**What is Exploratory Testing?**

* Exploratory testing is not random testing but it is ad-hoc testing with purpose of find bug
* Is highly teachable and manageable
* No documentation
* No plan
* Informal testing
* Tester should know application functionality
* Intension break to application and find out corner defects
* Testing is often not recorded
* Time consuming
* If there is any bug in app you will never know about it

**What is Boundary value testing?**

* It is the limits of valid ranges.
* It is a software testing technique in which tests are designed to include representative of boundary values in a range.
* Six parameters should all cover
* Ex:

13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28

---------------------!------------------------!-----------------

(Invalid) (Valid) (Invalid)

|  |  |  |
| --- | --- | --- |
| Min-1(Invalid) | (min, +min, -max, max)  (Valid) | Max+1(Invalid) |
| 17 | 18,19,20,21,22,23,24 | 25 |

**What is Equivalence partitioning testing?**

* Aim to target groups of input as equivalent selecting one representative input to test them all.
* Equivalence partitioning be used for all levels of testing
* Ex:

An OTP number

An opt number which contains only six digits,

Less or more than six digits will not be accepted, and the application will redirect the user to the error page.

|  |  |  |
| --- | --- | --- |
| Normal data | Equivalence class | Test data (Equivalence Partitioning) |
| 1 | Digits>=4 | 765419 (Invalid) |
| 2 | Digits>=5 | 3325 (Invalid) |
| 3 | Digits>=6 | 456789 (Valid) |
| 4 | Digits>=6 | 654126 (Valid) |

**What is Integration testing?**

* Integration Testing is a process where individual units are combined and tested as a group.
* There are two level of integration testing
* Component integration
* System integration
* There are two types of integration testing methods
* Top-down approach (stub)
* Bottom-up approach (drive)

**What is component testing?**

* Component testing also called as Unit testing.
* A minimal software item that can be tested in isolation. It means “A unit is the smallest testable part of software”.
* Component testing is smallest testable part of software
* Unit testing is performed by using white box testing methods
* The testing of individual software components
* Functional testing or non-functional
* Unit tests find problems early in the development cycle

**What is traceability matrix?**

* Traceability Matrix is a document that details the technical requirements for a given test scenario and its current state.

**What is functional system testing?**

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

**What is Non-Functional Testing?**

* Non Functional testing checks the performance, reliability, scalability and other non-functional aspect of the software system.

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

|  |  |  |
| --- | --- | --- |
| No | Functional Testing | Non-Functional Testing |
| 1 | is performed to determine the system behaviour as per the client requirement. | is the process to determine the system performance as per client expectations. |
| 2 | performed first with the help of Manual and Automation testing tools. | performed after functional testing with the effective tools required. |
| 3 | It is easy to perform manual testing as client requirements are the input in functional testing. | It is tough to do manual testing. |
| 4 | Requirements are the inputs to functional testing. | Performance parameters like speed, scalability are input to non-functional testing. |
| 5 | Functional testing describes what the product does. | Non-functional testing describes how good the product works. |
| 6 | Type of Functional testing are   * Black Box * White Box * Experience based * Smoke * Sanity * End to End | Type of Non-functional testing are   * Usability * Compatibility * GUI * Security * Performance * Stress * Load |

**What is GUI Testing?**

* Graphical user interface
* GUI is frontend testing
* Process of the user interface of an application
* It checks all the GUI elements like size, position, width, length.
* Check error messages are display correctly
* Check alignment of the text is proper
* Testing of the spelling
* The colour of the font
* Check that the image has good clarity
* UI elements like buttons, text box, check box, radio button etc...

**What is Ad-hoc testing?**

* Ad-hoc testing is an informal testing.
* This testing is usually an unplanned activity
* It is randomly done on any part of application.
* Tester should know the application functionality
* It is use after formal techniques have been applied.
* Three type of ad-hoc testing

buddy testing = developer and tester

Pair testing = tester and scriber

Monkey testing = randomly testing the product

**What is load testing?**

* Load testing is the process of subjecting a computer, server, network or application to a work level approaching the limits of its specifications.
* Gradually increase the load on app then check speed of the app
* Pre requisites for load testing

Server machines, Memory, Disk storage, Load machines configuration, network configuration, operating system, Server software.

