**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 ongoing maintenance and support. There are a number of different development models.
* A Software Development Life Cycle is essentially a series of steps, or phases that provide a model for the development and lifecycle management of an application or piece of software.

2) What is software testing?

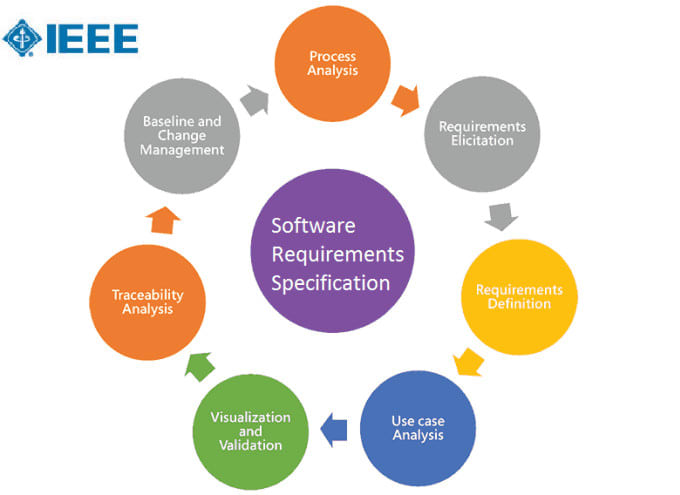
* Software Testing is a process used to identify the correctness, completeness, and quality of developed computer software.
* In simple words testing is executing a system in order to identify any gaps, errors or missing requirements in contrary to the actual desire or requirements.

3) What is agile methodology?

* Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product.
* Agile Methods break 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 areas like planning, requirements analysis,design, coding, unit testing, and acceptance testing.
* At the end of the iteration a working product is displayed to the customer and important stakeholders.

4) What is SRS ?

* A software requirements specification (SRS) is **a detailed description of a software system to be developed with its functional and non-functional requirements**.
* The SRS is developed based the agreement between customer and contractors. It may include the use cases of how user is going to interact with software system.
* Why do we need SRS?
* Creates agreement between all parties. An SRS contains all requirements that the client and the development company have discussed and agreed upon. If either party has doubts about the agreement, they can simply check the SRS. Clearly defines the project scope.



5) 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

behavior.

* Object-Oriented Programming System (OOPs) is **a programming concept that works on the principles of abstraction, encapsulation, inheritance, and polymorphism**.
* Why OOPs is important?
* OOP language **allows to break the program into the bit-sized problems that can be solved easily** (one object at a time).
* The new technology promises greater programmer productivity, better quality of software and lesser maintenance cost.

6) Write Basic Concepts of oops

* Object oriented programming is a type of programming which uses objects and classes its functioning.
* The object oriented programming is based on real world entities like inheritance, polymorphism, data hiding, etc. It aims at binding together data and function work on these data sets into a single entity to restrict their usage.

Some basic concepts of object oriented programming are −

* CLASS
* OBJECTS
* ENCAPSULATION
* POLYMORPHISM
* INHERITANCE
* ABSTRACTION

7) What is object ?

* Object can be anything in this world. Like car,subject,person…etc
* Object is a instance/example of a class.
* Ex. Apple is an object of fruit class.
* Tangible Things as a car, printer, ...
* Roles as employee, boss, ...
* Incidents as flight, overflow, ...
* Interactions as contract, sale, ...
* Specifications as colour, shape…
* An object represents an individual, identifiable item, unit, or entity,either real or abstract, with a well-defined role in the problem domain.

8) What is class ?

* Class is a blueprint for an object.
* Ex. Fruit is a class for apple,mango,cherry..etc
* A class represents an **abstraction of the object and abstracts the**

**properties and behavior of that object**.

9) What is Encapsulation?

* Wrapping up of data into a single unit.
* Provide a security of data from out side the class.
* Encapsulation enables **data hiding,** hiding irrelevant information from the users of a class and exposing only the relevant details required by the user.
* Consider the below real time example: Encapsulation: **As a driver you know how to start the car by pressing the start button and internal details of the starting operations are hidden from you**.
* So the entire starting process is hidden from you otherwise we can tell starting operation is encapsulated from you.

10) What is inheritance?

* Inheritance means that one class inherits the characteristics of another class. This is also called a “is a” relationship
* This is the process of adapting the properties and behavior of parent class to child class.
* This is a very important concept of object-oriented programming since this feature helps to reduce the code size.

11) What is Polymorphism ?

