designing:- The design phase is where you put pen to paper—so to speak. The original plan and vision are elaborated into a software design document (SDD) that includes the system design, programming language, templates, platform to use, and application security measures. This is also where you can flowchart how the software responds to user actions.

In most cases, the design phase will include the development of a prototype model. Creating a pre-production version of the product can give the team the opportunity to visualize what the product will look like and make changes without having to go through the hassle of rewriting code.

frontend part:-static development:-HTML,CSS.

4)----Coding / Implementation:- backend part:-Dynamic development:-

python,java,php,.net,nodejs.........

5)----Testing:- Before getting the software product out the door to the production environment, it’s important to have your quality assurance team perform validation testing to make sure it is functioning properly and does what it’s meant to do. The testing process can also help hash out any major user experience issues and security issues.

In some cases, software testing can be done in a simulated environment. Other simpler tests can also be automated.

----->The types of testing to do in this phase:-

1)Performance testing: Assesses the software's speed and scalability under different conditions.

2)Functional testing: Verifies that the software meets the requirements.

3)Security testing: Identifies potential vulnerabilities and weaknesses.

4)Unit-testing: Tests individual units or components of the software.

5)Usability testing: Evaluates the software's user interface and overall user experience.

6)Acceptance testing: Also termed end-user testing, beta testing, application testing, or field testing, this is the final testing stage to test if the software product delivers on what it promises.

6)----Maintenance:- The maintenance phase is the final stage of the SDLC if you’re following the waterfall structure of the software development process. However, the industry is moving towards a more agile software development approach where maintenance is only a stage for further improvement.

In the maintenance stage, users may find bugs and errors that were missed in the earlier testing phase. These bugs need to be fixed for better user experience and retention. In some cases, these can lead to going back to the first step of the software development life cycle.

The SDLC phases can also restart for any new features you may want to add in your next release/update.

-------------------------------------------------------------------------------

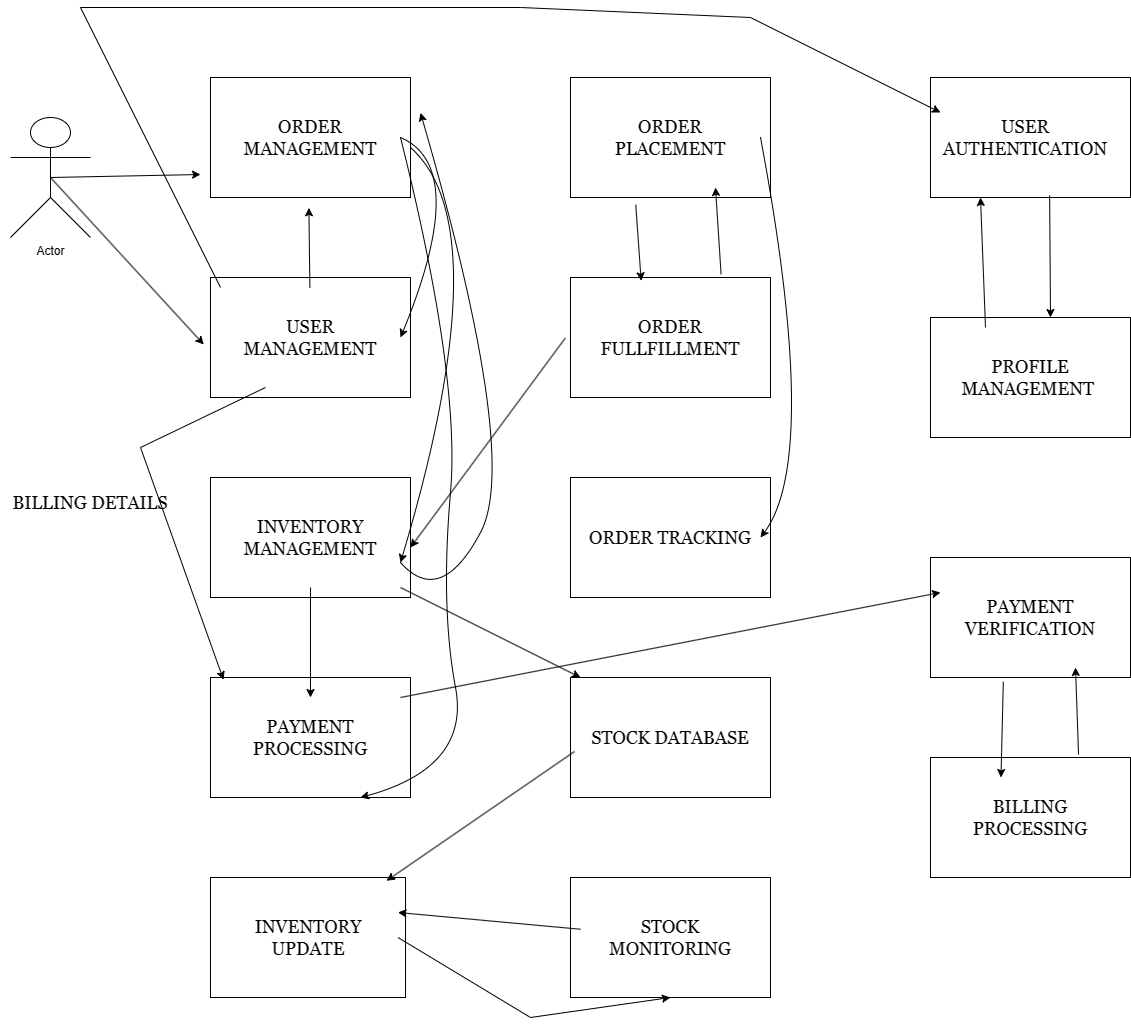
QUES:4-> What is DFD? Create a DFD diagram on Flipkart ?

