**Module–2(Manual Testing)**

1. **What is Exploratory Testing?**

* Exploratory testing is a concurrent process where

Test design, execution and logging happen simultaneously

Testing is often not recorded.

* Makes use of experience, heuristics and test patterns

Testing is based on a test charter that may include

Scope of the testing

1. **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.
* Traceability can be horizontal through all the test documentation for a given test level.

1. **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 which refines equivalence partitioning.
* Boundary value analysis generates test cases that highlight errors better than equivalence partitioning.

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

* Equivalence partitioning is the process of defining the optimum number of tests by Re viewing documents such as the Functional Design Specification and Detailed Design Specification, and identifying each input condition with in a function.

Equivalence partitioning testing can be used for all Levels of Testing.

1. **What is Integration testing?**

* System Integration Testing is testing between the ‘System’ and ‘Acceptance’ phases.
* The objective of System Integration Testing is to provide confidence that the system or application is able to interoperate successfully with other specified software systems and does not have an adverse affect on other systems that may also be present in the live environment*,* or vice versa.

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

* ‘A factor that could result in future negative consequences; usually expressed as impact and likelihood’.
* A Risk could be any future event with a negative consequence Need to identify the risks associated with your project.

1. **What is Alpha testing?**

* Alpha testing is always performed by the developers at the software development site.
* Alpha Testing is not open to the market and public.
* It is conducted for the software application and project.

1. **What is beta testing?**

* It is always performed by the customers at their own site.
* It is not performed by Independent Testing Team.
* Beta Testing is always open to the market and public.
* It is usually conducted for software product.

1. **What is component testing?**

* A minimal software item that can be tested in isolation. It means “A unit is the smallest testable part of software.”
* Component Testing – The testing of individual software components.
* Unit Testing is a level of the software testing process where individual units/components of a software/system are tested. The purpose is to validate that each unit of the software performs as designed.

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

* 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
* Requirement Based Functional Testing
* Process Based Testing

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

* Non-Functional Testing: Testing the attributes 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?**

* Graphical User Interface (GUI) testing is the process of testing the system’s GUI of the System under Test. 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.

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

* Ad hoc testing is an 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.

1. **What is load testing?**

* Load testing - Its a performance testing to check system behavior under load. 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.
* This testing usually identifies
* The maximum operating capacity of an application

1. **What is stress Testing?**

* Stress testing - System is stressed beyond its specifications to check how and when it fails. Performed under heavy load like putting large number beyond storage capacity, complex database queries, continuous input to system or database load.
* 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?**

* White Box Testing is *Testing based on an analysis of the internal structure of the component or system.*
* Structure-based testing technique is also known as ‘white-box’ or ‘glass- box’ testing technique because here the testers require knowledge of how the software is implemented, how it works.

**types of white box testing:-**

* glass testing
* open box testing

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

* Black-box testing is Testing, either functional or non-functional, without reference to the internal structure of the component or system.
* Functional Testing using Black Box Testing techniques against the Interfacing requirements for the component under test.
* Non-functional Testing (where appropriate, for performance or reliability testing of the component interfaces, for example)

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

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

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

* In Big Bang integration testing all components or modules is integrated simultaneously, after which everything is tested as a whole.
* Big Bang testing has the advantage that everything is finished before integration testing starts.

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

* Exit Criteria defines the items that must be completed before testing can be concluded
* Exit criteria is used to determine when testing at any stage is complete the set of generic and specific conditions, agreed upon with the stakeholders, for permitting a process to be officially completed

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

* When the system is stable and the system or the environment changes when testing bug-fix releases as part of the maintenance phase It should be applied at all Test Levels
* It should be considered complete when agreed completion criteria for regression testing has been met Regression test suites evolve over time and given that they are run frequently are ideal candidates for automation.

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

* **7 key principle:-** General Testing Principles
* Testing shows presence of Defects
* Exhaustive Testing is Impossible.
* Early Testing.
* Defect Clustering.
* The Pesticide Paradox.
* Testing is Context Dependent.
* Absence of Errors Fallacy.

## 1) **Exhaustive testing is not possible**

 Exhaustive testing is not possible. Instead, we need the optimal amount of testing based on the risk assessment of the application. and the million dollar question is, how do you determine this risk? To answer this let’s do an exercise

2) **Defect Clustering**

Defect Clustering which states that a small number of modules contain most of the defects detected. This is the application of the Pareto Principle to software testing: approximately 80% of the problems are found in 20% of the modules.

## 3) **Pesticide Paradox**

Repetitive use of the same pesticide mix to eradicate insects during farming will over time lead to the insects developing resistance to the pesticide Thereby ineffective of pesticides on insects. The same applies to software testing. If the same set of repetitive tests are conducted, the method will be useless for discovering new defects.

