

➤ What is SDLC & Explain the various Phases?

In software engineering, a **software development process** is a process of dividing software development work into smaller, parallel, or sequential steps or sub-processes to improve design, product management. It is also known as a **software development life cycle (SDLC)**. In This Life Cycle There Are Many Phases Which are as follows:

1. Planning & Information Gathering:

In this phase the company or the developer will discuss with client about the requirement of the application or software which they want to deploy or launch. And developer try to take as much as information possible about the software from client. And then Decide how much cost, time of development, resources, and final deliverable period.

2. Analysis:

In this Phase the company or the developer will analyze the requirement & information which the client has given about there software and then he Finalize the budget and the time of deployment of the software and then at end of this phase the developer or tester will ready **Software Requirement Specification (SRS)** Document. This Document Describe what is going to build in details & How is going to build.

3. Design:

In this phase the design document is created as per the requirement specification document which help to understand the structure of system and also help to provide proper design, demo, or beta layout of the system which provide the idea of output of the final system to client.

There are 2 Type of design document created in this phase which are as follows:

- **High-Level Design (HLD):**

- In this the developer provides detailed description and name of each module.
- Outline about the full functionality of every module.
- Detailed explanation of interface and process of development & deference between each module.
- Detailed data table with key feature of each module.
- Detailed drawing of the structure design of the software.

- **Low-Level Design (LLD):**

- Function of every module

- Basic data base table with type & size.
- Complete details of interface.
- Shows all dependency issue in each module.
- List of error messages.
- Complete input & output of every module.

4. Implementation /Coding:

Software construction is a software engineering discipline. It is the detailed creation of working meaningful software through a combination of coding, verification, unit testing, integration testing, and debugging. It is linked to all the other software engineering disciplines, most strongly to software design and software testing. And it is the Longest Phase of the **Software Development Life Cycle (SDLC)**. In this Phase the developer start to construct the software through coding and using per-determine coding language.

5. Testing Phase:

Software testing is the act of examining the artifacts and the behaviour of the software under test by validation and verification. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include, but not necessarily limited to

- analysing the product requirements for completeness and correctness in various contexts like industry perspective, business perspective, feasibility and viability of implementation, usability, performance, security, infrastructure considerations, etc.
- reviewing the product architecture and the overall design of the product
- working with product developers on improvement in coding techniques, design patterns, tests that can be written as part of code based on various techniques like boundary conditions, etc.
- executing a program or application with the intent of examining behaviour
- reviewing the deployment infrastructure and associated scripts & automation
- take part in production activities by using monitoring & observability techniques

Software testing can provide objective, independent information about the quality of software and risk of its failure to users or sponsors.

6. Deployment Phase:

Software deployment is all of the activities that make a software system available for use.

The general deployment process consists of several interrelated activities with possible transitions between them. These activities can occur on the producer side or on the consumer side or both. Because every software system is unique, the precise processes or procedures within each activity can hardly be defined. Therefore, "deployment" should be interpreted as a **general process** that has to be customized according to specific requirements or characteristics. Which include **User Acceptance Testing (UAT)** and also doing the final testing and any improvement if needed before the release of the final software to user.

7. Maintenance Phase:

A common perception of maintenance is that it merely involves fixing defects. However, one study indicated that over 80% of maintenance effort is used for non-corrective actions. This perception is perpetuated by users submitting problem reports that in reality are functionality enhancements to the system.

This is the Last Phase of the **Software Development Life Cycle (SDLC)**.

There are 3 Kind of Maintenance of software as follows:

- **Corrective Maintenance:** in this type of maintenance developer try to fix minor bugs and defect.
- **Adaptive maintenance:** in this type of maintenance developer upgrade to new version of the software
- **Perfective Maintenance:** in this type of maintenance developer try to enhance the software or adding some new changes to existing software.

➤ What is SRS?

Software Requirement Specification (SRS) Format as name suggests, is complete specification and description of requirements of software that needs to be fulfilled for successful development of software system. These requirements can be functional as well as non-functional depending upon type of requirement. The interaction between different customers and contractor is done because it's necessary to fully understand needs of customers.

Depending upon information gathered after interaction, SRS is developed which describes requirements of software that may include changes and modifications that is needed to be done to increase quality of product and to satisfy customer's demand.

Introduction:

(i) Purpose of this Document –

At first, main aim of why this document is necessary and what's purpose of document is explained and described.

(ii) Scope of this document –

In this, overall working and main objective of document and what value it will provide to customer is described and explained. It also includes a description of development cost and time required.

(iii) Overview –

In this, description of product is explained. It's simply summary or overall review of product.

➤ Explain the Phases Waterfall Model

The **waterfall model** is a breakdown of project activities into linear **sequential** phases, where each phase depends on the deliverables of the previous one and corresponds to a specialization of tasks. The approach is typical for certain areas of **engineering design**. In **software development**, it tends to be among the less iterative and flexible approaches, as progress flows in largely one direction ("downwards" like a **waterfall**) through the phases of conception, initiation, **analysis**, **design**, **construction**, **testing**, **deployment** and **maintenance**.

The waterfall development model originated in the **manufacturing** and **construction** industries, where the highly structured physical environments meant that design changes became prohibitively expensive much sooner in the development process. When first adopted for software development, there were no recognised alternatives for knowledge-based creative work.

In Royce's original waterfall model, the following phases are followed in order:

1. **System and software requirements:** captured in a product requirements document
2. **Analysis:** resulting in models, schema, and business rules
3. **Design:** resulting in the software architecture
4. **Coding:** the development, proving, and integration of software
5. **Testing:** the systematic discovery and debugging of defects
6. **Operations:** the installation, migration, support, and maintenance of complete systems

Thus, the waterfall model maintains that one should move to a phase only when its preceding phase is reviewed and verified.

Various modified waterfall models (including Royce's final model), however, can include slight or major variations on this process. These variations included returning

to the previous cycle after flaws were found downstream, or returning all the way to the design phase if downstream phases deemed insufficient.

➤ **What is agile methodology?**

The **Agile software development** methodology is one of the simplest and effective processes to turn a vision for a business need into software solutions. Agile is a term used to describe software development approaches that employ continual planning, learning, improvement, team collaboration, evolutionary development, and early delivery. It encourages flexible responses to change.

The agile software development emphasizes on four core values.

1. Individual and team interactions over processes and tools
2. Working software over comprehensive documentation
3. Customer collaboration over contract negotiation
4. Responding to change over following a plan.

➤ **Write phases of spiral model**

A spiral model has 4 phases described below:

1. Planning phase
2. Risk analysis phase
3. Engineering phase
4. Evaluation phase.

➤ **What is oops?**

Object-oriented programming (OOP) is a computer programming model that organizes software design around data, or objects, rather than functions and logic. An object can be defined as a data field that has unique attributes and behaviour.

OOP focuses on the objects that developers want to manipulate rather than the logic required to manipulate them. This approach to programming is well-suited for programs that are large, complex and actively updated or maintained. This includes programs for manufacturing and design, as well as mobile applications; for example, OOP can be used for manufacturing system simulation software.

The organization of an object-oriented program also makes the method beneficial to collaborative development, where projects are divided into groups. Additional benefits of OOP include code reusability, scalability and efficiency.

➤ Write Basic Concepts of oops

Concepts of OOPS in Selenium Automation Framework

1. ABSTRACTION
2. INTERFACE
3. INHERITANCE
4. POLYMORPHISM
 - METHOD OVERLOADING
 - METHOD OVERRIDING
5. ENCAPSULATION

➤ What is object?

An **object is** a philosophical term often used in contrast to the term subject. A subject **is** an observer and an **object is** a thing observed.

➤ What is class?

In object-oriented programming, a **class** is a blueprint for creating **objects** (a particular data structure), providing initial values for state (member variables or attributes), and implementations of behaviour (member functions or methods).

➤ What is encapsulation?

In object-oriented computer programming (OOP) languages, the notion of encapsulation (or OOP Encapsulation) refers to the bundling of data, along with the methods that operate on that data, into a single unit. Many programming languages use **encapsulation** frequently in the form of **classes**.

➤ What is inheritance?

Inheritance in OOP = When a class derives from another class. The child class will inherit all the public and protected properties and methods from the parent class.

In addition, it can have its own properties and methods. An inherited class is defined by using the extends keyword.

➤ What is polymorphism?

Polymorphism is one of the core concepts of object-oriented programming (OOP) and describes situations in which something occurs in several different forms. In computer science, it describes the concept that you can access objects of different types through the same interface.

➤ What is RDBMS?

RDBMS stands for Relational Database Management System. It is a program used to maintain a relational database. **RDBMS** is the basis for all modern database systems such as MySQL, Microsoft SQL Server, Oracle, and Microsoft Access. **RDBMS** uses **SQL queries** to access the data in the database.

➤ What is SQL?

SQL stands for Structured Query Language. SQL lets you access and manipulate databases.

➤ **Write SQL Commands?**

SQL Commands are as Follows:

- 1) DDL – Data Definition Language
- 2) DML – Data Manipulation Language
- 3) DCL – Data Control Language
- 4) DQL – Data Query Language

➤ **Write agile manifesto principles?**

The Agile Manifesto for Software Development was a declaration of a unifying philosophy for frameworks like Scrum, Extreme Programming, and Feature-Driven Development (FDD). The Agile Manifesto greatly departed from the waterfall-style project management approaches that were widely in use prior to that time.

And the Principle of the agile manifesto are as follows: -

- **Individuals and interactions** over processes and tools
- **Working software** over comprehensive documentation
- **Customer collaboration** over contract negotiation
- **Responding to change** over following a plan.

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

Agile Methodology is the process for managing a project that involves constant collaboration and working in iterations. Agile project management works off the basis that a project can be continuously improved upon throughout its life cycle, with changes being made quickly and responsively. Agile is one of the most popular approaches to project management due to its flexibility, adaptability to change, and high level of customer input.

➤ What is join?

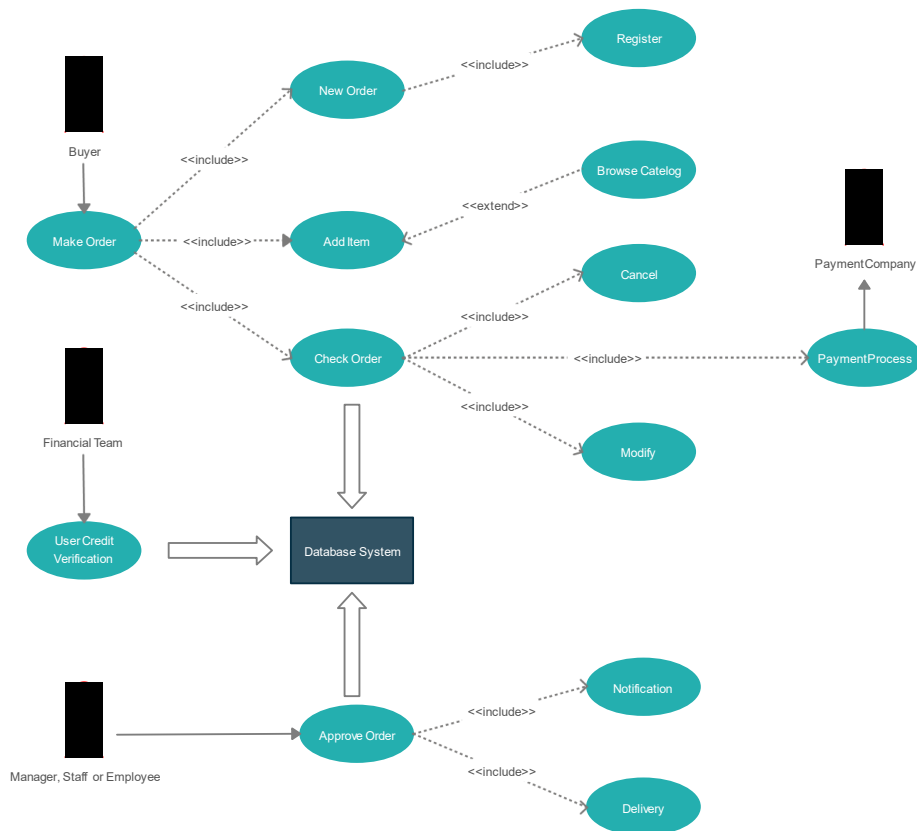
SQL joins are used to fetch or retrieve data from two or more data tables, based on a join condition. A join condition is a relationship among some columns in the data tables that take part in SQL join. Basically, data tables are related to each other with keys. We use these keys relationship in SQL joins. A primary key is a column or a combination of columns with a unique value for each row. Each primary key value must be unique within the table. The purpose is to bind data together, across tables, without repeating all of the data in every table.

➤ Write type of joins?

Here are the different types of the JOINS in SQL:

- ❖ **INNER JOIN:** Returns records that have matching values in both tables
- ❖ **LEFT OUTER JOIN:** Returns all records from the left table, and the matched records from the right table.
- ❖ **RIGHT OUTER JOIN:** Returns all records from the right table, and the matched records from the left table
- ❖ **FULL OUTER JOIN:** Returns all records when there is a match in either left or right table.

➤ Draw Use case on Online book shopping



➤ Draw Usecase on online bill payment system (paytm)

