**Software Testing Assignment**

**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 Ad-hoc testing with purpose of find bugs.
* Is structured and rigorous.
* Is cognitively (thinking) structured as compared to procedural structure of scripted testing. This structure comes from Charter, time boxing etc.
* Is highly teachable and manageable.

**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.

**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 says that by testing just one value we have tested the partition (typically a mid-point value is used). It assumes that:
* If one value finds a bug, the others probably will too.
* If one doesn't find a bug, the others probably won't either.

**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.

**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 increases when those defects are fixed.
* The Quality of systems can be improved through Lessons learned from previous projects.
* Analysis of root causes of defects found in other projects can lead to Process Improvement.
* Process Improvement can prevent those defects reoccurring.
* Which in turn, can improve the Quality of future systems.
* Testing should be integrated as one of the Quality assurance activities.

**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.
* Alpha Testing is definitely performed and carried out at the developing organization’s location with the involvement of developers.
* It comes under the category of both White Box Testing and Black Box 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.
* It is always performed outside the organization.
* It is also the form of Acceptance Testing.
* Beta Testing (field testing) is performed and carried out by users or you can say people at their own locations and site using customer data.
* It is only a kind of Black Box Testing.

**9. What is component testing?**

* Unit tests find problems early in the development cycle.
* Unit testing is performed by using the White Box Testing method.

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

* 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 are two types of Test Approach
* Requirement Based Functional Testing
* Process Based Testing

**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.
* To address this issue, performance testing is carried out to check & fine tune system response times. The goal of performance testing is to reduce response time to an acceptable level.
* Hence load testing is carried out to check systems performance at different loads i.e., number of users accessing the system.

**12. What is GUI Testing?**

* 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.

**13. 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!
* 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: Load Testing is done in order to check when the application fails by increasing the number of users and keeping the system resources as constant.
* Load testing - It’s 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.

**15. What is stress Testing?**

* Stress Testing: Stress Testing is done in order to check when the application fails by reducing the system resources such as RAM, HDD etc. and keeping the number of users as constant**.**

**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.
* In white-box testing the tester is concentrating on how the software does it.
* Types of white box testing.
* Statement Coverage: it’s also known as line coverage or segment coverage. It only covers the true condition.
* Branch Coverage: Also known as decision coverage or all edge coverages. It covers both the true & false conditions unlikely the statement coverage.
* Condition Coverage: This is closely related to decision coverage but has better sensitivity to the control flow.

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

* The technique of testing without having any knowledge of the interior workings of the application is Black Box testing.
* Types of Black Box Testing Techniques:
* 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?**

Following are the different categories of defects

* **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, Buttons like Save, Delete, Cancel not performing their intended functions, A missing functionality (or) a feature not functioning the way it is intended to continuous execution of loops.
* **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 Alignment problems

**19. 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.

**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)
* Exit Criteria typically measures:
* Thoroughness measures, such as coverage of requirements or of code or risk

Coverage.

* Estimates of defect density or reliability measures. (e.g., how many defects open by category).
* Cost.
* Residual Risks, such as defects not fixed or lack of test coverage in certain areas.
* Schedules - such as those based on time to market

**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.

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

**1.** Testing shows presence of 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.

**2.** Exhaustive Testing is Impossible: -Testing everything including all combinations of inputs and preconditions is not possible.

**3.** Early Testing: -Testing activities should start as early as possible in the software or system development life cycle, and should be focused on defined objectives.

**4.** Defect Clustering: - a) A small number of modules contain most of the defects discovered during pre-release testing, or are responsible for the most operational failures.

b) Defects are not evenly spread in a system.

c) They are ‘clustered’.

d) In other words, most defects found during testing are usually confined to a small number of modules.

**5.** The Pesticide Paradox: -a) If the same tests are repeated over and over again, eventually the same set of test cases will no longer find any new defects.

b) To overcome this “pesticide paradox”, the test cases need to be regularly reviewed and revised, and new and different tests need to be written to exercise different parts of the software or system to potentially find more defects.

**6.** Testing is Context Dependent: -a) 3 to 10 failures per thousand lines of code (KLOC) typical for commercial software.

b) 1 to 3 failures per KLOC typical for industrial software.

c) 0.01 failures per KLOC for NASA Shuttle code!

d) Also, different industries impose different testing standards.