* Polymorphism means “having many forms”.
* Poly refers to many. That is a single function or an operator functioning in many ways different upon the usage is called polymorphism.
* It allows different objects to respond to the same message in different

ways, the response specific to the type of the object.

* There is two types of polymorphism-

⚫ Compile time polymorphism(Overloading)

⚫ Runtime polymorphism(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 word and hiding their background details, i.e., to represent the needed information in program without presenting the details.
* **For example**, a database system hides certain details of how data is stored and created and maintained

13) Write SDLC phases with basic introduction

* SDLC is a process that defines the various stages involved in the development of software for delivering a high-quality product. SDLC stages cover the complete life cycle of a software i.e. from inception to retirement of the product.
* SDLC has defined its phases as, Requirement gathering, Designing, Coding, Testing, and Maintenance.



* SDLC Phases-

**1. Requirement:** In this phase, all the requirements are collected from the customer/client. They are provided in a document called Businessmen requirement specification (BRS) and System requirement specification (SRS). All the details are discussed with the customer/client in detail.

**2. Design:** It has two steps:

* **High-level design (HLD):** It gives the architecture of software products.
* **Low-level design (LLD):** It describes how each and every feature in the product should work and every component.
* **3. Implementation:** This is the longest phase.
* This phase consists of Front end + Middleware + Back-end.
* **In front-end:**Development of coding is done even SEO settings are done.
* **In Middleware:** They connect both the front end and back end.
* **In the back-end:** A database is created.

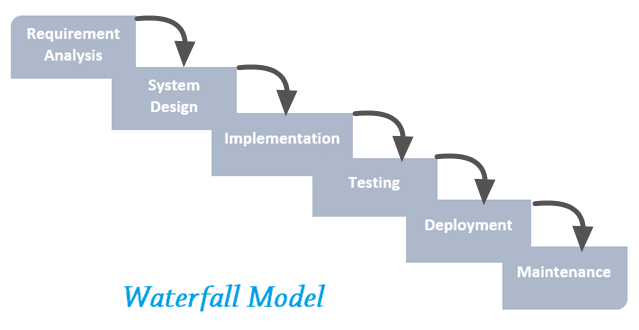
**4. Testing:** Testing is carried out to verify the entire system. The aim of the tester is to find out the gaps and defects within the system and also to check whether the system is running according to the requirement of the customer/client.

**5. Deployment:** After successful testing, the product is delivered/deployed to the client, and even clients are trained on how to use the product.

**6. Maintenance:** Once the product has been delivered to the client a task of maintenance starts as when the client will come up with an error the issue should be fixed from time to time.

14) Explain Phases of the waterfall model –

* The Waterfall Model, also known as the Linear-Sequential Life-cycle Model, is one of the first process models introduced for software development
* As the name implies, this model’s process of downward mechanism is similar to that of a waterfall. The whole process is divided into sequential stages, and it is imperative to complete each phase successfully in order to move onto the next one.
* the Waterfall Model consists of 6 phases:



* Requirements Analysis
* In this phase, all requirements of the project are analysed and documented in a specification document and a feasibility analysis is done to check if these requirements are valid.

## System Design

## In this phase, the system design is prepared which specifies hardware and system requirements, such as data layers, programming languages, network infrastructure, user interface etc. It helps define the overall system architecture

## Implementation

## As the name implies, in this phase the source code is written as per requirements. The physical design specifications are turned into a working code

## Testing

## The code is then handed over to the testing team. Testers check the program for all possible defects, by running test cases either manually or by automation. The client is involved in the testing phase as well, in order to ensure all requirements are met. All Flaws and bugs detected during this phase are fixed to ensure Quality Assurance.

