* **What is Exploratory Testing?**

This may be the only type of technique used for low-risk systems, but this approach may be particularly useful under extreme time pressure –in fact, this is one of the factors leading to exploratory testing.

* **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, To protect against changes you should be able to trace back from every System component to the original requirement that caused its presence.

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

* **What is Equivalence partitioning testing?**

Equivalence partitioning is the process of defining the optimum number of tests by reviewing documents such as the Functional Design Specification and Detailed Design Specification, and identifying each input condition within function.

* **What is Integration testing?**

Integration testing is associated with the architectural design phase. Integration tests are performed to test the coexistence and communication of the internal modules within the system.

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

A factor that could result in future negative consequences; usually expressed as impact and likelihood.

* **What is Alpha testing?**

The developers at the software development site always perform it, Alpha Testing is not open to the market and public.

* **What is beta testing?**

Beta Testing is always performed at the time when software product and project are marketed, Beta Testing is always open to the market and public.

* **What is component testing?**

A level of the software testing process where individual units/components of a software/system are tested.

* **What is functional system testing?**

Functional testing verifies that each function of the software application operates in conformance with the requirement specification.

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

Non-Functional Testing checks the performance, reliability, scalalibility and other non-functional aspects of software system.

* **What is GUI Testing?**

Graphical User Interface involves checking the screens with the controls like menus, buttons, icons, and all types of bars – tool bar, menu bar, dialog boxes etc ..

* **What is Adhoc testing?**

Adhoc testing main aim of this testing is to find defects by random checking, Adhoc testing can be achieved with the testing technique called Error Guessing.

* **What is load testing?**

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.

* **What is stress Testing?**

Stress Testing is done in order to check when the application fails by reducing the system resources such as RAM, HDD etc. and keeping the number of users as constant.

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

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

1. **Statement coverage**: The statement coverage covers only the true conditions
2. **Decision coverage**: Decision coverage also known as branch coverage, It covers both the true and false conditions.
3. **Condition Coverage:** Condition coverage reports the true or false outcome of each condition & Condition coverage reports the true or false outcome of each condition.

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

Black-box testing have no knowledge of how the system or component is structured inside the box.

1. **Equivalence partitioning:** Aim is to treat groups of inputs as equivalent and to select one representative input to test them all.
2. **Boundary Value Analysis:** Boundary value analysis is a methodology for designing test cases that concentrates software-testing effort on cases near the limits of valid ranges.
3. **Decision Table:** Decision Table the relationships between the inputs and the possible outputs are mapped together.
4. **State Transaction Testing:** A black body box test design techquine in which test cases are designed to execute valid and invalid state transitions.

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

1. **Data Quality/Database Defects:** Deals with improper handling of data in the database.
2. **Critical Functionality Defects:** The occurrence of these bugs hampers the crucial functionality of the application.
3. **Functionality Defects:** These defects affect the functionality of the application.
4. **Security Defects:** Application security defects generally involve improper handling of data sent from the user to the application.
5. **User Interface Defects:** As the name suggests, the bugs deal with problems related to UI are usually considered less severe.

* **Mention what bigbang 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?**

1. Executed Test Cases are documented
2. All High prioritized bugs fixed and closed
3. Technical documents to be submitted followed by release Notes
4. Limitations
5. Any condition not specified in integration tests, apart from the confirmation of the execution of the design items is usually not tested

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

1. Change in requirements and code is modified according to the requirement.
2. New feature is added to the software
3. Defect fixing
4. Performance issue

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

1. **Testing show presence of Defect:** Testing can show that defects are present, but cannot prove that there are no defects.
2. **Exhausting Testing Is in Possible:** Testing everything including all combinations of in puts and preconditions is not possible.
3. **Early Testing:** Testing Activities should start early phase of life cycle so that we can identify and fix bug or defects before it erode in to code.
4. **Defect Clustering:** Defects are not evenly spread in a system they are clustered, In other words, most defects found during testing are usually confined to a small number of modules similarly, most operational failures of a system are usually confined.
5. **Pesticides Paradox:** If the same tests are repeated overland over again eventually the same set of test cases will no longer find any new defects, the test cases need to be regularly reviewed and revised, and new and different tests need to be written to exercise different parts of the software or system to potentially find more defects.
6. **Testing is Context Dependent:** Testing is done differently in different contexts, Different kinds of sites & application are tested differently.
7. **Absence of Errors Fallacy:** If the system built is unusable and does not fulfil the user’s needs and expectations then finding and fixing defects does not help. Even after defects have been resolved, it may still be unusable and/or does not fulfil the users’ needs and expectations.

