What is Exploratory Testing?

* Test design, execution and logging happen simultaneously
* Testing is often not recorded
* Makes use of experience, heuristics and test patterns
* Is not random testing but it is Adhoc testing with purpose of find bugs
* Is structured and rigorous
* Is not a technique but it is an approach. What actions you perform next is governed by what you are doing currently

What is traceability matrix?

* Traceability matrix is a table type document that is used in the development of software application to trace requirements. It can be used for both forward (from Requirements to Design or Coding) and backward (from Coding to Requirements) tracing. It is also known as **Requirement Traceability Matrix (RTM) or Cross Reference Matrix (CRM).**
* It is prepared before the test execution process to make sure that every requirement is covered in the form of a Test case so that we don't miss out any testing. In the RTM document, we map all the requirements and corresponding test cases to ensure that we have written all the test cases for each condition.

What is Boundary value testing?

* The boundary value is **the minimum (or maximum) value that is at the boundary**. The number 0 is the maximum number in the first partition, the number 1 is the minimum value in the second partition, both are boundary values.

What is Equivalence partitioning testing?

* Equivalence partitioning is a technique of software testing in which input data is divided into partitions of valid and invalid values, and it is mandatory that all partitions must exhibit the same behavior. If a condition of one partition is true, then the condition of another equal partition must also be true, and if a condition of one partition is false, then the condition of another equal partition must also be false. The principle of equivalence partitioning is, test cases should be designed to cover each partition at least once. Each value of every equal partition must exhibit the same behavior as other.

What is Integration testing?

* In this testing, units or individual components of the software are tested in a group. The focus of the integration testing level is to expose defects at the time of interaction between integrated components or units.

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.

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.
* It can be a white box, or Black-box testing depends on the requirements - particular lab environment and simulation of the actual environment required for this testing.

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.

What is component testing?

* Another type of software testing is **Component Testing**. It is used to test all the components separately as well as the usability testing; interactive valuation is also done for each specific component. It is further known as **Module Testing or Program Testing and Unit Testing.**

What is functional system testing?

* It is a type of software testing which is used to verify the functionality of the software application, whether the function is working according to the requirement specification. In functional testing, each function tested by giving the value, determining the output, and verifying the actual output with the expected value. Functional testing performed as black-box testing which is presented to confirm that the functionality of an application or system behaves as we are expecting. It is done to verify the functionality of the application.
* Functional testing also called as black-box testing, because it focuses on application specification rather than actual code. Tester has to test only the program rather than the system.

What is Non-Functional Testing?

* Non-functional testing is a type of software testing to test non-functional parameters such as reliability, load test, performance and accountability of the software. The primary purpose of non-functional testing is to test the reading speed of the software system as per non-functional parameters. The parameters of non-functional testing are never tested before the functional testing.
* Non-functional testing is also very important as functional testing because it plays a crucial role in customer satisfaction.
* For example, non-functional testing would be to test how many people can work simultaneously on any software.

What is GUI Testing?

* It is one of the unique [types of **software testing**](https://www.javatpoint.com/types-of-software-testing) that is frequently used to check the Graphical user interface features for the application or the software.
* Usually, the GUI testing is used to assesses a design of elements or features like:
* Text boxes
* Font size
* Font color
* Buttons
* Menus
* Links
* Layout
* Labels
* Text Formatting
* Lists
* Captions
* Icons
* Content

What is Adhoc testing?

* When a software testing performed without proper planning and documentation, it is said to be Adhoc Testing. Such kind of tests are executed only once unless we uncover the defects.
* Adhoc Tests are done after formal testing is performed on the application. Adhoc methods are the least formal type of testing as it is NOT a structured approach. Hence, defects found using this method are hard to replicate as there are no test cases aligned for those scenarios.
* Testing is carried out with the knowledge of the tester about the application and the tester tests randomly without following the specifications/requirements. Hence the success of Adhoc testing depends upon the capability of the tester, who carries out the test. The tester has to find defects without any proper planning and documentation, solely based on tester's intuition.

What is load testing?