**7.** Absence of Errors Fallacy: -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 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. |

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

|  |  |  |
| --- | --- | --- |
| S. N | Smoke Testing | Sanity Testing |
| 1. | Smoke Testing is performed to ascertain that the critical functionalities of the new program is working fine. | Sanity Testing is done to check the functionality / bugs have been fixed. |
| 2. | The objective of this testing is to verify "stability" of the system in order to the with more rigorous testing. | The objective of the testing is to verify the "rationality" of the system in order proceed to proceed with more rigorous testing. |
| 3. | This testing is performed by the developers or tester. | Sanity testing is usually performed by testers. |
| 4. | Smoke testing is usually documented or scripted. | Sanity testing is usually not documented and is unscripted. |
| 5. | Smoke testing is a subset of Regression testing. | Sanity testing is a subset of Acceptance testing. |
| 6. | Smoke testing exercises the entire system from end to end. | Sanity testing exercises only the particular component of the entire system. |
| 7. | 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 fulfils 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 4. |

**26. Explain types of Performance testing?**

Types of Performance Testing

* Load testing
* Stress testing
* Endurance testing
* Spike testing
* Volume testing
* Scalability testing

1. Load testing: -

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

Definition: - “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 behaviour or with its internal features.
* **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.
* **Failure:** The inability of a system or component to perform its required functions within specified performance requirements. See: bug, crash, exception, and fault.

**28. Difference between Priority and Severity?**

|  |  |  |
| --- | --- | --- |
| **S. N** | **Priority** | **Severity** |
| **1.** | Priority is Relative and Business-Focused. Priority defines the order in which we should resolve a defect. | 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. |
| **2.** | Critical: Extremely urgent, resolve immediately. | Critical: The defect that results in the termination of the complete system or one or more component of the system and causes extensive corruption of the data. |
| **3.** | Major (High): The defect that results in the termination of the complete system or one or more component of the system and causes extensive corruption of the data. | High: The defect must be resolved as soon as possible because the defect is affecting the application or the product severely. The system cannot be used until the repair has been done. |
| **4.** | Medium: The defect should be resolved in the normal course of development activities. It can wait until a new build or version is created. | Moderate (Medium): The defect that does not result in the termination, but causes the system to produce incorrect, incomplete or inconsistent results then the severity will be stated as moderate. |
| **5.** | Low: The defect is an irritant which should be repaired, but repair can be deferred until after more serious defect has been fixed. | Minor (Low): The defect that does not result in the termination and does not damage the usability of the system and the desired results can be easily obtained by working around the defects then the severity is stated as minor. |

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

* Once the reported defect is fixed, the tester needs to re-test to confirm the fix. This is usually done by executing the possible scenarios where the bug can occur. Once retesting is completed, the fix can be confirmed and the bug can be closed. This marks the end of the bug life cycle.

**30. Explain the difference between Functional testing and Non-Functional testing?**

|  |  |  |
| --- | --- | --- |
| **S. N** | **Functional testing** | **Non-Functional testing** |
| **1.** | Functional testing is executed first. | Non-functional testing should be performed after functional testing. |
| **2.** | Manual testing or automation tools can be used for functional testing. | Using tools will be effective for this testing. |
| **3.** | Business requirements are the inputs to functional testing. | Performance parameters like speed, scalability are inputs to non-functional testing. |
| **4.** | Functional testing describes what the product does. | Non-functional testing describes how good the product works. |
| **5.** | Easy to do manual testing. | Tough to do manual testing. |
| **6.** | 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. |

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

|  |  |  |
| --- | --- | --- |
| **S. N** | **Software Testing Life Cycle** | **Software Development Life Cycle** |
| **1.** | Requirement Analysis | Requirements Collection/Gathering  : -Establish Customer Needs. |
| **2.** | Test Planning | Analysis Model and Specify the requirements- “What” |
| **3.** | Test case development | Design Model and Specify a Solution – “Why” |
| **4.** | Test Environment setup | Implementation Construct a Solution in Software |
| **5.** | Test Execution | Testing Validate the solution against the requirements. |
| **6.** | Test Cycle closure | Maintenance Repair defects and adapt the solution to the new requirements. |

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

|  |  |  |
| --- | --- | --- |
| **S. N** | **Test scenarios** | **Test cases** |
| **1.** | Test Scenario is ‘What to be tested’ | Test Case is ‘How to be tested’ |
| **2.** | Test scenario is nothing but test procedure. | Test case consist of set of input values, execution precondition, expected Results and executed post-condition developed to cover certain test Condition. |
| **3.** | The scenarios are derived from use cases. | Test cases are derived (or written) from test scenario. |
| **4.** | Test Scenario represents a series of actions that are associated | Test Case represents a single (low level) action by the user. |
| **5.** | Scenario is thread of operation. | Test cases are set of input and output given to the System. |

**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
* **Test Planning Factors: -**Factors which affect test planning

1) The organisation’s test policy

2) Scope of the testing being performed

3) Testing objectives

4) Project Risks – e.g., business, technical, people

5) Constraints – e.g., business imposed, financial, contractual etc

6) Criticality (e.g., system/component level)

7) Testability

8) Availability of resources

* Test plans are continuously refined

As more information becomes available

As new risks arise or others are mitigated

Not set in concrete, but changes must be carefully managed

**Test Planning Activities**

⚫ **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.

