**Software Testing Assignment**

**Module–1(Fundamental)**

* **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.
* **What is software testing?**
* The process of finding defects in the software is called Software Testing.
* Software Testing is a process used to identify the correctness, completeness, and quality of developed computer software.
* **What is agile methodology?**
* Agile SDLC model is a combination of iterative and incremental process models
* Agile Methods break the product into small incremental builds.
* **What is SRS**
* A software requirements specification (SRS) is a complete description of the behavior of the system to be developed.
* **What is oops**
* OOP stands for Object Oriented Programming.
* Object-oriented programming has a web of interacting objects, each house-keeping its own state.
* An object is like a black box.
* Messages are received by the methods of an object
* Objects of a program interact by sending messages to each other.
* The internal details are hidden.
* **Write Basic Concepts of oops**
* Object
* Class
* Encapsulation
* Inheritance
* Polymorphism
* Overriding
* Overloading
* Abstraction
* **What is object**
* Object is a basic runtime entity in an Object Oriented System.
* Object is an instance / variable of type of class. Each object contains the data and function to manipulate it.
* The general declaration of an object is.

Class\_ Name Object1, Object2…………. etc.

* **What is class**
* Class is the way of binding data and its associated function together.
* A class is the collection of object of similar type.
* When you define a class, you define a blueprint for an object.
* A class represent an abstraction of the object and abstracts the properties and behavior of that object.
* **What is encapsulation**
* Wrapping up of data and function in a single unit is known as encapsulation. The data encapsulation is the most striking feature of a class.
* The data is hidden, not accessible to the outside world , and only those function that are enclosed in the class can access it.
* **What is inheritance**
* The process of deriving a new car class from existing (old) class is called inheritance. The old class is referred a Base Class, and a new one is called Derived class.
* General Definition of Derived class is as follows.

Parent

Grandparent

Child

* Inheritance means that one class inherits the characteristics of another class. This is also called a “is a” relationship
* **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.
* The ability to make than 1 from is called polymorphism.
* There are **two** types of polymorphism

1. **Compile time polymorphism**
2. **Run time polymorphism**

* **Write SDLC phases with basic introduction**
* SDLC stand for **Software Development Life Cycle**
* SDLC is a structure imposed on the development of a software product that defines the process
* A Software Development Life Cycle is essentially a series of steps, or phases, that provide a model for the development
* *Software Development Life Cycle standards such as ISO/IEC* 12207
* There are a number of **six** different phases

1. ***Requirements Collection/Gathering***
2. ***Analysis***
3. ***Design***
4. ***Implementation***
5. ***Testing***
6. ***Maintenance***
7. ***Requirements Collection/Gathering***

* Features
* Client Requirements Understand
* Plan for change
* Requirements definitions usually consist of natural language, supplemented by (e.g., UML) diagrams and tables.
* Check Requirements Functional and Non-Functional

1. ***Analysis***

The Requirements collection phase all team Analysis

This analysis represents the “what” phase.

This phase defines the problem that the customer is trying to solve.

This phase starts with the requirement document delivered by the requirement phase and maps

1. ***Design***

* Design Architecture Document
* Implementation Plan
* Critical Priority Analysis
* Performance Analysis
* Test Plan
* The Design team can now expand upon the information established in the requirement document.
* The requirement document must guide this decision process.

1. ***Implementation***

In the implementation phase, the team builds the components either from scratch or by composition.

Implementation – Code

Critical Error Removal

The implementation phase deals with issues of quality, performance, baselines, libraries, and debugging.

1. ***Testing***

It is much easier to explain to a customer why there is a missing feature than to explain to a customer why the product lacks quality.

There are Different type Testing

 Regression Testing

 Internal Testing

 Unit Testing

 Application Testing

 Stress Testing

1. ***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 developing organization or team will have some mechanism to document and track

defects and deficiencies.

 configuration and version management

 reengineering (redesigning and refactoring)

 updating all analysis, design and user documentation

 Repeatable, automated tests enable evolution and refactoring

* **Explain Phases of the waterfall model**

Waterfall model

 Requirements must be “frozen” to early in the life cycle

 Requirements are validated too late

This is Diagram Below Waterfall model.

Requirements Collection

Analysis

Design

Implementation

Maintenance

Testing

* **Write phases of spiral model**

**Spiral Model** is one of the most important software Development Life Cycle model.

**Risk Analysis**



**Planning Alpha Demo**



<>

**Initial**  **Req.**



**First Prototype**

**Evaluation**  **Development & Testing**

**Evolving System**

1. Planning – Problem Solution, Determination of Objectives all Alternatives and Constraints
2. Risk Analysis – Working Develop / Software Trail / Analysis / Prototype
3. Development And Testing – Coding / Actual Software & Testing
4. Evaluation – Constmer Feedback / Assessment of the result of work

* **Write 7 key Principle of testing.**

**1.Testing shows presence of Defects**

* reduces the probability of undiscovered defects/ probability of undiscovered defects

**2.Exhaustive Testing is Impossible**

* Testing everything including all combinations of inputs and preconditions is not possible.
* That is we must Prioritise our testing effort using a Risk Based Approach.
* Then to test all the valid combinations you would need 30 517 578 125 (515) tests.

**3.Early Testing**

* Testing activities should start as early as possible in the development life cycle
* Remember from our Definition of Testing, that Testing doesn’t start once the code has been written!

**4.Defect Clustering**

* Defects are not evenly spread in a system
* They are ‘clustered’

**5.The Pesticide Paradox**

* If the same tests are repeated over and over again, eventually the same set of test cases will no longer find any new defects.
* Therefore we must learn, create and use new tests based on new techniques to catch new bugs
* Testing identifies bugs, and programmers respond to fix them

**6.Testing is Context Dependent**

* Testing is done differently in different contexts
* Different kinds of sites are tested differently.

1. **Absence of Errors Fallacy**

* **I**f the system built is unusable and does not fulfill the user’s needs and expectations then

finding and fixing defects does not help.

* If we build a system and, in doing so, find and fix defects....
* It doesn’t make it a good system
* **Explain working methodology of agile model and also write pros and cons.**
* Agile SDLC model is a combination of iterative and incremental
* Agile Methods break the product into small incremental builds.
* Every iteration involves cross functional teams working simultaneously on various areas like planning, requirements analysis, design, coding, unit testing, and acceptance testing.
* These builds are provided in iterations.

**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.
* Easy to manage

**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.