

Software Testing and Automation Assignments

MODULE - 1

1) What is Software Testing?

Software Testing Is Process Which is use to Verify Completeness, Correctness, and And Quality of any software developing by the developer.

Testing is Process to Evaluating the systems or its components with intend to it is fulfil the system requirement or not.

2) What is Manual Testing & Automation Testing?

Manual Testing

Manual Testing is a one type of software testing. Which test cases are solving by tester manually without use of automation tool it's known as manual testing.

Automation Testing

This also one type software testing. Which test cases are exacuted by tester with use of automation tool its known as automating testing

'SELENIUM'= is one automation tool use is automation testing in different type of languages.

3) What Is SDLC Model ?

'Software development life cycle' is the full form of SDLC.

SDLC is structure imposed on development of software product that defines the process for planning, analysing, designing, testing, implementation, documentation, and maintenance.

SDLC is also known as a Step by Step approach to develop any type of software in shortest possible time, standard Quality and use a lowest budget.

4) Write SDLC Phases with Basic Information

SDLC have total Six Type of Phases

1) Requirement Gathering/Collection

There are three types of Requirement gathering

- 1) Lack of clarity – it means customer doesn't clarify what they need actual about the software product
- 2) Requirement Confusion – its similar to lack of clarity because customer confuse about their actual need about any particular software/product.
- 3) Requirement Amalgamation - it means sometimes many requirement are coming in group. Customer say they need all this in one time.

2) ANALYSIS –

This is second type of SDLC. Tester need to analysis of the customer need with **SRS**(Systematic Requirement Specification)

SRS stand for Specification of whole system requirement. The merging of customer and system requirement at the one platform. "what actually customer need and what accept by system.

3) Design

After Use of SRS then Tester making design of the data which received from the Customer/Client/Company.

4) Implementation

A document from the design phase and the requirement information from the analysis phase, the tester should build exactly what has been requested.

5) TESTING

Without these phases of SDLC a successful software cannot be produce because we all making mistakes some of those are very difficult to solve but tester find defects and solve them so that the user use software easily.

6) Maintenance

The last Phases of the SDLC Structure. The maintenance mean daily running management of any software.

It have three types of

Corrective Maintenance, Adaptive Maintenance, Perfective Maintenance.

5) What Is OOPS

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

Identifying objects and assigning responsibilities to these objects.

6) WRITE BASIC CONCEPT OF OOPS

Object Oriented Programming is viewed as a collection of objects. It is used to structure the software program into simple reusable code. Here it is referred as Functional testing or Black Box Testing.

7) WHAT IS OBJECT ?

That is both data and function that operate on data are bundled as a unit called as 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 ?

Class is a collection of a data member (variables) and member function with its behavior.

Class is a blueprint or a template to describe the properties and behavior of the objects.

A class represents an abstraction of the object and abstracts the properties and behavior of that object.

9) What is encapsulation ?

A wrapping up of data and functions into a single unit is called Encapsulation. IT hide/include private access of data member & member function.

Encapsulation is placing the data and the functions that work on that data in the same place. While working with procedural languages, it is not always clear which functions work on which variables but object- oriented programming provides you framework to place the data and the relevant functions together in the same object.

10) What is inheritance ?

One class (Super, Base) inherits the properties of another class (Sub, Derived).

Inheritance describes the relationship between two own classes. A class can get some of its characteristics from a parent class and then add unique features of its.

11) Types of Inheritance:

- Single Inheritance
- Multilevel Inheritance
- Hierarchical Inheritance
- Hybrid Inheritance
- Multiple Inheritance

12) What is polymorphism ?

The ability to use an operator or function in different ways in other words giving different meaning or functions to the operators or functions is called **polymorphism**.

An ability to take one name having many different forms.

Compile time Polymorphism : (Operator Overloading)

Method name should be same in single class but its behavior (Arguments & Data type) is different.
Run time Polymorphism (Operator Overriding)
Method should be same in super class and sub class but its behavior is different.

- There is **two types of polymorphism** in Java
 - 1) Compile time polymorphism(Overloading),
 - 2) Runtime polymorphism(Overriding).

13) WHAT IS SRS ?

SRS is a complete description of the behavior of the system to be developed.

TYPES OF SRS

1) Customer Requirements

What exactly customer need about software which will be developed.

2) Functional Requirements

It known as what are the fix requirement of software which can not be changeable. Ex login page or regular functions use in any type of software.

3) Non-Functional Requirements

A requirements which are the changeable like colour. page layout etc...

14) Explain Phases of the waterfall model ?

Waterfall model is classical model software lifecycle that models the software

development as a step-by-step “waterfall” between the various development phases.

