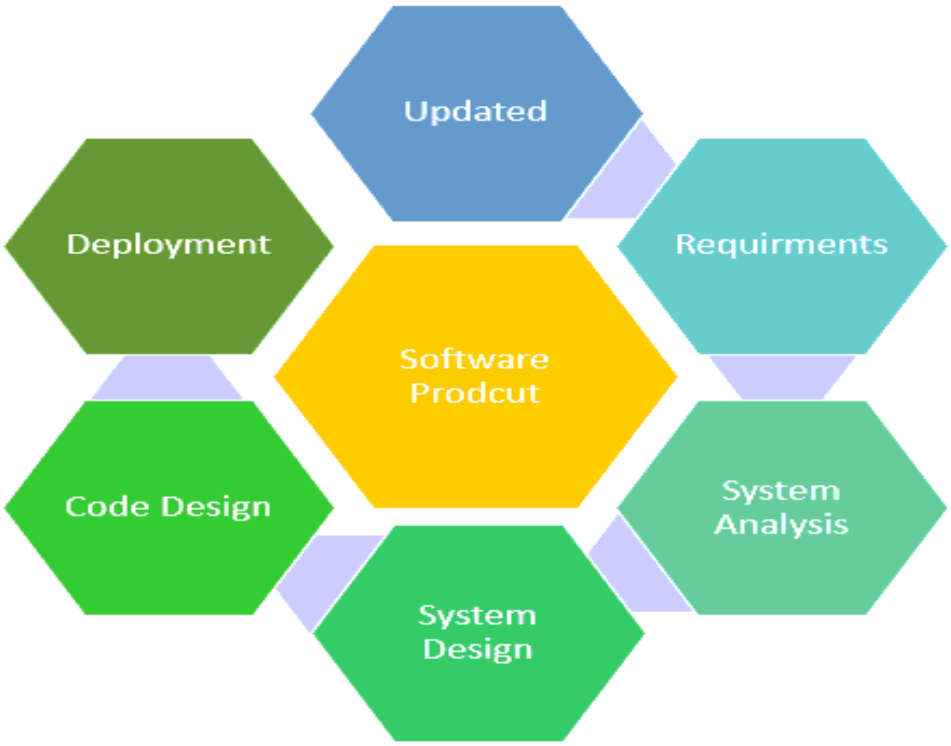
MODULE : 1

SE – Overview of IT Industry

1. Whatissoftware? What issoftware engineering?

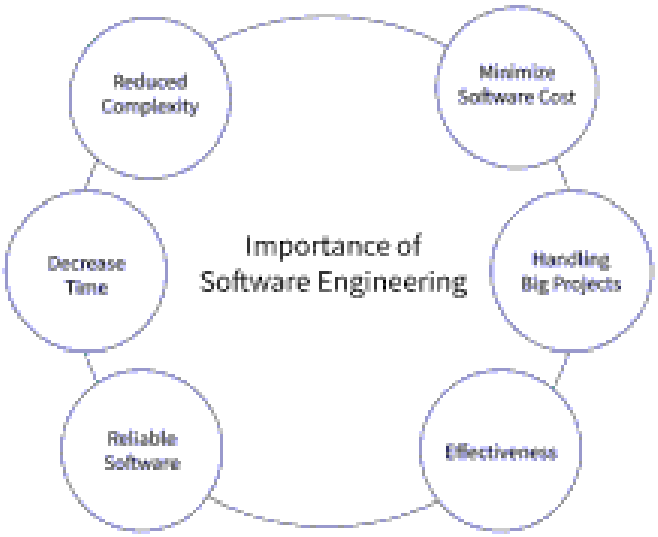
- **Ans)** Software refers to a set of instructions, programs, or data used to operate computers and execute specific tasks. It is intangible and consists of code written in various programming languages that tells a computer what to do Software Engineering is a discipline within computer science that focuses on the systematic approach to the design, development, testing, maintenance, and management of software systems. It applies engineering principles and practices to software development, aiming to produce high-quality, reliable, and scalable software solutions that meet user requirements within time and budget constraints.

