**Software Testing Assignment Module-2**

1-What is software testing?

Testing is the process of evaluating a system or its components with the intent to find that whether it satisfies the specified requirements or not.

2-What is Exploratory Testing?

In exploratory testing tester focuses more on how the software actually works, testers do minimum planning and maximum execution of the software by which they get in depth idea about the software functionality, once the tester starts getting insight into the software he can make decisions to what to test next. Exploratory testing is mostly used if the requirements are incomplete and time to release the software is less.

3-What is traceability matrix?

Traceability Matrix is a table which is used to trace the requirements during the Software development life Cycle. It can be used for forward tracing or backward tracing.

4- What is Boundary value testing?

Software testing technique in which tests are designed to include representatives of boundary values. It is performed by the QA testing teams.

5- What is Equivalence partitioning testing?

Software testing technique that divides the input data of a software unit into partitions of data from which test cases can be derived. it is usually performed by the QA teams.

6- What is Integration testing?

The phase in software testing in which individual software modules are combined and tested as a group. It is usually conducted by testing teams.

7- What determines the level of risk?

The likehood of an adverse event and the impact of the event.

8- What is Alpha testing?

First of all test newly developed hardware or software in a laboratory setting. When the first round of bugs has been fixed, the product goes into beta test with actual users. For custom software, the customer may be invited into the vendor's facilities for an alpha test to ensure the client's vision has been interpreted properly by the developer.

9- What is beta testing?

Test of new or revised hardware or software that is performed by users at their facilities under normal operating conditions. Beta testing follows alpha testing. Vendors of packaged software often offer their customers the opportunity of beta testing new releases.

10- What is component testing?

Testing technique similar to unit testing but with a higher level of integration - testing is done in the context of the application instead of just directly testing a specific method. Can be performed by testing or development teams.

11- What is functional system testing?

A requirement that specifies a function that a system or system component must perform.

There is two types of techniques

1 Requirement Based Functional Testing

2 Process Based Testing

12- What is Non-Functional Testing?

Testing the attributes of a component or system that do not relate to functionality, e.g. reliability, efficiency, usability, interoperability, maintainability and portability.

13- What is GUI Testing?

Graphical User Interface (GUI) testing is the process of testing the system’s GUI of the System under Test. GUI testing involves checking the screens with the controls like menus, buttons, icons, and all types of bars – tool bar, menu bar, dialog boxes and windows etc.

14- What is Adhoc testing?

Adhoc testing is an informal testing type with an aim to break the system.

It does not follow any test design techniques to create test cases.

This testing is primarily performed if the knowledge of testers in the system under test is very high.

Testers randomly test the application without any test cases or any business requirement document.

Adhoc Testing does not follow any structured way of testing and it is randomly done on any part of application.

The Error guessing is a technique where the experienced and good testers are encouraged to think of situations in which the software may not be able to cope.

There are different types of Adhoc testing

Buddy Testing

Pair testing

Monkey Testing.

15- What is white box testing and list the types of white box testing?

Testing based on an analysis of the internal structure of the component or system.

Testing based on an analysis of the internal structure of the component or system.

In white-box testing the tester is concentrating on how the software does it.

Types of White Box testing:

1 Path testing

2 Loop testing

3 Conditional testing

4 Unit testing

5 Mutation testing

6 Integration testing

16- What is black box testing? What are the different black box testing techniques?

Testing, either functional or non-functional, without reference to the internal structure of the component or system.

Specification-based testing technique is also known as ‘black-box’ or input/output driven testing techniques because they view the software as a black-box with inputs and outputs.

The testers have no knowledge of how the system or component is structured inside the box. In black-box testing the tester is concentrating on what the software does, not how it does it.

The technique of testing without having any knowledge of the interior workings of the application is Black Box testing.

Black Box testing technique

1 Equivalence partitioning

2 Boundary value analysis

3 Decision tables

4 State transition testing

5 State transition testing

17- Mention what are the categories of defects?

Arithmetic Defects

Logical Defects

Syntax Defects

Multithreading Defects

Interface Defects

18- Mention what bigbang testing is?

Big Bang Testing is an integration testing strategy wherein all units are linked at once, resulting in a complete system. When this type of testing strategy is adopted, it is difficult to isolate any errors found, because attention is not paid to verifying the interfaces across individual units.

19- What is the purpose of exit criteria?

Exit criterion is used to determine whether a given test activity has been completed or NOT. Exit criteria can be defined for all of the test activities right from planning, specification and execution. Exit criterion should be part of test plan and decided in the planning stage.

20- When should "Regression Testing" be performed?

Regression testing is done after functional testing has concluded, to verify that the other functionalities are working. In the corporate world, regression testing has traditionally been performed by a software quality assurance team after the development team has completed work.

21- What is 7 key principles? Explain in detail?

1. Testing shows presence of defects
2. Exhaustive testing is not possible
3. Early testing
4. Defect clustering
5. Pesticide paradox
6. Testing is context dependent
7. Absence of errors fallacy

1) Exhaustive testing is not possible

Yes! Exhaustive testing is not possible. Instead, we need the optimal amount of testing based on the risk assessment of the application.

