**Module-1 ( Fundamental )**

1. **What is SDLC**

SDLC is a structure imposed on the development of a software product that defines the process for **planning, implementation, testing, documentation , deployment , and maintenance and support.**

**SDLC divided into 6 phases :-**

* **Requirements Collection / Gathering :-** Establish Customer Need.
* **Analysis :**- Model and specify the requirement of client.
* **Design :**- Model and Specify a Solution.
* **Implementation :-** Construct a solution in software.
* **Testing :**- Validate the solution against the requirements.
* **Maintenance :**- Repair defects and adapt the solution to the new requirements.

1. **What is software testing?**

**Testing is a process of evaluating a system or its component with the intent to find that whether it satisfies the specified requirements or not.** Software testing is a process of executing a program or application with the intent of finding the software bugs. When asked people often think that testing only consists of running tests, i.e. executing the software. Test execution is only part of testing , but not all of the testing activities. Test activities exist before and after test execution. It can also be started as the **Process of validating and verifying** that a software program or application or product:

* Meets the business and technical requirement that guided it’s design and development.
* Works as expected.
* Can be implemented with the same characteristic.

**Let’s break the definition of software testing into the following parts :**

* **Process :-** Testing is a process rather than a single activity.
* **All Life Cycle Activities :**- Testing is a process that’s take place throughout the SDLC.
  + - * The Process of designing tests early in the life cycle can help to prevent defects from being introduced in the code. Something it’s referred as “**Verifying the test basis via the test design”**
* **Static Testing :-** It can test and find defects without executing code. Static testing is done during verification process. This testing reviewing of the document and static analysis.
* **Dynamic :-** In dynamic testing the software code is executed to demonstrate the result of running tests. It’s done during validation process.
* **Planning :**- We need to plan as what we want to do. We control the test activities , we report on testing progress and the status of the software under test.
* **Preparation :**- We need to choose what testing we will do , by selecting test conditions and designing test cases.
* **Evaluation :**- During evaluation we must check the results and evaluate the software under test and the completion criteria, which helps us to decide whether we have finished testing and whether the software product has passed the tests.
* **Software products and related work products :-** Along with the testing of code the testing of requirement and design specifications and also the related documents like operation, user and training material is equally important.

1. **What is OOPS ?**

* OOPs Stand for **Object Oriented Programming Language.** And OOPs have main Concepts is **Class , Object , Encapsulation , Inheritance , Polymorphism , Abstraction.** 
  + **Class :-** In Class Include Data Type , Variable and Constructor.
  + **Object :-** Object means Bunch of Things which we have available part in data.
  + **Inheritance :-** Inheritanceit means Child class can use all the functionality of parent class. And Inheritance divided into 5 types.
    - **Single**
    - **Multiple**
    - **Multilevel**
    - **Hybrid**
    - **Hierarchical**
* **Encapsulation :-** Ecnapsulation it means it’s wraping the data it is called Encapsulation.
* **Polymorphism :**- Polymorphism it means One thing and multiple implementation and there are two type of polymorphism
* **Method Overloading**
* **Method Overriding**
  + - * + **Method Overloading :-** Overloading it means method name are same but different parameters
        + **Method Overloading :-** Overloading it means method name and parameters both are same
* **Abstraction :-** Abstraction it means it hide sensitive data it is called Abstraction.

1. **Write SDLC phases with basic introduction**

* SDLC is a structure imposed on the development of a software product for planning , Implementation, Documentation , Deployment , and ongoing maintenance and support
* A SDLC is essentially a series of step , or phases , that provide a model for the development and lifecycle management of an application of piece of software.
* **SDLC Phases :-** 
  + Requirement Collection / Gathering
  + Analysis
  + Design
  + Implementation
  + Testing
  + Maintenance

1. **What is agile methodology ?**

* Agile Methodology is a system where we store the daily work report and it follow the all it MNC company and it very useful for the team leader or project manager they know everyone work.

