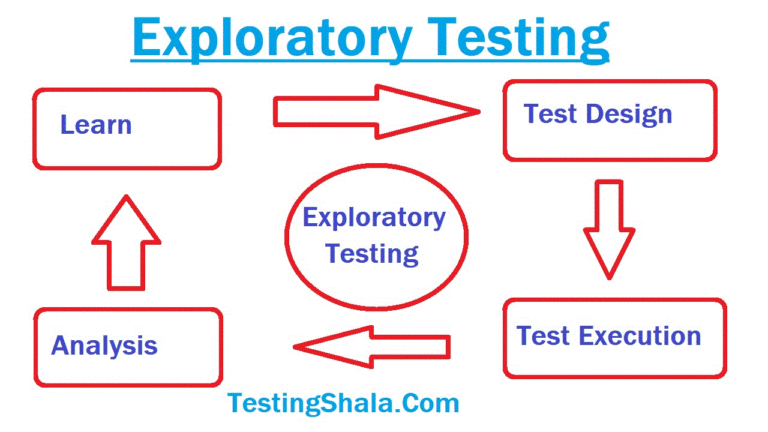
**Assignment – 2(Manual Testing)**

1. **What is Exploratory Testing?**

Ans: - Exploratory testing is a part of experience-based testing.

* Exploratory testing is a software testing method that involves exploring software to identify issues and assess user experienced without using pre-defines test case.
* Exploratory testing is a concurrent process where test design, execution and learning as simultaneously.



**When to use Exploratory Testing.**

* Testing unclear, incomplete or frequently changing requirement.
* Testing complex, high-risk areas within the software system.
* Covering testing gaps and providing feedback.
* Discovering new defects
* Saving on testing time and cost
* Ensuring end user satisfaction.

**When to say no to exploratory testing.**

* Time – sensitive project with tight deadlines.
* Project with rigorous requirement and stringent compliance.

1. **What is a traceability matrix?**

**Ans: -** A traceability matrix is a document that compares two baselined documents that have a many-to-many relationship.

* To protect against changes, you should be able to trace back from every system component to the original requirement that caused its presence.
* **T**raceability matrix is also known as RTM (Requirement Traceability Matrix).
* There are two types of RTM/TM

1. Forward Traceability – Mapping of requirement to test cases
2. Backward Traceability – Mapping of test case to requirement
3. Bi- Directional Traceability – A good traceability matrix is the references from test case to basic document and vice versa.

**Pros: -**

* To make sure that all requirement included in the test case.
* Easy to identify the missing functionalities.
* If there is a change request for a requirement, then we can easily find out which test case need to update.

**Cons: -**

* Poor or unknown test coverage, more defects found in production.
* Difficult project planning and tracking, misunderstandings between different teams over project dependencies, delays, etc.

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

**Ans**: -BVA is a software testing technique that identifies errors and bugs in software program by testing the boundaries of its input domain, including the minimum, maximum, or exact values and the transitions between different states.

* BVA is a methodology for designing test cases that concentrates software testing effort on case near the limits of valid ranges.
* BVA is a method which refines equivalence partitioning.
* BVA generates test cases that highlight errors better than equivalence partitioning.
* At those points when input values change from valid to invalid errors are most likely to occur.

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

**Ans: -** Equivalence partitioning testing is a Blackbox testing method that reduces the number of test cases while still ensuring effective test coverage.

* It involves dividing input data into group of equivalent data is called equivalence partitioning or classes.
* Aim is to treat group of input as equivalent and select one representative input to test them all.
* EP can be used for all levels of testing.
* EP always one positive scenario and two negative scenarios.

1. **What is Integration testing?**

**Ans: -**Testing performed to expose defects in the interactions between integrated components or systems.

Integration testing is a level of the software testing process where individual units are combined and tested as a group.

**Two level of Integration testing:**

1. Component integration testing.
2. System integration testing.

* **Component integration testing:** testing performed to expose defects in the interfaces and interaction between integrated components.
* **System integration testing:** it tests the interactions between different systems and may be done after system testing.

**There are two types of integration testing methods:**

1.Bing Bang integration testing

2. Incremental integration testing

* Top-down approach
* Botton up approach

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

Ans: - It depends on below evaluated points:

1. Business levels.

2. Technological level

3. Project level

4. Testing level – it depends on where to start testing and where more testing is needed

5. Financial cost of software

6. Potential loss of life like safety systems

7. Potential loss of face (improper implementations)

8. Late delivery to the market

1. **What is Alpha testing?**

**Ans: -** Alpha Testing is a type of software testing performed to identify bugs before releasing the product to real users or the public.

* Alpha Testing is one of user acceptance testing. This is referred to as Alpha Testing.
* It is always performed by the developer at the software development site.
* Sometimes it is always performed by the independent testing teams.
* Alpha testing is not open to the market and public.
* It is conducted for the software application and project
* It comes under the category of both White Box Testing and Black Box Testing.

