1. **What is Software testing?**

**Answer**: Software testing is process use to identify the correctness, completeness and quality of develop computer software.

1. **What is traceability matrix?**

**Answer**: Test condition should be able to be linked back to their source in the test basis this is known as traceability.

1. **What is integration testing?**

**Answer**: Integration testing is a level of the software testing process where individual units are combined and tested as a group. Integration testing is done by a specific integration tester or test team.

1. **What is alpha testing?**

**Answer**: Alpha Testing is always performed by the developers at the software development site. It is not open to the market and public and always performed in Virtual Environment.

1. **What is beta testing?**

**Answer:** Beta Testing is always performed by the customers at their own site. It is always open to the market and public and performed in Real Time Environment.

1. **What is Component testing?**

**Answer:** Component testing is the testing of individual software components.

* The goal of unit testing is to isolate each part of the program and show that the individual parts are correct.
* Unit testing is performed by using the White Box Testing method
* It also known as Unit Testing, Module Testing or Program Testing
* Unit tests are typically written and run by software developer to ensure that code meets its design and behave as intended with debugging tool.

1. **What is Exploratory testing?**

**Answer:** Exploratory testing is a concurrent process where test design, execution and logging happen simultaneously.

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

**Answer:** This is one of the software testing technique in which the test cases are designed to include values at the boundary. If the input data is used within the boundary values limits, then it is said to be positive testing. If the input data is picked outside the boundary values limits then it is said to be negative testing,

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

**Answer:** This is a software testing technique which divides the input date into many partition. Values from each partition must be tested at least once. Partitions with valid values are used for positive testing. While partition with invalid values are used for negative testing.

1. **What is determines the level of risk?**

**Answer:** A factor that could result in future negative consequences usually expressed and impact likelihood.

1. **What is Functional System testing?**

**Answer:** Functional testing mainly involves black box testing and it is not concerned about the source code of application. This testing checks user interface, APIs, Database, Security, Client/Server communication and other functionality of the application under test. The testing is done either manually or using automation.

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

**Answer:** Non- Functional testing the attribute of a component or system that do not relate to functionality, **e.g**. reliability, efficiency, usability, interoperability, maintainability and portability.

1. **What is GUI testing?**

**Answer:** GUI testing is a software testing type that checks the Graphical User Interface of the software. The purpose of Graphical User Interface (GUI) testing is to ensure the functionality of software application work as per specification by checking screens and controls like menus, button, icons, etc.

1. **What is Ad hoc testing?**

**Answer**: Ad hoc testing is an informal or unstructured software testing type that aims to break the testing process in order to find possible defect or error at an early possible stage. Ad hoc testing is done randomly and it is usually an unplanned activity which does not follow any document and test design techniques to create test cases.

It also known as Random testing, Monkey testing, Error guessing, Gorilla testing, Negative testing.

1. **What is white box testing and list the type of white box testing?**

**Answer:** White box testing based on an analysis of the internal structure of the component or system.

White box testing is also known as Structure-based testing technique, open box, clear box, code-based, Transparent box or glass box testing technique because here the tester required knowledge of how the software is implemented, how it works.

**Type of white box testing**

1. Path Testing
2. Loop Testing
3. Condition Testing
4. **What is black box testing? What are the different black box testing techniques?**

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

Black box testing also known as Specification-based testing technique or input/output driven testing technique.

**Technique of Black Box Testing**

* Equivalence partitioning: Aim is to treat groups of inputs as equivalent and to select one representative input to test them all.
* Boundary value analysis: That concentrates soft. Testing efforts on cases near the limit of valid ranges.
* Decision tables: using the decision table the relationships between the input and the possible outputs are mapped together.
* State transition testing: Any system where you get different output for the same input, depending on what happened before

1. **Mention what are the Categories of defect?**

**Answer**: Category of Defect

* Data quality / Database Defect
* Critical Functionality Defect
* Functional Defect
* Security Defect
* User interface defect

1. **Mention what big bang testing is?**

**Answer**: In big bang integration testing all component or module is integrated simultaneously after which everything is tested as whole. Here all component are integrated together at once and then tested.

