**Name: Shah Shrey Mahendrabhai**

**Module 1:**

Q1: 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](https://www.techtarget.com/whatis/definition/script) 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.

🡪**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.

1. Software engineering includes a variety of techniques, tools, and methodologies, including requirements analysis, design, testing, and maintenance.
2. It is a rapidly evolving field, and new tools and technologies are constantly being developed to improve the software development process.
3. By following the principles of software engineering and using the appropriate tools and methodologies, software developers can create high-quality, reliable, and maintainable software that meets the needs of its users.
4. Software Engineering is mainly used for large projects based on software systems rather than single programs or applications.
5. The main goal of Software Engineering is to develop software applications for improving quality, budget, and time efficiency.
6. Software Engineering ensures that the software that has to be built should be consistent, correct, also on budget, on time, and within the required requirements.

Q2: Explain types of software?

Ans-Types of software:

* **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, computer language translators and system utilities.
* **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 other software programs. Examples of programming software include assemblers, compilers, debuggers and interpreters.

Q3: What is SDLC? Explain each phase of SDLC

Ans: The Software Development Life Cycle (SDLC) is a process used by software development organizations to plan, design, develop, test, deploy, and maintain software applications.

**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.

**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) or [product specification](https://theproductmanager.com/topics/product-specification/), a Use Case document, and a Requirement Traceability Matrix document.

**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.

**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 and [specialized development tools](https://theproductmanager.com/tools/product-development-software/). 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.

**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.

**6. Deployment**

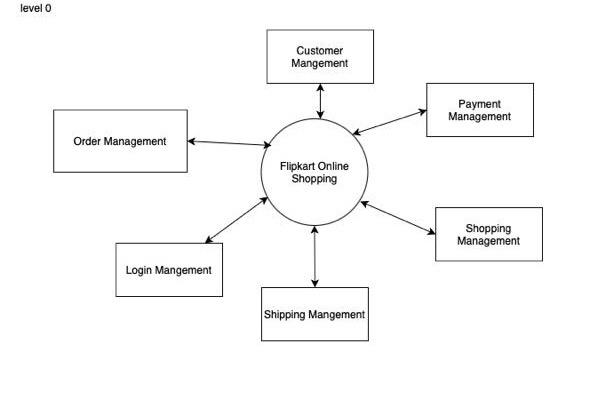
During the deployment phase, your final product is delivered to your intended user. You can automate this process and schedule your deployment depending on the type. For example, if you are only deploying a feature update, you can do so with a small number of users (canary release). If you are creating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

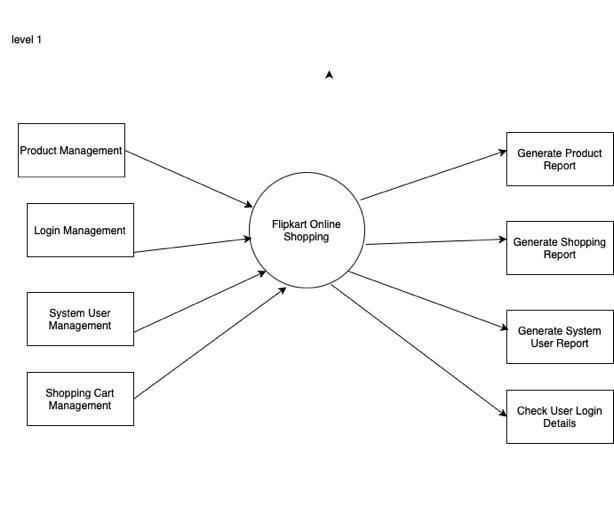
**7. Maintenance**

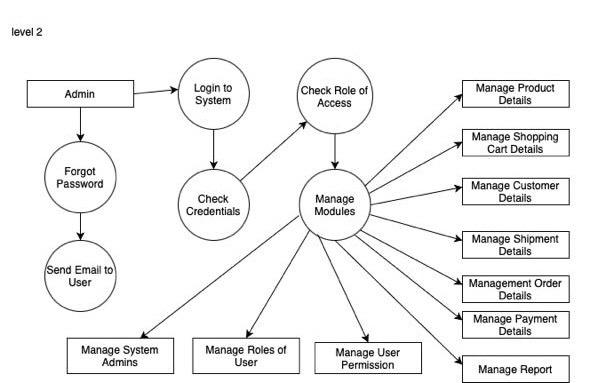
The maintenance phase is the final stage of the SDLC if you’re following the [waterfall](https://theproductmanager.com/topics/agile-vs-waterfall-methodology/) structure of the software development process. However, the industry is moving towards a more agile software development approach where maintenance 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.

Q4: What is DFD? Create a DFD diagram on Flipkart

Ans: Data flow diagram (DFD) is a diagram being used frequently in software design. It visually represents the flow of data throughout processes in a given system. DFD shows the kind of information that will be input to and output from processes as well as where the data will be stored.

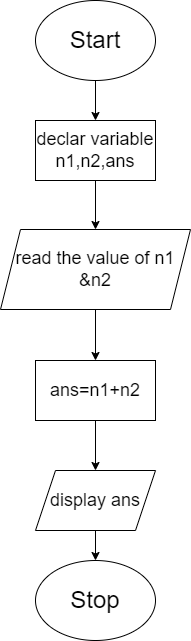






Q5: What is Flow chart? Create a flowchart to make addition of two numbers

Ans: A flow chart is a graphical or symbolic representation of a process. Each step in the process is represented by a different symbol and contains a short description of the process step. The flow chart symbols are linked together with arrows showing the process flow direction.



Q6: What is Use case Diagram? Create a use-case on bill payment on Paytm.

Ans: A Use Case Diagram is a vital tool in system design, it provides a visual representation of how users interact with a system. It serves as a blueprint for understanding the functional requirements of a system from a user’s perspective, aiding in the communication between stakeholders and guiding the development process.

