OVERVEIW OF IT INDUSTRY

MODULE 1

JANUARY 19, 2025

- What is software? What is software engineering?

Software is a set of instructions that tells a device what to do. It's the opposite of hardware, which is the physical components of a device. Software is essential for making a device useful and allowing users to interact with it.

Software engineering is the branch of computer science that deals with the design, development, testing, and maintenance of software applications. Software engineers apply engineering principles and knowledge of programming languages to build software solutions for end users.

Software engineers design and develop computer games, business applications, operating systems, network control systems, and middleware—to name just a few of the many career paths available.

- Explain types of software

There are two types of software:

1. SYSTEM SOFTWARE

It is a software that directly operates the computer hardware and provides the basic functionality to the users as well as to the other software to operate smoothly. Or in other words, system software basically controls a computer’s internal functioning and also controls hardware devices such as monitors, printers, and storage devices, etc.

Types of system software:

a) Operating System: It is the main program of a computer system. When the computer system ON it is the first software that loads into the computer's memory. Basically, it manages all the resources such as computer memory, CPU, printer, hard disk, etc., and provides an interface to the user, which helps the user to interact with the computer system. It also provides various services to other computer software. Examples of operating systems are Linux, Apple macOS, Microsoft Windows, etc.

b) Language Processor: As we know that system software converts the human-readable language into a machine language and vice versa. So, the conversion is done by the language processor. It converts programs written in high-level programming languages like Java, C, C++, Python (known as source code), into sets of instructions that are easily readable by machines (known as object code or machine code).

c) Device Driver: A device driver is a program or software that controls a device and helps that device to perform its functions. Every device like a printer, mouse, modem, etc. needs a driver to connect with the computer system eternally. So, when you connect a new device with your computer system, first you need to install the driver of that device so that your operating system knows how to control or manage that device.

2. APPLICATION SOFTWARE:

Software that performs special functions or provides functions that are much more than the basic operation of the computer is known as application software. Or in other words, application software is designed to perform a specific task for end-users. It is a product or a program that is designed only to fulfill end-users' requirements. It includes word processors, spreadsheets, database management, inventory, payroll programs, etc.

Types of application software

a) General Purpose Software: This type of application software is used for a variety of tasks and it is not limited to performing a specific task only. For example, MS-Word, MS-Excel, PowerPoint, etc.

b) customized Software: This type of application software is used or designed to perform specific tasks or functions or designed for specific organizations. For example, railway reservation system, airline reservation system, invoice management system, etc.

c) Utility Software: This type of application software is used to support the computer infrastructure. It is designed to analyze, configure, optimize and maintains the system, and take care of its requirements as well. For example, antivirus, disk fragment, memory tester, disk repair, disk cleaners, registry cleaners, disk space analyzer, etc.

- What is SDLC? Explain each phase of SDLC

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

The phases of SDLC are:

a) Planning and 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 feature prioritization

framework that takes into account the value of the software/update, the cost, the time it takes to build, and other factors.

b) 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, a Use Case document, and a Requirement Traceability Matrix document.

c) 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 a 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.

d) 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. 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.

e) 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:

i. Performance testing: Assesses the software's speed and scalability under different conditions.
ii. Functional testing: Verifies that the software meets the requirements.
iii. Security testing: Identifies potential vulnerabilities and weaknesses.
iv. Unit-testing: Tests individual units or components of the software
v. Usability testing: Evaluates the software's user interface and overall user experience.
vi. Acceptance testing: Also termed end-user testing, beta testing, application testing, or field testing, this is the final testing stage to test

