**Software Testing Assignment**

**Module-2 (Manual Testing)**

1. What is Exploratory Testing?

In exploratory testing tester focuses more on how the software actually works, testers do minimum planning and maximum execution of the software by which they get in depth idea about the software functionality, once the tester starts getting insight into the software he can make decisions to what to test next. Exploratory testing is mostly used if the requirements are incomplete and time to release the software is less.

2. What is traceability matrix?

Traceability Matrix (also known as Requirement Traceability Matrix - RTM) is a table which is used to trace the requirements during the Software development life Cycle. It can be used for forward tracing (i.e. from Requirements to Design or Coding) or backward (i.e. from Coding to Requirements). There are many user defined templates for RTM.

3. What is Boundary value testing?

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

4. What is Equivalence partitioning testing?

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

5. What is Integration testing?

Testing performed to expose defects in the interfaces and in the interactions between integrated components or systems

6. What determines the level of risk?

A properly designed test that passes, reduces the overall level of Risk in a system

7. What is Alpha testing?

First of all test newly developed hardware or software in a laboratory setting. When the first round of bugs has been fixed, the product goes into beta test with actual users. For custom software, the customer may be invited into the vendor's facilities for an alpha test to ensure the client's vision has been interpreted properly by the developer.

8. What is beta testing?

Test of new or revised hardware or software that is performed by users at their facilities under normal operating conditions. Beta testing follows alpha testing. Vendors of packaged software often offer their customers the opportunity of beta testing new releases.

9. 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.

10. What is functional system testing?

A requirement that specifies a function that a system or system component must perform

11. What is Non-Functional Testing?

Testing of those requirements that do not relate to functionality • Emphasis on non-functional requirements: • Performance • Load • Data volumes • Storage • Recovery • Usability • Stress • Security\*

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.

13. What is Adhoc testing?

Adhoc testing is an informal testing type with an aim to break the system.

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

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

White Box Testing is a testing technique in which software’s internal structure, design, and coding are tested to verify input-output flow and improve design, usability, and security. In white box testing, code is visible to testers, so it is also called Clear box testing, Open box testing, Transparent box testing, Code-based testing, and Glass box testing.

* Unit Testing
* Static Analysis
* Dynamic Analysis
* Statement Coverage
* Branch Testing Coverage
* Security Testing
* Mutation Testing

16. What is black box testing? What are the different black box testing techniques?

Testing, either functional or non-functional, without reference to the internal structure of the component or system.

* Equivalence partitioning
* Boundary value analysis
* Decision tables
* State transition testing
* Use-case Testing
* Other Black Box Testing
* Syntax or Pattern Testing

17. Mention what are the categories of defects?

* Functional defects
* Performance defects
* Usability defects
* Compatibility defects
* Security defect

18. Mention what bigbang 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.

19. What is the purpose of exit criteria?

Based on the risk assessment of the project we will set the criteria for each test level against which we will measure the “enough testing”. These criteria vary from project to project and are known as exit criteria

Exit criteria come into picture, when:

* Maximum test cases are executed with certain pass percentage.
* Bug rate falls below certain level.
* When achieved the deadlines.
* To check the test logs against the exit criteria specified in test planning.
*  To assess if more test are needed or if the exit criteria specified should be changed.
*  To write a test summary report for stakeholders.

20. When should "Regression Testing" be performed?

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.

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

21. 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 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 remaining in a system
* reduces.

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.

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

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

22. Difference between QA v/s QC v/s Tester

|  |  |  |
| --- | --- | --- |
| Quality Assurance | Quality Control | Testing |
| 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. |
| 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. |
| Process oriented activities. | Product oriented activities. | Product oriented activities. |
| Preventive activities. | It is a corrective process. | It is a preventive process. |
| 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. |

23. 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 the "stability" of the system in order to proceed 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 |
| 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 is 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 Gneral Health Check Up | Sanity Testing is like specialized health check up |

24. Difference between verification and Validation

|  |  |
| --- | --- |
| Verification | Validation |
| 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. |
| 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. |
| Are we building the product right? | Are we building the right product? |
| Plans, Requirement Specs, Design Specs, Code, Test Cases | The actual product/software. |
| Reviews, Walkthroughs, Inspections | Testing |

25. Explain types of Performance testing.

Types of Performance Testing

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

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.

Load testing gives confidence in the system & its reliability and performance.