    - Software engineering includes a variety of techniques, tools, and methodologies, including requirements analysis, design, testing, and maintenance.
    - It is a rapidly evolving field, and new tools and technologies are constantly being developed to improve the software development process.
    - 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.
    - Software Engineering is mainly used for large projects based on software systems rather than single programs or applications.
    - The main goal of Software Engineering is to develop software applications for improving quality, budget, and time efficiency.
    - 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.



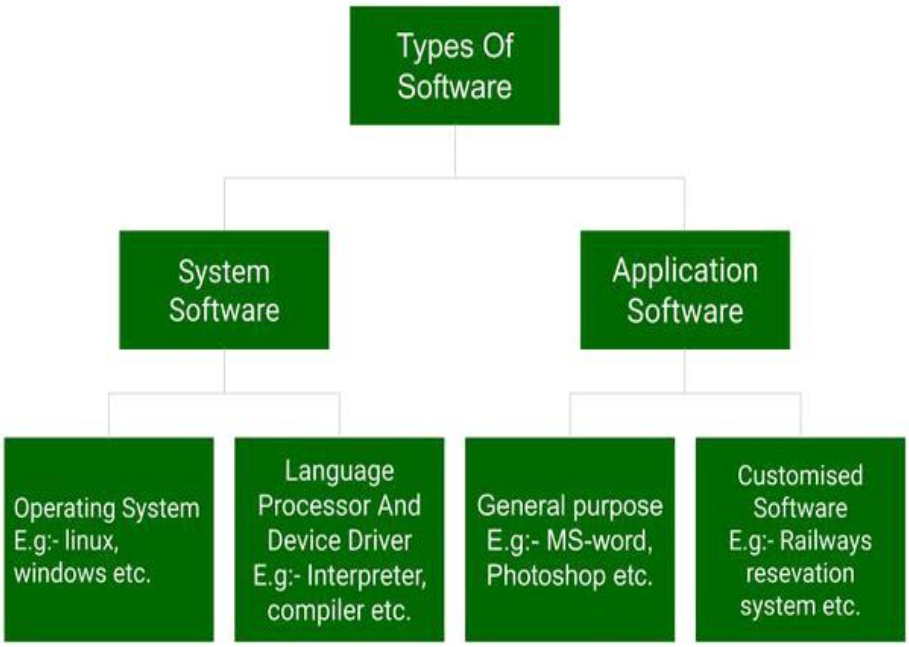
Software Engineering is a subfield of engineering concerned with the development of software products by the use of precise scientific methods, procedures, and concepts. A dependable and efficient software product is the end product of software engineering.

Importance of Software Engineering :



Q2. Explain types of software.

It is an assortment of information sent to the computer to do a specific duty. The types of software are described in the chart below:

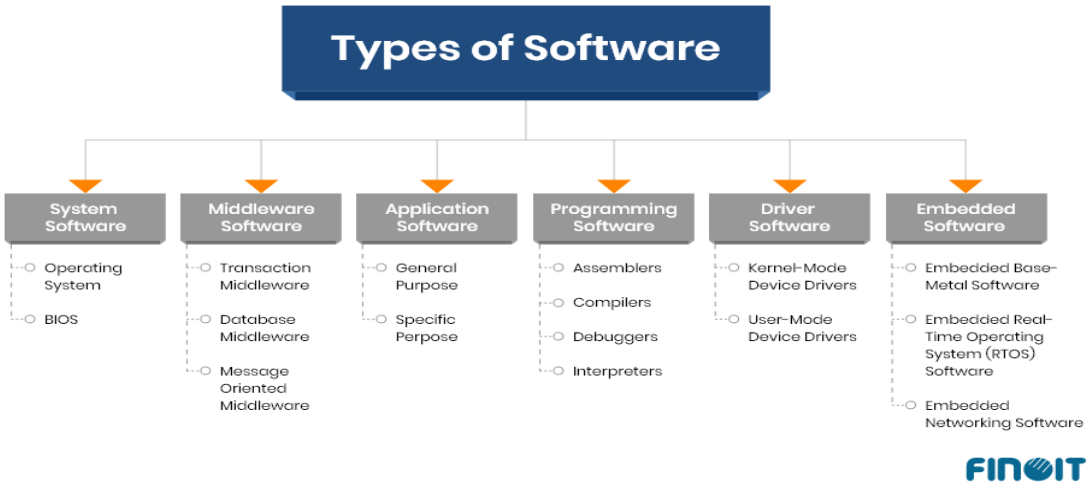


Above is the diagram of types of software. Now we will briefly describe each type and its subtypes:

