# Software Testing Assignment 2

## Module – 2(Manual Testing)

**1. What is Exploratory Testing?**

- **Though the current trend in testing is to push for automation, exploratory testing is a new way of thinking.**
- **Automation has its limits Is not random testing but it is Adhoc testing with purpose of find bugs Is structured and rigorous Is cognitively (thinking) structured as compared to procedural structure of scripted 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 (in and out)**

**2. 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.**

- **A software process should help you keeping the virtual table up-to-date. Simple technique may be quite valuable (naming convention)**

**3. 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.**
- **The trick is to concentrate software testing efforts at the extreme ends of the equivalence classes.**

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

- **Aim is to treat groups of inputs as equivalent and to select one representative input to test them all**
- **EP can be used for all Levels of 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 a function,**

- **Selecting input data that is representative of all other data that would likely invoke the same process for that particular condition.**

**5. What is Integration testing?**

- **Integration Testing - Testing performed to expose defects in the interfaces and 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.**
- **Integration testing is done by a specific integration tester or test team. Components may be code modules, operating systems, hardware and even complete systems**
- **There are 2 levels of Integration Testing**

    **1) Component Integration Testing**
    **2) System Integration Testing**

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

- **A properly designed test that passes, reduces the overall level of Risk in a system**
- **Risk – 'A factor that could result in future negative consequences; usually expressed as impact and likelihood' When testing does find defects, the Quality of the software system**

- **Types of Risk :**
  - **A Risk could be any future event with a negative consequence**
  - **You need to identify the risks associated with your project**
  - **Risks are of two types**

    **1)Project Risks**
    **2)Product Risk**

**7. What is Alpha testing?**
- **It is always performed by the developers at the software development site.**
- **Sometimes it is also performed by Independent Testing Team. Alpha Testing is not open to the market and public**
- **It is conducted for the software application and project. It is always performed in Virtual Environment.**
- **It is always performed within the organization. It is the form of Acceptance Testing.**

**8. 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.**
- **It is performed in Real Time Environment.**

- Beta Testing is always performed at the time when software product and project are marketed.
- It is always performed at the user’s premises in the absence of the development team.

9. What is component testing?

- Component(Unit) – 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.
- Unit testing is the first level of testing and is performed prior to Integration Testing.
- Unit testing frameworks, drivers, stubs and mock or fake objects are used to assist in unit testing.
- Unit testing is performed by using the White Box Testing method.

10. What is functional system testing?

  - Functional Testing: 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.

- **This testing mainly involves black box testing and it is not concerned about the source code of the application.**

**11. 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**
- **Non-functional testing includes, but is not limited to, performance testing, load testing, stress testing, usability testing, maintainability testing, reliability testing and portability testing.**
- **It is the testing of "how" the system works. Non-functional testing may be performed at all test levels.**
- **To address this issue, performance testing is carried out to check & fine tune system response times.**

**12. 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.**

- **WHAT DO YOU CHECK IN GUI TESTING?**

  - **Check all the GUI elements for size, position, width, length and acceptance of characters or numbers. For instance, you must be able to provide inputs to the input fields.**
  - **Check you can execute the intended functionality of the application using the GUI Check Error Messages are displayed correctly**
  - **Check for Clear demarcation of different sections on screen Check Font used in application is readable**
  - **Check the alignment of the text is proper**
  - **Check the Color of the font and warning messages is aesthetically pleasing Check that the images have good clarity**
  - **Check that the images are properly aligned**
  - **Check the positioning of GUI elements for different screen resolution.**

**13. What is Adhoc testing?**

- **Adhoc 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.
- Main aim of this testing is to find defects by random checking. Adhoc testing can be achieved with the testing technique called Error Guessing.
- The Error guessing is a technique where the experienced and good testers are encouraged to think of situations in which the software may not be able to cope.
- Some people seem to be naturally good at testing and others are good testers because they have a lot of experience either as a tester or working with a particular system and so are able to find out its weaknesses.