Ans--->

----->DFD stands for Data Flow Diagram. It's a graphical representation of the flow of data through a system, illustrating how input data is transformed into output data through various processes. DFDs are commonly used in software engineering to model the flow of data within systems.

---->Creating a DFD for Flipkart, a popular e-commerce platform, involves identifying the main components of the system and illustrating how data flows between them.

---->Here's a simplified DFD for Flipkart:-



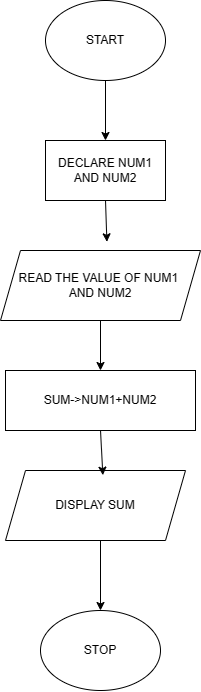
-------------------------------------------------------------------------------

QUES:5-> What is Flow chart? Create a flowchart to make addition of two numbers.

Ans---->

---🡪A flowchart is a graphical representation of a process or algorithm, typically using standardized symbols to illustrate the steps involved and the flow of control or data within the system. It's commonly used in various fields such as software engineering, business process modeling, and education.

Here's a simple flowchart to illustrate the process of adding two numbers:



1.Start: The beginning of the process.

2.Input first number (NUM1): This step involves the user inputting the first number to be added.

3.Input second number (num2): The user inputs the second number to be added.

4.read the value of num1 and num2: it displays the values of num1 and num2.

5. sum (num1 + num2): The two input numbers are added together to calculate the sum.

6.Display sum: The sum of the two numbers is displayed.

7.stop: The end of the process.

QUES:6-> What is Use case Diagram? Create a use-case on bill payment on patym.

ANS-🡪

--🡪 A use case diagram is a graphical representation of the interactions between actors (users or external systems) and a system, showing how the system will be used in various scenarios. It helps to visualize the functionalities of a system from the perspective of its users.