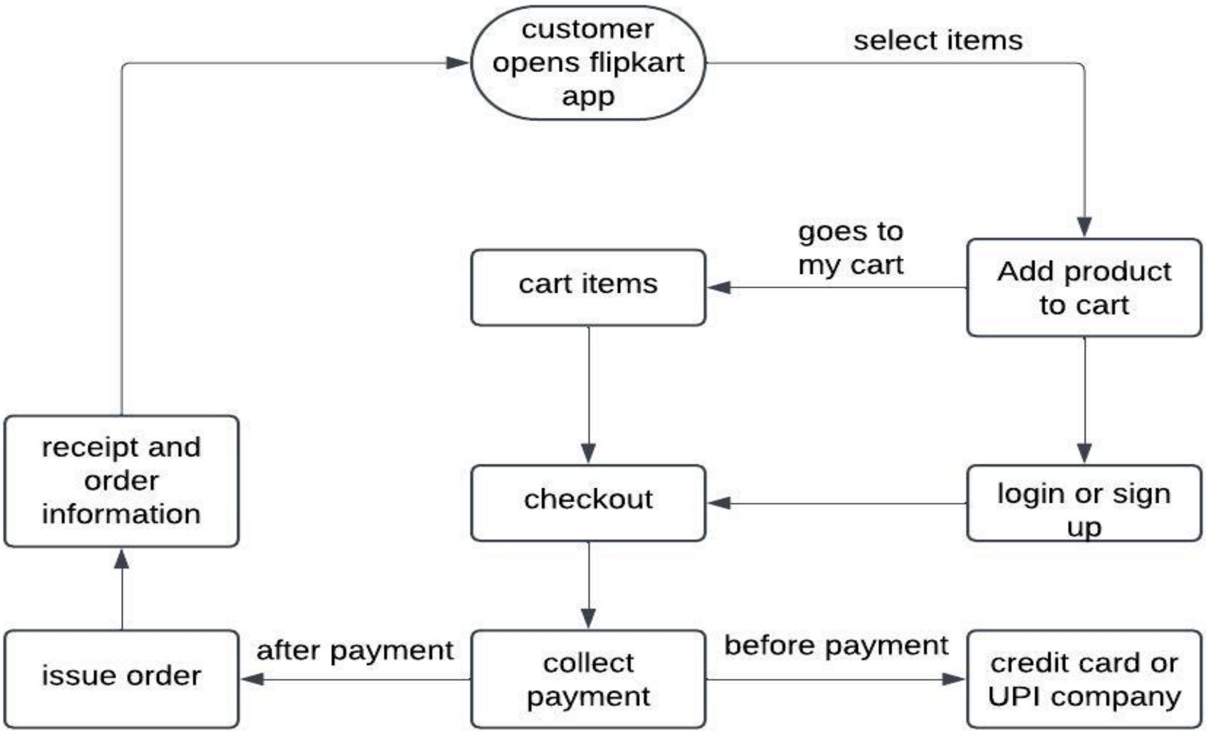
if the software product delivers on what it promises.

f) 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 (SRLC).

g) 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.

- What is DFD? Create a DFD diagram on Flipkart

DFD is the abbreviation for Data Flow Diagram. The flow of data in a system or process is represented by a Data Flow Diagram (DFD). It also gives insight into the inputs and outputs of each entity and the process itself. Data Flow Diagram (DFD) does not have a control flow and no loops or decision rules are present. Specific operations, depending on the type of data, can be explained by a flowchart. It is a graphical tool, useful for communicating with users, managers and other personnel. it is useful for analyzing existing as well as proposed systems. t should be pointed out that a DFD is not a flowchart. In drawing the DFD, the designer has to specify the major transforms in the path of the data flowing from the input to the output. DFDs can be hierarchically organized, which helps in progressively partitioning and analyzing large systems.


customer opens flipkart app
select items
Add product to cart
goes to my cart
cart items
login or sign up
checkout
collect payment
before payment
credit card or UPI company
after payment
issue order
receipt and order information

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

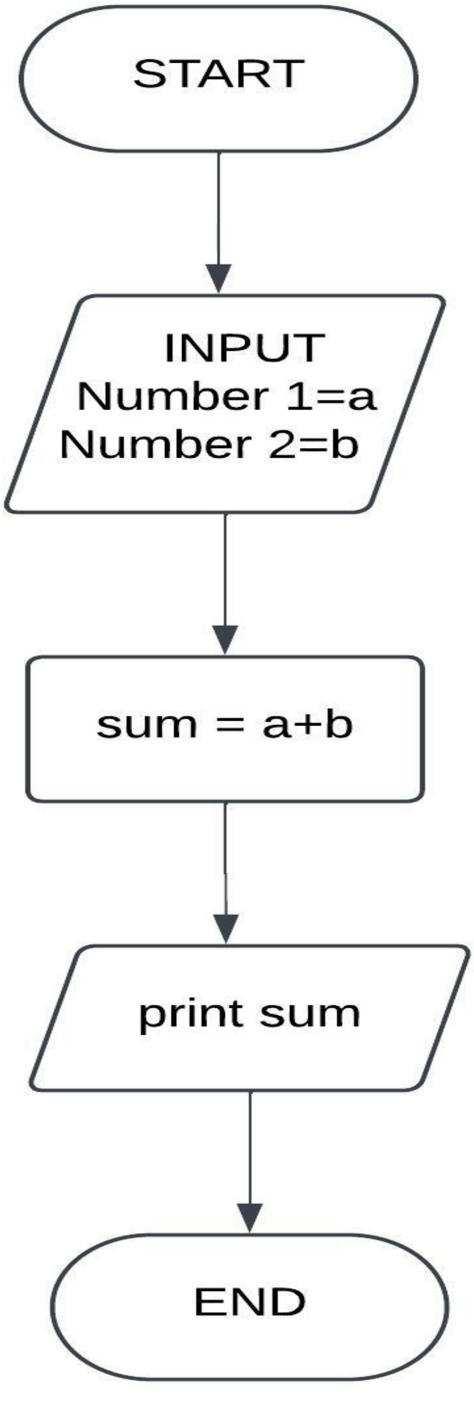
  A flowchart is a diagram that uses symbols to represent the steps in an algorithm. It's a graphical representation of a problem or process. They use geometrical shapes and arrows to show processes, relationships, and data/process flow. In other words, flowcharts depict decisions and results of them. In different fields, flowcharts are often used to analyze and manage processes.

