1. **What is Exploratory Testing?**

**Ans.** – Exploratory testing is an approach to software testing that is often described as simultaneous learning, test design and execution. It focuses on discovery and relies on the guidance of the individual tester to uncover defects that are not easily covered in the scope of other sets.

1. **What is Traceability Matrix**?

**Ans.** – A traceability matrix is a document that details the technical requirement for a given test scenario and its current state. It helps the testing team understand the level of testing that is done for a given product. The traceability process itself is used to review the test cases that were defined for any requirement.

1. **What is Boundary Value Analysis?**

**Ans**. – Boundary value analysis involves testing the input parameters of a software system, focusing on the boundaries of their defined ranges. The purpose of this technique is to identify defects that may arise due to the boundary conditions, such as minimum and maximum limits or overflow/underflow situations.

1. **What is Equivalence Partitioning Testing?**

**Ans.** – Equivalence partitioning is technique that divides the input domain of a system into partitions or classes that are expected to produce the same output or behavior. For example, if a system accepts an integer between 1 and 100 as input, you can create four partitions: 1-10, 11-50, 51-99 and 100. Each partition represents a set of equivalent values that should trigger the same response from the system. You can select one value from each partition as a test case, rather than testing all 100 values.

1. **What is Integration Testing?**

**Ans.** – Integration testing is the phase in software testing in which the whole software modules is tested or if it consists of multiple software modules they are combined and then tested as a group. Integration testing is conducted to evaluate the compliance of system or component with specified functional requirement.

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

**Ans. -** Several factors determine the level of risk in software testing, including the complexity of the software, the criticality of the system being tested the experience and expertise of the testing team, the thoroughness of test coverage, the quality of the development environment, and the potential impact of defects on end users or business operations.

1. **What is Alpha Testing?**

**Ans.** – Alpha testing is the initial phase of software testing where developers test the software in a controlled environment before releasing it to externals testers or customers. It is always performed in virtual environment and within the organization.

1. **What is Beta Testing?**

**Ans.** – Beta testing is the final round of testing before releasing a product to a wide audience. The objective is to uncover as many bugs or usability issues as possible in this controlled setting. It is performed in Real- time environment.

1. **What is Component Testing?**

**Ans. -** Component Testing is a level of the software testing process where individual components of a software/system are tested. The purpose is to validate that each unit of the software performs as designed. Components tests are typically written and run by software developers to ensure that code meets its design and behaves as intended with debugging tool.

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

**Ans. -** Functional System Testing is a requirement that specifies a function that a system or system component must perform A Requirement may exist as a text document and/or a model. There is two types of Test Approach –

1. Requirement Based Functional Testing

2. Process Based Testing

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

**Ans.** – Non – functional testing is a type of software testing that verifies non functional aspects of the product, such as performance, stability, and usability. Whereas functional testing verifies whether or not the product does what it is supposed to, non functional testing verifies how well the product performs. For example- the website page should load in 3 seconds with the total numbers of simultaneous user <5 thousand. The system should be able to handle 20 million users without performance deterioration.

1. **What is GUI testing?**

**Ans.** - Graphic User Interface Testing (GUI) is the process of ensuring proper functionality of the graphical user interface for a specific application. GUI testing involves checking the screens with the controls like menus, buttons, icons, and all types of bars – tool bar, menu bar, dialog boxes and windows etc. For example- Microsoft window, Mac OS, GNOME Shell for desktop environment and Android, Apple’s iOS, Blackberry OS, Windows 10 and Firefox OS for smart phone.

1. **What is Adhoc Testing?**

**Ans.** – Adhoc testing is a software testing technique performed without any specific test plan or predefined set of steps. Instead, testers use their intuition, experience and creativity to identify defects and issues that more formal testing methods may not find. Main aim of this testing is to find defects by random checking. Adhoc testing can be achieved with the testing technique called Error Guessing.

1. **What is Load Testing?**

**Ans.-** 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. Testing an application under heavy loads, such as testing of a web site under a range of loads to determine at what point the system’s response time degrades or fails.

1. **What is Stress Testing?**

**Ans.** - Stress testing is used to test the stability & reliability of the system. This test mainly determines the system on its robustness and error handling under extremely heavy load conditions. It even tests beyond the normal operating point and evaluates how the system works under those extreme conditions. Stress Testing is done to make sure that the system would not crash under crunch situations. Stress testing is also known as Endurance testing.