And the million dollar question is, how do you determine this risk?

To answer this let’s do an exercise

In your opinion, Which operation is most likely to cause your Operating system to fail?

I am sure most of you would have guessed, Opening 10 different application all at the same time.

So if you were testing this Operating system, you would realize that defects are likely to be found in multi-tasking activity and need to be tested thoroughly which brings us to our next Defect Clustering.

## 2) Defect Clustering

Defect Clustering which states that a small number of modules contain most of the defects detected. This is the application of the Pareto Principle to software testing: approximately 80% of the problems are found in 20% of the modules.

By experience, you can identify such risky modules. But this approach has its own problems

If the same tests are repeated over and over again, eventually the same test cases will no longer find new bugs.

## 3) Pesticide Paradox

Repetitive use of the same pesticide mix to eradicate insects during farming will over time lead to the insects developing resistance to the pesticide Thereby ineffective of pesticides on insects. The same applies to software testing. If the same set of repetitive tests are conducted, the method will be useless for discovering new defects.

To overcome this, the test cases need to be regularly reviewed & revised, adding new & different test cases to help find more defects.

Testers cannot simply depend on existing test techniques. He must look out continually to improve the existing methods to make testing more effective. But even after all this sweat & hard work in testing, you can never claim your product is bug-free. To drive home this point, let’s see this video of the public launch of Windows 98

You think a company like MICROSOFT would not have tested their OS thoroughly & would risk their reputation just to see their OS crashing during its public launch!

## 4) Testing shows a presence of defects

Hence, testing principle states that – Testing talks about the presence of defects and don’t talk about the absence of defects. i.e. Software Testing reduces the probability of undiscovered defects remaining in the software but even if no defects are found, it is not a proof of correctness.

But what if, you work extra hard, taking all precautions & make your software product 99% bug-free. And the software does not meet the needs & requirements of the clients.

This leads us to our next principle, which states that- Absence of Error.

## 5) Absence of Error – fallacy

It is possible that software which is 99% bug-free is still unusable. This can be the case if the system is tested thoroughly for the wrong requirement. Software testing is not mere finding defects, but also to check that software addresses the business needs. The absence of Error is a Fallacy i.e. Finding and fixing defects does not help if the system build is unusable and does not fulfill the user’s needs & requirements.

To solve this problem, the next principle of testing states that Early Testing.

## 6) Early Testing

Early Testing – Testing should start as early as possible in the Software Development Life Cycle. So that any defects in the requirements or design phase are captured in early stages. It is much cheaper to fix a Defect in the early stages of testing. But how early one should start testing? It is recommended that you start finding the bug the moment the requirements are defined. More on this principle in a later training tutorial.

## 7) Testing is context dependent

Testing is context dependent which basically means that the way you test an e-commerce site will be different from the way you test a commercial off the shelf application. All the developed software’s are not identical. You might use a different approach, methodologies, techniques, and types of testing depending upon the application type. For instance testing, any POS system at a retail store will be different than testing an ATM machine.

22- Difference between QA v/s QC v/s Tester

* Quality Assurance:-

1, Activities which ensure the implementation of processes, procedures and standards in context to verification of developed software and intended requirements.

2, Focuses on processes and procedures rather than conducting actual testing on the system.

3, Process oriented activities.

4, Preventive activities.

* Quality Control:-

1, Activities which ensure the verification of developed software with respect to documented (or not in some cases) requirements.

2, Focuses on actual testing by executing Software with intend to identify bug/defect through implementation of procedures and process.

3, Product oriented activities.

4, It is a corrective process.

* Tester:-

1, Activities which ensure the identification of bugs/error/defects in the Software.

2, Focuses on actual testing.

3, Product oriented activities.

4, It is a preventive process.

23- Difference between Smoke and Sanity?

* Smoke Testing has a goal to verify “stability” whereas Sanity Testing has a goal to verify “rationality”.
* Smoke Testing is done by both developers or testers whereas Sanity Testing is done by testers.
* Smoke Testing verifies the critical functionalities of the system whereas Sanity Testing verifies the new functionality like bug fixes.
* Smoke testing is a subset of acceptance testing whereas Sanity testing is a subset of Regression Testing.
* Smoke testing is documented or scripted whereas Sanity testing isn’t.
* Smoke testing verifies the entire system from end to end whereas Sanity Testing verifies only a particular component.

24- Difference between verification and Validation

**Verification** is the process of checking that a software achieves its goal without any bugs. It is the process to ensure whether the product that is developed is right or not. It verifies whether the developed product fulfills the requirements that we have. Verification is static testing.   
Verification means **Are we building the product right?**

**Validation** is the process of checking whether the software product is up to the mark or in other words product has high level requirements. It is the process of checking the validation of product i.e. it checks what we are developing is the right product. it is validation of actual and expected product. Validation is the dynamic testing.

25- Explain types of Performance testing.

### Load Testing

Load testing is sort of the simplest form of performance testing. You induce a normal or expected workload to a system under test and observe it. You can use load tests to determine general system behavior, latency and throughput. In general load tests are used to verify your quality criteria.