15) WHEN TO USE WATERFALL MODEL

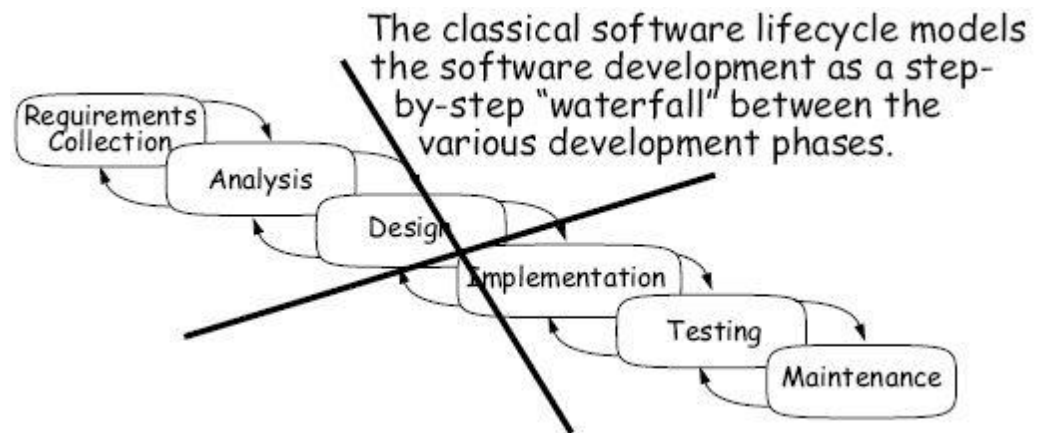
Requirements are very well documented, clear & fixed.

Product definition is stable.

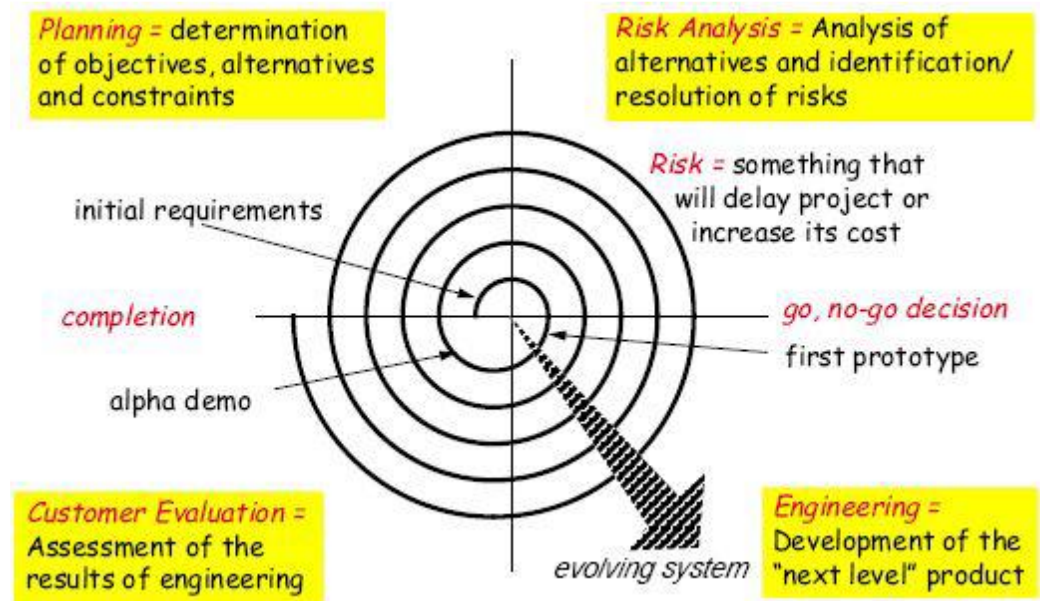
Technology is understood & not dynamic.

The project is short.

PRO	CONS
Simple and easy to understand and use	Lack of flexibility & adaptability (No changing requirements & no parallel work).
Phases are processed and completed one at a time.	High amounts of risk and uncertainty.
Works well for smaller projects where requirements are very well understood.	Poor model for long and on-going projects.
Process and results are well documented.	Not good model for complex & object oriented projects
Clearly defined stages.	
Easy to arrange tasks.	



16) Write phases of spiral model



17) When costs there are a budget constraint and risk evaluation is important For medium to high-risk projects.

- Long-term project commitment because of potential changes to economic priorities as the requirements change with time.
- Customer is not sure of their requirements which are usually the case.

- Requirements are complex and need evaluation to get clarity. New product line which should be released in phases to get enough customer feedback.
- Significant changes are expected in the product during the development cycle.

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

Agile thought process had started early in the software development and started becoming popular with time due to its flexibility and adaptability.

19) Write agile manifesto principles ?

- 1) Individuals and interactions over processes and tools.
- 2) Working software over comprehensive documentation.

3) Customer collaboration over contract negotiation.

4) Responding to change over following a plan

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

<i>Pros:</i> 1. Very realistic approach	<i>Cons:</i> 1. More risk of sustainability, maintainability and extensibility.
2. Rapid delivery.	2. Depends heavily on customer interactions.
3. Functionality can be developed rapidly	3. Very high individual dependency.
4. Resource requirements are minimum.	4. Minimum documentation generated.
5. Little or no planning required	5. Not useful for small projects.
6. Promotes teamwork and cross training.	6. Not suitable for handling complex dependencies.
7. Suitable for fixed or changing requirements	
8. Gives flexibility to developers	