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

**Ans.** – White box testing is a software testing technique that is based on the application’s internal code structure. Following are the types of white box testing –

* Unit testing
* Integration testing
* Code coverage

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

**Ans.** - : Testing, either functional or non-functional, without reference to the internal structure of the component or system called black box testing. Following are the black box testing techniques –

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

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

**Ans.** – 1.Data Quality/Database Defects: Deals with improper handling of data in the database. Examples: Values not deleted/inserted into the database properly. Improper/wrong/null values inserted in place of the actual values.

2. Critical Functionality Defects: The occurrence of these bugs hampers the crucial functionality of the application. Examples: - Exceptions

3. Functionality Defects: These defects affect the functionality of the application. Examples: All JavaScript errors Buttons like Save, Delete, Cancel not performing their intended function.

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

**Ans.-** Big bang integration testing is a testing approach where all components or modules are integrated and tested as a single unit. This is done after all modules have been completed and before any system level testing is performed.

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

**Ans.** - 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. Executed Test Cases are documented. All High prioritized bugs are fixed and closed. Technical documents have been submitted.

1. **When should "Regression testing" be performed?**

**Ans.** – Regression testing should be performed whenever there are changes or updates to a software application. This includes after bug fixes, feature addition, or code modifications. The goal is ensure that new changes haven’t adversely affected existing functionality.

1. **What is7 key principles? Explain in details.**

**Ans. –** Following are 7 key principles -

1. Testing shows presence of defects- Testing can show that defects are present, but cannot prove that there are no defects. As we find more defects, the probability of undiscovered defects remaining in a system reduces. However Testing cannot prove that there are no defects
2. Exhaustive testing is impossible - . Testing everything including all combinations of inputs and preconditions is not possible. So, instead of doing the exhaustive testing we can use risks and priorities to focus testing efforts.
3. Early Testing – Testing activities should start as early as possible in the software or system development life cycle, and should be focused on defined objectives. Testing activities should start as early as possible in the development life cycle.
4. Defect Clustering – A small number of modules contain most of the defects discovered during pre-release testing, or are responsible for the most operational failures. Defects are not evenly spread in a system.
5. The Pesticide Paradox – If the same tests are repeated over and over again, eventually the same set of test cases will no longer find any new defects. To overcome this “pesticide paradox”, 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 basically context dependent. Testing is done differently in different contexts. Different kinds of sites are tested differently.
7. Absence of Errors Fallacy - If the system built is unusable and does not fulfill the user’s needs and expectations then finding and fixing defects does not help.
8. **Difference between QA/QC/Tester.**

**Ans.-**

|  |  |  |
| --- | --- | --- |
| Quality Assurance (QA) | Quality Control (QC) | Testing |
|  |  |  |
|  |  |
| Process-oriented focuses on making the process of creating software better. | 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. |

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

**Ans. –**

|  |  |
| --- | --- |
| 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 to verify the stability of the system in order to proceed with more rigorous testing. | The objectives of the testing is to verify the rationality of the system in order to proceed with more rigorous testing |
| Smoke testing is performed by the developers or testers. | Sanity testing in software testing is usually by testers. |
| Smoke testing is usually documented or scripted. | Sanity testing is usually not documented and is unscripted. |
| Smoke testing is 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. |

1. **Difference between Verification and Validation.**

**Ans.**–

|  |  |
| --- | --- |
| Verification | Validation |
| The process of evaluating work-products (not the actual final product) of a development phase to determine whether they meet the specified requirements for that phase. | The process of evaluating software during or at the end of the development process to determine whether it satisfies specified business requirements. |
| To ensure that work products meet their specified requirements. | To demonstrate that the product fulfills its intended use when placed in its intended environment. |
| It is done by Developer. | It is done by Tester. |
| Concerned with phase containment of errors. | Aim is to make final product error free. |
| It involves review, inspection, unit testing and integration testing. | It involves system testing. |

1. **Explain type of performance testing.**

**Ans.** - Software performance testing is a means of quality assurance. It involves testing software applications to ensure they will perform well under their expected workload.

