

MODULE : 1

What is software? What is software engineering?

Software is a set of instruction, 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.

Software engineering is a systematic engineering approach to software development. A software engineer is a person who applies the principles of software engineering to design, develop, maintain, test, and evaluate computer software.

 Explain types of software.

There are total three types of software =>

1. System software :If you think of software as being in layers, the system software is the bottom layer. It sits betweenthe hardware and the application software.

Operating systems like Windows, macOS, Android and iOS are examples of system software. Operating systems are loaded into RAM when the device starts up, and have access to the hard drive.

2. Utility software : Utility software is part of the system software and performs specific tasks to keep the computer running. Utility software is always running in the background. Examples of utility software are security and optimisation programs.

Security programs include anti-virus software that scans and removes viruses. Most computers will include some sort of anti-virus software, but you can add your own.

Optimisation programs can include tools for system clean-up, disk defragmentation, and file comprission. These tools are typically installed as part of the operating system. They heve access to the hard drive to keep it tidy.

3. Application software : This is everything else! Anything that is not an operating system or a utility is an application or app. So a word processor, spreadsheet, web browser, and graphics software are all examples of application software, and they can do many specific tasks.

You can remove and add applications on your computer using the operating system.Application software like a word processor regularly directs the operating system to load and save files from and to the hard drive. When you are working on a file, it is saved temporarily in the RAM. It is only when you choose to save that it is written to the hard drive.

This is why, if the computer crashess while you're working on a file, you may lose any changes you didn't save. Data stored in the RAM is volatile. The data is lost when the RAM loses power.

What is SDLC? Explain each phase of SDLC

SDLC or the Software Development Life Cycle is a process that produces software with the highest quality and lowest cost in the shortest time possible.

SDLC provides a well-structed flow of phases that help an organization to quickly produce high-quality software which is well-tested and ready for production use. 

There are total seven(7) phases of SDLC :

1. Planning => In the Planning phase, project leaders evaluate the terms of the project. This includes calculating labor and material costs, creating a timetable with target, and creating the project's teams and leadership structure.

Planning can also include feedback from stakeholders. Stackholders are anyone who stands to benefit from the application.

Try to get feedback from potential customers, developers, subject matter experts, and sales reps.

Planning should clearly define the scope and purpose of the application. It plosts the course and provisions the team to effectively create the software. It also sets boundaries to help keep the project from expanding or shufting from its original purpose.

2. Define Recuirments=>

Defining requirements is considered part of planning to determine what the application is supposed to do and its requirements. For example, a social media application would require the ability to connect with a friend. An inventory program might require a search feature. Requirements also include defining the resources needed to build the project. For example, a team might develop software to control a custom manufacturing machine. The machine is a requirement in the process.

3. Design and Prototyping=>

The Design phase models the way a software application will work. Some aspects of the design include:

Architecture – Specifies programming language, industry practices, overall design, and use of any templates or boilerplate

User Interface – Defines the ways customers interact with the software, and how the software responds to input

Platforms – Defines the platforms on which the software will run, such as Apple, Android, Windows version, Linux, or even gaming consoles

Programming – Not just the programming language, but including methods of solving problems and performing tasks in the application

Communications – Defines the methods that the application can communicate with other assets, such as a central server or other instances of the application

Security – Defines the measures taken to secure+ the application, and may include SSL traffic encryption, password protection, and secure storage of user credentials Prototyping can be a part of the Design phase. A prototype is like one of the early versions of software in the Iterative software development model. It demonstrates a basic idea of how the application looks and works. This “hands-on” design can be shown to stakeholders. Use feedback o improve the application. It’s less expensive to change the Prototype phase than to rewrite code to make a change in the Development phase.

4. Software Development =>

This is the actual writing of the program. A small project might be written by a single developer, while a large project might be broken up and worked by several teams. Use an Access Control or Source Code Management application in this phase. These systems help developers track changes to the code. They also help ensure compatibility between different team projects and to make sure target goals are being met.

The coding process includes many other tasks. Many developers need to brush up on skills or work as a team. Finding and fixing errors and glitches is critical. Tasks often hold up the development process, such as waiting for test results or compiling code so an application can run. SDLC can anticipate these delays so that developers can be tasked with other duties.

Software developers appreciate instructions and explanations. Documentation can be a formal process, including wiring a user guide for the application. It can also be informal, like comments in the source code that explain why a developer used a certain procedure. Even companies that strive to create software that’s easy and intuitive benefit from the documentation.

Documentation can be a quick guided tour of the application’s basic features that display on the first launch. It can be video tutorials for complex tasks. Written documentation like user guides, troubleshooting guides, and FAQ’s help users solve problems or technical questions.

5. Testing => It’s critical to test an application before making it available to users. Much of the testing can be automated, like security testing. Other testing can only be done in a specific environment – consider creating a simulated production environment for complex deployments. Testing should ensure that each function works correctly. Different parts of the application should also be tested to work seamlessly together—performance test, to reduce any hangs or lags in processing. The testing phase helps reduce the number of bugs and glitches that users encounter. This leads to a higher user satisfaction and a better usage rate.

6. Development =>

In the development phase, the application is made available to users. Many companies prefer to automate the development phase. This can be as simple as a payment portal and download link on the company website. It could also be downloading an application on a smartphone.

Deployment can also be complex. Upgrading a company-wide database to a newly-developed application is one example. Because there are several other systems used by the database, integrating the upgrade can take more time and effort.

7. Operations and Maintenance =>

At this point, the development cycle is almost finished. The application in done and being used in the field. The Operation and Maintenance phase is still important, though. In this phase, users discover bugs that weren't found during testing. These errors need to ne resolved,which can spawn new development cycles.

In addition to bug fixes, models like Iterative development plan additional features in future releases. For each new release, a newDevelopment Cycle can be launched.

What is DFD? Create a DFD diagram on Flipkart

A data flow diagram(DFD) is a way of representing a flow of data through a process or a system. The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow - there are no decision rules and no loops.

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

Flow chart => A diagram that shows the connections between different stages of a process or parts of a system is known as flow chart.

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

Use case Diagram =>

A use case diagram is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use

cases and different types of users the system has and will often be accompanied by other types of diagrams as well. The use cases are

represented by either circles or replies.