**SOFTWARE TESTING ASSIGNMENT**

**Module-2**

1. **What is Exploratory Testing?**

**Exploratory Testing** is a type of software testing where Test cases are not created in advance but testers check system on the fly.

**2. What is traceability matrix?**

Test conditions should be able to be linked back to their sources

in the test basis, this is known as traceability

**3.What is Boundary value testing?**

Boundary value analysis is a methodology for designing test cases that

concentrates software testing effort on cases near **the limits of valid**

**ranges**

**4.What is Equivalence partitioning testing?**

Aim is to treat groups of inputs as equivalent and to select one representative input to test them all

**5.What is Integration testing?**

Integration Testing *-* Testing performed to expose defects in the interfaces and in the interactions between integrated components or systems

**6. What determines the level of risk?**

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?**

Alpha testing is conducted in the organization and tested by a representative group of end-users at the developer's side and sometimes by an independent team of testers.

Alpha testing is simulated or real operational testing at an in-house site. It comes after the unit testing, integration testing, etc. Alpha testing used after all the testing are executed.

**8. What is beta testing?**

Beta testing is a type of **U**ser **A**cceptance **T**esting among the most crucial testing, which performed before the release of the software. Beta Testing is a type of Field Test. This testing performs at the end of the ***software*** testing life cycle. This type of testing can be considered as external user acceptance testing. It is a type of salient testing. Real users perform this testing. This testing executed after the alpha testing. In this the new version, beta testing is released to a limited audience to check the accessibility, usability, and functionality, and more.

**9. What is component testing?**

Component (Unit) – A minimal software item that can be tested in isolation. It means “A unit is the smallest testable part of software.”

**10. What is functional system testing?**

Testing based on an analysis of the specification of the functionality of a component or system.

**11. What is Non-Functional Testing?**

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

**12. 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.

**13. What is Adhoc testing?**

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.

**14. What is 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.

**15. What is 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.

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

**White Box Testing**: Testing based on an analysis of the internal structure of the component or system.

**Types of white box testing**

1. Path Testing

2. Loop Testing

3. Conditional Testing

4. Unit Testing

5. Mutation Testing

6. Integration Testing

7. Penetration Testing

8. Testing based on Memory Perspective

9. Test Performance of the Program

**17. What is black box testing? What are the different black box testing techniques?**

**Black-box testing**: Testing, either functional or non-functional, without reference to the internal structure of the component or system.

**Black-Box Testing Techniques**

* Equivalence partitioning
* Boundary value analysis
* Decision tables
* State transition testing
* Use-case Testing
* Other Black Box Testing
* Syntax or Pattern Testing

**18. Mention what are the categories of defects?**

* Arithmetic Defects
* Logical Defects
* Syntax Defects
* Multithreading Defects
* Interface Defects
* Performance Defects

**19. Mention what big-bang testing is?**

In Big Bang integration testing all components or modules is integrated simultaneously, after which everything is tested as a whole.

**20. 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.

**21. 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.

**22. What is 7 key principles? Explain in detail?**

Here is the 7 key principle

1. **Testing Shows Presence Of Defects**
2. **Exhaustive Testing Is Impossible**
3. **Early Testing**
4. **Defect Clustering**
5. **Pesticide Paradox**
6. **Testing Is Context Dependent**
7. **Absence Of Error Fallacy**

**1. Testing Shows Presence Of Defects**

- testing reduces the probability of undiscovered defects remaining in the software but, even if no defect are found is not a proof of correctness

**2. Exhaustive Testing Is Impossible**

- testing everything including all combinations of inputs and preconditions is not possible

**3. Early Testing**

- testing activities should starts as early as possible in the software or system developed life cycle and should be focused on defined objectives

**4. Defect Clustering**

- defect are not evenly spread in system

-they are clustering

EXAMPLE – corona virus duration – green zone – orange zone – red zone

**5. Pesticide Paradox**

- if the same tests are repeated over and over again , eventually the same sets of test cases will no longer find any new defects