Types of performance testing-

* + 1. 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.
    2. Stress testing: Performed under heavy load like putting large number beyond storage capacity, complex database queries, continuous input to system or database load.
    3. Endurance testing: This type of performance test is meant to test the system performance for longer period of time, in order to reveal other type of problems.
    4. Spike testing: Spike testing is a type of performance testing in which an application receive a sudden and extreme increase or decrease in load.
    5. Volume testing: Volume testing is a type of non- functional testing that refers to testing that refers to testing the data load and capabilities of a product.
    6. Scalability testing: It tests how the system is going to perform during a sudden spike or fall of user request loads.

1. **What is Error, defects, bug and failure ?**

**Ans. -** A mistake in coding is called error, error found by tester is called defect, defect accepted by development team then it is called bug, build does not meet the requirements then it is failure.

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

**Ans. -**

|  |  |
| --- | --- |
| **Priority** | **Severity** |
| Priority has specified the order in which the developer should fix a defect. | Severity is specified as the degree of impact that a defect has on the operation of the product. |
| Priority means how soon the bud should be fixed. | Severity means the seriousness of the defect in the product functionality. |
| Priority of defects is decided in discussion with the manager/client. | The test engineer determines the severity level of defect. |
| It is driven by business value. | It is driven by functionality. |
| Priority status is established on customer requirements. | Severity status is established on the technical aspect of the product. |

1. **What is Bug life cycle?**

**Ans.** – 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 ‘Bug Life Cycle’. Following are stages in bug life cycle-

1. New

2. Assigned

3. Open

4. Fixed

5. Pending retest

6. Retest

7. Verified

8. Reopen

9. Closed

10. Duplicate

11. Rejected

12. Deferred

13. Not a bug

1. **Explain the difference between Functional testing and non- functional testing.**

**Ans. -**

|  |  |
| --- | --- |
| **Functional Testing** | **Non – Functional Testing** |
| Test the functionality of the software. | Test the non – functional aspect or readiness of the software including performance, usability, reliability. |
| It has to be done before non functional testing. | It will be done after functional testing. |
| It can be done manually, through test cases can be automated once application is stable | It is hard to do it manually. It usually need already existing applications to measure and test application performance. |
| Types of functional testing includes unit testing, smoke testing, integration testing, regression testing, system testing, user acceptance testing. | Types of non- functional testing includes volume testing, load testing, stress testing, scalability testing, security testing. |
| Test data can be prepared using the business or functional requirements of the application. | Test data can be prepared using the performance requirement of the application. |
| Testing tools used for functional testing includes UFT(previously QTP), Selenium, Telerik test studio, micro focus, test complete, IBM rational | Testing tools used for non- functional testing includes j Meter, load runner, web load, neo load, load complete. |

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

**Ans.** -

|  |  |
| --- | --- |
| Software Testing Life Cycle | Software Development Life Cycle |
| STLC is related to software testing. | The SDLC is primarily concerned with software development. |
| STLC ensures that anything we produce meets customer needs and that the products are of high quality. | SDLC ensures that we are building the correct thing in the correct manner. |
| STLC is concerned with both the development and testing processes, but it is primarily concerned with the testing process. | SDLC assures that we deliver high- quality software which is as per client needs. |
| In STLC QA team analysis all the requirement from the requirement document and create a system test plan. | In SDLC business analyst gather all project – related requirements from a stockholder and create a development plan. |
| In software testing life cycle, the Test Architect or test manager create a strategy to test a software application. | In SDLC, the life cycle development team create the high- level design of project based on clients requirement. |

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

**Ans. –** A test plan is a detailed document outlining the approach, objectives, resources and schedule for testing a software application. It typically includes information such as test objectives, scope, test strategies, test environment, test deliverables and exit criteria. It should cover aspects like test scenario, test cases, test data, risk assessment and responsibilities of team members involved in testing. The goal is to ensure that all aspects of the software are thoroughly tested to meet quality standards and fulfill user requirements.

1. **What is priority?**

**Ans.** – Priority refers to the order in which defects should be addressed, fixed and tested based on their importance and impact on project’s goals. Priority levels are typically categorized in to several levels such as immediate, high, medium and low.

1. **What is severity?**

**Ans.** – The impact of the bug on the application is known as severity. Severity is basically a parameter that donates the total impact of a given defect on any software.