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

|  |  |  |
| --- | --- | --- |
| **Quality Assurance (QA)** | **Quality Control (QC)** | **Testing** |
| **Process-oriented focuses on making the process of creating software better.** | **A product-oriented approach is a way to make sure the software meets all its requirements.** | **Testing the software system is about finding any mistakes or issues.** |
| **It works with the development process to help stop mistakes and ensure the software is of good quality.This means setting up and keeping standards, processes, procedures, and tools in place to ensure we’re consistently producing high-quality software.** | **It’s done after the development process and involves running test cases and seeing how the software reacts.** | **This usually happens after the software has been created, and it’s all about ensuring that the software’s quality is up to standard.** |
| **The goal is to keep improving our software development process for the best possible results.** | **The goal is to find any defects or errors in the software and fix them.** | **It involves running tests and looking at what comes out of them, finding any problems with the software, and ensuring that it does everything it’s supposed to do.** |

* **Difference between Smoke and Sanity?**

|  |  |
| --- | --- |
| **Smoke Testing** | **Sanity Testing** |
| Smoke Testing is performed to ascertain that the critical functionalities of the program is working fine | Sanity Testing is done to check the new functionality/bugs have been fixed |
| The objective of this testing is to verify the stability of the system in order to proceed with more rigorous testing | The objective of the testing is to verify the rationality of the system in order to proceed with more rigorous testing |
| This testing is performed by the developers or testers | Sanity testing in software testing is usually performed by testers |
| Smoke testing is usually documented or scripted | Sanity testing is usually not documented and is unscripted |
| Smoke testing is a subset of Acceptance testing | Sanity testing is a subset of Regression Testing |
| Smoke testing exercises the entire system from end to end | Sanity testing exercises only the particular component of the entire system |
| Smoke testing is like General Health Check Up | Sanity Testing is like specialized health check up |

* **Difference between verification and Validation**

|  |  |
| --- | --- |
| **Verification** | **Validation** |
| It includes checking documents, design, codes and programs. | It includes testing and validating the actual product. |
| Verification is the static testing. | Validation is the dynamic testing. |
| It does *not* include the execution of the code. | It includes the execution of the code. |
| 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. |
| 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. |
| 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. |
| Quality assurance team does verification. | Validation is executed on software code with the help of testing team. |
| The goal of verification is application and software architecture and specification. | The goal of validation is an actual product. |
| After a valid and complete specification the verification starts. | Validation begins as soon as project starts. |
| Verification is about process, standard and guideline. | Validation is about the product. |

* **Explain types of Performance testing**

1. **Load testing:** Load testing is the process that simulates actual user load on any application or website. It checks how the application behaves during normal and high loads. This type of testing is applied when a development project nears to its completion.
2. **Stress testing:** stress testing is to determine the limit, at which the system, software, or hardware breaks.
3. **Load Testing:** 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.
4. **Endurance testing:** It is done to determine if the system can sustain the continuous expected load for a long duration. Issues like memory leakage are found with endurance testing.
5. **Spike Testing:** we analyze the behavior of the system on suddenly increasing the number of users. It also involves checking if the application is able to recover after the sudden burst of users.
6. **Volume testing:** The volume testing is performed by feeding the application with a high volume of data. The application can be tested with a large amount of data inserted in the database or by providing a large file to the application for processing. Using volume testing, we can identify the bottleneck in the application with a high volume of data.