1. **What is Beta Testing?**

**Ans:-** Beta testing is the process of testing a software product or service in a real-world environment before its official release.

* It helps identify bugs and errors that may have been missed during the development process.
* Beta testing is always performed by the customer at their own sites.
* It is not performed by the independent testing team.
* Beta testing is always open to the market or public.
* It is usually conduct for the software product.
* It is performed by the Real time environment.
* It is always performed outside organization.
* It is only kind of black box testing.

1. **What is component testing?**

**Ans:-** A minimal software item that can be tested in isolation. That means a small unit to be tested as a part of software.

* It is the first level of testing and is performed prior to integration testing.
* Developer also conduct the testing which is called unit/component testing.
* It’s also known as unit testing, module testing or program testing.
* Unit testing is performed by using the white box testing method.
* Unit testing frameworks, drivers, stubs and fake objects are used to assist in unit testing.
* Unit testing are typically written and run by software developers to ensure that code meets its design and behaves as intended with debugging tool.
* Unit tests find problems early in the development cycle.

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

**Ans:-** Function means – what the system does work

* Functional Testing is a type of software testing that validates the software system against the functional requirements/specifications.
* The purpose of Functional testing is to test each function of the software application, by providing appropriate input, verifying the output against the Functional requirements.
* Testing based on an analysis of the specification of the functionality of a component or system.
* Functional testing verifies that each **function** of the software application operates in conformance with the requirement specification.
* The testing mainly involves Black-box testing and it is not concerned about the source code of the application.
* Each and every functionality to be tested by providing the input, verifying and output and compare to the actual result with the expected result.
* There are different types of functional testing: -
  + - 1. Unit Testing ∙
      2. Smoke Testing
      3. Sanity Testing
      4. Integration Testing
      5. White box testing
      6. Black Box testing
      7. User Acceptance testing
      8. Regression Testing

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

**Ans:-** Non – functional testing is one type of behaviour testing that is performed to verify the non-functional requirements of the application.

* It verifies whether the behaviour of the system is as per the requirement or not.
* It tests all the aspects that are not tested in functional testing.
* Non – functional testing that performed reliability, efficiency, usability, performance, maintainability and portability.
* It performed after the functional testing.
* Non- functional testing describe- how good the product works.
* Using tools will be effective for this testing.
* There are different types of non- functional testing.

Performance Testing

Load Testing

Volume Testing

Stress Testing ∙

Security Testing ∙

Installation Testing ∙

Penetration Testing ∙

Compatibility Testing ∙

Migration Testing

1. **What is GUI Testing?**

**Ans:-** GUI testing is the process of testing the system’s GUI of the system under test.

* In this we involve checking the screens with the controls like menus, buttons, icons, and all types of bars – toolbar, menu bar, dialog boxes and windows etc.

Common aspects verified in GUI testing:

1. Layout
2. Style and aesthetics
3. Responsiveness
4. Behaviour
5. Error messages

Type of GUI testing

1. Manual testing
2. Automated Testing
3. Cross-platform testing
4. **What is Adhoc testing?**

**Ans:-** It is informal testing type with an aim to break the system.

* It does not follow any test design techniques to create test cases.
* In fact is does not create test cases altogether.
* This testing is primarily performed if the knowledge of testers in the system under test is very high.
* Main aim of this testing is to find defects by random checking.
* Ad-hoc testing can be achieved with the testing technique called error guessing.

Types of ad-hoc testing

1. Buddy testing
2. Pair testing
3. Monkey testing
4. **What is load testing?**

**Ans:-** Load testing is to test the system behaviour under normal workload conditions, and it is just testing or simulating with the actual workload.

* Load testing does not break the system
* Load testing identifies the bottlenecks breaking the system under various workload and checks how the system reacts when the load is gradually increased.
* Load testing gives confidence in the system & its reliability and performance.
* Load testing is a kind of performance testing which determines a system’s performance under real-life load conditions.

1. **What is stress Testing?**

**Ans:-** Stress testing is to test the system behaviour under extreme conditions and is carried out till the system failure.

* In the stress testing determines the point of the system to revel the maximum point after which it breaks.
* Stress testing tries to break the system by testing with overwhelming data or resources.
* Stress testing is done in order to check when the application fails by reducing the resources such as RAM, HDD etc. and also the keeping the number of users constant.