**14. What is load testing?**

- Load testing - Its a performance testing to check system behaviour 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

- **Determine whether current infrastructure is sufficient to run the application Sustainability of application with respect to peak user load**
- **Number of concurrent users that an application can support, and scalability to allow more users to access it.**
- **It is a type of non-functional testing. Load testing is commonly used for the Client/Server, Web based applications – both Intranet and Internet.**

**15. 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.**
- **Most prominent use of stress testing is to determine the limit, at which the system or software or hardware breaks.**

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

- **White Box Testing: 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.**
- **White box testing is the detailed investigation of internal logic and structure of the code.**
- **White box testing is also called glass testing or open box testing. In order to perform white box testing on an application, the tester needs to possess knowledge of the internal working of the code.**

- **Types of white-box testing :**

**1)Web Based Testing**

**2)Desktop Based Testing**

**3)Mobile Based Testing**

**4)Game Based Testing**

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

- **Black-box testing: Testing, either functional or non-functional, without reference to the internal structure of the component or system.**
- **Specification-based testing technique is also known as 'black-box' or input/output driven testing techniques because they view the software as a black-box with inputs and outputs.**
- **The testers have no knowledge of how the system or component is structured inside the box.**
- **The technique of testing without having any knowledge of the interior workings of the application is Black Box testing.**

- **Techniques of Black Box Testing :**

- **There are four specification-based or black-box technique:**
    - **Equivalence partitioning**
    - **Boundary value analysis Decision tables**
    - **State transition testing Use-case Testing**
    - **Other Black Box Testing**
    - **Syntax or Pattern Testing**

## 18. Mention what are the categories of defects?

- **Types of Defect :**
- **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**
- **Critical Functionality Defects:**
  - **The occurrence of these bugs hampers the crucial functionality of the application.**
  - **Examples: - Exceptions**
- **Functionality Defects:**
  - **These defects affect the functionality of the application.**
  - **Examples:**
  - **All JavaScript errors**
- **Security Defects:**
  - **Application security defects generally involve improper handling of data sent from the user to the application. These defects are the most severe and given highest priority for a fix.**
  - **Examples:**
  - **Authentication: Accepting an invalid username/password**
  - **Authorization: Accessibility to pages though permission not given**

- **User Interface Defects:**
    - **As the name suggests, the bugs deal with problems related to UI are usually considered less severe.**
    - **Examples:**
    - **Improper error/warning/UI messages Spelling mistakes**

**19. Mention what big-bang testing is?**

- **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.**
- **The major disadvantage is that in general, it is time-consuming and difficult to trace the cause of failures because of this late integration. Here all components are integrated at once and then tested.**

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

- **Purpose of exit criteria is to define when we STOP testing either at the:**
    - **End of all testing – i.e. product Go Live**
    - **End of phase of testing (e.g. hand over from System Test to UAT)**

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

- **Regression Testing: Testing of a previously tested program following modification to ensure that defects have not been introduced or uncovered in unchanged areas of the software, as a result of the changes made. It is performed when the software or its environment is changed.**
- **If the test is re-run and passes you cannot necessarily say the fault has been resolved because ..**
- **You also need to ensure that the modifications have not caused unintended side-effects elsewhere and that the modified system still meets its requirements – Regression Testing**
- **Regression testing should be carried out:**
- **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 have been met**
- **Regression test suites evolve over time and given that they are run frequently are ideal candidates for automation**

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

- **Testing shows presence of Defects**
- **Exhaustive Testing is Impossible!**
- **Early Testing**
- **Defect Clustering**
- **The Pesticide Paradox**
- **Testing is Context Dependent**
- **Absence of Errors Fallacy**

- **Testing shows the presence of Defects :**
  - **Testing can show that defects are present, but cannot prove that there are no defects.**
  - **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.**
  - **We test to find Faults**
  - **As we find more defects, the probability of undiscovered defects**