1. **What is purpose of exit criteria?**

**Answer**: Purpose of exit criteria to define when we stop testing either at the:

* End of all testing – i.e. product go live
* End of phase of testing

1. **When should “Regression Testing” be performed?**

**Answer**: Regression testing can be performed on a new build when there is a significant change in the original functionality. It ensure that the code still work even when the changes are occurring. Regression means Re-test those part of the application, which are unchanged.

1. **What is 7 key principle? Explain in detail?**

**Answer:** 1**.** Testing shows presence of Defects

2. Exhaustive Testing is Impossible

3. Early Testing

4. Defect Clustering

5. The Pesticide Paradox

6. Testing is Context Dependent

7. Absence of Errors Fallacy

1. **Testing shows presence of Defects:** Testingcan show that defects are present, but cannot prove that there are no defect. Testing reduces the probability of undiscovered defects remaining in the software but, even if no defect are found, it is not a proof of correctness
2. **Exhaustive Testing is Impossible:** Exhaustive testing usually tests and verifies all functionality of a software application while using both valid and invalid input and pre-conditions. No matter how hard you try, testing everything is pretty much impossible. The inputs and outputs alone have an infinite number of combination, so it is 100% not possible to test an application from every angle.
3. **Early Testing:** To find the defect in the software, early test activity shall be started. The defect detected in the early phases of SDLC will be very less expensive. For better performance, software testing will start at the initial phase i.e. testing will perform at the requirement analysis phase.
4. **Defect Clustering:** During testing, it is observed that most of the defect found are related to a small number of modules. There might be multiple reasons for this like module may be complex, coding related to such module may be complicated, etc.

This is the Pareto principle of software testing where you can detect 80% of the problems in 20% of the modules.

1. **The Pesticide Paradox:** Repeating the same test cases, again and again, will not find new bugs. So it is necessary to review the test cases and add or update test cases to find new bugs.
2. **Testing is Context Dependent:** The testing approach depends on the context of the software development. Different types of software need to perform different types of testing. For example, the testing of the e-commerce site is different from the testing of the android application.
3. **Absence of Errors Fallacy:** If a built software is 99% bug-free but it does not follow the user requirement then it is unusable. It is not only necessary that software is 99% bug-free but it is also mandatory to fulfill all the customer requirement
4. **Difference between QA v/s QC v/s Tester**

|  |  |  |
| --- | --- | --- |
| **QA** | **QC** | **Tester** |
| QA is subset of SDLC | QC is a subset of QA | Testing is subset of QC |
| QA is a process oriented | QC is a product oriented | Tester is a product oriented |
| QA is a preventive Activity | QC is a Corrective activity | Testing is a corrective activity |
| Focus on process | Focus on required quality | Focus on actual testing |
| Verifies the quality | Validates the quality | Validates the quality |
| The whole project team is involved | Testing team is involved | Testing team involved |
| Make sure that right thing are being done | Make sure the things are being done | Evaluates the result of done things |
| Proactivity Process | Reactive process | Reactive Process |

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

|  |  |
| --- | --- |
| **Smoke** | **Sanity** |
| Smoke testing is performed to ascertain that the critical functionality of the program working fine | Sanity testing is done to check the new functionality / bugs have been fixed |
| This testing is performed by the developers and testers | Sanity testing is usually perform by tester |
| Smoke testing is usually documented or scripted | Sanity testing is usually not documented and is unscripted |
| Smoke testing is subset of Regression testing | Sanity testing is subset of Acceptance testing |
| Smoke testing exercise 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 |