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

**Ans:-** Testing based on an analysis of the internal structure of the component or system.

* Structure-based testing technique is also known as white-box testing or glass box testing technique because here the testers require knowledge of how the software is implemented, how it works.
* White box testing done by the developer ends.
* There are different types of white box testing:

1. Statement coverage
2. Decision coverage
3. Condition coverage
4. **What is black box testing? What are the different black box testing techniques?**

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

* Testers have no knowledge of how the system or component is structured inside the box.
* In black box testing the tester is concentrating on what the software does, not how it does it.
* In this testing tester does not require coding knowledge.
* There are different type of Black box testing techniques:

1. Equivalence partitioning
2. Boundary value analysis
3. Decision tables
4. State transition testing
5. Use-case testing
6. **Mention what are the categories of defects?**

**Ans:-** Functional defects or Critical functional defects

* Performance defects
* GUI Defects
* Security defects
* Logical defects
* Base on lifecycle phase defects like requirement defects, design defects, coding defects, testing defects, deployment defects etc.
* Database defects

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

**Ans:-** Big-bang testing is one type of integration testing.

* In this testing all components or modules is integrated simultaneously, after which everything is tested as a whole.
* In this first we integrated all modules and then after we tested.
* It’s convenient for small systems.
* Main problem is fault localization is difficult.

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

**Ans:-** Successful testing of integrated application.

* Executed test cases are documented.
* All high prioritized bugs fixed and closed.
* Technical documents to be submitted followed by release notes.

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

**Ans:-** In this testing of a previously tested program following modification to ensure that defects have not been introduced or uncovered in unchanged area of the software, as a result of the changes made. It is performed when the software or its environment changed.

- When we use Regression testing :

* Change in requirements and code is modified according to requirement.
* New feature is added to the software
* Defect fixing
* Performance issue fix

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

**Ans:-** There are 7 key principles:-

1. **Testing shows presence of defects:**

* Testing shows the defect are present, but cannot prove that there are no defects.
* Testing reduce the probability of undiscovered defects remaining in the software but, even if no defect are found, it is not a proof of correctness.
* However testing cannot prove that there are no defect present.

1. **Exhaustive testing is impossible**

* Testing everything including all combinations of inputs and preconditions is not possible
* We don’t have too much time to check all modules in the software or project so we must priorities our testing effort using a risk based approach.

1. **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.

1. **Defect clustering**

* A small number of modules contain most of defects discovered during pre-release testing, or are responsible for the most operational failures.
* Defects are not evenly spread in system
* They are clustered.

1. **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.
* In this stage we 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.

1. **Testing is context dependent**

* Testing is basically context dependent.
* Testing is done differently in different contexts
* Different kinds of sites are tested differently

1. **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.
* The defect have been resolved it may still be unusable and does not fulfil the user’s needs and expectations.

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

**Ans:-**

|  |  |  |
| --- | --- | --- |
| **Quality Assurance** | **Quality Control** | **Tester** |
| QA is a set of activities for ensuring quality in the process by which products are developed. | QC is a set of activities for ensuring quality in products. The activities focus on identifying defects in the actual products produced. | Activities which ensure the identification of bugs/error/defects in the Software. |
| Focuses on processes and procedures rather than conducting actual testing on the system. | Focuses on actual testing by executing Software with intend to identify bug/defect through implementation of procedures and process. | Focuses on actual testing. |
| QA is preventing defects | QC is correcting defects | Tester is correcting defects |
| QA gives confidence to you | QC is gives expected results |  |
| It is a subset of Software Test Life Cycle (STLC). | QC can be considered as the subset of Quality Assurance. | It is a subset of Quality control(QC) |
| QA involves the whole team. | QC involves specific team member. | Tester involves testing team |
| QA is proactive process | QC is reactive process | Tester is a reactive |

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

**Ans:-**

|  |  |
| --- | --- |
| **Smoke Testing** | **Sanity Testing** |
| Smoke testing is used to perform to ensure that the critical functionalities of the application are working fine. | Sanity testing is used to verify the newly added functionalities/bugs etc are working fine. |
| This testing is done at the initial level | This testing is done when the build is relatively stable. |
| Smoke testing is performed by the developer or testers. | Sanity testing is usually performed by testers. |
| Smoke testing is called a subset of acceptance testing | Sanity testing is called a subset of regression testing |
| Smoke testing is done on every build | Sanity testing is done after the completion of regression testing. |
| Smoke testing is used to test all over the functionality of the application. | Sanity testing is used in the of only updated or defect functions of the application. |
| Smoke testing is may be stable/unstable | Sanity testing is always stable. |

1. **Difference between verification and Validation**

**Ans:-**

|  |  |
| --- | --- |
| **Verification** | **Validation** |
| Verification means “Are we building the software right?” | Validation means” Are we building the right software?” |
| 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. |
| The goal of verification is the specification of the application. | The goal of validation process is an actual product. |
| It is used to find the bugs in the beginning phase of the software | It finds the bug which cannot find in the verification process. |
| It consists of checking documents and program | It consists of validating and testing the actual product. |

1. **Explain types of Performance testing**

**Ans:-** Performance testing is a meaning of quality assurance. It involves the testing software application to ensure they will perform well under their expected workload.

