1. **What is SDLC?**

A software development life cycle is a structure imposed on the development of a software product that define the process. It is a structured approach or process used to design, develop, test, deploy and maintain software applications.

These are the number of SDLC phases:

- Analysis
- Design
- Implementation
- Testing
- Maintenance

2. **What is software testing?**

Software testing is executing a system in order to identifying gaps, errors or missing requirements in contrary to the actual desire or requirements.

- Software testing is the process of evaluating a software product or an application to ensure it functions as intended, meets requirements and is free from defects before it is released.
- The primary goal of software testing is to identify and fix bugs/errors to ensure the software delivers the expected functionality.
- It is the process of finding errors in the developed product.

3 **What is agile methodology**?

- Agile SDLC model is a combination of iterative and incremental process model with focus on process adaptability and customer satisfaction by rapid delivery of working software product.
- Agile Method breaks the product into small incremental builds.
- These builds are provided in iterations.
- Each iteration typically lasts from about one to three weeks.
- Every iteration involves cross functional teams working simultaneously on various area like planning, requirements analysis, designing, coding, unit testing, acceptance testing.
- At the end of the iteration a working product is displayed to the customer and important stakeholders.
- Agile model believes that every project needs to be handled differently and existing methods need to be tailored to best suit the project requirements. In agile the tasks are divided to time boxes (small time frames) to deliver specific features for a release.
- Iteration approach is taken and working software build is delivered after each iteration. Each build is incremental in terms of features, the final build holds all the features required by the customer.
- Agile though process had started early in the software development and started becoming popular with time due to its flexibility and adaptability

4 What is SRS?

- A software requirements specification is a complete description of the behavior of the system to be developed.
- SRS (Software Requirement Specification) is a complete specification and description of requirements of the software that need to be fulfilled for the successful development of the software system.
- It includes a set of use cases that describe all of the interactions that the users will have with the software.
- Types of requirements
- Functional Requirements
- Non-Functional Requirements

5. What is OOPS?

Object oriented programming has a web of interacting objects. OOP stands for object-oriented programming paradigm based on the concept of objects which are instances of classes.

6. **Write basic concepts of OOPS**.

- Object
- Class
- Encapsulation
- Inheritance

1. Polymorphism (Overriding) (Overloading)
2. Abstraction

7 What is an Object:

- An object represent on individuals identifiable item, unit or entity either real or abstract with a well-defined role in the problem domain.
- An object has the responsibility to know and the responsibility to do. It's a particular instance of a class.

8 What is an Class:

- It is a blue print for an object a class represent an abstraction of the object and abstract the properties and behavior of that object. It is a blueprint for an object.
- A class represents an abstraction of the object and abstracts the properties and behaviors of that object.
- In real case, in the case of car, these will be a blueprint or design created first and then the actual car will be built based on that.

9 What is Encapsulation

- Encapsulation is a practice of including in an object everything it needs hidden from other objects. The internal state is usually not accessible by other objects.
- Encapsulation is defined as the wrapping up of data under a single unit.
- It is the practice of including in an object everything it needs hidden from other objects.

10 What is Inheritance?

Inheritance means that one class inherits the characteristics of another class.

11 What is Polymorphism?

It means having many forms it allows different objects to respond to the same message in different ways. The response specific to the type of the objects.

The ability to change form is known as Polymorphism. Polymorphism is described as a situation in which something occurs in several different forms. Same function but having different functionality

There are two types of polymorphism:

1 Overloading
2 Overriding

12 What is Abstraction?

Abstraction is the representation of the essential features of an object, These are encapsulated into an abstract data type.

Data abstraction refers to providing only essential information to the outside wordand hiding their background detail.

13 Draw Usecase on online bill payment system (paytm)

Online bill payment usecase: click here

14 Draw Usecase on banking system for customers.

Banking system: Click here

15 Draw Usecase on Broadcasting System.

Broadcasting system: Click here

16 Write SDLC phases with basic introduction

The **Software Development Life Cycle (SDLC)** is a structured process used by software developers and project managers to design, develop, test, and maintain software systems. It ensures high-quality software is delivered efficiently.

There are various type of sdlc phase

1 Requirement gathering

2 Analysis

3 Design

4 Implementation

5 Testing

6 Maintenance

1 Requirement gathering- Understand what the client or user needs from the Software . Documenting functional and non-functional requirements in the SRS document

2 Analysis - This phase defines the problem that the customer is trying to solve. The deliverable result at the end of this phase is a requirement document. Ideally, this document states in a clear and precise fashion what is to be built. This analysis represents the **"what"** phase

3 Design- Plan how the software will be built based on the requirements.

Architectural design or system design. Interface design (UI/UX). Database and data flow design. In the output we get result in Design Specification Document (DSD) or High-Level Design (HLD) and Low-Level Design (LLD).

