Software Testing Assignment Rupali Barchha

**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 testing ,the focus of exploratory testing is more on testing as a “thinking”’ activity, A brief description of how tests will be performed, Expected problems
* Is carried out in time boxed intervals

1. **What is traceability matrix?**

A traceability matrix is a document that details the technical requirements for a given text 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 testing?**

Software testing technique in which tests are designed to include representatives of boundary values. It is performed by QA testing teams.

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

Software testing technique that divides the input data of software unit into partitions of data from which test cases can be derived. It is usually performed by the QA teams.

1. **What is Integration testing?**

The phase in software testing in which individual software modules are combined and tested as a group. It is usually conducted by testing teams.

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

A properly designed test phases, reduces the overall level of risk in system.

There are two types of risk: (1) Project Risk (2) Product Risk

1. **What is Alpha testing?**

Types of testing a software product or system conducted at the developer’s site. Usually it is performed by end user.

1. **What is beta testing?**

Final testing before releasing application for commercial purpose, It is typically done by end-user or others.

1. **What is component testing?**

Testing technique similar to unit testing but with a higher level of integration testing is done in the context of the application instead of just directly testing a specific method. It can be performed by testing or development teams.

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

Testing based on an analysis of the specification of the functionality of component or system.

It involve black box testing and it is not concerned about the source code of the application.

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

Testing the attribute of a component or system that do not relate to functionality e.g. reliability, efficiency, usability, interoperability, maintainability and portability. It is the testing of how system works.

Testing technique which focuses on testing of software application for its non-functional requirements. Can be conducted by the performance engineers or by manual testing teams.

1. **What is GUI Testing?**

Graphical User Interface testing is the process of testing the system’s GUI of the System under test. It involves checking the screens with the control like menus, buttons, icons and all types of bars etc.

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

Ad hoc testing is informal types of testing performed without planning and documentation-the tester tries to break the system by randomly trying the system’s functionality. It is performed by the testing teams.

1. **What is load testing?**

Load testing is a kind of performance testing which determines a system’s performance under real-life load condition. This testing helps determine how the application behaves when multiple users access it simultaneously.

1. **What is stress Testing?**

Testing technique which evaluates a system or component at or beyond the limits of its specified requirements.

It is used to test the stability and reliability of the system. It is also known as endurance testing.

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

Testing technique 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’ , ‘open –box’ testing technique because here the testers require knowledge of how the software is implemented, how it works.

**Techniques of White – box testing are**

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

A method of software testing that verifies the functionality of an application without having specific knowledge of the application’s code/internal structure. Tests are based on requirements and functionality.

Specification-based testing technique is also known as ‘black-box’ or input/output driven testing technique.

**Black box testing techniques are:**

1. Equivalence partitioning
2. Boundary value analysis
3. Decision table
4. State transition testing
5. Use case testing
6. Other black box testing (Syntax or pattern testing)
7. **Mention what are the categories of defects?**

Following are the categories of defects:

1. Data quality/Database defects
2. Critical functionality defects
3. Functionality defects
4. Security defects
5. User interface defects
6. **Mention what big bang testing is?**

Big-bang integration testing is a type of integration testing that combines all the modules or components of a system into a single unit and tests them as a whole.

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

The purpose of exit criteria is to determine if a plan or project can exit to the next stage or be considered complete.

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

Regression testing is performed before each new instance of the product is deployed, guaranteeing that the program works perfectly in each setting. For instance, we need to make sure the product continues to perform properly in a production environment before we release it.

or

Regression testing should be performed to ensure that previous functionality of the application works effectively and new changes have not introduced new bugs.

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

**7 Key principles are:**

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

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

1. **Exhaustive testing is impossible**

* Testing everything including all combinations of inputs and preconditions is not possible. So, instead of doing 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, to test the all the valid combinations you would need 515 = 30,517,578,125 tests.
* This is very unlikely that the project timescales would allow for this number of tests. So, accessing and managing risk is one of the most important activities and reason for testing in any project.
* We have learned that we cannot test everything (i.e. all combinations of inputs and pre-conditions).that is we must priorities our testing efforts using a risk based approach.

1. **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.

1. **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.

By experience, you can identify such risky modules. But this approach has its own problem.

1. **The pesticide paradox**

If the same test are repeated over and over again, eventually the same test cases will no longer find new bugs.