⚫ **Making decisions about**: what to test?, who do test? i.e., what roles will perform the test

activities, when and how the test activities should be done and when they should be stopped (exit criteria – see next slides), how the test results will be evaluated

⚫ Assigning resources for the different tasks defined.

⚫ **Test ware**: Defining the amount, level of detail, structure and templates for the test

documentation.

⚫ Selecting metrics for monitoring and controlling test preparation and execution, defects

resolution & risk issues.

⚫ **Process**: Setting the level of detail for test procedures in order to provide enough

information to support reproducible test preparation and execution.

**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.

For example: If the company name is misspelled in the home page of the website, then the priority is high and severity is low to fix it.

Priority can be of following types:

**Low**: The defect is an irritant which should be repaired, but repair can be deferred until after more serious defect has been fixed.

**Medium**: The defect should be resolved in the normal course of development activities. It can wait until a new build or version is created.

**High**: The defect must be resolved as soon as possible because the defect is affecting the application or the product severely. The system cannot be used until the repair has been done. **Critical**: Extremely urgent, resolve immediately

**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.

For example: If an application or web page crashes when a remote link is clicked, in this case clicking the remote link by a user is rare but the impact of application crashing is severe. So, the severity is high but priority is low.

Severity can be of following types:

**Critical**: The defect that results in the termination of the complete system or one or more component of the system and causes extensive corruption of the data. The failed function is unusable and there is no acceptable alternative method to achieve the required results then the severity will be stated as critical.

**Major** (High): The defect that results in the termination of the complete system or one or more component of the system and causes extensive corruption of the data. The failed function is unusable but there exists an acceptable alternative method to achieve the required results then the severity will be stated as major.

**Moderate** (Medium): The defect that does not result in the termination, but causes the system to produce incorrect, incomplete or inconsistent results then the severity will be stated as moderate. **Minor** (Low): The defect that does not result in the termination and does not damage the usability of the system and the desired results can be easily obtained by working around the defects then the severity is stated as minor.

**Cosmetic**: The defect that is related to the enhancement of the system where the changes are related to the look and field of the application then the severity is stated as cosmetic.

**36. Bug categories are?**

There are four types of Bugs, they are Security, Database, Functionality (Critical/General), UI

**37. Advantage of Bug zila?**

Bugzilla is an open-source issue/bug tracking system that allows

developers effectively to keep track of outstanding problems with them

product. It is written in Perl and uses MYSQL database.

* + This open bug-tracker enables users to stay connected with their clients or employees, to communicate about problems effectively throughout the data management chain.
  + Key features of Bugzilla include
  + Advanced search capabilities
  + E-mail Notifications
  + Modify/file Bugs by e-mail
  + Time tracking
  + Strong security
  + Customization Localization

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

The Agile methodology is a way to manage a project by breaking it up

into several phases. It involves constant collaboration with stakeholders and

continuous improvement at every stage. Once the work begins, teams’ cycle

through a process of planning, executing, and evaluating. Agile is a philosophy,

i.e., a set of values and principles to make a decision developing software.

There are 5 main Agile methodologies:

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

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

|  |  |  |
| --- | --- | --- |
| **S. N** | **Authorization** | **Authentication** |
| **1.** | Authorization determines what  resources a user can access. | Authentication verifies who the user  is. |
| **2.** | Authorization works through settings  that are implemented and  maintained by the organization. | Authentication works through  passwords, one-time pins, biometric information, and other information provided or entered by the user. |
| **3.** | Authorization always takes place  after authentication. | Authentication is the first step of a  good identity and access  management process. |
| **4.** | Authorization isn’t visible to or  changeable by the user. | Authentication is visible to and  partially changeable by the user. |
| **5.** | The user authorization is not visible  at the user end. | The user authentication is identified  with username, password, face  recognition, retina scan,  fingerprints, etc. |

Below are five web application testing challenges faced by web developers

during the development process.

* Integration. Integration testing exposes problems with interfaces among different program components before
* deployment. ...
* Interoperability. ...
* Security. ...
* Performance. ...

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

If possible, usability testing can and should be conducted on the

current iteration of a product before beginning any new design

work, after you’ve begun the strategy work around a brand new

site or app.

**41. What is the procedure for 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 Colour 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.