  The symbols used in flow chart are:

  i. Oval: Represents the start and end of the flowchart.
  ii. Rectangle: Represents a processing step.
  iii. Diamond: Represents a decision point or branching point.

iv. Parallelogram: Represents input or output operations.

v. Arrow lines: Show the direction of flow between operations.

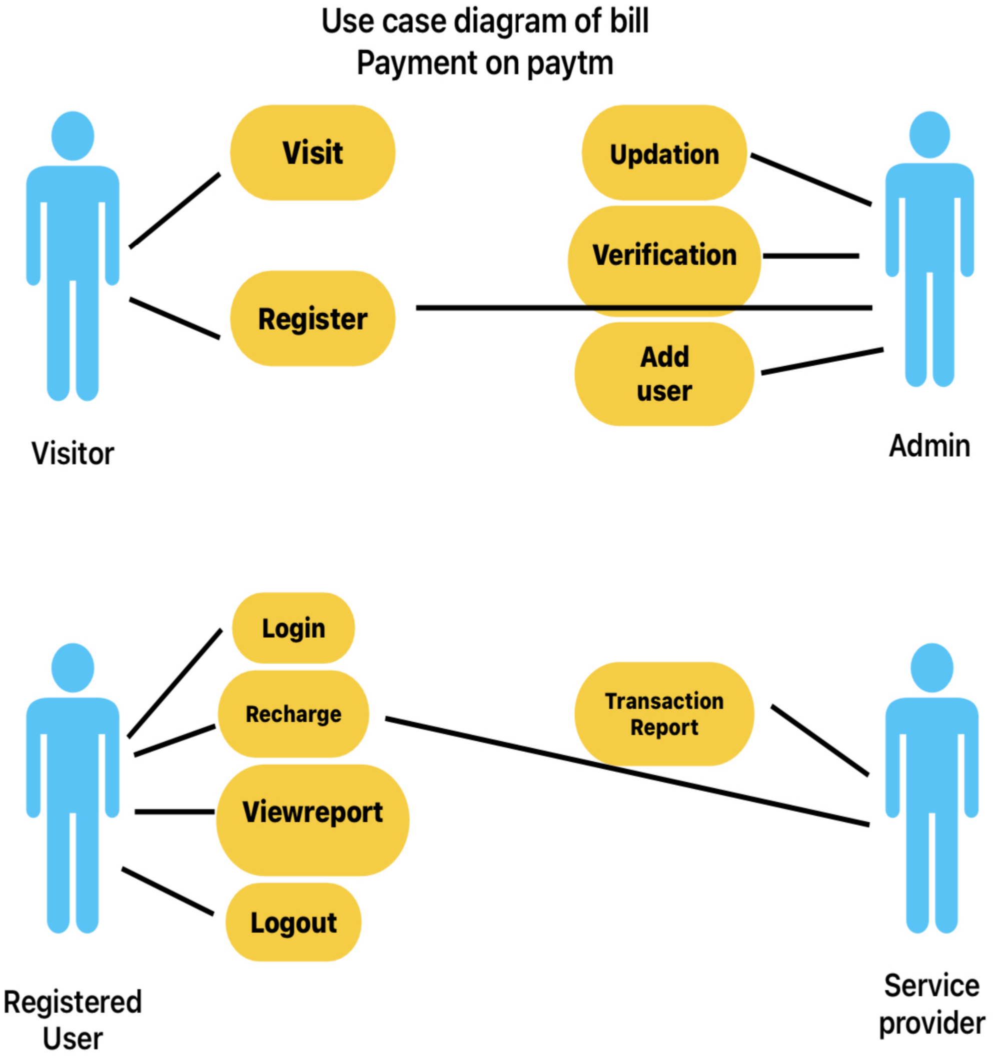


- What is Use case Diagram? Create a use-case on bill payment on paytm.

A UML use case diagram is the primary form of system/software requirements for a new software program underdeveloped. Use cases specify the expected behavior (what), and not the exact method of making it happen (how). Use cases once specified can be denoted both textual and visual representation (i.e. use case diagram). A key concept of use case modeling is that it helps us design a system from the end user's perspective. It is an effective technique for communicating system behavior in the user's terms by specifying all externally visible system behavior. A use case diagram is usually simple. It does not show the detail of the use cases.

i. It only summarizes some of the relationships between use cases, actors, and systems.

ii. It does not show the order in which steps are performed to achieve the goals of each use case.


Use case diagram of bill
Payment on paytm
Visit
Register
Visitor
Updation
Verification
Add user
Admin
Login
Recharge
Viewreport
Logout
Registered User
Transaction Report
Service provider