- **System Software**
    - Operating System
    - Language Processor
    - Device Driver
- **Application Software**
    - General Purpose Software
    - Customize Software
    - Utility Software

System Software is a subset of software that controls computer hardware resources and offers an operating environment for application software. It facilitates communication and interaction between the user and the hardware by acting as a go-between. System software is necessary for a computer system to function and carry out activities successfully and efficiently.

Application Software refers to a class of computer software intended to carry out particular functions or applications for users. It is not to be confused with system software, which includes the operating system, which controls and operates the computer itself. Application software is created to meet the requirements of users, be they companies, organisations, or individuals. Word processors, spreadsheets, email clients, web browsers, games, and multimedia players are a few examples of application software.



Q3. What is SDLC? Explain each phase of SDLC.

- **SDLC (Software Development Life Cycle)** Software Development Life Cycle is referred to as SDLC. Software developers use an organised process to efficiently design, create, test, and implement high-quality software. There are multiple stages in the SDLC process, and each has its own set of tasks, products, and goals. Usually, the phases consist of:

  To offer an orderly and systematic method for developing software: A framework for managing the software development life cycle (SDLC) is provided, aiding in making sure that all essential procedures are followed and that the finished product satisfies the specifications.

1.To ensure that the software is of high quality: Testing and quality assurance stages of the SDLC assist guarantee that the programme is error-free and

complies with specifications.

2.To manage risks and costs: Early risk identification and management during the development process is made possible by the SDLC, which may help organisations cut costs and lessen the effect of any problems that do occur.

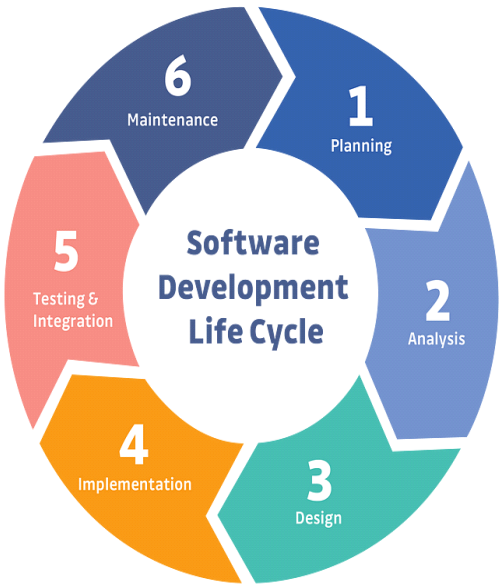
3.To improve communication and collaboration: The Software Development Life Cycle (SDLC) facilitates the involvement of all stakeholders in the development process and ensures that their needs are met, including clients, end users, and developers.

4.To improve efficiency and productivity: Organisations can increase productivity and efficiency by optimising resource usage and streamlining the development process with the aid of the SDLC.

5.To increase the likelihood of a successful project outcome: The likelihood of a project succeeding can be considerably increased by adhering to a well-defined SDLC process, since it provides the team with efficient and methodical guidance towards the objective.

All things considered, the software development life cycle (SDLC) is an invaluable resource that businesses can employ to ensure that software applications are developed to a high standard, that specifications are met, and that the project is completed on schedule and within budget.

The following phases are usually included in the SDLC:



- **Requirements analysis and collection:** In this stage, information regarding the software requirements is gathered from clients, business

analysts, and end users, among other stakeholders.

- **Design:** This stage involves creating the software design, which includes the interfaces, data structures, and general programme architecture. There are two phases to it:
- **High-level design (HLD):** It provides software product architecture.
  - **Low-level design (LLD):** It describes how each and every feature in the product should work and every component.

4 . **Implementation or coding:** After that, the idea is put into code, usually over the course of multiple iterations. This stage is also known as development.
things you need to know about this phase:

- In the SDLC paradigm, this is the longest phase.
- Middleware, Backend, and Front End make up this phase.
- **In front-end:** Coding development is complete, including the creation of SEO settings.
- **In Middleware:** They connect both the front end and back end.
- **In the back-end:** A database is created.

5.Testing: The programme is extensively tested to make sure it satisfies the specifications and functions as intended.

6. Deployment: The programme is released to end users and put into a production environment following successful testing.

7. Maintenance: Updates to the programme, bug patches, and continuous support are included throughout this phase.
Organisations can utilise a variety of approaches, including Waterfall, Agile, Scrum, V-Model, and DevOps, to implement the SDLC.

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

- The acronym for Data Flow Diagram is DFD. The data flow within a system is shown graphically, showing the input, processing, storing, and output of data. In software engineering and business analysis, DFDs are frequently used to simulate the structure and For Flipkart to create a DFD diagram, a thorough understanding of its own internal systems would be necessary. I can, however,

give you a broad idea of the elements that might be present in such a diagram:

- **External Entities :** These stand in for external entities that communicate with the system. This could apply to Flipkart's clients, vendors, and delivery providers.
- **Process:** These stand in for the different processes or features that the system uses to manipulate data. Processes at Flipkart could involve handling orders, managing inventories, processing payments, and more.
- **Data stores:** These show the locations of the system's data storage. This could include databases containing order histories, product catalogues, and consumer information for Flipkart.
- **Data flows:** These show how data moves between the data stores, processes, and external entities. Data flows could involve things like customers placing orders, retrieving product details from the database, updating customers on their orders, etc.

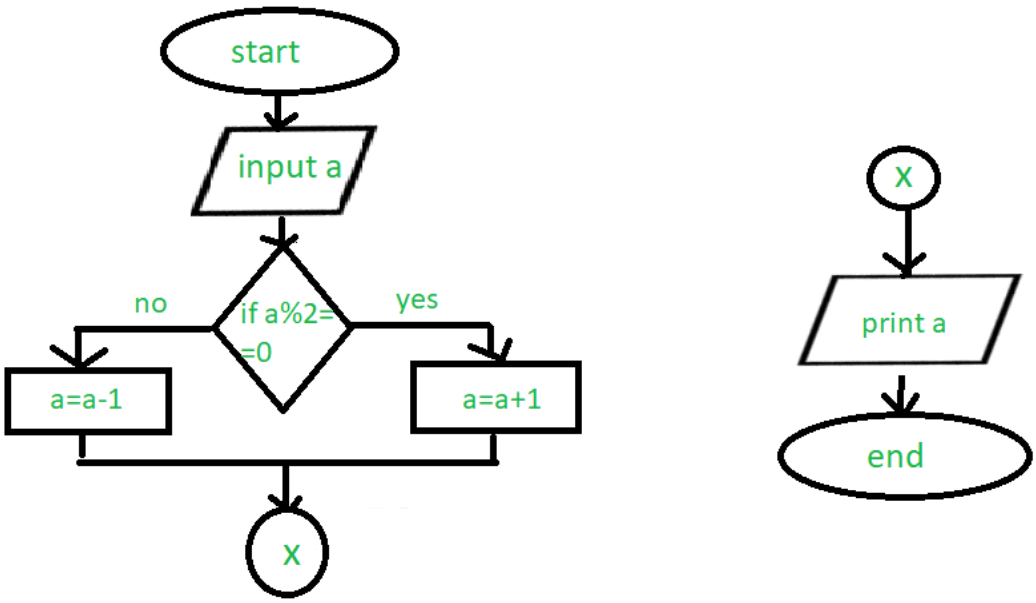
Here's a very basic and simplified example of a DFD diagram for Flipkart:

USER

Level 0 DFD [Data Flow Diagram] of Flip cart

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

- A flowchart is a graphic depiction of an algorithm or process that usually shows the steps in the algorithm's flow by connecting symbols with arrows. It's frequently used to show workflows, procedures, or decision-making processes in a variety of industries.
- This straightforward flowchart shows how to add two numbers:



In this flowchart:

- "Start" is represented by an oval or rounded rectangle.
- "Input num1 and num2" is represented by a parallelogram.
- "Add num1 and num2" is represented by a rectangle.
- "Display result" is represented by a parallelogram.
- "End" is represented by an oval or rounded rectangle.