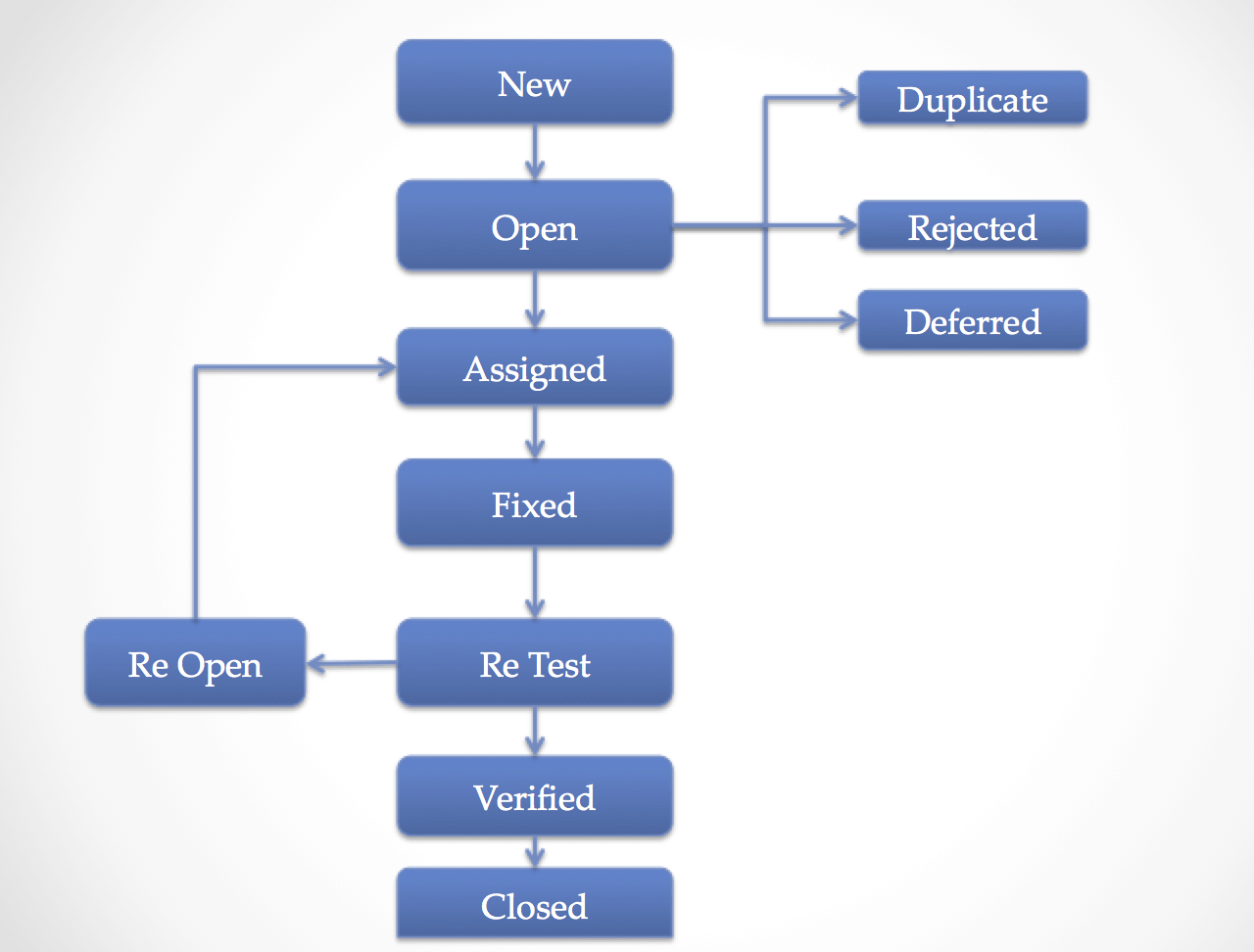
* **What is Error, Defect, Bug and failure?**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Basis** | **Error** | **Defect** | **Bug** | **Failure** |
| **Definition** | An Error is a mistake made in the code due to which compilation or execution fails, | A Defect is a deviation between the actual and expected output | A bug refers to defects which means that the software product or the application is not working as per the adhered requirements set | Failure is the accumulation of several defects that ultimately lead to Software failure and results in the loss of information in critical modules thereby making the system unresponsive. |
| **Raised by** | Developers and automation test engineers | The defect is identified by The Testers And is resolved by developers in the development phase of SDLC. | Test Engineers | The failure is found by the test engineer during the  development cycle of SDLC |
| **Different types** | Syntactic Error | Defects are classified as follows: | Logical bugs |  |
| UI screen error | **Based on Priority:** | Algorithmic bugs |
| Error handling error | High | Resource bugs |
| Flow control error | Medium |  |
| Calculation error | Low |  |
| Hardware error | **Based on Severity:** |  |
|  | Critical |  |
|  | Major |  |
|  | Minor |  |
|  | Trivial |  |
| **Reasons behind** | Error in code. | Receiving & providing incorrect input | Missing Logic | Environment variables |
| Inability to compile/execute a program | Coding/Logical Error leading to the breakdown of software | Erroneous Logic | System Errors |
| Ambiguity in code logic |  | Redundant codes | Human Error |
| Misunderstanding of requirements |  |  |  |
| Faulty design and architecture |  |  |  |
|  |  |  |  |