* Type of performance testing.

1. Load testing
2. Stress testing
3. Endurance testing
4. Spike testing
5. Volume testing
6. Scalability testing
7. **What is Error, Defect, Bug and failure?**

Ans:- **Error** – Mistake made by a programmer during coding is called an error.

**Defect** – An error found during the unit testing in the development phase is called defect

**Bug** – An error found during the testing phase is called a bug

**Failure** – An error is found at an end user’s end is called as the failure.

1. **Explain the difference between Functional testing and Nonfunctional testing**

**Ans:-**

|  |  |
| --- | --- |
| **Functional testing** | **Non-Functional testing** |
| Functional testing is performed using functional specification provided by the client and verifies the system against the functional requirements. | the Non-Functional testing checks the Performance, and reliability, scalability and other non-functional aspects of the software system. |
| Functional testing is executed first | Non - functional testing should be performed after functional testing |
| Manual testing or automation tools can be used for functional testing | Using tools will be effective for this testing |
| Business requirements are the inputs to functional testing | Performance parameters like speed, scalability are inputs to non-functional testing. |
| Easy to do manual testing | Tough to do manual 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 Nonfunctional testing are  ∙ Performance Testing  ∙ Load Testing  ∙ Volume Testing  ∙ Stress Testing  ∙ Security Testing  ∙ Installation Testing  ∙ Penetration Testing  ∙ Compatibility Testing  ∙ Migration Testing |

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

**Ans:-**

|  |  |
| --- | --- |
| **Software Development life cycle (SDLC)** | **Software Testing Life Cycle  (STLC)** |
| SDLC is mainly related to software development | STLC is mainly related to software testing |
| Besides development other phases like testing is also included. | It focuses only on testing the software |
| It help in developing good quality software | It help in making the software defect free. |
| SDLC phases are completed before the STLC phases. | STLC phases are performed after SDLC phases. |
| SDLC involves total six phases | STLC involved only five phase or step |
| In SDLC, more number of members are required for the whole process (included developer) | In STLC, less number of members are needed (including tester) |
| In SDLC, development team makes the plans and design based on the requirements. | In STLC, testing team (Test Lead or Test Architect) makes the plans and design. |

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

**Ans:-**

|  |  |  |
| --- | --- | --- |
| **Test Scenarios** | **Test Cases** | **Test Script** |
| Test scenarios is any functionality that can be tested. | Test case is a set of actions executed to verify particular features or functionality | Test Script is a set of instructions to test an app automatically |
| It derived from test artifacts like BRS and SRS | It is mostly derived from test scenarios | It is mostly derived from test cases |
| It helps to test the end-to-end functionality in Agile way | It helps to exhausting testing in an app | It helps to test specific things repeatedly. |
| It is more focus on what to test | It is focused on what to test and how to test. | It is focused on expected result |
| Take a less time and fewer resources to create | Requires more resources and time | Required less time for testing but more resources for scripts creating and updating |
| Included an end-to-end functionality to be tested | Included test step, data expected results for testing | Include different commands to develop a script |
| Allows quickly assessing the testing scop | Allows detecting error and defects | Allows carrying out an automatic execution of test cases |

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

**Ans:-** A document describing the scope, approach, resources and schedule of intended test activities.

* Key information in a plan

1. Test objectives
2. Test scope
3. Test strategy
4. Entry and exit criteria
5. Test deliverables
6. Test environment
7. Role and responsibilities
8. Risk and mitigation
9. **What are the different Methodologies in Agile Development Model?**

**Ans:-** There are some different methodologies in the agile development model:

* + - 1. Scrum
      2. Kanban
      3. Extreme programming (XP)
      4. Adaptive Project Framework (APF)
      5. Feature Driven Development (FDD)
      6. Dynamic systems Development Method (DSDM)
      7. Adaptive Software Development (ASD)

Scram frameworks:

* Product backlog
* Sprint planning
* Sprint backlog
* Daily scram
* Sprint retrospective
* Sprint review

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

**Ans:-** The procedure for GUI testing, also knows as UI testing, involves examining the user interface(UI) of a software application to ensure it works as expected.

Here are some methods for GUI testing:

* Manual Testing
* Record and replay
* Model based testing
* Code based testing
* Hybrid tests

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

**Ans:-** You should use usability testing at various stages of a product’s development, including:-

* Before designing
* Once you have a wireframe or prototype
* Before launching
* At regular intervals after launch

Usability testing help you:-

* To determine if tester can complete tasks successfully and independently
* Assess how well your design works by evaluating testers performance and mental state
* See how much users enjoy using your product
* Make changes based on the results to improve your product’s usability