Arrows connecting each step show how the steps flow into one another.

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

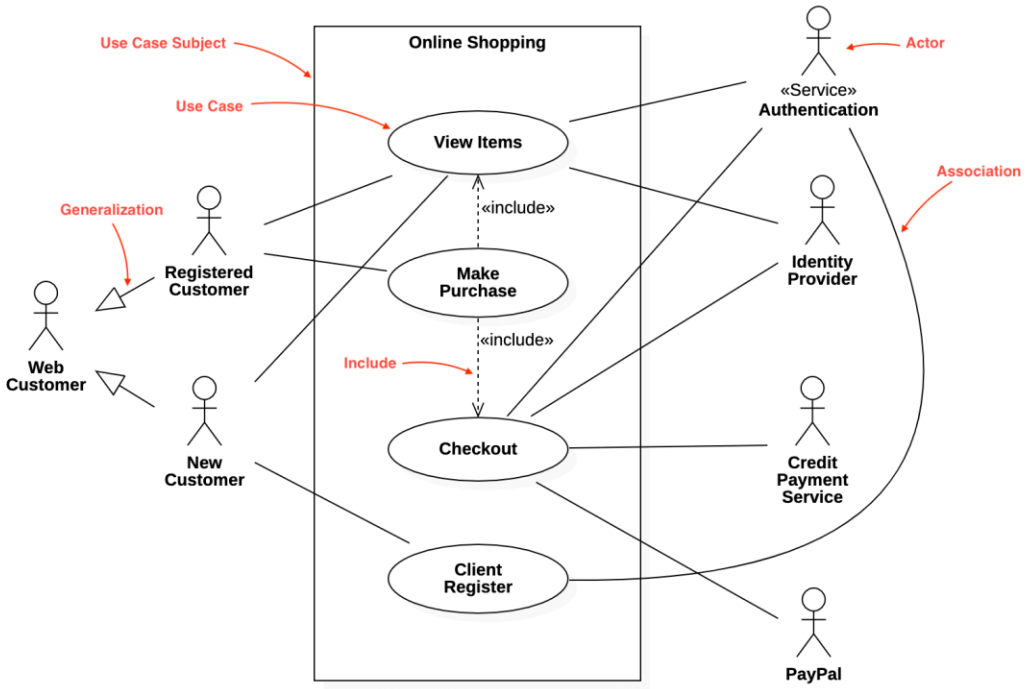
- A visual depiction of the interactions between a system and its users, a use case diagram shows how users work with the system to accomplish specific objectives. It is made up of use cases (tasks or goals), actors (users or other systems), and the connections between them.

  Here's an example of a use case diagram for bill payment on Paytm

In this Diagram:

- **Actors:**
- **User:** The person using the Paytm application to pay bills.
- **Use Cases:**
- **Login to Paytm:** The user logs in to their Paytm account.
- **Select Bill Payment:** The user selects the option for bill payment.
- **Enter Bill Details:** The user enters the details of the bill they wish to pay.
- **Choose Payment Method:** The user chooses the payment method from options like wallet balance, credit/debit card, net banking, etc.
- **Authorize Payment:** The user confirms and authorizes the payment.
- **Process Payment:** Paytm processes the payment transaction.
- **Update Payment Status:** Paytm updates the payment status and provides confirmation to the user.
- **Relationships:**
- **Actor-Use Case Relationships:** Arrows connecting actors to use cases represent interactions. For example, the User interacts with all the listed use cases.
- **Association between Use Cases:** Lines connecting use cases indicate relationships between them. For instance, all the use cases are linked to the "Process Payment" use case, indicating that they are steps involved in the payment process.

- This use case diagram illustrates the flow of actions involved in bill payment through the Paytm application, outlining the interactions between the user and the system to achieve the goal of paying a bill.


Use Case Subject
Online Shopping
Actor
Use Case
«Service»
Authentication
View Items
Association
Generalization
«include»
Registered
Customer
Make
Purchase
Identity
Provider
Web
Customer
«include»
Include
New
Customer
Checkout
Credit
Payment
Service
Client
Register
PayPal