To overcome this, the test cases need to be regularly reviewed & revised, adding new and different test cases to help find more defects.

Testers cannot simply depend on existing test techniques. He must look out continually to improve the existing methods to make testing more effective. But even after all this sweat and hard work in testing, you can never claim your product is bug-free.

1. **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. **Absence of Errors 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 Fallacy i.e Finding and fixing defects does not help if the system build is unusable and does not fulfill the user’s needs & requirements.

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

|  |  |  |
| --- | --- | --- |
| **QA(Quality Assurance)** | **QC (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 developed software with 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. |

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

|  |  |
| --- | --- |
| **Smoke** | **Sanity** |
| 1. Smoke testing is performed to ascertain that the critical functionality of the program is working fine. | Sanity testing is done to check the new functionality/bugs have been fixed. |
| 1. The objective of this testing is to verify “stability” of the system in order to with more rigorous testing. | The objective of the testing is to verify the “rationality” of the system in order to proceed with more rigorous testing. |
| 1. This testing is performed by the developers or testers | Sanity testing is usually performed by testers. |
| 1. Smoke testing is usually documented or scripted. | Sanity testing is usually not documented and is unscripted. |
| 1. It is a subset of regression testing | It is a subset of Acceptance testing |
| 1. It exercises the entire system from end to end. | It exercises only the particular component of the entire system |
| 1. Smoke testing is like General Health checkup | Sanity testing is like specialized health checkup |

1. **Difference between verification and validation**

|  |  |  |
| --- | --- | --- |
| **Criteria** | **Verification** | **Validation** |
| **Definition** | 1. 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** | 1. 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 places in its intended environment. |
| **Question** | 1. Are we building the product right? | Are we building the right product? |
| **Evaluation Items** | 1. Plans, Requirement specs, Design specs, code ,test cases | The actual product/software. |
| **Activities** | 1. Reviews ,Walkthrough , Inspection | Testing |

1. **Explain types of performance testing.**

**Types of performance testing are:**

1. **Load testing**

It is performance testing to check system behavior under load. Testing an application under heavy loads, such as testing of a website under a range of loads to determine at what point the system’s response time or fails.

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.

1. **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.

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 condition.

Stress testing is done to make sure that the system would not crash under crunch situations.

Types of stress testing are: Application stress testing, Transactional stress testing, Systemic stress testing, exploratory stress testing.

1. **Endurance testing**

Stress testing is also known as endurance testing.

1. **Spike testing**

Spike testing is a type of performance testing in which an application receives a sudden and extreme increase or decrease in load. The goal of spike testing is to determine the behavior of software application when it receives extreme variations in traffic.

1. **Volume testing:** Testing which confirms that any values that may become large over time (such as accumulated counts, logs, and data files) can be accommodated by the program and will not cause the program to stop working or degrade its operation in any manner. It is usually conducted by the performance engineer.
2. **Scalability testing**

Part of the battery of non-functional test which tests a software application for measuring its capability to scale up-be it the user load supported, the number of transactions, the data volume etc.it is conducted by the performance engineer.

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

**A mistake in coding is called error, error found by tester is called defects, 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 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. Error found by tester is called defect.

**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 a terminology of Tester.

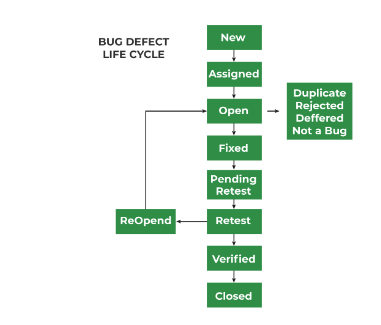
**Failure:** The inability of a system or component to perform its required functions within specified performance requirements. See: bug, crash, exception, and fault.