- **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.**
  - **For example: In an application in one screen there are 15 input fields, each having 5 possible values, then to test all the valid combinations you would need 30 517 578 125 (515) tests.**

- **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**
  - **These activities should be focused on defined objectives – outlined in the Test Strategy**
- **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 They are 'clustered'**
- **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.**
- **Testing is Context Dependent :**
  - **Testing is context-dependent. Testing is done differently in different contexts**
  - **Different kinds of sites are tested differently.**
  - **For example :**

- Safety – critical software is tested differently from an e-commerce
- site.
- Whilst, Testing can be 50% of development costs, in NASA's Apollo program it was 80% of testing

**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.
- If we build a system and, in doing so, find and fix defects .... It doesn't make it a good system
- Even after defects have been resolved it may still be unusable and/or does not fulfil the users' needs and expectations

## 23. Difference between QA v/s QC v/s Tester

| S.N. | Quality Assurance | Quality Control | Testing |
|---|---|---|---|
| 1 | **Activities which ensure the implementation of processes, procedures and standards in context to verification of developed software and intended requirements.** | **Activities which ensure the verification of software with developed respect to documented (or not in some cases) requirements.** | **Activities which ensure the identification of bugs/error/defects in the Software.** |
| 2 | **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.** |
| 3 | **Process oriented activities.** | **Product oriented activities.** | **Product oriented activities.** |
| 4 | **Preventive activities.** | **It is a corrective process** | **It is a preventive process.** |
| 5 | **It is a subset of Software Test Life Cycle (STLC).** | **QC can be considered as the subset of Quality Assurance.** | **Testing is the subset of Quality Control.** |

## 24. 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 "stability" of the system in order to the rigorous testing** | **The objective of the testing is to verify the "rationality" of the system in order proceed with more to proceed with more rigorous testing** |
| **This testing is performed by the developers or testers** | **Sanity testing is usually performed by testers** |
| **Smoke testing is usually documented  or scripted** | **Sanity testing is usually not documented and unscripted** |
| **Smoke testing is a subset of Regression testing** | **Sanity testing is a subset of Acceptance 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** |

## 25. 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 Plans, Requirement Specs, Design Specs, Code, Test Cases · Reviews · Walkthroughs 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<br>· Walkthroughs<br>· Inspections** | **· Testing** |

## 26. Explain types of Performance testing.

- 1. Understand the Work
- 2. Write Test Cases
- 3. Get Ready for Testing
- 4. Start Testing
- 5. Find and Report Problems
- 6. Re-Test the Fixed Problems
- 7. Check Other Features (Regression Testing)
- 8. Final Testing (User Check)

## 27. What is Error, Defect, Bug and 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.**
- **Failure: The inability of a system or component to perform its required functions within specified performance requirements. See: bug, crash, exception, and fault.**
- **Bug: A fault in a program which causes the program to perform in an unintended or unanticipated manner. See: anomaly, defect, error, exception, and fault. Bug is terminology of Tester.**

- **Defect: Commonly refers to several troubles with the software products, with its external behaviour or with its internal features.**

**28. Difference between Priority and Severity**

- **High Priority & High Severity: An error which occurs on the basic functionality of the application and will not allow the user to use the system. (Eg. A site maintaining the student details, on saving record if it, doesn't allow to save the record then this is high priority and high severity bug.)**
- **High Priority & Low Severity: The spelling mistakes that happens on the cover page or heading or title of an application.**
- **High Severity & Low Priority: An error which occurs on the functionality of the application (for which there is no workaround) and will not allow the user to use the system but on click of link which is rarely used by the end user.**
- **Low Priority and Low Severity: Any cosmetic or spelling issues which is within a paragraph or in the report (Not on cover page, heading, title).**