**6. Testing Is Context Dependent**

- testing is basically context dependent

- testing is done differently in different contexts

-different kind of sites are tested differently

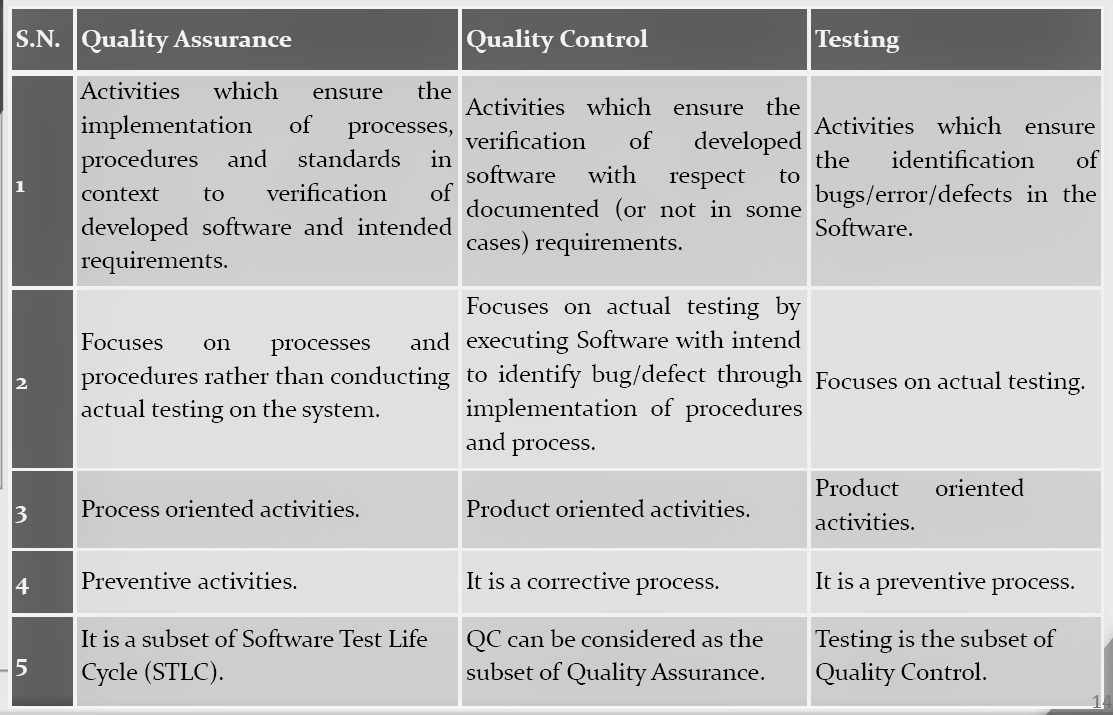
EXAMPLE – safety -critical software is tested differently from an e-commerce

- 3 to 10 failure per KLOC typical for program it was 80% testing

**7. Absence Of Errors Fallacy**

- if the system built is unusable and does not fulfil the users needs and expectations the find and fixing defects does not help.

**23. Difference between QA v/s QC v/s Tester**

****

**24. Difference between Smoke and Sanity?**

|  |  |  |
| --- | --- | --- |
| **Sr.no** | **Smoke Testing** | **Sanity Testing** |
|  |  |  |
| 1 | [Smoke testing](https://www.geeksforgeeks.org/smoke-testing-software-testing/) is done to assure that the acute functionalities of program is working fine. | [Sanity testing](https://www.geeksforgeeks.org/sanity-testing-software-testing/) is done to check the bugs have been fixed after the build. |
| 2 | Smoke testing is also called subset of acceptance testing. | Sanity testing is also called subset of regression testing. |
| 3 | Smoke testing is documented. | Sanity testing isn’t documented. |
| 4 | Smoke testing is performed by either developers or testers. | Sanity testing is normally performed by testers. |
| 5 | Smoke testing may be stable or unstable. | Sanity testing is stable. |
| 6 | Smoke testing is scripted. | Sanity testing is usually not scripted. |
| 7 | Smoke testing is done to measures the stability of the system/product by performing testing. | Sanity testing is done to measures the rationality of the system/product by performing testing. |
| 8 | Smoke testing is used to test all over function of the system/product. | Sanity testing is used in the case of only modified or defect functions of system/products. |
| 9 | Smoke testing can be performed either manually or by using automation tools. | Sanity testing is commonly executed manually, not by using any automation approach. |
| 10 | Smoke testing is performed when new product is built. | Sanity testing is conducted after the completion of regression testing. |
| 11 | It includes all of the system’s essential basic functionality. | It includes only those modules where change in code is made. |
| 12 | Smoke Testing firstly performs on the initial build. smoke testing is done first. | Sanity Testing is done on stable builds or for the introduced new features in the software. |
| 13 | Smoke testing can be carried out either way-manually or automatically. | Without using test cases or scripts sanity testing can be carried out |
| 14 | There is end-to-end system verification done in smoke testing. | A specific component gets verified in sanity testing. |
| 15 | In the smoke testing process, the software build could be stable or unstable. | During sanity testing, the software build is comparatively stable. |

