

## **MODULE 1 - ASSIGNMENT**

### **Que: What is SDLC?**

Software Development Life Cycle (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.

### **Que: What is software testing?**

According to ANSI/IEEE 1059 standard, Testing can be defined as a process of analyzing a software item to detect the differences between existing and required conditions (that is defects/errors/bugs) and to evaluate the features of the software item.

### **Que: What is agile methodology?**

Agile methodology is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product. This model breaks the product into small incremental builds which are provided in iterations.

### **Que: What is SRS?**

A Software Requirements Specification (SRS) is a complete description of the behaviour of the system to be developed. It includes a set of use cases that describe all the interactions that the users will have with the software.

### **Que: What is OOPS?**

Object-oriented programming (OOP) is a programming model that organizes software design around data, or objects, rather than functions and logic.

### **Que: Write Basic Concepts of OOPS?**

There are six basic concepts of OOPS which includes object, class, encapsulation, inheritance, polymorphism and abstraction.

### **Que: What is object?**

An "Object" is anything to which a concept applies. Both data and function that operate on data are bundled as a unit called as Object.

### **Que: What is class?**

Class can be define as the blueprint or definition or a template for an object and describes the properties and behaviour of that object, but without any actual existence. This doesn't actually define any data, but it does define what the class name means, i.e., what an object of the class will consist of and what operations can be performed on such an object.

### **Que: What is Encapsulation?**

Encapsulation is the practice of including in an object everything it needs hidden from other objects. The internal state is usually not accessible by other objects.

### **Que: What is Inheritance?**

Inheritance means that one class inherits the characteristics of another class. As the name suggests, Inheritance is the process of forming a new class from an existing class that is from the existing class called as base class, new class is formed called as derived class.

### **Que: What is Polymorphism?**

Polymorphism 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 object. Giving different meaning or functions to the operators or functions is called polymorphism.

**Que: Draw Usecase on Online book shopping?**

<https://drive.google.com/file/d/1RLZcwaQ3uA-r0BL6558SdsGe17rUaRAf/view?usp=sharing>

**Que: Draw Usecase on online bill payment system (paytm)**

<https://drive.google.com/file/d/1V44UHhMqe2fSdzCkAfWs11rvbN1O2nzR/view?usp=sharing>

**Que:- Write SDLC phases with basic introduction?**

The Software Development Life Cycle includes following six phases.

- 1) Requirements gathering- In this first phase, the requirements of the customer is documented in written form which are may be incomplete, unambiguous or incorrect. User and business needs may change during the project. It usually consist of natural language and supplemented by diagrams and tables. If the functional & non-functional requirements are not met, the system is useless.
- 2) Analysis- This phase defines the requirements of the system, independent of how these requirements will be accomplished. This phase defines the problem that the customer is trying to solve.
- 3) Design- In this phase, the architecture document is being designed to implement the plans and the design team can expand upon the information established in the requirement document.
- 4) Implementation- In this phase, the team builds the components either from scratch or by composition. From the architecture and requirement document, the team should build exactly what has been requested, though there is still room for innovation and flexibility.
- 5) Testing- This is a separate phase performed by a different team after the implementation is completed. Regardless if testing is done after the-fact or continuously, testing is usually based on a regression technique split into several major focuses, namely internal, unit, application, and stress.

6) Maintenance- Software maintenance is one of the activities in software engineering, and is the process of enhancing and optimizing deployed software (software release), as well as fixing defects. The maintenance phase is the phase which comes after deployment of the software into the field.

**Que: Explain Phases of the waterfall model.**

- Requirement collection- It provides input material for the developed product, and the expected impact is studied, finalized, and documented. we need to know and understand what we have to design, what we have to develop, its processes, what its functionality will be, etc.
- 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.
- Design- It helps in specifying software and hardware requirements for product design. It also helps in the overall architecture of the system design. So the requirement specification is mainly studied and verified in this phase.
- Implementation- During the development phase, small programs called units are created using system design inputs, then tested and implemented in the next phase. Sometimes, functionality of each unit is tested before integration, which is called Unit Testing.
- Testing- 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.
- Maintenance- It includes making the appropriate modification to the product or system or enhancing, changing, or modifying attributes related to performance issues related to the system. Its primary role is to improve the system's performance with the maximum accuracy result of the software output.

**Que: Write phases of spiral model.**

- 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- In this phase, the risks associated with the project are identified and evaluated.
- Customer evaluation- In the evaluation phase, the software is evaluated to determine if it meets the customer's requirements and if it is of high quality.
- Engineering- In the engineering phase, the software is developed based on the requirements gathered in the previous iteration. It includes testing, coding and deploying software at the customer site.

**Que: Write agile manifesto principles.**

The following 12 Principles are based on the Agile Manifesto.

- Customer satisfaction through early and continuous software delivery – Customers are happier when they receive working software at regular intervals, rather than waiting extended periods of time between releases.
- Accommodate changing requirements throughout the development process – The ability to avoid delays when a requirement or feature request changes.
- Frequent delivery of working software – Scrum accommodates this principle since the team operates in software sprints or iterations that ensure regular delivery of working software.
- Collaboration between the business stakeholders and developers throughout the project – Better decisions are made when the business and technical team are aligned.
- Support, trust, and motivate the people involved – Motivated teams are more likely to deliver their best work than unhappy teams.

- Enable face-to-face interactions – Communication is more successful when development teams are co-located
- Working software is the primary measure of progress – Delivering functional software to the customer is the ultimate factor that measures progress.
- Agile processes to support a consistent development pace – Teams establish a repeatable and maintainable speed at which they can deliver working software, and they repeat it with each release.
- Attention to technical detail and design enhances agility – The right skills and good design ensures the team can maintain the pace, constantly improve the product, and sustain change.
- Simplicity – Develop just enough to get the job done for right now.
- Self-organizing teams encourage great architectures, requirements, and designs – Skilled and motivated team members who have decision-making power, take ownership, communicate regularly with other team members, and share ideas that deliver quality products.
- Regular reflections on how to become more effective – Self-improvement, process improvement, advancing skills, and techniques help team members work more efficiently.

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

Agile methodology is a combination of iterative and incremental process models with the focus on process adaptability and customer satisfaction by rapid delivery of working software product. This method break the product into small incremental builds and these builds are provided in iterations. 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.

- **Pros:**

- In this model, resource requirements are minimum and also suitable for fixed or changing requirements.
- There is no planning is required which gives flexibility to developers.
- The functionality of the model can be developed rapidly and demonstrated.
- This model promotes teamwork and cross training.

- **Cons:**

- This method is heavily depends on customer interaction, so if customer is not clear, team can be driven in the wrong direction.
- Transfer of technology to new team members may be quite challenging due to lack of documentation.
- It is not suitable for handling complex dependencies.
- There is very high individual dependency, since there is minimum documentation generated.

**Que: Draw usecase on Online shopping product using COD.**

[https://drive.google.com/file/d/1lZnMZi\\_TOBaf6rMxCouZuGcs6s9tEpce/view?usp=drive\\_link](https://drive.google.com/file/d/1lZnMZi_TOBaf6rMxCouZuGcs6s9tEpce/view?usp=drive_link)

**Que: Draw usecase on Online shopping product using payment gateway.**

<https://drive.google.com/file/d/1PqTfX-quX6mENpwYH8wuXsl6mWPbrLcm/view?usp=sharing>