Name : Dhruvik sutariya

Modul:1

1.What is software? What is software engineering?

:-software is a programming or set of programs containing instructions that provide desired functionality. And engineering is the process of designing and building that serves a particular purpose and find a cost-effective solution to problem

Software engineering is the process of designing ,developing ,testing , and maintaining software it is a systematic and disciplined approach to software development that aims to create high-quality ,reliable and maintainable software. software engineering includes a variety of techniques ,tools ,and methodologies ,including requirement analysis ,design , testing and maintenance.

Some key principles of software engineering include:

1. modularity : breaking the software into smaller , reusable components that can be developed and tested independently.

2. abstraction: hiding the implementation details of a component and exposing only the necessary functionality to other parts of the software.

3. encapsulation : wrapping up the data and function of an object into a single unit and protecting

2.explen type of software?

Among the various categories of software, the most common types include the following:

* **Application software.**The most common type of software, application software is a computer software package that performs a specific function for a user, or in some cases, for another application. An application can be self-contained, or it can be a group of programs that run the application for the user. Examples of [modern applications](https://www.techtarget.com/searchcio/feature/The-rise-of-modern-applications-Why-you-need-them) include office suites, graphics software, databases and database management programs, web browsers, word processors, software development tools, image editors and communication platforms.
* **System software.** These software programs are designed to run a computer's application programs and hardware. System software coordinates the activities and functions of the hardware and software. In addition, it controls the operations of the computer hardware and provides an environment or platform for all the other types of software to work in. The OS is the best example of system software; it manages all the other computer programs. Other examples of system software include the [firmware](https://www.techtarget.com/whatis/definition/firmware), computer language translators and system [utilities](https://www.techtarget.com/whatis/definition/utility).
* **Driver software.**Also known as device drivers, this software is often considered a type of system software. Device drivers control the devices and peripherals connected to a computer, enabling them to perform their specific tasks. Every device that is connected to a computer needs at least one device driver to function. Examples include software that comes with any nonstandard hardware, including special game controllers, as well as the software that enables standard hardware, such as USB storage devices, keyboards, headphones and printers.
* **Middleware.**The term *middleware* describes software that mediates between application and system software or between two different kinds of application software. For example, middleware enables Microsoft Windows to talk to Excel and Word. It is also used to send a remote work request from an application in a computer that has one kind of OS, to an application in a computer with a different OS. It also enables newer applications to work with legacy ones.
* **Programming software.** Computer programmers use programming software to write code. Programming software and programming tools enable developers to develop, write, test and [debug](https://www.techtarget.com/searchsoftwarequality/definition/debugging) other software programs. Examples of programming software include assemblers, compilers, debuggers and interpreters.

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

Ans : he software development life cycle (SDLC) is the process of planning, writing, modifying, and maintaining software. Developers use the methodology as they design and write modern software for computers, cloud deployment, mobile phones, video games, and more. Adhering to the SDLC methodology helps to optimize the final outcome.

**1. Planning & 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 feture 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.

**2. Define Requirements**

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.

Top of Form

Bottom of Form

**3. Design**

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 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.

**4. Development**

The actual development phase is where the development team members divide the project into software modules and turn the software requirement into code that makes the product.

This SDLC phase can take quite a lot of time. It’s important to have a set timeline and milestones so the software developers understand the expectations and you can keep track of the progress in this stage.

In some cases, the development stage can also merge with the testing stage where certain tests are run to ensure there are no critical bugs.

**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:

* **Performance testing:** Assesses the software's speed and scalability under different conditions
* **Functional testing:** Verifies that the software meets the requirements
* **Security testing:** Identifies potential vulnerabilities and weaknesses
* **Unit-testing:**Tests individual units or components of the software
* **Usability testing:**Evaluates the software's user interface and overall user experience
* **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. Deployment**

During the deployment phase, your final product is delivered to your intended user. You can automate this process and schedule your deployment depending on the type. For example, if you are only deploying a feature update, you can do so with a small number of users (canary release). If you are creating brand-new software, you can learn more about the different stages of software release life cycle  (SRLC).

**7. 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.

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

A picture is worth a thousand words. 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.

It is usually beginning with a context diagram as level 0 of the DFD diagram, a simple representation of the whole system. To elaborate further from that, we drill down to a level 1 diagram with lower-level functions decomposed from the major functions of the system. This could continue to evolve to become a level 2 diagram when further analysis is required. Progression to levels 3, 4 and so on is possible but anything beyond level 3 is not very common. Please bear in mind that the level of detail for decomposing a particular function depending on the complexity that function.

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

Ans : A **flowchart** is a graphical representation of a procedure or algorithm in the form of a diagram. You can convert a complex process into a bright and straightforward method using a flowchart and make it understandable. Besides, if you need to frame a flowchart, you do not need a professional. Instead, you can create it in your way. Flowchart symbols like a diamond, round, parallelogram, give life to a dead diagram. So, let us move forward to know the types of flowcharts and how to make a flowchart in the upcoming sections.

\*Addition of two number in flowchart



.

6. what is use case ? drowa use case diagram on payment on paytm.

Ans: A use case diagram is a way to summarize details of a system and the users within that system.

Create a use-case on bill payment on paytm.

