Module -2

1. What is exploratory testing.?

Ans. 1. What is Exploratory Testing?

* 2. Exploratory testing is about learning the product, designing and executing tests, and interpreting the test results, all at the same time. It is a plan-as-you-test approach.
* 3. In contrast to formal testing (e.g. manual testing), where the purpose is to monitor known risks, the purpose of exploratory testing is to analyze potential risks.

This means that exploratory testing starts with a focus on the things you do not know, and the illusions you are holding true without any empirical evidence.

The ultimate goal is to explore the unknown to avoid the unexpected happening.

* 4. In exploratory testing, your next test is always influenced by the result of the last test you did. This implies that you become more exploratory the more you can’t tell what test

should be run in advance of the test cycle. Finally, exploratory testing is not so much a thing you do, it is much more a way you think

2. what is traceability matrix.?

Ans. A traceability matrix is a document that details the technical requirements for a given test scenario and its current state. It helps the testing team understand the level of testing that

is done for a given product. The traceability process itself is used to review the test cases that were defined for any requirement

3.what is a boundary value testing.?

Ans. Boundary value analysis is a methodology for designing test cases that

concentrates software testing effort on cases near the limits of valid ranges.

* Boundary value analysis is a method which refines equivalence partitioning
* Boundary value analysis generates test cases that highlight errors better

than equivalence partitioning.

* The trick is to concentrate software testing efforts at the extreme ends of the equivalence classes.
* At those points when input values change from valid to invalid errors are most likely to occur.
* Boundary Value Analysis (BVA) uses the same analysis of partitions as EP and is usually used in conjunction with EP in test case design

4. what is equivalence partitioning testing.?

Ans. Equivalence Partitioning Method is also known as Equivalence class partitioning (ECP). It is a software testing technique or black-box testing that divides input domain into classes of data

and with the help of these classes of data, test cases can be derived. An ideal test case identifies class of error that

5. what is integration testing.?

Ans. Integration Testing is a level of the software testing process where individual units are combined and tested as a group.

The purpose of this level of testing is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in Integration Testing.

* Integration testing tests integration or interfaces between components, interactions to different parts of the system such as an operating system, file

system and hardware or interfaces between systems.

* Integration testing is done by a specific integration tester or test team.
* Components may be code modules, operating systems, hardware and Components may be code modules, operating systems, hardware and
* There are 2 levels of Integration Testing

1. There are 2 levels of Integration Testing
2. System integration testing

6.what is determines the level of risk.?

Ans. As Risk is determined by a combination of Probability and Severity, the main area of the Matrix reveals the Risk Levels. The levels are Low, Medium, High, and Extremely High.

To have a low level of risk, we must have a somewhat limited probability and level of severity

7. what is alpha testing.?

Ans. It is always performed by the developers at the software development site

* Sometimes it is also performed by Independent Testing Team.
* Alpha Testing is not open to the market and public
* It is conducted for the software application and project.
* It is always performed in Virtual Environment.
* It is always performed within the organization.
* It is the form of Acceptance Testing.

8. what is beta testing?

Ans. It is always performed by the customers at their own site.

* It is not performed by Independent Testing Team.
* Beta Testing is always open to the market and public.
* It is usually conducted for software product.
* It is performed in Real Time Environment.
* It is always performed outside the organization.
* It is also the form of Acceptance Testing.

9. what is component testing.?

Ans.

10. what is a functional system tasting.?

Ans. Functional System Testing : A requirement that specifies a function that a system or system component must perform

* A Requirement may exist as a text document and/or a model
* There is two types of Test Approach

1. Requirement Based Functional Testing
2. Process based testing
3. Requirement Based Functional Testing

* Testing against requirements and specifications
* Test procedures and cases derived from:
* system requirements functional specification
* high level System design
* User documentation/instructions

1. Process based testing

* Test procedures and cases derived from:
* Business scenarios
* Use cases

11. what is non-functional testing.?