* Load Testing helps identify the bottlenecks in the system under heavy user stress scenarios

before they happen in a production environment.

* Load testing gives excellent protection against poor performance and accommodates

complementary strategies for performance management and monitoring of a production

environment.

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 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.
* The goal of stress testing is to analyze the behavior of the system after failure. For stress

testing to be successful, system should display appropriate error message while it is under

extreme conditions.

* To conduct Stress Testing, sometimes, massive data sets may be used which may get lost

during Stress Testing.

* Testers should not lose this security related data while doing stress testing.
* The main purpose of stress testing is to make sure that the system recovers after failure

which is called as recoverability.

Endurance testing:- Stress testing is also known as endurance testing.

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 a software application when it receives extreme variations in traffic.

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.

Scalability testing :- Part of the battery of non-functional tests 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

26. What is Error, Defect, Bug and failure?

Error :- A mistake in coding is called error

Defect :- error found by tester is called defect

Bug :- defect accepted by development team then it is called bug

Failure :- build does not meet the requirements then it is failure

27. Difference between Priority and Severity

|  |  |  |
| --- | --- | --- |
| **SR No** | **Severity** | **Priority** |
| 1 | Defined by the impact on the application’s functionality. | Defined by the impact on business. |
| 2 | Category decided by testers. | Category decided by developers or product owners. |
| 3 | Deals with the technical aspects of the application. | Deals with the timeframe or order to fix the defects. |
| 4 | The value does not change with time, it’s fixed. | Value of priority is subjective and may change after comparison with other defects. |
| 5 | Defect Priority has defined the order in which the developer should resolve a defect | Defect Severity is defined as the degree of impact that a defect has on the operation of the product |
| 6 | Priority is categorized into three types | Severity is categorized into five types |
|  | Low | Critical |
|  | Medium | Major |
|  | High | Moderate |
|  |  | Minor |
|  |  | Cosmetic |
| 7 | Priority is associated with scheduling | Severity is associated with functionality or standards |
| 8 | Priority indicates how soon the bug should be fixed | Severity indicates the seriousness of the defect on the product functionality |
| 10 | Priority of defects is decided in consultation with the manager/client | QA engineer determines the severity level of the defect |
| 11 | Priority is driven by business value | Severity is driven by functionality |
| 12 | Its value is subjective and can change over a period of time depending on the change in the project situation | Its value is objective and less likely to change |
| 13 | High priority and low severity status indicates, defect have to be fixed on immediate bases but does not affect the application | High severity and low priority status indicates defect have to be fixed but not on immediate bases |
| 14 | Priority status is based on customer requirements | Severity status is based on the technical aspect of the product |
| 15 | During UAT the development team fix defects based on priority | During SIT, the development team will fix defects based on the severity and then priority |

28.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’

The different phases of Bug life cycle are,

* + New or Opened
  + Assigned
  + Fixed
  + Tested
  + Closed

29. Explain the difference between Functional testing and NonFunctional testing

|  |  |  |
| --- | --- | --- |
| SR No | Functional testing | Non-functional testing |
| 1 | It is performed before non-functional testing. | It is performed after the functional testing. |
| 2 | It is based on customer’s requirements. | It focusses on customer’s expectation. |
| 3 | It is easy to define functional requirements. | It is difficult to define the requirements for non-functional testing. |
| 4 | Helps to validate the behavior of the application. | Helps to validate the performance of the application. |
| 5 | Carried out to validate software actions. | It is done to validate the performance of the software. |
| 6 | Functional testing is carried out using the functional specification. | This kind of testing is carried out by performance specifications |
| 7 | Functional testing is easy to execute by manual testing. | It’s very hard to perform non-functional testing manually. |
| 8 | It describes what the product does. | It describes how the product works. |
| 9 | Check login functionality. | The dashboard should load in 2 seconds. |
| 10 | Examples of Functional Testing Types | Examples of Non-functional Testing Types |
|  |  |  |
|  | Unit testing | Performance Testing |
|  | Smoke testing | Volume Testing |
|  | User Acceptance | Scalability |
|  | Integration Testing | Usability Testing |
|  | Regression testing | Load Testing |
|  | Localization | Stress Testing |
|  | Globalization | Compliance Testing |
|  | Interoperability | Portability Testing |
|  |  | Disaster Recover Testing |