* We are going to understand **load testing**, which is the important part of **Performance testing** and used to check the performance of the software by applying some load.
* And we also learn about its process, why we need to perform the load testing, the objective of load testing, example, various strategies of load Testing, advantage and disadvantage.

What is Stress Testing?

* In software testing, stress testing is an important part of performance testing under non-functional testing.
* **Stress Testing** is testing used to check the accessibility and robustness of software beyond usual functional limits. It mainly considers for critical software but it can also be used for all types of software applications.
* It is also known as ***Endurance Testing, fatigue testing*** or **Torture Testing**.
* The stress testing includes the **testing beyond standard operational size**, repeatedly to a **breaking point**, to get the outputs.
* It highlights the error handling and robustness under a heavy load instead of correct behavior under regular conditions.
* In other words, we can say that **Stress testing** is used to verify the constancy and dependability of the system and also make sure that the system would not crash under disaster circumstances.
* To analyses how the system works under extreme conditions, we perform **stress testing** outside the normal load.

**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.
* Structure-based testing technique is also known as ‘white-box’ or ‘glass-box’ testing technique because here the testers require knowledge of how the software is implemented, how it works.
* **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

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

* Black box testing techniques apply to all levels of testing, as well as functional and non-functional testing types. There are four main black box testing techniques:
* Equivalence partitioning.
* Boundary value analysis.
* Decision table testing.
* State transition testing.

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

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

**Mention what Big bang testing is?**

* Big Bang Integration 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.

**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.

**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.

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

**Testing shows the presence of defects**

* The test engineer will test the application to make sure that the application is bug or defects free. While doing testing, we can only identify that the application or software has any errors. The primary purpose of doing testing is to identify the numbers of unknown bugs with the help of various methods and testing techniques because the entire test should be traceable to the customer requirement, which means that to find any defects that might cause the product failure to meet the client's needs.
* By doing testing on any application, we can decrease the number of bugs, which does not mean that the application is defect-free because sometimes the software seems to be bug-free while performing multiple types of testing on it. But at the time of deployment in the production server, if the end-user encounters those bugs which are not found in the testing process.

**Exhaustive Testing is not possible**

* Sometimes it seems to be very hard to test all the modules and their features with effective and non- effective combinations of the inputs data throughout the actual testing process.
* Hence, instead of performing the exhaustive testing as it takes boundless determinations and most of the hard work is unsuccessful. So we can complete this type of variations according to the importance of the modules because the product timelines will not permit us to perform such type of testing scenarios.

**Early Testing**

* Here early testing means that all the testing activities should start in the early stages of the software development life cycle's **requirement analysis stage** to identify the defects because if we find the bugs at an early stage, it will be fixed in the initial stage itself, which may cost us very less as compared to those which are identified in the future phase of the testing process.
* To perform testing, we will require the requirement specification documents; therefore, if the requirements are defined incorrectly, then it can be fixed directly rather than fixing them in another stage, which could be the development phase.

**Defect clustering**

* The defect clustering defined that throughout the testing process, we can detect the numbers of bugs which are correlated to a small number of modules. We have various reasons for this, such as the modules could be complicated; the coding part may be complex, and so on.
* These types of software or the application will follow the **Pareto Principle**, which states that we can identify that approx. Eighty percent of the complication is present in 20 percent of the modules. With the help of this, we can find the uncertain modules, but this method has its difficulties if the same tests are performing regularly, hence the same test will not able to identify the new defects.

**Pesticide paradox**

* This principle defined that if we are executing the same set of test cases again and again over a particular time, then these kinds of the test will not be able to find the new bugs in the software or the application. To get over these pesticide paradoxes, it is very significant to review all the test cases frequently. And the new and different tests are necessary to be written for the implementation of multiple parts of the application or the software, which helps us to find more bugs.

**Testing is context-dependent**

* Testing is a context-dependent principle states that we have multiple fields such as e-commerce websites, commercial websites, and so on are available in the market. There is a definite way to test the commercial site as well as the e-commerce websites because every application has its own needs, features, and functionality. To check this type of application, we will take the help of various kinds of testing, different technique, approaches, and multiple methods. Therefore, the testing depends on the context of the application.