Ans. It is the testing of “how” the system works. Non-functional testing may be performed at all test levels.

* The term non-functional testing describes the tests required to measure characteristics of systems and software

that can be quantified on a varying scale, such as response times for performance testing. To address this issue, performance testing is carried out to check & fine tune

system response time

* load testing is carried out to check systems performance at different loads i.e. number of users accessing the system

12. what is GUI testing.?

Ans. 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

* WHAT DO YOU CHECK IN GUI TESTING?

Check all the GUI elements for size, position, width, length and acceptance of characters or numbers. For instance, you must be able to provide inputs to the input fields.

* Check you can execute the intended functionality of the application using the GUI
* Check Error Messages are displayed correctly
* Check for Clear demarcation of different sections on screen
* Check Font used in application is readable
* Check the alignment of the text is proper
* Check the Color of the font and warning messages is aesthetically pleasing Check that the images have good clarity
* Check that the images are properly aligned

13.what is adhoc testing.?

Ans. 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.
* Main aim of this testing is to find defects by random checking.
* Adhoc testing can be achieved with the testing technique called Error Guessin

15. what is load testing.?

Ans. Load testing - Its a performance testing to check system behavior under load. Testing an application under heavy loads, such as testing of a web site under a

range of loads to determine at what point the system’s response time degrades or fails.

Load testing is a kind of performance testing which determines a system’s performance under real-life load conditions. This testing helps determine how the application

behaves when multiple users access it simultaneously

16. what is stress testing.?

Ans. Stress testing - System is stressed beyond its specifications to check how and when it fails. Performed under heavy load like putting large number beyond storage capacity,

complex database queries, continuous input to system or database load.

Stress testing is used to test the stability & reliability of the system. This test mainly determines the system on its robustness

and error handling under extremely heavy load conditions

17. What is white box testing and list the types of white box testing?

Ans. Types of white box testing

* Path Testing. Path Testing is a white-box testing approach based on a program's control structure.
* Loop Testing.
* Conditional Testing.
* Unit Testing.
* Mutation Testing.
* Integration Testing.
* Penetration Testing. .
* Testing based on Memory Perspective

18. What is black box testing?

Ans. What are the different black box testing techniques? Black box testing is a method for writing test cases that is frequently utilized. It may be beneficial for condensing

a large number of potential inputs into a smaller number of more effective ones. It is accomplished by categorizing inputs into classes a assigning a value to each class.

19. Mention what bigbang testing is?

Ans. What is Big-Bang Testing? Big Bang Integration Testing is an integration testing strategy wherein all units are linked at once, resulting in a complete system.

20. Mention what are the categories of defects?

Ans. 3 Types of defects every importer needs to know. Quality control⎝ professionals typically classify quality defects into three main categories: minor, major and critical.

The nature and severity of a defect determines in which of the three categories it belongs

21. What is the purpose of exit criteria?

Ans. 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.

22. When should "Regression Testing" be performed?

Ans. Regression testing is necessary after any feature (or application) enhancement, bug fix, or configuration changes. For example, when developers add a new widget to an application.

As more regressions are found in software products are found companies are moving towards test

23. What is 7 key principles? Explain in detail?

Ans. 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.

* 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 where 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 principle 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

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

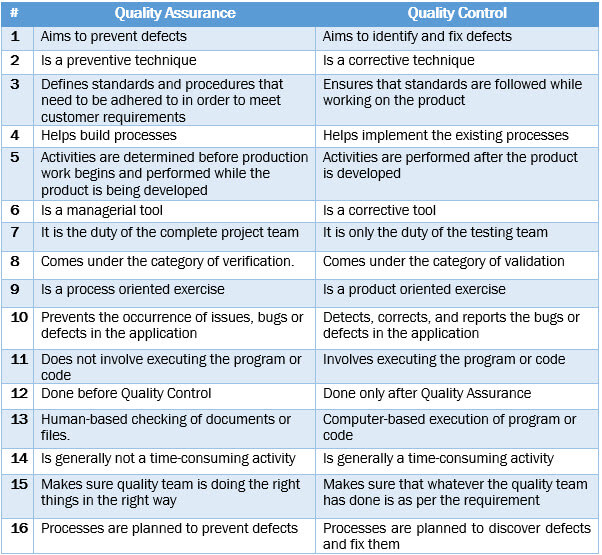
* Testing is context dependent which basically means that the way you test an e-commerce site will be different from the way you test a commercial off the shelf application.

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

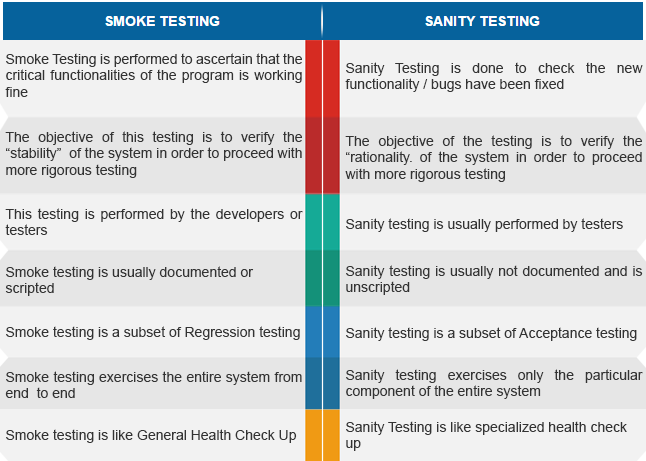
24. Difference between QA v/s QC v/s Tester

Ans.



25. Difference between Smoke and Sanity?

Ans.



24 Difference between verification and Validation?

Ans. 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.

Validation means Are we building the right product

25. Explain types of Performance testing?

Ans. Load Testing. Load testing measures system performance as the workload increases.

* Stress Testing.
* Spike Testing.
* Endurance Testing.
* Scalability Testing.
* Volume Testing.
* Identify the Testing Environment.
* Identify Performance Metrics

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

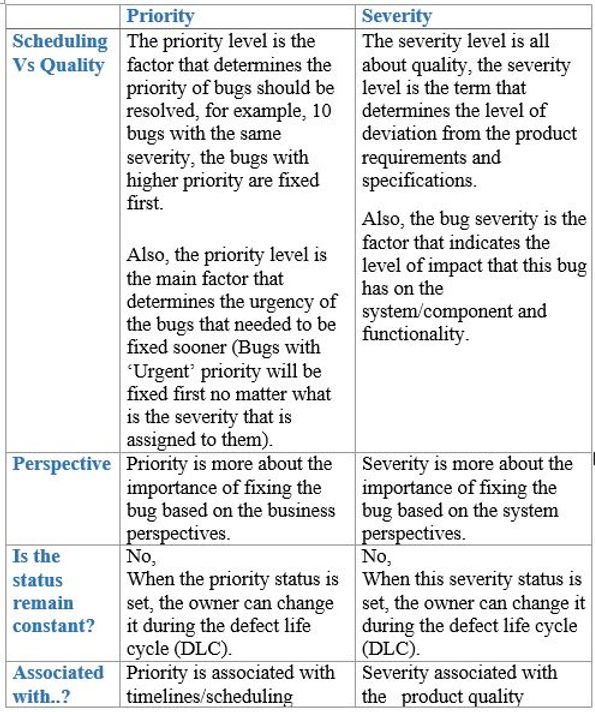
Ans. In software testing, a bug is the informal name of defects, which means that software or application is not working as per the requirement.

When we have some coding error, it leads a program to its breakdown, which is known as a bug. The test engineers use the terminology Bug.