**What is stress Testing?**

* Stress testing is the process of determining the ability of a computer, network, program or device to maintain a certain level of effectiveness under unfavourable conditions.
* Suddenly increase/decrease the load on the app and check the speed of app
* Types of stress testing

Application stress testing

Transactions stress

Systemic stress testing

Exploratory stress testing

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

* White box testing is testing based on an analysis of the internal structure of the Component or system.
* Also known as glass box
* Tester has knowledge of how the software is implemented, how it’s work.
* It is detailed investigation of internal logic and structure of the code

Types of White Box

* Statement Coverage
* Decision Coverage
* Condition Coverage

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

* Black box testing is that the testers have no knowledge of how the system or component is structured inside the box.
* Don’t show code
* Testers don’t have knowledge of how the system or component is structure inside the box

**Different Black box testing techniques**

* Equivalence Partitioning
* Boundary value analysis
* Decision table
* State transition
* Use Case

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

* Data Quality/Database Defects
* Critical Functionality Defects
* Functionality Defects
* Security Defects
* User Interface Defects

**Mention what big bang testing is?**

* Big bang integration testing all components or modules are integrated and tested as a single unit or whole
* All components are integrated at once, then tested
* Don’t wait for integration (finished before integration testing starts)
* Convenient for small system

**When should "Regression Testing" be performed?**

* Change in requirements and code is modified according to the requirement
* New feature is added to the software
* Defect fixing
* Performance issue fix
* Regression testing can be performed on a new build when there is a significant change in the original functionality.

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

* **1. Show presence of error** = important to design test which find as many defects as possible
* **2. exhaustive testing is impossible** = for this reason risk and priorities used to concentrate important aspects to test

Testing including all combinations of input and preconditions is not possible

* **3. Early testing** = testing activities should start early as possible. focused on defined objectives
* **4. Defects clustering** = a small number of modules contain most of the defects discovered pre-release testing most operating failure
* defects are not evenly spread in a system they are clustered
* **5. The pesticide paradox** = if the same tests are repeated over again the same set of test cases will no longer find any new defects
* If you keep running the same set of tests over and over again chances are no more new defects
* **6. Testing is context dependent** = different kind of sites are tested differently
* Ex: medical testing same testing in gaming that not possible in medical we all thing try to understand to small of small thing in gaming we should testing hardware, ram support to software that is context dependent
* **7. Absence of error fallacy** = all the test are done if there are no defects that not mean the software app are defects free or good in that cases your testing design does not match the software design

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

|  |  |  |
| --- | --- | --- |
| QA | QC | Tester |
| * Processes oriented activities * introducing the standards of quality to prevent the error and flaws in the product | * making sure the product corresponds to the requirement * and specs before it is released | * Deleting and solving software error and flaws * Activities which ensure the identification of bugs/error/defects in the software |
| * Focuses on process | * Focuses on product as whole | * Focuses on source code and design * Focuses on actual testing` |
| * Process oriented activities | * Product oriented activities | * Product oriented activities |
| * The team including the stakeholders | * The team | * Test engineers, developers |
| * Preventive activities | * Verification activities * It is a corrective process | * Detection activities * It is a preventive process |
| * Throughout the process | * Before the release | * At the tasting stage or along with the development 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?**

|  |  |  |
| --- | --- | --- |
| **No** | **Smoke Testing** | **Sanity Testing** |
| 1 | Check the critical functionality. | Check the new functionality. |
| 2 | It is done in initial stage. | It is done after 30 build |
| 3 | It checks the stability | It checks the sanity/rationality |
| 4 | Part of acceptance testing | Part of 30 build |
| 5 | General health check up | Advance health check up |
| 6 | It checks the system end to end | It checks only a particular function of entire system |
| 7 | Done by tester and developer | Done by tester |
| 8 | Part of acceptance testing | Part of regression testing |

**Difference between verification and Validation**

|  |  |  |
| --- | --- | --- |
| No | Verification | Validation |
| 1. | It is a process of checking, if a product is developed as per the specifications. | It is a process of ensuring that the product meets the needs and expectations of stakeholders. |
| 2. | It tests the requirements, architecture, design, and code of the software product. | It test the usability, functionalities, and reliability of the end product. |
| 3. | To ensure that the product is being built according to the requirements and design specification. | To ensure that the product actually meets the user’s needs. |
| 4. | A few verification methods are inspection, code review, desk-checking, and walkthroughs. | A few widely-used validation methods are black box testing, white box testing, integration testing, and acceptance testing. |
| 5. | It targets internal aspects such as requirements, design, architecture, database, and code. | It targets the end product that is ready to be deployed. |

**Difference between Priority and Severity.**

* Severity is the degree of impact that a defect has on the development or operation of a component or system.
* Priority is the level of importance assigned to an item.
* Severity relates to the standards of Quality.
* Priority relates to the scheduling of defects to resolve them in software.

**Explain types of Performance testing.**

* Types of performance testing

1. Load testing
2. Stress testing
3. Endurance testing
4. Spike testing
5. Volume testing
6. Scalability testing

**What is Error, Defects, Bug and Failure?**

* Error: error is a mistake in cording.
* Defect: Error found by tester is called defect.
* Bug: Defect accepted by development team then it is called bug.
* Failure: Build does not meet the requirement then it is failure.