## 4) **Testing shows a presence of defects**

Hence, testing principle states that – Testing talks about the presence of defects and don’t talk about the absence of defects. i.e. Software Testing reduces the probability of undiscovered defects remaining in the software but even if no defects are found, it is not a proof of correctness.

### 5) Absence of Error – fallacy

### It is possible that software which is 99% bug-free is still unusable. This can be the case if the system is tested thoroughly for the wrong requirement. Software testing is not mere finding defects, but also to check that software addresses the business needs. The absence of Error is a Fallacy i.eFinding and fixing defects does not help if the system build is unusable and does not fulfill the user’s needs & requirements.

### 6) Early Testing

### Early Testing – Testing should start as early as possible in the Software Development Life Cycle. So that any defects in the requirements or design phase are captured in early stages. It is much cheaper to fix a Defect in the early stages of testing. But how early one should start testing? It is recommended that you start finding the bug the moment the requirements are defined. More on this principle in a later training tutorial.

## 7) **Testing is context dependent**

Testing is context dependent which basically means that the way you test an e-commerce site will be different from the way you test a commercial off the shelf application. All the developed software’s are not identical. You might use a different approach, methodologies, techniques, and types of testing depending upon the application type. For instance testing, any POS system at a retail store will be different than testing an ATM machine.

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

|  |  |  |
| --- | --- | --- |
| **S.N** | **Quality Assurance** | **Quality Control** |
| **1.** | **Activity which ensures the implementation of processes, products and standards in context to verification of develop software and intended requirements.** | **Activities which ensures the verifications of developed software with respect to documented.** |
| **2.** | **Focuses on processes and products rather than conducting actual testing in the system.** | **Focuses on actual testing by executing software with intend to identify bug/defects through implementation of procedures and process.** |
| **3.** | **Process oriented activity.** | **Product oriented activity.** |
| **4.** | **Preventive activities.** | **It is a corrective process.** |
| **5.** | **It is a subset of STLC.** | **It can be considered as the subset of QA.** |

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

|  |  |  |
| --- | --- | --- |
| **S.N** | **Smoke** | **Sanity** |
| **1.** | Smoke Testing is performed to ascertained that the critical functionalities of the program is working fine. | Sanity Testing is done to check the new functionality/bugs have been fixed. |
| **2.** | The objectives of this testing is to verify “stability” of the system in order to with more rigorous testing. | The objectives of the testing is to verify the “rationality” of the system in order proceed with more rigorous testing. |
| **3.** | This testing is performed by the developers or testers. | Sanity is usually performed by testers. |
| **4.** | Smoke testing usually documented or scripted. | Sanity testing is usually not documented and unscripted. |
| **5.** | Smoke testing is a subset of Regression Testing. | Sanity testing is a subset of Accepted Testing. |
| **6.** | 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**

|  |  |  |
| --- | --- | --- |
| **Criteria** | **Verification** | **Validation** |
| **Definition** | 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. |
| **Objective** | To ensure that the product is  being built according to  the requirements and  design specifications. In other words, to ensure that work products meet their  specified requirements. | To ensure that the product actually meets  the user’s needs, and that the  specifications were correct in the first place.  In other words, to demonstrate that the product fulfills its intended use  when placed in its intended environment. |
| **Question** | Are we building the product right? | Are we building the right product? |
| **Evaluation**  **Items** | Plans, Requirement Specs, Design  Specs, Code, Test Cases | The actual product/software. |
| **Activities** | Reviews  Walkthroughs  Inspections | Testing |

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

* **Types of Performance Testing :-**
* Load testing
* Stress testing
* Endurance testing
* Spike testing
* Volume testing
* Scalability testing

1. **Difference between Priority and severity.**

|  |  |  |
| --- | --- | --- |
| **S.N** | **Priority** | **Severity** |
| **1.** | Severity is a parameter to denote the impact of a particular defect on the software. | Priority is a parameter to decide the order in which defects should be fixed. |
| **2.** | Severity means how severe defect is affecting the functionality. | Priority means how fast defect has to be fixed |
| **3.** | Severity is related to the quality standard. | Priority is related to scheduling to resolve the problem. |
| **4.** | Testing engineer decides the severity level of the defect. | Product manager decides the priorities of defects. |
| **5.** | Its value is objective. | Its value is subjective. |
| **6.** | Its value doesn’t change from time to time. | Its value changes from time to time. |
| **7.** | Severity is of 5 types: Critical, Major, Moderate, Minor, and Cosmetic. | Priority is of 3 types: Low, Medium, and High |

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

* “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”
* **Error**: A discrepancy between a computed, observed, or measured value or condition and the true, specified, or theoretically correct value or condition. This can be a misunderstanding of the internal state of the software, an oversight in terms of memory management, confusion about the proper way to calculate a value, etc.
* **Defect**: Commonly refers to several troubles with the software products, with its external behavior or with its internal features.
* **Failure**: The inability of a system or component to perform its required functions within specified performance requirements. See: bug, crash, exception, and fault.