1. **Difference between priority and severity**

|  |  |  |
| --- | --- | --- |
| **Features** | **Priority** | **Severity** |
| **Definition** | 1. Priority is parameter to decide the order in which defects should be fixed. | Severity is a parameter to denote the impact of particular defect on the software. |
| **Purpose** | 1. Priority means how fast the defect has to be fixed. | Severity means how severe the defect is affecting the functionality. |
| **Relation** | 1. Priority relates to the scheduling of defects to resolve them in software. | Severity relates to the standards of quality. |
| **Based on** | 1. It is based on customer’s requirements. | It is based on technical aspect of the product. |
| **Who decide the defect** | 1. The product manager basically decides a defects priority level. | The testing engineer basically decides a defect’s severity level. |
| **Driving factor** | 1. It is driven by business value | It is driven by functionality. |
| **Types** | 1. There are 3 types of priorities: High, Medium, Low and Critical | There are 5 types of severities: Cosmetic, Minor, Moderate, Major and Critical |

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”.**
* The duration or time span between the first time defects is found and the time that is closed successfully, rejected, postponed or deferred is called as **‘Defect Life Cycle’.**
* When a bug is discovered, it goes through several states and eventually reaches one of the terminal states, where it becomes inactive and closed.
* The process by which the defect moves through the life cycle is depicted here.



* 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.
* This is to ensure that all necessary steps are taken to resolve or investigate that defect. For Example, a bug should not move from submitted state to resolved state without having it open.
* In a typical scenario, as soon as a bug is identified, it is logged into the bug tracking system with status as submitted. After ascertaining the validity of the defect, it is given the “open” status.

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

|  |  |
| --- | --- |
| **Functional Testing** | **Non Functional Testing** |
| Functional Testing is performed using the non-functional specification provided by the client and verifies the system against the functional requirements. | Non-functional testing checks the performance, reliability, scalability and other nonfunctional aspect of the software system. |
| Functional testing is executed first. | Nonfunctional 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 input to non-functional testing |
| Easy to do manual testing | Tough to do manual testing |
| Types of functional testing :   1. Unit testing 2. Smoke testing 3. Sanity testing 4. Integration 5. White box testing 6. Black box testing 7. User Acceptance 8. Regression testing | Types of Non-Functional testing:   1. Performance testing 2. Load testing 3. Volume 4. Stress 5. Security 6. Installation 7. Penetration 8. Compatibility 9. Migration |

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

**(2) Facebook Login page**

**https://github.com/rupalibarchha/Rupali-Manual-Testing/blob/main/Assignment/HLR\_TESTCASE\_Assignement2.xlsx**

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

|  |  |  |
| --- | --- | --- |
| **Features** | **STLC** | **SDLC** |
| **Domain** | STLC is mainly related to software testing | SDLC is mainly related to software development. |
| **Focus** | It focuses only on testing the software | Besides development other phases like testing is also included. |
| **Phases** | **STLC**  has six phases:   1. Requirement analysis 2. Test planning 3. Test case development 4. Test environment setup 5. Test execution 6. Test cycle closure | **SDLC** has six phases:   1. Requirement gathering 2. Analysis 3. Design 4. Implementation 5. Testing 6. Maintenance |
| **Number of Member** | **In STLC,** less number of members (testers) are needed. | **In SDLC,** more number of members (developers) are required for whole process. |
| **Team Involved** | **In STLC,** testing team make the plans and designs. | **In SDLC,** development team makes the plans and designs based on the requirements. |
| **Objective** | Goal of STLC is to complete successful testing of software. | Goal of SDLC is to complete successful development of software. |
| **Result** | It helps in making the software defects free | It helps in developing good quality software. |
| **End Result** | A tested software system is the end result of STLC. | Creation of reusable software system is the end result of SDLC. |

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

|  |  |  |
| --- | --- | --- |
| **Test Scenario** | **Test case** | **Test Script** |
| Is any functionality that can be tested | Is a set of actions executed to verify particular features or functionality | Is a set of instructions to test an app automatically |
| Is derived from test artifacts like business requirement specification (BRS) and software requirement specification (SRS) | Is mostly derived from test scenarios | Is mostly derived from test cases |
| Helps test the end-to –end functionality in an Agile way | Helps in exhaustive testing of an app. | Helps to test specific things repeatedly. |
| Is more focused on what to test | Is focused on what to test and how to test. | Is focused on the expected result. |
| Takes less time and fewer resources to create | Requires more resources and time | Requires less time for testing but more resources for scripts creating and updating |
| Includes an end-to-end functionality to be tested. | Includes test steps, data, expected results for testing. | Includes different commands to develop a script. |
| The main task is to check the full functionality of a software application. | The main task is to verify compliance with the applicable standards, guidelines and customer requirements | The main task is to verify that nothing is skipped, and the results are true as the desired testing plan. |
| Allows quickly assessing the testing scope. | Allows detecting errors and defects | Allows carrying out an automatic execution of test cases. |

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

A document describing the scope, approach, resources and schedule of intended test activities is called test planning.