**What is Bug Life Cycle?**

* Bug life cycle is also known as defect life cycle.
* It is a process in which defect goes through different stages in its entire life.
* Bug(Defect) Life Cycle

New

Assigned

---------- > Open --------------------------🡪 duplicated, rejected, deferred, not a bug

| Fixed

| Pending reset

Reopened ----🡪 Retest

Verified

Closed

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

|  |  |  |
| --- | --- | --- |
| No | SDLC | STLC |
| 1 | It is a sequence of different activities performed during the software development process. | It is a sequence of different activities performed during the software testing process. |
| 2 | Phases:   * Requirement * Analysis * Design * Implication * Testing * Maintenance | Phases:   * Planning * Test Case Development * Test Environment Setup * Test Execution * Test Closure |
| 3 | In SDLC, more number of members (Developer) are required for the whole process. | In STLC, less number of members (testers) are needed. |
| 4 | In SDLC, development team makes the plans and designs based on the requirements. | In STLC, testing team makes the plans and designs. |
| 5 | Goal of SDLC is to complete successful development of software. | Goal of STLC is to complete successful testing of software. |
| 6 | It helps in developing good quality software. | It helps in making the software defects free. |

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

* A test plan is document that describe the test scope, test strategy, objectives, deliverables and resources requirement
* the test plan strategy along with efforts and cost estimates for the project.
* Moreover, the resources, test environment, test limitations and the testing schedule are also determined.
* The Test Plan gets prepared and finalized in the same phase.
* Below information should be covered
* Overview
* Scope
* Inclusions
* Test environment
* Exclusions
* Test strategy
* Defect reporting procedure
* Roles/responsibilities
* Test schedule
* Pricing
* Tools
* Approvals
* Risk and mitigations
* Test deliverables
* Test tools selections
* Test formations
* Test effort estimation

**Bug categories are…**

* Bug categories are
* **Duplicate:** If the defect is repeated twice or the defect corresponds to the same concept of the bug it called duplicate
* **Rejected**: If the developer feels the defect is not a genuine defect then it changes the defect to “rejected.”
* **Differed**: If the present bug is not of a prime priority and if it is expected to get fixed in the next release it called differed
* **Not a bug**: If it does not affect the functionality of the application then it called not a bug

**Write agile manifesto principles**

* **Customer satisfaction through early and continuous software delivery** – Customers are happier when they receive working software at regular intervals, rather than waiting extended periods of time between releases.
* **Accommodate changing requirements throughout the development process** – The ability to avoid delays when a requirement or feature request changes.
* **Frequent delivery of working software** – Scrum accommodates this principle since the team operates in software sprints or iterations that ensure regular delivery of working software.
* **Collaboration between the business stakeholders and developers throughout the project** – Better decisions are made when the business and technical team are aligned.
* **Support, trust, and motivate the people involved** – Motivated teams are more likely to deliver their best work than unhappy teams.
* **Enable face-to-face interactions** – Communication is more successful when development teams are co-located.
* **Working software is the primary measure of progress** – Delivering functional software to the customer is the ultimate factor that measures progress.
* **Agile processes to support a consistent development pace** – Teams establish a repeatable and maintainable speed at which they can deliver working software, and they repeat it with each release.
* **Attention to technical detail and design enhances agility** – The right skills and good design ensure the team can maintain the pace, constantly improve the product, and sustain change.
* **Simplicity** – Develop just enough to get the job done for right now.
* **Self-organizing teams encourage great architectures, requirements, and designs** – Skilled and motivated team members who have decision-making power, take ownership, communicate regularly with other team members, and share ideas that deliver quality products.
* **Regular reflections on how to become more effective** – Self-improvement, process improvement, advancing skills, and techniques help team members work more efficiently.

**Explain the difference between Authorization and Authentication in Web testing.**

|  |  |  |
| --- | --- | --- |
| NO | Authorization | Authentication |
| 1. | Authorization determines what resources a user can access. | Authentication verifies who the user is. |
| 2. | Authorization isn’t visible to or changeable by the user. | Authentication is visible to and partially changeable by the user. |
| 3. | Accessibility to pages through permission not given | Accepting an invalid username and password |

**What is Alpha testing**

* It is always performed by the developer at the software development site
* Something it is also performed by the independent testing team
* Alpha testing is not open the market and public
* It conducted for software application and project
* It performed in a virtual environment
* Always performed in a virtual environment
* It always performed within the organization
* It is a form of acceptance testing
* It comes under the category of both white-box testing and black-box testing
* Alpha testing is definitely performed and carried out at the developing organization locations with the involvement of the developer

**What is beta testing**

* It always performed by the customers at their own site
* Not performed by the independent testing team
* Always open to the market and public
* Usually conducted for software product
* Performed in a real-time environment
* Always performed outside the organization
* It is also the form of acceptance testing
* Beta testing can be considered “pre-release”
* It is a only kind of black box testing
* It is a performed in real time environment
* Beta testing is always performed at the time when software product and project are marketed
* It is also considered as the user acceptance testing (UAT) which is done at customers or user area