1. **What is Bug Life Cycle?**

* “A computer bug is an error, flaw, mistake, failure, or fault in a computer program that prevents it from working correctly or produces an incorrect result. Bugs arise from mistakes and errors, made by people, in either a program’s source code or its design.”

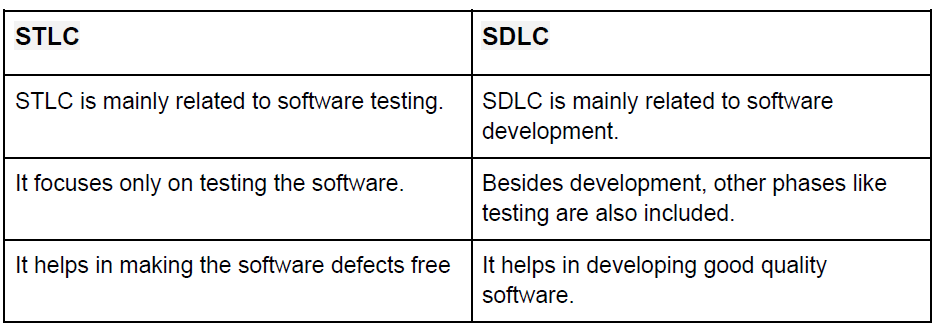
1. **Explain the difference between Functional testing and Non Functional testing**

|  |  |  |
| --- | --- | --- |
| **S.N** | **Functional Testing** | **Non Functional Testing** |
| **1.** | Functional Testing is performed using the functional specification provided by the client and verifies the system against the functional requirements. | Non-Functional testing checks the performance, reliability, scalability and other Non- functional aspects of the software system. |
| **2.** | Functional Testing is executed first. | Non-Functional should be performed after functional testing. |
| **3.** | Manual testing or automation tools can be used for functional testing. | Using tools will be effective for this testing. |
| **4.** | Business requirements are the inputs to functional testing. | Performance parameters like speedscalability are inputs to non-functional testing. |
| **5.** | Functional Testing is describes what the product does. | Non-Functional Testing is describes how good the product work. |
| **6.** | Easy to do manual testing. | Tough to do manual testing. |
| **7.** | 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.Regrassion testing | Types of Non-functional testing  1.Load testing  2.performance testing  3.Volume testing  4.Stress testing  5.Secuirity testing  6.Installation testing  7.Panastration testing  8.Competibility testing  9.migration testing |

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

* Test Planning in STLC is a phase in which a Senior QA manager determines 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.

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

****

1. **What is priority?**

* The quality or state of coming before another in time or importance.

1. **What is the severity?**

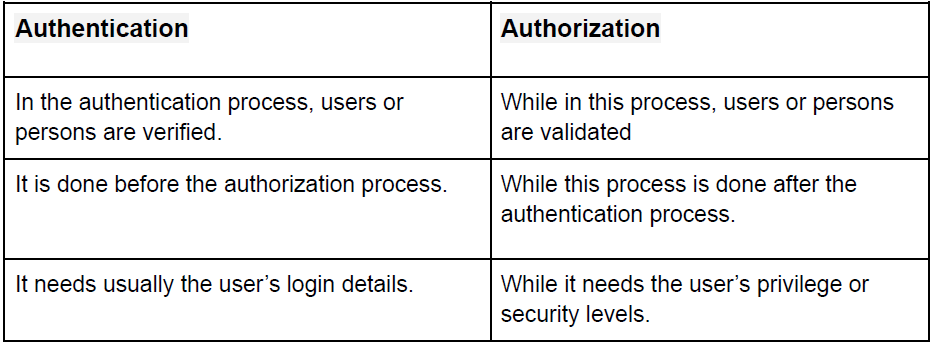
* The quality or state of being severed

1. **What are the different Methodologies in Agile Development Model?**

* Agile methodology is a “step by step” dynamic focused on short-term visibility but never

losing the long-term product goal.

* Scrum,
* Kanban
* Extreme Programming (XP),
* Lean Development e Crystal.

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

* Integration.
* Interoperability.
* Security.
* Performance.
* Usability.
* Quality Testing, Exceptional Services.