**Software Testing Assignments**

**Module - 1**

* What is SDLC?

Ans. - SDLC (Software development life cycle) 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.

- A Software Development Life Cycle is essentially a series of steps, or phases, that provide a model for the development and life cycle management of an application or piece of software.

* What is agile methodology?

Ans.- 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.

- Agile model believes that every project needs to be handled differently and the existing methods need to be tailored to best suit the project requirements.

* What is SRS?

Ans. - 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 (functional requirements) and in addition to use cases, the SRS also contains non-functional (Or supplementary) requirements that describe all of the interactions that the users will have with the software.

- Use cases are also known as functional requirements. In addition to use cases, the SRS also contains non-functional (or supplementary) requirements.

* What is oops?

Ans. - An object-based programming language is one which easily supports object-orientation.

* Write basic concepts of oops?

Ans. - Object, Class, Encapsulation, Inheritance, Polymorphism (Overriding & Overloading), Abstraction are the basic concepts of oops.

* What is object?

Ans. - Both data and function that operate on data are bundled as a unit called as object.

- Object is an instance/example of a class.

* What is Class?

Ans. - Class is blueprint for an object.

- Class represents an abstraction of the object and abstracts the properties and behaviour of that object.

- An object is a particular instance of a class.

* What is encapsulation?

Ans. - Encapsulation is the practice of including in an object everything it needs hidden from other objects.

- Encapsulation is the practice of wrapping up of data.

* What is inheritance?

Ans. - Inheritance means that one class inherits the characteristics of another class. This also called a "is a" relationship.

* What is Polymorphism?

Ans. - Polymorphism means "having many forms"

- it allows different objects to respond to the same message in different ways.

- The ability too change form is known as polymorphism.

* Draw usecase on online book shopping? **online book shopping**

Ans. -

registration

Admin

Seller

Trace product

Payment

Receive feed back

profile manage

sell book

search

Log in

Give feed back

Online

COD

buy product

Add to cart

Customer

* Draw Usecase on online bill payment system (paytm)

**online bill payment system (Paytm)**

Receive payment

Search

Mo. Recharge

Add payment to wallet

fund transfer

login

ticket booking

bill payment

Sign up

Bank

Admin

User

* Write SDLC phases with basic introduction

Ans.- 1. Requirement collection/Gathering - Establish Customer Needs

2. Analysis - Model and specify the requirements

3. Design - Model and specify a solution

4. Implementation - Construct a solution in software

5. Testing - Validate the solution against the requirements

6. Maintenance - Repair defects and adapt the solution to the new requirements.

* Explain phases of the waterfall model?

Ans. - In waterfall model requirements must be "frozen" to early in the life cycle

- Requirement are validated too late.

Phases of waterfall model are:

1. Requirements collection

2. Analysis

3. Design

4. Implementation

5. Testing

6. Maintenance

* Write phases of spiral model

Ans. - The important phases of spiral model are:

1. Planning - determination of objectives, alternatives and constraints.

2. Risk Analysis - Analysis of alternatives and identification resolution of risks.

3. Engineering - Development of the "next level" product

4. Customer Evaluation - Assessment of the results of engineering.

* Write agile manifesto principles

Ans. - Individuals and Interactions - In agile development, self-organization and motivation are important, as are interactions like co-location and pair programming.

- Working software - Demo Working software is considered the best means of communication with the customer to understand their requirement instead of just depending on documentation

- Customer Collaboration- As the requirements cannot be gathered completely in the beginning of the project due to various factor, continuous customer interaction is very important to get proper product requirements.

Responding to change- agile development is focused on quick responses to change and continuous development.

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

Ans. - **working methodology of agile model:**

- 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 improtant stakeholders.

**pros:**

- 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 stedily.

- Minimal rules, documentation easily employed.

- Enables concurrent development and delivery within an overall planned context.

- Little or no planning required.

- Easy to manage.

-Give flexibility to developers.

**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

Communication.

* Draw usecase on Online shopping product using COD.

**Online shopping product using COD.**

registration

login

search product

Delivery boy

Seller

Admin

Costumer

receive cash

pay COD

trace product

buy product

receive feedback

give feedback

prof. management

Add to cart

* Draw usecase on Online shopping product using payment gateway.

**Online shopping product using payment gateway**

seller

Customer

Admin

Give feedback

trace product

payment (online)

buy product

prof. manage

add to cart

search product

login

registration

Receive feedback

Sell product