**25. Difference between verification and Validation**

|  |  |  |
| --- | --- | --- |
| Sr no | **Verification** | **Validation** |
| 1 | It includes checking documents, design, codes and programs. | It includes testing and validating the actual product. |
| 2 | Verification is the static testing. | Validation is the dynamic testing. |
| 3 | It does *not* include the execution of the code. | It includes the execution of the code. |
| 4 | Methods used in verification are reviews, walkthroughs, inspections and desk-checking. | Methods used in validation are Black Box Testing, White Box Testing and non-functional testing. |
| 5 | It checks whether the software conforms to specifications or not. | It checks whether the software meets the requirements and expectations of a customer or not. |
| 6 | It can find the bugs in the early stage of the development. | It can only find the bugs that could not be found by the verification process. |
| 7 | The goal of verification is application and software architecture and specification. | The goal of validation is an actual product. |
| 8 | Quality assurance team does verification. | Validation is executed on software code with the help of testing team. |
| 9 | It comes before validation. | It comes after verification. |
| 10 | It consists of checking of documents/files and is performed by human. | It consists of execution of program and is performed by computer. |

**26. Explain types of Performance testing.**

* Load testing
* Stress testing
* Endurance testing
* Spike testing
* Volume testing
* Scalability testing

**27. What is Error, Defect, Bug and failure?**

* We can say that a mistake made by a programmer during coding is called an **error**,
* An error found during the unit testing in the development phase is called a **defect**,
* An error found during the testing phase is called a **bug** and when an error is found at an end user's end is called as the **failure**.

**28. Difference between Priority and Severity**

|  |  |  |
| --- | --- | --- |
| **Sr.no** | **Severity** | **Priority** |
| 1 | Severity is a parameter to denote the impact of a particular defect on the software. | Priority is a parameter to decide the order in which defects should be fixed. |
| 2 | Severity means how severe defect is affecting the functionality. | Priority means how fast defect has to be fixed. |
| 3 | Severity is related to the quality standard. | Priority is related to scheduling to resolve the problem. |
| 4 | Testing engineer decides the severity level of the defect. | Product manager decides the priorities of defects. |
| 5 | Its value is objective. | Its value is subjective. |
| 6 | Its value doesn’t change from time to time. | Its value changes from time to time. |
| 7 | Severity is of 5 types: Critical, Major, Moderate, Minor, and Cosmetic. | Priority is of 3 types: Low, Medium, and High. |

**29. What is Bug Life Cycle?**

The Defect Life Cycle, also known as the Bug Life Cycle, is **a cycle of defects from which it goes through covering the different states in its entire life**. This starts as soon as any new defect is found by a tester and comes to an end when a tester closes that defect assuring that it won't get reproduced again.

**30. Explain the difference between Functional testing and Non-Functional testing**

|  |  |  |
| --- | --- | --- |
| **Sr. no** | **Functional Testing** | **Non-functional Testing** |
| **1** | It verifies the operations and actions of an application. | It verifies the behavior of an application. |
| **2** | It is based on requirements of customer. | It is based on expectations of customer. |
| **3** | It helps to enhance the behavior of the application. | It helps to improve the performance of the application. |
| **4** | Functional testing is easy to execute manually. | It is hard to execute non-functional testing manually. |
| **5** | It tests what the product does. | It describes how the product does. |
| **6** | Functional testing is based on the business requirement. | Non-functional testing is based on the performance requirement. |
| **7** | **Examples:**  **1.** Unit Testing  **2.** Smoke Testing  **3.** Integration Testing  **4.** Regression Testing | **Examples:**  **1.** Performance Testing  **2.** Load Testing  **3.** Stress Testing  **4.** Scalability Testing |

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

|  |  |  |
| --- | --- | --- |
| **Sr.no** | **SDLC** | **STLC** |
| 1 | SDLC is mainly related to software development. | STLC is mainly related to software testing. |
| 2 | Besides development other phases like testing is also included. | It focuses only on testing the software. |
| 3 | SDLC involves total six phases or steps. | STLC involves only five phases or steps. |
| 4 | In SDLC, more number of members (developers) are required for the whole process. | In STLC, less number of members (testers) are needed. |
| 5 | In SDLC, development team makes the plans and designs based on the requirements. | In STLC, testing team(Test Lead or Test Architect) makes the plans and designs. |
| 6 | Goal of SDLC is to complete successful development of software. | Goal of STLC is to complete successful testing of software. |
| 7 | It helps in developing good quality software. | It helps in making the software defects free. |
| 8 | SDLC phases are completed before the STLC phases. | STLC phases are performed after SDLC phases. |
| 9 | Post deployment support , enhancement ,and update are to be included if necessary. | Regression tests are run by QA team to check deployed maintenance code and maintains test cases and automated scripts. |
| 10 | Creation of reusable software systems is the end result of SDLC. | A tested software system is the end result of STLC. |

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

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

**33. What is priority?**

Priority is defined as the order in which the defects should be resolved. The priority status is usually set by the testing team while raising the defect against the dev team mentioning the timeframe to fix the defect. The Priority status is set based on end users requirement.