**24. Difference between verification and validation**

|  |  |
| --- | --- |
| **Verification** | **Validation** |
| Involves all static technique | Involves all dynamic technique |
| It is development level | It is testing level |
| It is perform on paper | It is perform on Component |
| It does not involve executing the code | It always involves executing the code |
| Are we building the product right…? | Are we building the right product…? |
| It finds bug early in the development cycle | It can find bugs that the verification process can’t catch |
| Its human based checking of document and files | It is computer based executing of program |
| It comes before validation | It comes after verification |
| Ex: levels like   * Business Analysis * System Design * Architecture Design (Technical specification)   Program specification | Ex: Test levels like   * Unit Testing * Integration Testing * System Testing * Acceptance Testing |
| Activity is review, Walkthrough, inspection | Activity is testing |

1. **Explain types of Performance testing?**

**Answer:**

* **Load Testing:** Confirms that the system can handle the required number of users and still operate at a high level of performance. This ensure that there is no day to day issue in performance
* **Stress Testing:** It involves testing a product under extreme workload to see whether it handle high traffic or not. The objectives is to identify the breaking point of a software product.
* **Endurance Testing:** it is performed to ensure the software can handle the expected load over a long period of time.
* **Spike Testing:** Spike testing is to determine the behavior of the system under sudden increase of load (a large number of users) on the system.
* **Volume Testing:** Check that the software can handle and process a large amount of data at once without breaking , slowing down, or losing any information
* **Scalability Testing:** Inscalability testing, software application’s effectiveness is determine in scaling up to support an increase in user load. It helps in planning Capacity addition to your software system

1. **What is Error, Defect, Bug and Failure?**

**Answer:** Error: a mistake in coding is called error.

Defect: Error found by tester is called defect

Bug: Defect accepted by developer team than it is called bug

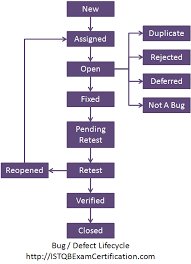
Failure: build does not meet the requirement then it is failure

1. **Difference between Priority and Severity?**

|  |  |
| --- | --- |
| **Priority** | **Severity** |
| Priority is associated with scheduling | Severity is associated with functionality |
| Category decided by developers or product owners | Category decided by testers. |
| It indicate how soon the bug should be fixed | It indicate the seriousness of defect |
| Priority of defect is consultation with the client | QA engineer determine the severity level |
| Priority is driven by business level | Severity is driven by functionality |
| Priority levels are: Critical, high, medium, low | Severity levels are: Critical, major, minor, moderate & Cosmetic |
| Its value is subjective | Its value is objective. |
| Its value changes from time to time. | Its value doesn’t change from time to time. |
| Priority is related to scheduling to resolve the problem. | Severity is related to the quality standard. |
| Priority is a parameter to decide the order in which defects should be fixed. | Severity is a parameter to denote the impact of a particular defect on the software. |

1. **What is bug life Cycle?**

**Answer:** The duration or time span between the first time defects is found and the time that it is closed successfully, rejected, postponed or deferred is called as Defect life cycle /bug life cycle.



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

|  |  |
| --- | --- |
| **Functional Testing** | **Non-Functional testing** |
| Functional testing is performed using the functional specification provided by the client and verifies the system against the functional requirement | Non-functional testing check the performance, reliability, scalability and other non-functional aspects of the software system |
| It verifies the operations and actions of an application. | 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 |
| It checks the operations and actions of an Application. | It checks the behavior of an application |
| Functional testing is executed first | Non-functional testing should be performed after functional testing |
| Types of Functional testing are  ● Unit Testing  ● Smoke Testing  ● Sanity Testing  ● Integration Testing  ● White box testing  ● Black Box testing  ● User Acceptance testing  ● Regression Testing | Types of Non-functional testing are  ● Performance Testing  ● Load Testing  ● Volume Testing  ● Stress Testing  ● Security Testing  ● Installation Testing  ● Compatibility Testing  ● Migration Testing |
| Example: A Login page must show textboxes to Enter the username and password. | Example: Test if a Login page is getting loaded in 5 seconds. |

