Software Testing Assignment

Module–1(Fundamental)

1. What is SDLC ?

-> SDLC Stands for Software Development Life Cycle (make Software of User Requirements)

2. What is software testing ?

-> Software Testing is useing for finding Errors in the developed product (Match ecpected results, Missing requirements etc)

3. What is agile methodology?

-> It is use for Project management to continuous collaboration and improvement for best result.

4. What is SRS ?

-> SRS Stands for Software Requirements Specification. There are two requirements - Functional and Non-Functional.

5. What is oops ?

-> OOPS Stands for Object Oriented Principles, Objects communicate to other objects by sending messages.

Internal details are hidden.

6. Write Basic Concepts of oops

-> There are 7 concepts -1) Abstraction

2) Encapsulation

3) Polymorphism

4) Inheritance

5) Association

6) Aggregation

7) composition

7. What is Object ?

-> 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 ?

-> A class represents an abstraction of the object and abstracts the properties and

behavior of that object.

An object is a particular instance of a class which has actual existence and there can be

many objects (or instances) for a class.

9. 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.

10. What is inheritance ?

-> Inheritance means that one class inherits the characteristics of another class.

11. What is polymorphism ?

-> It allows different objects to respond to the same message in different ways, the

response specific to the type of the object.

Polymorphism means “having many forms”.

12. Write SDLC phases with basic introduction

-> 1. Requirements Gathering

2. Analysis

3. Design

4. Implementation

5. Testing

6. Maintenance

> SDLC is a structure imposed on the development of a software product that defines the

process.

13. Explain Phases of the waterfall model

-> The Classical Software Life Cycle Models the software development

as a step-by-step "Waterfall" between the various development phase.

14. Write phases of spiral model

-> The spiral model is systems development lifecycle (SDLC) method used for risk management

that combines the iterative development process model with elements of the Waterfall model.

The spiral model is used by software engineers and is favored for large, expensive and complicated projects.

15. Write agile manifesto principles

-> 1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

2. 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.

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

-> 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.

Little or no planning required

Easy to manage

Gives 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.

Depends heavily on customer interaction, so if customer is not clear, team can be driven inthe wrong direction.

There is very high individual dependency, since there is minimum documentation generated.