**Low -**This defect can be fixed after the critical ones are fixed.

**Medium -**The defect should be resolved in the subsequent builds.

**High -**The defect must be resolved immediately because the defect is affecting the application to a considerable extent and the relevant modules cannot be used until it's fixed.

**Urgent -**The defect must be resolved immediately because the defect is affecting the application or the product severely and the product cannot be used until it has been fixed.

**34. What is severity?**

The impact of the bug on the application is known as severity.  
It can be a **blocker, critical, major, and minor** for the bug.

**Blocker:** if the severity of a bug is a blocker, which means we cannot proceed to the next module, and unnecessarily test engineer sits ideal.

There are two types of **blocker** bug, which are as follows:

**A major feature is not working:** Login to HDFC, amount transfer is not working

**The major flow is not working:** Login and signup itself not working in HDFC application.

**Critical:** if it is critical, that means the main functionality is not working, and the test engineer cannot continue testing.

**Major:** if it is major, which means that the supporting components and modules are not working fine, but test engineer can continue the testing.

**Minor:** if the severity of a bug is major, which means that all the U.I problems are not working fine, but testing can be processed without interruption.

**35. Bug categories are…?**

Thus, bugs can be classified as urgent, high-, medium-, and low-priority.

**36. Advantage of Bugzila?**

* Open source, free bug tracking tool.
* Automatic [Duplicate Bug Detection](https://cloudinfrastructureservices.co.uk/how-to-setup-bugzilla-issue-tracker-on-azure-aws-gcp/).
* Search option with advanced features.
* File/Modify Bugs By Email.
* Move Bugs Between Installs.
* Multiple [Authentication](https://cloudinfrastructureservices.co.uk/adfs-vs-azure-ad-how-authentication-has-evolved/) Methods ([LDAP](https://cloudinfrastructureservices.co.uk/radius-vs-ldap-vs-kerberos/),[Apache server](https://cloudinfrastructureservices.co.uk/how-to-setup-apache-web-server-mysql-server-on-linux-in-azure-aws-gcp/)).
* Time Tracking.
* Automated bug reporting; has an API to interact with system.
* Integrated email capabilities.
* Detailed permissions system.
* Optimized database structure to enhance performance.
* Robust security.
* Powerful query tool.
* Ideal for small projects.

**37. What are the different Methodologies in Agile Development Model?**

* Scrum
* Crystal
* Dynamic Software Development Method(DSDM)
* Feature Driven Development(FDD)
* Lean Software Development
* eXtreme Programming(XP)

**38. Explain the difference between Authorization and Authentication in Web testing.What are the common problems faced in Web testing?**

|  |  |
| --- | --- |
| **Authentication** | **Authorization** |
| Authentication is the process of identifying a user to provide access to a system. | Authorization is the process of giving permission to access the resources. |
| In this, the user or client and server are verified. | In this, it is verified that if the user is allowed through the defined policies and rules. |
| It is usually performed before the authorization. | It is usually done once the user is successfully authenticated. |
| It requires the login details of the user, such as user name & password, etc. | It requires the user's privilege or security level. |
| Data is provided through the Token Ids. | Data is provided through the access tokens. |
| **Example:** Entering Login details is necessary for the employees to authenticate themselves to access the organizational emails or software. | **Example:** After employees successfully authenticate themselves, they can access and work on certain functions only as per their roles and profiles. |
| Authentication credentials can be partially changed by the user as per the requirement. | Authorization permissions cannot be changed by the user. The permissions are given to a user by the owner/manager of the system, and he can only change it. |

**39. When to used Usablity Testing?**

**40 . What is the procedure for GUI Testing?**

* Testing the size, position, height, width of the visual elements
* Verifying and testing the error messages are displayed or not
* Testing different sections of the display screen
* Verifying the usability of carousel arrows
* Checking the navigation elements at the top of the page
* Checking the message displayed, frequency and content
* Verifying the functionality of proper filters and ability to retrieve results.
* Checking alignment of radio buttons, drop downs
* Verifying the title of each section and their correctness
* Cross-checking the colors and its synchronization with the theme