**Absence of errors fallacy**

* Once the application is completely tested and there are no bugs identified before the release, so we can say that the application is 99 percent bug-free. But there is the chance when the application is tested beside the incorrect requirements, identified the flaws, and fixed them on a given period would not help as testing is done on the wrong specification, which does not apply to the client's requirements. The absence of error fallacy means identifying and fixing the bugs would not help if the application is impractical and not able to accomplish the client's requirements and needs.

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

|  |  |  |
| --- | --- | --- |
| **Quality Assurance** | **Quality Control** | **Testing** |
| QA includes activities that ensure the implementation of processes, procedures and standards in context to verification of developed software and intended requirements. | It includes activities that ensure the verification of a developed software with respect to documented (or not in some cases) requirements. | It includes activities that ensure the identification of bugs/error/defects in a software. |
| Focuses on processes and procedures rather than conducting actual testing on the system. | Focuses on actual testing by executing the software with an aim to identify bug/defect through implementation of procedures and process. | Focuses on actual testing. |
| Process-oriented activities. | Product-oriented activities. | Product-oriented activities. |
| Preventive activities. | It is a corrective process. | It is a preventive process. |
| It is a subset of Software Test Life Cycle (STLC). | QC can be considered as the subset of Quality Assurance. | Testing is the subset of Quality Control. |

**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. |
| 16 | For every new build release smoke testing is carried out. | Sanity testing is carried out when in-depth testing is not possible because of short time. |

**Difference between verification and Validation?**

|  |  |  |
| --- | --- | --- |
| srno | **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. |

**Explain types of Performance testing**?

* Stress Testing
* Spike Testing
* Load Testing
* Endurance Testing
* Volume Testing
* Scalability Testing

**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**.

**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. |

**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.

**Explain the difference between Functional testing and NonFunctional 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 |

**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. |

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

|  |  |  |
| --- | --- | --- |
| **Test Case** | **Test Script** | **test scenarios** |
| Test Case is a step by step procedure to test any functionality of the software application/product. | Test Script is set of instructions or a short program to test any functionality of software application/product. | The test Scenarios is just a document that is detailed and provides details about the assessment method, testing process, preconditions, and anticipated output. The test scenarios are the ones based on the use situation and give one-line information ona  what to check. |
| Test Case is a manual approach of software testing. | Test Script is an automatic approach of software testing. | Test scenarios are one-liner statement, however,the  it is linked to a few test instances. |
| It is a set up that is used by the tester to test any specific function of the software product. | It is a program developed by the tester, intended to test any specific function of the software product. | These are high-level actions. |
| Point by point test case configuration encourages tester to test viably. | Automatic testing approach is beneficial for constant execution. | Writing the test scenario’s primary objective is an address end to get rid of functionality of a software program. |
| Test Cases are written manually. | Test Scripting is done by scripting format. | It will take less time as compared to test cases. |
| Test case is developed in form of templates. | Test script is developed in form of scripting. | Test scenarios are really easy to maintain due to their high-level design. |
| If the tester does not have a good understanding of how the program is used or about the recent risks to the program, then it will be difficult to use the test cases properly. | Active software projects frequently change. So testers have to make a continuous effort to update the scripts to match the changes of the new product. | The test scenario will help us in a way that is nimble of through the functionality. |
| Test Case is used in manual testing environment. | Test Script is used in automatic testing environment. | The test Scenarios tend to be work on the essential to “things to be tested”. |
| Test Cases are classified as delegated, positive, reusable, negative and UI test cases. | Test Script are characterized as manual test script and automation test scripts. | A lot fewer sources are sufficient to publish test circumstances in comparison with the test instances. |
| Test Case comprises of many test qualities like conditions, test data, environment, test suite id, name and others. | In Test Script different commands can be used to develop scripts which is used as a part of performing repeatable and regression testing. | Requires fewer resources and less time. |
| Requires more resources and time. | Requires less time for testing scripts. |  |
|  |  |  |

**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.

**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.

**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.

**Bug categories are…?**

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

**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.

**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)

**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. |

**When to used Usablity Testing?**

**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