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

|  |  |  |
| --- | --- | --- |
| **SR No** | **SDLC** | **STLC** |
| 1 | Development Life Cycle | Testing Life Cycle |
| 2 | The main object of SDLC life cycle is to complete successful development of the software including testing and other phases. | The only objective of the STLC phase is testing. |
| 3 | In SDLC the business analyst | In STLC, the QA team analyze requirement documents like functional and non-functional documents and create System Test Plan |
| 4 | gathers the requirements and |
| 5 | create Development Plan |
| 6 | In SDLC, the development team creates the high and low-level design plans | In STLC, the test analyst creates the Integration Test Plan |
| 7 | The real code is developed, | The testing team prepares the test environment and executes them |
| 8 | and actual work takes place as per the design documents. |
| 9 | SDLC phase also includes post-deployment supports and updates. | Testers, execute regression suits, usually automation scripts to check maintenance code deployed. |

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

|  |  |  |  |
| --- | --- | --- | --- |
| **SR No** | **TEST SCENARIO** | **TEST CASE** | **TEST SCRIPT** |
| 1 | It is a process to test an application with all possible ways. | It is a step by step by procedure that is used to test an application | The term Test Script is used in automation testing environment. |
| 2 | Test scenarios are an input for the creation of test cases. | The term Test Case is used in the manual testing environment. | It is done by scripting format. |
| 3 | Test scenario covers all possible cases to test an application. | It is done manually. | It is developed in the form of scripting. |
| 4 | It reduces the complexity. | It is developed in the form of templates. | In Test Scrip,t we can use different commands to develop script. |
| 5 | Test scenario can be a single or a group of test cases. | Test case template includes Test Suit ID, Test Data, Test procedure, Actual results, Expected results etc. | It is also used to test an application. |
| 6 | By writing scenarios it will be easy to understand the functionality of an application. | Is used to test an application. | Once we develop, the script will run it multiple times until the requirement is changed. |
| 7 | These are one line statements to explain what we are going to test. | It is the base form to test an application in sequence. |  |

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

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

A test plan is a detailed document that describes the test strategy, objectives, schedule, estimation, deliverables, and resources required to perform testing for a software product.

Test Plan helps us determine the effort needed to validate the quality of the application under test.

The test plan serves as a blueprint to conduct software testing activities as a defined process, which is minutely monitored and controlled by the test manager.

1. Analyze the product
2. Design the Test Strategy
3. Define the Test Objectives
4. Define Test Criteria
5. Resource Planning
6. Plan Test Environment
7. Schedule & Estimation
8. Determine Test Deliverables

33. What are the different Methodologies in Agile Development Model?

1.Kanban

2.Scrum

3.Extreme Programming (XP)

4.Feature-driven development (FDD)

5.Dynamic Systems Development Method (DSDM)

6.Crystal

7.Lean

34. Explain the difference between Authorization and Authentication in Web testing.

|  |  |  |
| --- | --- | --- |
| **SR No** | **Authentication** | **Authorization** |
| 1 | Authentication is the process of identifying a user to provide access to a system. | Authorization is the process of giving permission to access the resources. |
| 2 | In this, the user or client and server are verified. | In this, it is verified that if the user is allowed through the defined policies and rules. |
| 3 | It is usually performed before the authorization. | It is usually done once the user is successfully authenticated. |
| 4 | It requires the login details of the user, such as user name & password, etc. | It requires the user's privilege or security level. |
| 5 | Data is provided through the Token Ids. | Data is provided through the access tokens. |
| 6 | Authentication credentials can be partially changed by the user as per the requirement. | Authorization permissions cannot be changed by the user. The permissions are given to a user by the owner/manager of the system, and he can only change it. |
| 7 | **Example:** Entering Login details is necessary for the employees to authenticate themselves to access the organizational emails or software. | **Example:** After employees successfully authenticate themselves, they can access and work on certain functions only as per their roles and profiles. |
|  |  |  |

35. What are the common problems faced in Web testing?

### Insufficient testing for browser compatibility

### Failing to conduct thorough functional testing across mobile

### Failing to conduct thorough functional testing across desktop

### Poor data security

### Failing to provide an intuitive experience

### Not performing testing frequently enough

### Leaving digital accessibility to the last minute

### Releasing new features breaks the existing live system

### Localisation and the global experience

### The most common bugs

36. To create HLR & TestCase of WebBased (WhatsApp web , Instagram) 1. WhatsApp Web : <https://web.whatsapp.com/>