1. **What are the bug categories?**

**Ans.** – Software bugs can be classified into multiple categories based on their nature and impact. These categories include –

* Functional bugs
* Compatibility bugs
* Usability bugs
* Unit Level Bugs
* Logical Bugs
* Security

1. **Advantages of Bugzilla.**

**Ans.** – Following are the advantages of bugzilla:

1. Effective Bug Tracking – Bugzilla provides a centralized platform for tracking and managing software defects.
2. Customization and Flexibility –Bugzilla is highly customizable, allowing organizations to tailor it to their specific needs.
3. Integrated Collaboration – Bugzilla supports collaborations among team members by providing a platform for discussions and attachments related to each bug report.
4. Automation and Workflow Management – Bugzilla offers automation features that streamline bug tracking processes.
5. Comprehensive Reporting and Analysis – Bugzilla includes robust reporting tools that enable users to generate a variety of reports, charts and graphs related to bug statistics and project progress.
6. **What are the different Methodologies in Agile Development Model?**

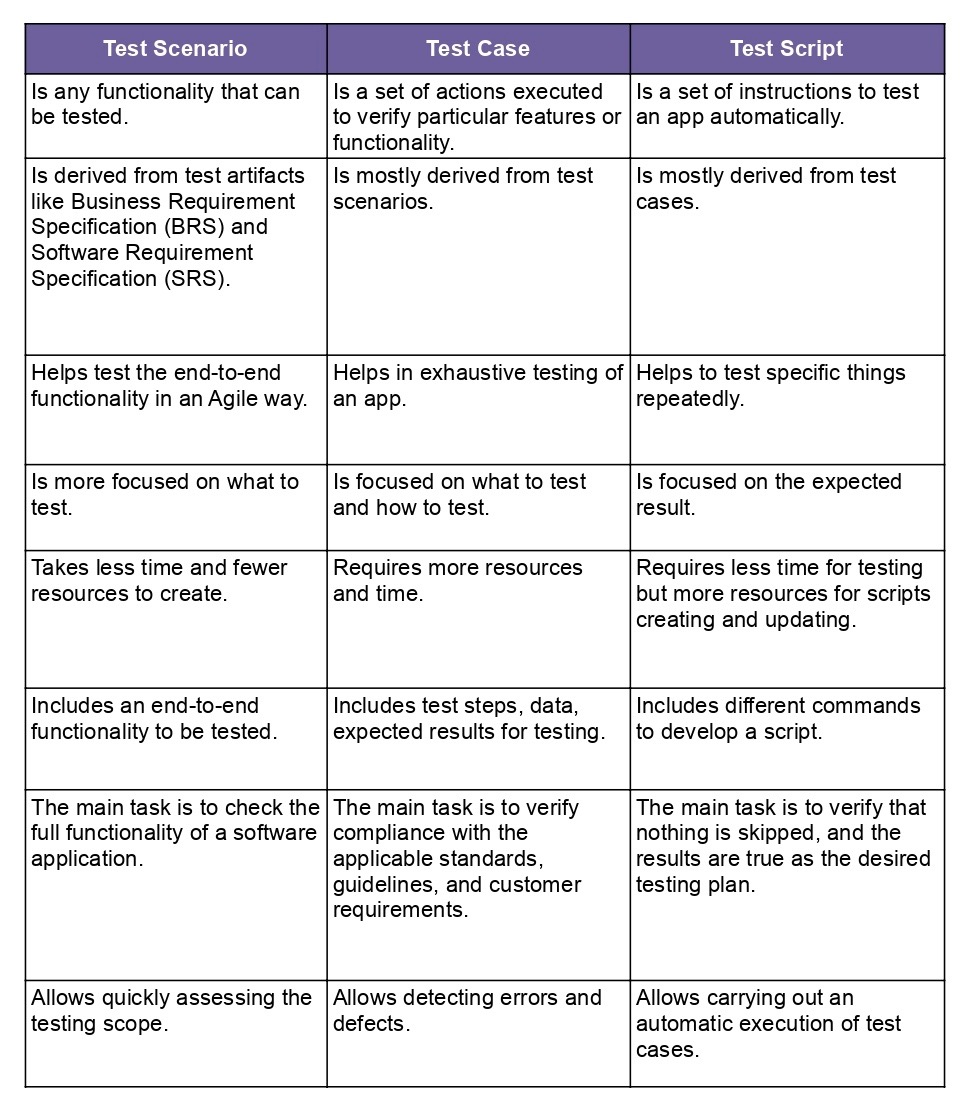
**Ans.** – Following are the different methodology in agile development model -

