What is SDLC

SDLC or the Software Development Life Cycle is a process that produces software with the highest quality and lowest cost in the shortest time possible. SDLC provides a well-structured flow of phases that help an organization to quickly produce high-quality software which is well-tested and ready for production use.

What is software testing?

Software testing is a method to check whether the actual software product matches expected requirements and to ensure that software product is[Defect](https://www.guru99.com/defect-management-process.html)free. It involves execution of software/system components using manual or automated tools to evaluate one or more properties of interest. The purpose of software testing is to identify errors, gaps or missing requirements in contrast to actual requirements.

What is agile methodology?

Agile is an iterative approach to project management and software development that helps teams deliver value to their customers faster and with fewer headaches. Instead of betting everything on a "big bang" launch, an agile team delivers work in small, but consumable, increments. Requirements, plans, and results are evaluated continuously so teams have a natural mechanism for responding to change quickly.

What is SRS

A software requirements specification (SRS) is a document that captures complete description about how the system is expected to perform. It is usually signed off at the end of requirements engineering phase.

What is oops

OOP stands for **Object-Oriented Programming**.Procedural programming is about writing procedures or methods that perform operations on the data, while object-oriented programming is about creating objects that contain both data and methods.

Write Basic Concepts of oops

* Object
* Class
* Abstraction
* Inheritance
* Polymorphism
* Encapsulation

What is object

Objects are key to understanding *object-oriented* technology. Look around right now and you'll find many examples of real-world objects: your dog, your desk, your television set, your bicycle.

Real-world objects share two characteristics: They all have *state* and *behavior*. Dogs have state (name, color, breed, hungry) and behavior (barking, fetching, wagging tail). Bicycles also have state (current gear, current pedal cadence, current speed) and behavior (changing gear, changing pedal cadence, applying brakes). Identifying the state and behavior for real-world objects is a great way to begin thinking in terms of object-oriented programming.

What is class

**A class** **is a template definition of the method s and variable s in a particular kind of object** .

What is encapsulation

Encapsulation is a way to restrict the direct access to some components of an object, so users cannot access state values for all of the variables of a particular object. Encapsulation can be used to hide both data members and data functions or methods associated with an instantiated class or object.

What is inheritance

Inheritance is the process by which genetic information is passed on from parent to child. This is why members of the same family tend to have similar characteristics.

What is polymorphism

Polymorphism is a feature of object-oriented programming languages that allows a specific routine to use variables of different types at different times. It is the ability of a programming language to present the same interface for several different underlying data types and different objects to respond in a unique way to the same message.

Write SDLC phases with basic introduction

* Requirement gathering and analysis
* Design
* Implementation or coding
* Testing
* Deployment
* Maintenance

**1) Requirement Gathering and Analysis**

During this phase, all the relevant information is collected from the customer to develop a product as per their expectation. Any ambiguities must be resolved in this phase only.

Business analyst and Project Manager set up a meeting with the customer to gather all the information like what the customer wants to build, who will be the end-user, what is the purpose of the product. Before building a product a core understanding or knowledge of the product is very important.

**For Example,** A customer wants to have an application which involves money transactions. In this case, the requirement has to be clear like what kind of transactions will be done, how it will be done, in which currency it will be done, etc.

Once the requirement gathering is done, an analysis is done to check the feasibility of the development of a product. In case of any ambiguity, a call is set up for further discussion.

Once the requirement is clearly understood, the SRS (Software Requirement Specification) document is created. This document should be thoroughly understood by the developers and also should be reviewed by the customer for future reference.

**2) Design**

In this phase, the requirement gathered in the SRS document is used as an input and software architecture that is used for implementing system development is derived.

**3) Implementation or Coding**

Implementation/Coding starts once the developer gets the Design document. The Software design is translated into source code. All the components of the software are implemented in this phase.

**4) Testing**

Testing starts once the coding is complete and the modules are released for testing. In this phase, the developed software is tested thoroughly and any defects found are assigned to developers to get them fixed.

Retesting, regression testing is done until the point at which the software is as per the customer’s expectation. Testers refer SRS document to make sure that the software is as per the customer’s standard.

**5) Deployment**

Once the product is tested, it is deployed in the production environment or first [UAT (User Acceptance testing)](https://www.softwaretestinghelp.com/what-is-user-acceptance-testing-uat/) is done depending on the customer expectation.

In the case of UAT, a replica of the production environment is created and the customer along with the developers does the testing. If the customer finds the application as expected, then sign off is provided by the customer to go live.

**6) Maintenance**

After the deployment of a product on the production environment, maintenance of the product i.e. if any issue comes up and needs to be fixed or any enhancement is to be done is taken care by the developers.

Explain Phases of the waterfall model

[Waterfall model](https://www.softwaretestinghelp.com/what-is-sdlc-waterfall-model/) is the very first model that is used in SDLC. It is also known as the linear sequential model.

In this model, the outcome of one phase is the input for the next phase. Development of the next phase starts only when the previous phase is complete.

* First, Requirement gathering and analysis is done. Once the requirement is freeze then only the System Design can start. Herein, the SRS document created is the output for the Requirement phase and it acts as an input for the System Design.
* In System Design Software architecture and Design, documents which act as an input for the next phase are created i.e. Implementation and coding.
* In the Implementation phase, coding is done and the software developed is the input for the next phase i.e. testing.
* In the testing phase, the developed code is tested thoroughly to detect the defects in the software. Defects are logged into the defect tracking tool and are retested once fixed. Bug logging, Retest, Regression testing goes on until the time the software is in go-live state.
* In the Deployment phase, the developed code is moved into production after the sign off is given by the customer.
* Any issues in the production environment are resolved by the developers which come under maintenance.

Write phases of spiral model

It has four stages or phases: The planning of objectives, risk analysis, engineering or development, and finally review. A project passes through all these stages repeatedly and the phases are known as a Spiral in the model.

1. **Determine objectives and find alternate solutions –** This phase includes requirement gathering and analysis. Based on the requirements, objectives are defined and different alternate solutions are proposed.
2. **Risk Analysis and resolving –**In this quadrant, all the proposed solutions are analyzed and any potential risk is identified, analyzed, and resolved.
3. **Develop and test:** This phase includes the actual implementation of the different features. All the implemented features are then verified with thorough testing.
4. **Review and planning of the next phase –**In this phase,the software is evaluated by the customer. It also includes risk identification and monitoring like cost overrun or schedule slippage and after that planning of the next phase is started

Write agile manifesto principles

#### 1. Individuals and interactions over processes and tools

This value of the Agile manifesto**focuses on giving importance to communication with the clients.** There are several things a client may want to ask and it is the responsibility of the team members to ensure that all questions and suggestions of the clients are promptly dealt with.

#### 2. Working product over comprehensive documentation

In the past, more focus used to be on proper documentation of every aspect of the project. There were several times when this was done at the expense of the final product. The Agile values dictate that **the first and foremost duty of the project team is completing the final deliverables**as identified by the customers**.**

#### 3. Customer collaboration over contract negotiation

Agile principles **require customers to be involved in all phases of the project**. The [Waterfall approach](https://kissflow.com/project/agile/traditional-vs-agile-project-management/) or Traditional methodologies only allow customers to negotiate before and after the project. This used to result in wastage of both time and resources. If the customers are kept in the loop during the development process, team members can ensure that the final product meets all the requirements of the client.

#### 4. Responding to change over following a plan

Contrary to the management methodologies of the past, Agile values are against using elaborate plans before the start of the project and continue sticking to them no matter what. Circumstances change and sometimes customers demand extra features in the final product that may change the [project scope](https://kissflow.com/project/project-scope-management/) **.**

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

[Agile Software Development Methodology](https://www.geeksforgeeks.org/software-engineering-agile-software-development/) is a process of software development (such as other software development methodologies – waterfall model, V-model, iterative model, etc.), however, the Agile development model is also a type of incremental model. Software develops in incremental, rapid cycles. In English, Agile means ‘the ability to move quickly and easily’ and respond to change rapidly – this is an important aspect of Agile software development.

**Advantages of Agile Methodology :**

1. In Agile methodology the delivery of software is unremitting.
2. The customers are satisfied because after every Sprint working feature of the software is delivered to them.
3. Customers can have a look of the working feature which fulfilled their expectations.
4. If the customers has any feedback or any change in the feature then it can be accommodated in the current release of the product.
5. In Agile methodology the daily interactions are required between the business people and the developers.
6. In this methodology attention is paid to the good design of the product.
7. Changes in the requirements are accepted even in the later stages of the development.
8. An Agile/Scrum approach can improve organizational synergy by breaking down organizational barriers and developing a spirit of trust and partnership around organizational goals.

**Disadvantages of the Agile Methodology :**

1. In Agile methodology the documentation is less.
2. Sometimes in Agile methodology the requirement is not very clear hence it’s difficult to predict the expected result.
3. In few of the projects at the starting of the software development life cycle it’s difficult to estimate the actual effort required.
4. Because of the ever-evolving features, there is always a risk of the ever-lasting project.
5. For complex projects, the resource requirement and effort are difficult to estimate.
6. In Agile methodology the documentation is less.
7. Sometimes in Agile methodology the requirement is not very clear hence it’s difficult to predict the expected result.
8. In few of the projects at the starting of the software development life cycle it’s difficult to estimate the actual effort required.
9. Because of the ever-evolving features, there is always a risk of the ever-lasting project.

Draw usecase on Online shopping product using COD.

ONLINE SHOPPING PRODUCT

**CUSTOMER**

Draw usecase on Online shopping product using payment gateway.

ONLINE SHOPPING GATWAY

CUSTOMER

Draw Usecase on online bill payment system (paytm)

ONLINE BILL PAYMENT PAYTM

Customer

Draw Usecase on Online book shopping

E-Book

Customer