* **Difference between Priority and Severity**

|  |  |
| --- | --- |
| Priority | Severity |
| Defined by the impact on business. | Defined by the impact of a specific problem on any application’s functionality. |
| Category decided by developers or product owners. | Category decided by testers. |
| Deals with the timeframe or order to fix the defects. | Deals with the technical aspects of the application. |
| The priority value is subjective and may change after comparing with other defects. | The value does not change with time, it is fixed. |

* **What is Bug Life Cycle?**

****

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

|  |  |
| --- | --- |
| **Functional Testing** | **Non-functional Testing** |
| It verifies the operations and actions of an application. | It verifies the behavior of an application. |
| It is based on requirements of customer. | It is based on expectations of customer. |
| It helps to enhance the behavior of the application. | It helps to improve the performance of the application. |
| Functional testing is easy to execute manually. | It is hard to execute non-functional testing manually. |
| It tests what the product does. | It describes how the product does. |
| Functional testing is based on the business requirement. | Non-functional testing is based on the performance requirement. |

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

|  |  |
| --- | --- |
| **STLC** | **SDLC** |
| STLC is mainly related to software testing. | SDLC is a mainly related to software development. |
| It focuses only on testing. | SDLC phases are completed before the STLC. |
| STLC involves only five phases or steps. | Help in developing good quality software. |
| In STLC less of member (tester) are needed. | Goal of SDLC is to complete successful development of software. |
| Goal of the STLC is to completed successful testing of software. | SDLC more number of members are required for the whole process. |
| It helps in making the software defect free. | SDLC involves total six phases or steps. |
| STLC phases are performed after SDLC phases. | Besides development other phases like testing is also. |
| The STLC phases are completed after SDLC phases. | The SDLC phases are done before the STLC phases. |
| The STLC will helps to create the software bug-free. | The SDLC will help us to develop a good quality software product. |

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

|  |  |  |
| --- | --- | --- |
| **Test Scenarios** | **Test Cases** | **Test Script** |
| A scenario is any functionality that can be tested. it is also called test condition, or test possibility. | Test case involve the set of steps, conditions and inputs data which can be used performing the testing tasks. | A set of sequential instruction that detail how to execute a core business function. |

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

A Document describing the scope, approach, resources, and schedule of intended test activities.

* **What is priority?**

Priority is defined as the order in which a defect should be fixed. Higher the priority the sooner the defect should be resolved.

* **What is severity?**

Defect has on the software application under test. A higher effect of bug/defect on system functionality will lead to a higher severity level.

* **Bug categories are…**

Security bugs, Database bugs, Functionality bugs, UI based bug.

* **Advantage of Bugzila**

Bugzilla is an open-source issue/bug tracking system that allows developers effectively to keep track of outstanding problems with their produce.

* **Difference between priority and severity**

|  |  |
| --- | --- |
| **Priority** | **Severity** |
| Priority has defined the order in which the developer should resolve a defect | Severity is defined as the degree of impact that a defect has on the operation of the product |
| Priority indicates how soon the bug should be fixed | Severity indicates the seriousness of the defect on the product functionality |
| Priority status is based on customer requirements | Severity status is based on the technical aspect of the product |

* **What are the different Methodologies in Agile Development Model?**
* Scrum

Empowering the development team and advocates working in small teams

(say- 7 to 9members).

* Kanban

It provides a transparent way of visualizing the tasks and work capacity of a team.

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

|  |  |
| --- | --- |
| **Authorization** | **Authentication** |
| **While in authorization process, a the person’s or user’s authorities are checked for accessing the resources.** | **In the authentication process, the identity of users are checked for providing the access to the system.** |
| **While in this process, users or persons are validated.** | **In the authentication process, users or persons are verified.** |
| **The user authentication is visible at user end.** | **The user authorization is not visible at the user end.** |
| **The user authentication is identified with username, password, face recognition, retina scan, fingerprints, etc** | **The user authorization is carried out through the access rights to resources by using roles that have been pre-defined** |