* When the application is not working as per the requirement is knows as defects. It is specified as the aberration from the actual and expected result of the application or software
* The Problem in code leads to errors, which means that a mistake can occur due to the⎝ developer's coding error as the developer misunderstood the requirement or the requirement was not defined correctly

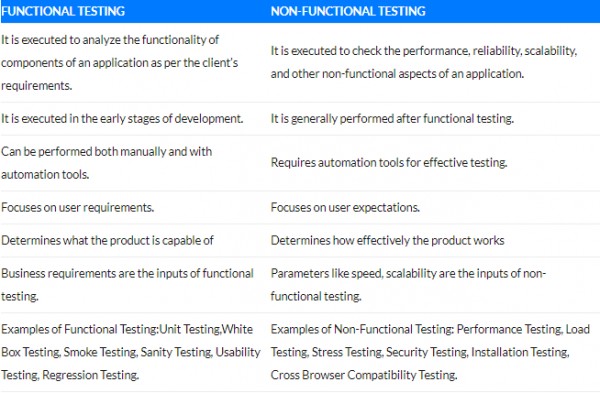
27 Difference between Priority and Severity?

Ans.



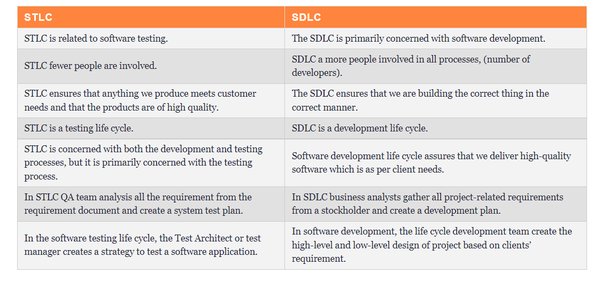
28 Explain the difference between Functional testing and NonFunctional testing?

Ans.



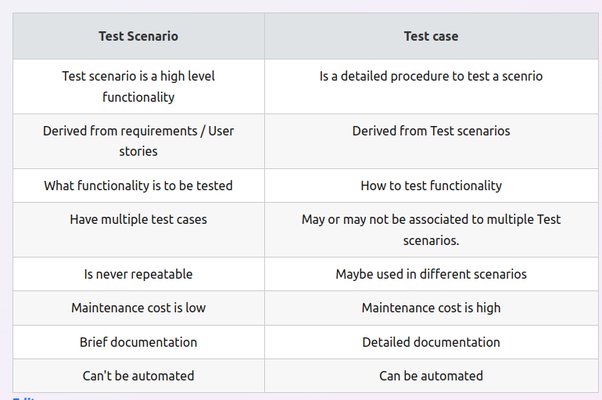
29.What is the difference between the STLC (Software Testing Life Cycle) and SDLC (Software Development Life Cycle)?

Ans.



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

Ans.



31. What is priority?

Ans. What is Priority in Testing? One can define Priority as a parameter for⎝ deciding the order in which one can fix the defect.

In this, the defect with a higher priority first needs to get fixed. Priority basically defines the order in which one would resolve any given defect.

32. what is Severity?

Ans. What is Severity in Testing? One can define Severity as the extent to⎝ which any given defect can affect/ impact a particular software.

Severity is basically a parameter that denotes the impact of any defect and its implication on a software's functionality

33 Difference between priority and severity?

Ans. What is Priority in Testing? One can define Priority as a parameter for deciding the order in which one can fix the defect. In this, the defect with

a higher priority first needs to get fixed. Priority basically defines the order in which one would resolve any given defect

* What is Severity in Testing? One can define Severity as the extent to which any given defect can affect/ impact a particular software

Severity is basically a parameter that denotes the impact of any defect and its implication on a software's functionality

34 What are the different Methodologies in Agile Development Model?

Ans.

* 1) Kanban.
* 2) Scrum.
* 3) Extreme Programming (XP)
* 4) Crystal.
* 5) Dynamic Systems Development Method (DSDM)
* 6) Feature-Driven Development (FDD)