It should cover the information of

1. Test planning
2. Test planning strategy
3. Test planning factor
4. Test planning activity
5. Exit Criteria
6. **What is priority?**

**Priority** defines the order in which we should resolve a defect. This priority status is set by the tester to the developer mentioning the time frame to fix the defect. Priority means how fast the defect has to be fixed.

1. **What is severity?**

**Severity** is a parameter to denote the impact of a particular defect on the software. Severity means how severe the defect is affecting the functionality.

1. **Bug categories are..**

Following are the categories of bug:

1. Data quality/Database defects
2. Critical functionality defects
3. Functionality defects
4. Security defects
5. User interface defects
6. **Advantage of Bugzilla**

* Advance search capabilities
* E-mail Notification
* Modify/File bugs by e-mail
* Time tracking
* Strong security
* Customization
* Localization

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

Following are the most popular methodologies in Agile Development model:

1. Scrum
2. Kanban
3. Extreme programming (XP)
4. Lean product development
5. Feature driven development (FDD)
6. Dynamic systems development method (DSDM)
7. Crystal Family
8. **Explain the difference between Authorization and Authentication in Web testing.**

|  |  |
| --- | --- |
| **Authorization** | **Authentication** |
| While in authorization process, a person’s or user’s authorities are checked for accessing the resources. | In the authentication process, the identity of users are checked for providing the access to the system. |
| In Authorization process, users or persons are validated | In the authentication process, users or persons are verified |
| Authorization process is done after the authentication process | While authentication is done before the authorization process |
| Authorization needs the user’s privilege or security levels. | Authentication needs the users login details |
| Authorization determines what permission does the user have? | Authentication determines whether the person is user or not |
| The authorization permission cannot be changed by users as these are granted by the owner of the system and only he has the access to change it | The authentication credential can be changed in part as and when required by the user. |

* **What are the common problems faced in Web testing?**

Common problem can arise during web testing are:

1. Cross-Browser compatibility issues
2. Responsive Design issues
3. Performance problem
4. Security vulnerabilities
5. Functionality issues
6. Usability issues
7. Accessibility problem
8. Integration issues
9. Database issues
10. Session and state management
11. Concurrency issues
12. **To create HLR & test case of Web based (WhatsApp web, Instagram)**

**1. WhatsApp web (https://web.whatsapp.com/)**

**2. Instagram web**

**https://github.com/rupalibarchha/Rupali-Manual-Testing/blob/main/Assignment/HLR\_TESTCASE\_Assignement2.xlsx**

1. **To create HLR and Test case on this link.**

[**https://artoftesting.com/**](https://artoftesting.com/) **(contact us page)**

**https://github.com/rupalibarchha/Rupali-Manual-Testing/blob/main/Assignment/HLR\_TESTCASE\_Assignement2.xlsx**

1. **Write a scenario of only WhatsApp chat messages**
2. **Write a Scenario of pen**
3. **Write a scenario of pen stand**
4. **Write a scenario of Door**
5. **Write a scenario of ATM**

**https://github.com/rupalibarchha/Rupali-Manual-Testing/blob/main/Assignment/Test\_Scenario\_Assignement2.xlsx**

1. **When to used usability testing?**

Usability testing is used when we want to test following:

1. Effectiveness of the system
2. Efficiency
3. Accuracy
4. User friendliness
5. **What is procedure for GUI testing?**

Following will be the procedure for GUI testing

* Check all the GUI elements for size, position, width, length and characters or numbers.
* 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 test is proper
* Check the color of the font and warning messages in 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.

1. **Write a scenario of Microwave oven**
2. **Write a scenario of coffee vending machine**
3. **Write a scenario of chair**
4. **To create scenario (positive & negative) of Gmail and flip kart**
5. **Write a scenario of wrist watch**
6. **Write a scenario of lift (Elevator)**
7. **Write a scenario of WhatsApp group (generate group)**
8. **Write a scenario of WhatsApp payment**

**https://github.com/rupalibarchha/Rupali-Manual-Testing/blob/main/Assignment/Test\_Scenario\_Assignement2.xlsx**