4 Implementation-

In the implementation phase, the team builds the components either from Scratch or by composition The implementation phase deals with issues of quality , performance, baselines, libraries, and debugging. In this phase Output is working software

5 Testing-

This Testing Phase ensures the software works correctly and meets Requirement. the separate team tests the system to find and fix problems It includes different types of testing, such as

Regression Testing
Internal Testing
Unit Testing
Application Testig

Stress Testing

6 **Maintenance**-

Software maintenance is one of the activities in software engineering, and is the process of optimizing deployed software (software release), as well as fixing defects. Maintenance is the process of changing a system after it has been deployed.

There are different part of maintenance.

1 corrective maintenance- identify and repairing defect
2 Adaptive maintenance- adapting the existing solution to the new platform
3 Perfective maintenance-implementing the new requirement

17 Explain Phases of the waterfall model

The **Waterfall Model** is a linear and sequential software development methodology, where each phase must be completed before the next one begins. It is one of the earliest models used in software engineering.

1 Requirement gathering- Understand what the client or user needs from the Software . Documenting functional and non-functional requirements in the SRS document
2 Analysis - This phase defines the problem that the customer is trying to solve. The deliverable result at the end of this phase is a requirement document. Ideally, this document states in a clear and precise fashion what is to be built. This analysis represents the “what” phase
3 Design- Plan how the software will be built based on the requirements. Architectural design (system structure). Interface design (UI/UX).Database and data flow design. In the output we get result in Design Specification Document (DSD) or High-Level Design (HLD) and Low-Level Design (LLD).
4 Implementation- In the implementation phase, the team builds the components either from Scratch or by composition .The implementation phase deals with issues of quality , performance, baselines, libraries, and debugging. In this phase Output is working software
5 Testing- This Testing Phase ensures the software works correctly and meets Requirement. the separate team tests the system to find and fix problems

It includes different types of testing, such as

Regression Testing
Internal Testing
Unit Testing
Application Testig
Stress Testing

6 Maintenance- Software maintenance is one of the activities in software engineering, and is the process of optimizing deployed software (software release), as well as fixing defects. Maintenance is the process of changing a system after it has been deployed.
There are different part of maintenance.

corrective maintenance- identify and repairing defect
Adaptive maintenance- adapting the existing solution to the new platform
Perfective maintenance-implementing the new requirement

18 Write phases of spiral model

The **Spiral Model** is a risk-driven software development process model that combines elements of both **iterative** and **waterfall** models. It emphasizes **risk analysis** and **repeated refinement** through multiple iterations (called **spirals** or **cycles**).

1. **Planning phase:**

- Gather detailed requirements from stakeholders
- Analyze requirements for feasibility, scope, and
- Define system and software requirements.
- Identify project constraints (budget, timeline, technology, etc.).
- Break down work into manageable tasks.

2 Risk analysis phase:

- Identify possible project risks (technical, schedule, cost, etc.)
- Perform risk analysis (impact and likelihood).
- Explore alternative approaches (e.g., use of prototypes).
- Develop and validate prototypes to reduce uncertainty.
- Prepare risk mitigation and resolution strategies.

3 Engineering phase:

- Design system architecture and module components.
- Develop code based on refined requirements and design.
- Unit and integration testing of the components.
- Apply verification and validation techniques.
- Create partially working versions of the system

4 Evaluation phase:

- Present the developed prototype or product increment to stakeholders.
- Evaluate output and get user feedback.
- Analyze progress against plans.

- Decide whether to continue, change direction, or terminate the project
- Plan for the next iteration (refine schedule, requirements, risks).

19 Write agile manifesto principles.

1 Individual and Interaction

- People are more important than tools and processes.

- Teamwork, communication, and motivation matter a lot.
- Things like pair programming and working in the same place help better teamwork.

2 Working Software

- A running, working version of the software is the best way to show progress.
- Instead of only writing long documents, showing real software helps the customer understand better.

3 Customer Collaboration

- Its hard to know all the requirements at the beginning.
- So, keep talking to the customer regularly to understand what they want as the project moves forward.

4 Responding to Change

- Agile, change is normal and welcome.
- If customer needs or market trends change, the team adjusts quickly and keeps improving the product.

20 Explain working methodology of agile model and also write pros and cons.

The **Agile Model** is an **iterative and incremental** approach to software development. It focuses on **collaboration**, **flexibility**, and **customer satisfaction** through continuous delivery of working software.

Pros

- Is a very realistic approach to software development Promotes teamwork and cross training.

- Functionality can be developed rapidly and demonstrated.
- Resource requirements are minimum.
- Suitable for fixed or changing requirements Delivers early partial working solutions.
- Good model for environments that change steadily

Cons

- Not suitable for handling complex dependencies.
- More risk of sustainability, maintainability and extensibility.
- An overall plan, an agile leader and agile PM practice is a must without which it will not work.
- Strict delivery management dictates the scope, functionality to be delivered, and adjustments to meet the deadlines.
- Depends heavily on customer interaction, so if customer is not clear, team can be driven in the wrong direction

21 Draw usecase on OTT Platform

OTT platefrom- click here

22 Draw usecase on E-commerce application

E commerce app-click here

23 Draw usecase on online book shopping

Shopping book - click here