Module 1

**Software**

Software is a collection of instructions, data, or computer programs that are used to run machines and carry out particular activity .

Software engineering

Software engineering is the systematic approach to designing, developing, testing, and managing software systems. Software engineering involves the implementation of computer science, engineering principles, and mathematical concepts to create, test, and analyze software applications and systems that are efficient, reliable, and secure.

**types of software:**

**Application software**

The most frequently used software is application software, which 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** :- graphics software, databases

**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.

**Examples :-**operating system

**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, helping them perform their specific tasks. Every device that's connected to a computer needs at least one device driver to function

**Examples** :- keyboards,

**Middleware**.

. The term middleware describes software that mediates between application and system software or between two different kinds of application software

.**Examples** :- Excel and Word.

**Programming software.**

Computer programmers use programming software to write code. Programming software and programming languages, such as Java or Python, let developers develop, write, test and debug other software programs.

**Examples** :- compilers,

**SDLC**

The **software development lifecycle** (SDLC) is the cost-effective and time-efficient process that development teams use to design and build high-quality software.

1 **PLANNING** ➖

The planning phase typically includes tasks like cost-benefit analysis, scheduling, resource estimation, and allocation. The development team collects requirements from several stakeholders such as customers, internal and external experts, and managers to create a software requirement specification document.

2 **ANALYSIS:**-

Analysis is also performed by the developers of the organization. This is attained from customer inputs, and sales department/market surveys. The information from this analysis forms the building blocks of a basic project. The quality of the project is a result of planning

3 **DESIGN**➖

In the design phase, software engineers analyze requirements and identify the best solutions to create the software. For example, they may consider integrating pre-existing modules, make technology choices, and identify development tools. They will look at how to best integrate the new software into any existing IT infrastructure the organization may have.

4 **IMPLEMENTATION:**-

In the implementation phase, the development team codes the product. They analyze the requirements to identify smaller coding tasks they can do daily to achieve the final result.

5 **TESTING**:-

The development team combines automation and manual testing to check the software for bugs. Quality analysis includes testing the software for errors and checking if it meets customer requirements. Because many teams immediately test the code they write, the testing phase often runs parallel to the development phase.

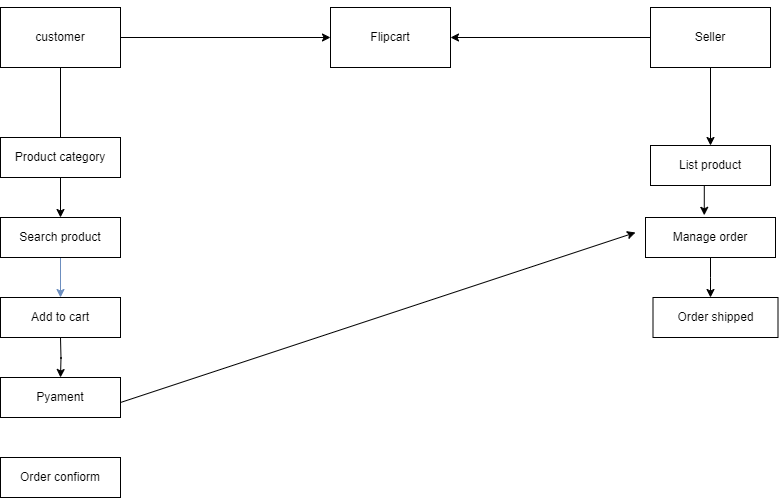
6 **Maintenance:**-

In the maintenance phase, among other tasks, the team fixes bugs, resolves customer issues, and manages software changes. In addition, the team monitors overall system performance, security, and user experience to identify new ways to improve the existing software.

**DFD**

A **data-flow diagram** 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.

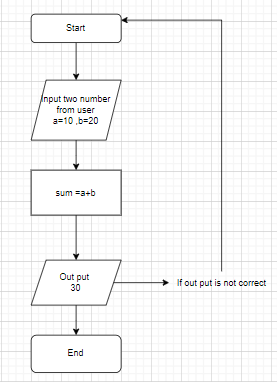
**DFD DIAGRAM ON FLIPKART**



## **What is a flowchart?**

A flowchart is a diagram that depicts a process, system or computer algorithm. They are widely used in multiple fields to document, study, plan, improve and communicate often complex processes in clear, easy-to-understand diagrams. use rectangles, ovals, diamonds and

potentially numerous other shapes to define the type of step, along with connecting arrows to define flow and sequence.



**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 ellipses. The actors are often shown as stick figures