Software engineering assignment

Module – 1(software)

1. What Is Software? Explain types of software.

→ Software is a set of data, programs, directives, and attestation that perform various predetermined tasks on a computer system.
→ They empower users to interact with the computer system.
→ Computer software usually interacts with the operating system to perform specific tasks. And the operating system itself is the leading computer software..

❖ Types of software:-Computer software consists of two primary software types:

1. System Software
2. Application Software

1.System Software:-

→ System software controls the computer's inner functioning, mainly through an Operating System.
→ It also monetizes external devices such as monitors, printers, and storage devices.
→ OS acts as the principal system software for every machine. We use System software to operate the computer itself.
→ System Software executes in the backdrop, managing the computer's elementary functions with the aim that a user can manage to control higher-level applications to accomplish a particular task.

- Examples related to System Software are as follows:

I. Operating System (OS)

→ The most well-known category of system software is the Operating system. Being a collection of software helps the users to manage their resources, providing comprehensive services for the separate applications that operate over them.
→ Examples of Operating systems are MS Windows, Android, iOS, Mac OS, CentOS, Linux, etc.

II. Device Drivers

→ The principal function of device drivers is to administer specific hardware connected to the system. Some hardware devices that involve drivers to tie up with any system include sound cards, displays, mice, and hard disks.
→ Some examples of Device Drivers are Display Drivers, ROM Drivers, Printer Drivers, BIOS Drivers, USB Drivers, Motherboard Drivers, VGA Drivers, etc.

III. Utility software

→ Utility Software is that system software that helps users sustain the decent and stable functioning of a Computer System.
→ It also aids the Operating System in managing, organizing, and maintaining the computer system's functionality.
→ It executes duties such as; file backup, virus observation, deleting rejected data, installation and uninstallation, etc.
→ Some examples are antivirus software, file management tools, compression tools, disk management tools, etc.
→ Examples of utility software are filing administration devices, disk management devices, software for antivirus, compression tools, etc.

IV. Programming Language Translators

→ These type of translators helps users to address instructions in a distinct programming language, and the language translator converts them into a machine code. The computer system then examines the given instructions in machine code and executes them.
→ Examples related to Programming Language Translators are Compiler, Interpreter, and Assemblers.

2. Application Software:-

→ Application software is different from system software for many reasons, also identified as end-user programs or productivity programs.
→ It helps users accomplish their tasks such as research online, setting an alarm clock, graphics design, logging into an account, doing calculations, etc. All these functionalities prevail much above the system software.

- The various categories of Application Software are:

I. Word Processors

→ The term word processor indicates that it processes words with paragraphs and pages.
→ There are three types of Word processors: mechanical, software, and electronic.
→ Further, along with these functionalities, it also helps save, format, and print documents. A few examples of word processors are Google Docs, MS Word, Apple iWork- Pages, etc.

II. Database Software

→ Users use this software to formulate and execute their databases.
→ The term database software is also known as the Database Management System (DBMS).
→ They also help users to organize their data. Examples are FoxPro, Clipper, etc.

III. Multimedia Software

→ Multimedia software can design or record images play audio and video files.
→ Users use this software to edit videos and create graphics and animations.

→ Some examples related to multimedia software are Picasa, CorelDRAW, Adobe Photoshop, Adobe After Effects, VLC Media Player, etc.

IV. Web Browser

→ Web browsers help users to browse the Internet.
→ They help find and retrieve data crosswise across the web.
→ Examples of web browsers are Internet Explorer, Microsoft Edge, Google Chrome, Safari, etc.

2. Explain the SDLC Each phase process.

- **What is SDLC?**

→ One of the fundamental procedures of developing software in a step by step manner is by following the Software Development Life Cycle (SDLC).
→ SDLC is a popular practice that is followed by different organizations for designing and developing high-quality software applications.
→ It acts as a framework that holds some specific tasks to be achieved at every phase during the software development progression..

- **Phases of the Software Development Life Cycle:**

The various phases of SDLC are explained below:

1. First Phase: Requirement Collection or Planning Phase

→ The prime focus of this phase is to gather the essential requirements from the customer. This information gets collected by the business analyst from their target customer(s) and plans the BRS (Business requirement Specification) for the development of the product.

→ The team of all the designers and BA will do brainstorming to extract all the requirements and plan accordingly for the new system to be developed.

→ Some popular questions that this meeting picks up are - Who will use the product? What must be the output data by the product?

2. Second Phase: Defining or Feasibility Study Phase - Analysis

→ When the BRS documentation is done, there are another set of employees like Human Resource (HR), Finance Analyst, Architect, a Business analyst as well as Project manager will sit jointly discuss as well as analyze how to proceed and whether it is feasible and possible in the allotted budget.

→ Such decisions are taken depending on the cost, resources, time, etc. Documentation is made, which is the SRS (Software Requirement Specification) document, which contains a detailed explanation of product requirements, right from design to development.

3. Third Phase: Designing Phase

→ This phase is when the design specification is organized from the prerequisite document when the project is approved to go further.

→ This phase contributes to the next phase for development.

→ This phase portrays a blueprint of the product, which helps to specify the hardware and requirements of your system as well as assist in crafting a significant architecture of your system.

4. Fourth Phase: Building or Coding Phase - Implementation

→ As you are preparing with the design document, this phase deals with the developers to start writing the code or prepare for the engineering so that a prototype of the product can be created using some specific tools and techniques.

→ This is considered the longest phase of SDLC.

5. Fifth Phase: Testing Phase

→ As your product is prepared for deployment, it needs a prior testing environment by the test engineers to check for bugs and run-time errors, and they check in this phase whether the functionality of the product is working as per the requirement or not.

→ The bugs or defects which are encountered in the test phase are reported to the developers, who fix the bug and revert to the test engineers for further testing.

→ This is an iterative process that continues until your application is free from bugs and defects and works stably.

6. Sixth Phase: Deployment Phase

→ Once your prototype or product is developed, tested, and completely in working form as per the requirement, and then it is installed or deployed in the customer's workplace or system for their use.

7. Seventh Phase: Maintenance Phase

→ This is an additional phase, and in many cases, this phase does not come under the count of SDLC, when your customer(s) begin using your product and encounter with some issues which they want us (as developers) to fix from time to time.

→ The developer fixes the issue, and software testers test the product and hand it over the back to the customer.