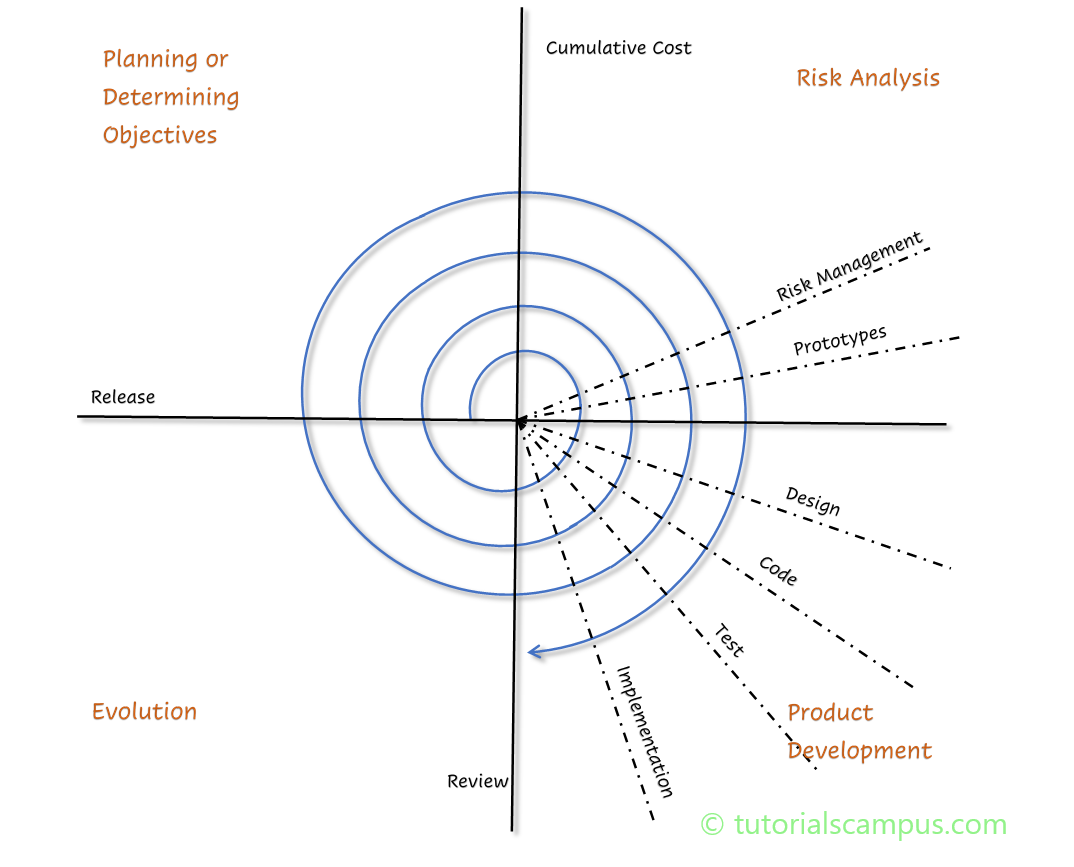
## Deployment

## In this phase, the software is deployed into a live environment (client’s server) in order to test its performance. Once the [software is deployed](https://technet.microsoft.com/en-us/library/bb496997.aspx), it becomes available to end-users. Sometimes, this phase also includes training of real-time users to communicate benefits of the system.

## Maintenance

## After the deployment phase, the next step is to provide support and maintenance for the software, making sure it runs smoothly. If the client and users come across errors/defects/bugs during use, fixing them is the main purpose of this stage.

**15)** Write phases of spiral model



* The spiral model is a systems development lifecycle ([SDLC](https://www.techtarget.com/searchsoftwarequality/definition/systems-development-life-cycle)) method used for [risk management](https://searchcompliance.techtarget.com/definition/risk-management) that combines the [iterative development](https://www.techtarget.com/searchsoftwarequality/definition/iterative-development) process model with elements of the [Waterfall model](https://www.techtarget.com/searchsoftwarequality/definition/waterfall-model). The spiral model is used by software engineers and is favored for large, expensive and complicated projects.

| **Spiral Model Phases** | **Activities performed during phase** |
| --- | --- |
| **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** | * Identification 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 |

16) Write agile manifesto principles

* The Agile Manifesto for Software Development was a declaration of a unifying philosophy for frameworks like [Scrum](https://www.workfront.com/project-management/methodologies/scrum), Extreme Programming, and Feature-Driven Development (FDD). The Agile Manifesto greatly departed from the [waterfall-style project management approaches](https://www.workfront.com/project-management/methodologies/waterfall) that were widely in use prior to that time.

## The Four Values of The Agile Manifesto-

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

 Key principles –

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity--the art of maximizing the amount of work not done--is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

17) Explain working methodology of agile model and also write pros and

Cons.

* The Agile methodology is **a way to manage a project by breaking it up into several phases**. It involves constant collaboration with stakeholders and continuous improvement at every stage. Once the work begins, teams cycle through a process of planning, executing, and evaluating.

* Pros –
* Little or no planning required
* Easy to manage
* Gives flexibility to developers
* 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.
* Minimal rules, documentation easily employed.
* 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.

* There is very high individual dependency, since there is minimum

documentation generated.

* Transfer of technology to new team members may be quite challenging due to lack of documentation.

**18)** Draw Use case on Online book shopping



111  
20) Draw Usecase on online bill payment system (paytm)