**29. 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."**
- **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'.**
- **As you can see from above diagram, a defect's state can be divided into Open or Closed.**
- **When a bug reaches one of the Closed or Terminal states, its lifecycle ends. Each state has one or more valid states to move to.**

## 30. 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 verifies the system against the functional requirements. | Non-Functional testing check the Performance, 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. |
| Functional testing describes what the product does | Non functional testing describes how good the product works |
| Easy to do manual testing | Tough to do manual testing |
| Types of Functional testing are<br>· Unit Testing<br>· Smoke Testing<br>· Sanity Testing<br>· Integration Testing<br>· White box testing<br>· Black Box testing<br>· User Acceptance testing<br>· Regression Testing | Types of Nonfunctional testing are<br>· Performance Testing<br>· Load Testing<br>· Volume Testing<br>· Stress Testing<br>· Security Testing<br>· Installation Testing<br>· Penetration Testing<br>· Compatibility Testing<br>· Migration Testing |

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

| SDLC | STLC |
|---|---|
| **Requirements Collection/Gathering** | **Requirement Analysis** |
| **Analysis** | **Test Planning** |
| **Design** | **Test case development** |
| **Implementation** | **Test Environment setup** |
| **Testing** | **Test Execution** |
| **Maintenance** | **Test Cycle closure** |

## 32. 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 cases involve the set of steps, conditions and inputs which can be used while performing the testing tasks.** | **A set of sequential instruction that detail how to execute a core business function** |
| Test Scenario is 'What to be tested' | **Test Case is 'How to be tested'** | **One script is written to explain how to simulate each business scenario** |
| Test scenario is nothing but test procedure. The scenarios are derived from use cases. | **Test case consist of set of input values, execution precondition, expected Results and executed post-condition developed to cover certain test Condition.** | **A test script in software testing is a set of instructions that will be performed on the system under test to test that the system functions as expected.** |

33. 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**
- **Determining the scope and risks, and identifying the objectives of testing.**
- **Integrating and coordinating the testing activities into the software life cycle activities:**
- **acquisition, supply, development, operation and maintenance.**
- **Factors that affect test planning**
- **The organization's test policy**
- **Scope of the testing being performed Testing objectives**
- **Project Risks – e.g. business, technical, people**
- **Constraints – e.g. business imposed, financial, contractual etc Criticality (e.g. system/component level)**
- **Testability**
- **Availability of resources**
- **Approach: Defining the overall approach of testing (the test strategy), including the definition of the test levels and entry and exit criteria.**
- **Integrating and coordinating the testing activities into the software life cycle activities: acquisition, supply, development, operation and maintenance.**
- **All projects require a set of plans and strategies which define how the testing will be conducted.**

**There are number of levels at which these are defined:**

- **Test Policy**
- **Master Test Plan/Test Strategy**
- **Functional Test Plan**
- **System Integ Test Plan**
- **UAT Test Plan**

**34. What is priority?**

- **Priority is Relative and Business-Focused. Priority defines the order in which we should resolve a defect. Should we fix it now, or can it wait? This priority status is set by the tester to the developer mentioning the time frame to fix the defect. If high priority is mentioned then the developer has to fix it at the earliest. The priority status is set based on the customer requirements.**

**35. What is severity?**

- **Severity is absolute and Customer-Focused. It is the extent to which the defect can affect the software. In other words it defines the impact that a given defect has on the system.**

**36. Bug categories are…**

- **Security, Database, Functionality (Critical/General), UI**

**37. Advantage of Bugzilla .**

- **Advanced search capabilities**
- **E-mail Notifications**
- **Modify/file Bugs by email**
- **Time tracking**
- **Strong security**
- **Customization**
- **Localization**

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

- **Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product.**
- **Agile Methods break the product into small incremental builds. These builds are provided in iterations.**
- **Each iteration typically lasts from about one to three weeks.**
- **Every iteration involves cross functional teams working simultaneously on various areas like planning, requirements analysis, design, coding, unit testing, and acceptance testing.**