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

|  |  |
| --- | --- |
| **STLC** | **SDLC** |
| STLC stands for Software Testing Life Cycle | SDLC stands for Software Development Life Cycle |
| STLC mainly related to software testing | SDLC is mainly related to software development |
| It focuses only on testing the software | Beside Development other phase like testing is also include |
| STLC involves only 5 phases or Steps | SDLC involve total 6 phases or steps |
| IN STLC testing team (test lead or test architect) makes the plan and design | Development team makes the plans and design based on the requirement |
| Focuses on testing a product. | Focuses on building a product. |
| A child of SDLC process. | A parent process. |
| STLC phases start after SDLC phases are completed. | SDLC phases are completed before testing. |
| Ensuring the product is working as expected. | Building a product as user requirement. |
| Goal of STLC is to complete successful testing of software | Goal of SDLC is to complete successful development of software |
| It helps in making the software defect free | It helps in developing good quality software |

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

|  |  |  |
| --- | --- | --- |
| **Test Scenarios** | **Test Case** | **Test Script** |
| A Test Scenario is any functionality that can be tested. It is also called Text Condition or Test Possibility | Test Case involve the set of steps, condition and inputs which can be used while performing the testing task | A set of sequential instruction that detail how to execute a core business function |
| Is more focused on what to test | Is focused on what to test and how to test | Is focused on the expected result |
| Take less time and fewer resources to create | Required more resources and time | Requires less time for testing but more resources for script creating and updating |
| The main task is to check the full functionality of software application | The main task is to verify compliance with the applicable standards, guidelines, and customer requirement | The main task is to verify that nothing is skipped, and the result are true as the desired testing plan |
| Allow quickly assessing the testing scope | Allows detecting errors and defects | Allows carrying out an automatic execution of test cases |
|  |  |  |

1. **Explain what test plan is? What is information that should be covered?**

**Answer:** A test plan documents the strategy that will be used to verify and ensure that a product or system meets its design specifications and other requirements. A test plan is usually prepared by or with significant input from test engineers.

Information that should be covered

* Introduction of the test plan document.
* Assumptions when testing the application
* List of test cases included in testing the application.
* List of features to be tested.
* What sort of approach to use when testing the software.
* List of deliverables that need to be tested.
* The resources allocated for testing the application.
* Any risk involved during the testing process.
* A schedule of task and milestone as testing is started.

1. **What are the difference methodologies in Agile Development model?**

**Answer:** Methodologies in agile Development model

* **Scrum**: scrum is a framework for effective collaborations among teams working on complex products. Scrum is a type of agile technology that consists of meetings, roles, and tools to help teams working on complex projects collaborate and better structure and manage their workload
* **eXtreme** **Programming**: this is light weight agile testing methodology I which development and testing happen in parallel. Business requirement are gathered in terms of stories. All those stories are stored in a place called parking lot.in this type of methodology release are based on the shorter cycles called iterations with span of 14days time period. Each iteration include phases like coding , unit testing and system testing where at each phases some minor or major functionality will be built in application.

1. **Explain the difference Authorization and Authentication in web testing. What are the common problem faced in web testing?**

|  |  |
| --- | --- |
| **Authorization** | **Authentication** |
| Validates access permission | Verify user identities |
| In this, it is verified that if the user is allowed through the defined policies and rules. | In this, the user or client and server are verified |
| Performed after Authentication | Performed before authorization |
| Data is transmitted through access token | Data is transmitted through token IDs |
| It requires the user's privilege or security level. | It requires the login details of the user, such as user name & password, etc. |
| **Example**: After employees successfully authenticate themselves, they can access and work on certain functions only as per their roles and profiles. | **Example**: Entering Login details is necessary for the employees to authenticate themselves to access the organizational emails or software. |
| 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. | Authentication credentials can be partially changed by the user as per the requirement. |

**Problem faced in Web testing**

* Cross Browser Compatibility
* Responsiveness
* Cross-Device Compatibility
* Integration Testing
* Security
* Performance Testing
* Dynamic environment
* Very short development time