1. **Explain Phases of the waterfall model ?**

* Waterfall Model is a sequential model that divides software development into pre-defined phases. Each phase must be completed before the next phase can begin with no overlap between the phases. Each phase is designed for performing specific activity during the SDLC phase. It was introduced in 1970 by Winston Royce.
* **Different Phases of Waterfall Model in Software Engineering**

**1-Requirement Gathering stage**

During this phase, detailed requirements of the software system to be developed are gathered from client

**2-Design Stage**

Plan the programming language, for Example Java, PHP, .net or database like Oracle, MySQL, etc. Or other high-level technical details of the project

**3-Built Stage**

After design stage, it is built stage, that is nothing but coding the software

**4-Test Stage**

In this phase, you test the software to verify that it is built as per the specifications given by the client.

**5-Deployment stage**

Deploy the application in the respective environment

**6-Maintenance stage**

Once your system is ready to use, you may later require change the code as per customer request

**When to use SDLC Waterfall Model?**

Requirements are not changing frequently

Application is not complicated and big

Project is short

Requirement is clear

Environment is stable

Technology and tools used are not dynamic and is stable

Resources are available and trained

**Advantages Waterfall Model**

Before the next phase of development, each phase must be completed

Suited for smaller projects where requirements are well defined

They should perform quality assurance test (Verification and Validation) before completing each stage

Elaborate documentation is done at every phase of the software’s development cycle

Project is completely dependent on project team with minimum client intervention

Any changes in software is made during the process of the development

**DisAdvantages Waterfall Model**

Error can be fixed only during the phase

It is not desirable for complex project where requirement changes frequently

Testing period comes quite late in the developmental process

Documentation occupies a lot of time of developers and testers

Clients valuable feedback cannot be included with ongoing development phase

Small changes or errors that arise in the completed software may cause a lot of problems

1. **Write Phases of the Spiral model ?**

* Spiral Model is a risk-driven software development process model. It is a combination of waterfall model and iterative model. Spiral Model helps to adopt software development elements of multiple process models for the software project based on unique risk patterns ensuring efficient development process.
* Each phase of spiral model in software engineering begins with a design goal and ends with the client reviewing the progress. The spiral model in software engineering was first mentioned by Barry Boehm in his 1986 paper.
* The development process in Spiral model in SDLC, starts with a small set of requirement and goes through each development phase for those set of requirements. The software engineering team adds functionality for the additional requirement in every-increasing spirals until the application is ready for the production phase.

**Spiral Model Phases**

* **Planning - It includes estimating the cost, schedule and resources for the iteration. It also involves understanding the system requirements for continuous communication between the system analyst and the customer**
* **Risk Analysis - dentification of potential risk is done while risk mitigation strategy is planned and finalized**
* **Engineering - It includes testing, coding and deploying software at the customer site**
* **Evaluation - Evaluation of software by the customer. Also, includes identifying and monitoring risks such as schedule slippage and cost overrun**

**When to use Spiral Model?**

**A Spiral model in software engineering is used when project is large**

**When releases are required to be frequent, spiral methodology is used**

**When creation of a prototype is applicable**

**When risk and costs evaluation is important**

**Spiral methodology is useful for medium to high-risk projects**

**When requirements are unclear and complex, Spiral model in SDLC is useful**

**When changes may require at any time**

**When long term project commitment is not feasible due to changes in economic priorities**

**Spiral Model Advantages**

**Additional functionality or changes can be done at a later stage**

**Cost estimation becomes easy as the prototype building is done in small fragments**

**Continuous or repeated development helps in risk management**

**Development is fast and features are added in a systematic way in Spiral development**

**There is always a space for customer feedback**

**Spiral Model DisAdvantages**

**Risk of not meeting the schedule or budget**

**Spiral development works best for large projects only also demands risk assessment expertise**

**For its smooth operation spiral model protocol needs to be followed strictly**

**Documentation is more as it has intermediate phases**

**Spiral software development is not advisable for smaller project, it might cost them a lot**