### Stress Testing

Stress testing is basically a load test, but we are applying a higher-than-expected workload and see how the system behaves under serious stress and when exceeding the design limits. You want to learn when your system breaks and how it starts to fail when being in a serious traffic situation.

A typical approach is to steadily increase the load to see where the system under test begins to violate its non-functional requirements. You can use this “tipping point” to describe the capacity of the given system, like “we can handle 1000 concurrent users per application server before we start to violate our quality requirements”.

### Scalability Testing

With scalability testing you are changing the perspective to answer the question: How effective can I grow? You can run a series of stress tests and gather data on how effective you really are.

Using stress tests in a series where you steadily increase the system’s resources, you can easily tell if your system can translate this into additional capacity.

26- What is Error, Defect, Bug and failure?

* Error – It is the Deviation from actual and the expected value.
* Defect – It is found in the product itself after it is shipped to the respective customer
* Bug – It is found in the development environment before the product is shipped to the respective customer.

27- Difference between Priority and Severity

Defect Priority:-

Priority is Relative and Business-Focused. Priority defines the order in which we should resolve a defect. Should we fix it now, or can it wait? This priority status is set by the tester to the developer mentioning the time frame to fix the defect. If high priority is mentioned then the developer has to fix it at the earliest. The priority status is set based on the customer requirements.

Defect Severity:-

Severity is absolute and Customer-Focused. It is the extent to which the defect can affect the software. In other words it defines the impact that a given defect has on the system.

28- What is Bug Life Cycle?

Bug life cycle is nothing but the various phases a bug under goes after it is raised or reported.

The different phases of Bug life cycle are,

- New or Opened

- Assigned

- Fixed

- Tested

- Closed

29- Explain the difference between Functional testing and NonFunctional testing

**Functional Testing:**  
Functional testing is a type of software testing in which the system is tested against the functional requirements and specifications. Functional testing ensures that the requirements or specifications are properly satisfied by the application. This type of testing is particularly concerned with the result of processing. It focuses on simulation of actual system usage but does not develop any system structure assumptions.

It is basically defined as a type of testing which verifies that each function of the software application works in conformance with the requirement and specification. This testing is not concerned about the source code of the application. Each functionality of the software application is tested by providing appropriate test input, expecting the output and comparing the actual output with the expected output.

**Non-functional Testing:**  
Non-functional testing is a type of software testing that is performed to verify the non-functional requirements of the application. It verifies whether the behavior of the system is as per the requirement or not. It tests all the aspects which are not tested in functional testing.

Non-functional testing is defined as a type of software testing to check non-functional aspects of a software application. It is designed to test the readiness of a system as per nonfunctional parameters which are never addressed by functional testing. Non-functional testing is as important as functional testing.

30- What is the difference between the STLC (Software Testing Life Cycle) and SDLC (Software Development Life Cycle)?

SDLC:-

Software Development Life Cycle (SDLC) defines all the standard phases which are involved during the software development process. SDLC life cycle is a process of developing software through a phased manner in the following order

1. Requirements Gathering
2. Design the software
3. Build the Software
4. Test
5. Deployment
6. Maintenance.

Each stage has definite entry and exit criteria along with deliverables.

## STLC:-

## Software Testing Life Cycle (STLC) is the testing process that is executed in a well-planned manner. In the STLC process, various activities are carried out to improve the quality of the product. However, STLC phases only deal with testing and detecting errors but not development itself.

Different companies define different phases in STLC. However, generic Software Test Life Cycle has the following stages.

1. Requirement Analysis
2. Test Planning
3. Test Development
4. Test Environment Setup
5. Test Execution & Closure

31-What is the difference between test scenarios, test cases, and test script?

**Test Scenarios:**  A Test Scenario is any functionality that can be tested. It is also called Test Condition or Test Possibility.

**Test Cases:**  It is a document that contains the steps that has to be executed, it has been planned earlier.

**Test Script:**It is written in a programming language and it's a short program used to test part of functionality of the software system. In other words a written set of steps that should be performed manually.

32- Explain what Test Plan is? What is the information that should be covered.

A Test Plan refers to a detailed document that catalogs the test strategy, objectives, schedule, estimations, deadlines, and the resources required for completing that particular project. Think of it as a blueprint for running the tests needed to ensure the software is working properly – controlled by test managers.

33- What are the different Methodologies in Agile Development Model?

1, Scrum

2, eXtreme Programming

3, eXtreme Programming

4, Test Driven Development (TDD)

5, Feature Driven Development

6, XBreed

7, Crystal

34- Explain the difference between Authorization and Authentication in Web testing.

Authentication:-

Authentication verifies who the user is.

Authentication works through passwords, one-time pins, biometric information, and other information provided or entered by the user.

Authentication is the first step of a good identity and access management process.

Authentication is visible to and partially changeable by the user.

Authorization:-

Authorization determines what resources a user can access.

Authorization works through settings that are implemented and maintained by the organization.

Authorization always takes place after authentication.

Authorization isn’t visible to or changeable by the user.

34- What are the common problems faced in Web testing?

## Integration

## Interoperability

## Security

## Performance

## Usability

## Quality Testing, Exceptional Services.