* Scrum
* Extreme Programming (XP)
* Feature-Driven Development (FDD)
* Adaptive Software Development (ASD)
* Kanban

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

**Ans**.- One of the major challenges in web applications testing is achieving cross browser compatibility. Web applications need to function correctly across different web browsers such as Chrome, Firefox, Safari and Microsoft Edge. Following are difference between authorization and authentication in web testing-

|  |  |
| --- | --- |
| Authorization | Authentication |
| Access to resources is granted or denied based on verified identity. | Verifies the identity of a user or device. |
| Work based on assigned roles or permission by admin or security user. | Work based on password, OTP, biometrics, security questions etc. |
| Authorization occurs during the session as the user request access to different resources. | Takes place at the beginning of a session. |
| It is the second step, and it is essentials for maintaining the confidentiality of an application. | It is the first step in security, and it is essentials for maintaining the integrity of an application. |
| Entire authorization process take place in the background. | Parts of authentication process is visible to users. |

1. **What are difference between test scenario, test cases and test script?**

Ans.-

1. **To create HLR & Test Case of ( Instagram, Facebook ) only first page.**

**Ans. –** [**https://docs.google.com/spreadsheets/d/18wKntS3J\_QynKMKDYFV77gLjURA9J4g8JezHqE8Dfzs/edit?usp=sharing**](https://docs.google.com/spreadsheets/d/18wKntS3J_QynKMKDYFV77gLjURA9J4g8JezHqE8Dfzs/edit?usp=sharing)

[**https://docs.google.com/spreadsheets/d/1M2ulLo1kYz8Xf3nyuFXr0ysWX4jOsao73gmmcQyZVA4/edit?usp=sharing**](https://docs.google.com/spreadsheets/d/1M2ulLo1kYz8Xf3nyuFXr0ysWX4jOsao73gmmcQyZVA4/edit?usp=sharing)

1. **When to used Usability Testing?**

**Ans. -** Usability testing should be conducted on the current iteration of a product before beginning any new design work, after you’ve begun the strategy work around a brand new site or app. This will quickly identify areas for opportunity, and reduce the amount of assumptions your design team will make with regard to what the user wants. Additionally, after the usability test analysis, the team should have the ability to pinpoint the steps needed to achieve the project goals with as little disruption as possible.

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

**Ans. -** Following are the procedure for GUI testing –

1. Testing the size, position, height, width of the visual elements.
2. Verifying and testing the error message are displayed or not.
3. Testing different section of the display screen.
4. Verifying the usability of carousel arrows.
5. Checking the message displayed, frequency and content
6. Verifying the functionality of proper filters and ability to retrieve results.
7. Checking alignment of radio button, drop downs.
8. Verifying the title of each section and their correctness.
9. Cross- checking the colors and its Synchronization with the theme.
10. **Test Scenario.**

**Ans**. - <https://docs.google.com/spreadsheets/d/18Q342iAuy2zFg13jpcUg3CGdOI5Qv6t1PSc2Tnnkunk/edit?usp=sharing>

1. **To create HLR & Test Case of Web Based (WhatsApp web , Instagram).**

**Ans. -** [**https://docs.google.com/spreadsheets/d/15MCi2o34c0xng-bL6FExbVfxZuE\_1JewoxIC-oLRNrc/edit?usp=sharing**](https://docs.google.com/spreadsheets/d/15MCi2o34c0xng-bL6FExbVfxZuE_1JewoxIC-oLRNrc/edit?usp=sharing)

[**https://docs.google.com/spreadsheets/d/1qWEDIcFForJy8Z2kjGykCc0xvRMVj8v-EexGt7SeqMk/edit?usp=sharing**](https://docs.google.com/spreadsheets/d/1qWEDIcFForJy8Z2kjGykCc0xvRMVj8v-EexGt7SeqMk/edit?usp=sharing)

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |