**Software Testing Assignment**

**Module – 2 (Manual Testing)**

1. **What is Exploratory testing?**

* Exploratory testing is a type of software testing where the test cases are not created in advance but tester check system on the fly.
* The test design and test execution activities are performed in parallel typically without formally documenting the test conditions, test cases or test scripts.

1. **What is traceability Matrix?**

* A Traceability Matrix is a table type document that is used in the development of software application to trace requirements.
* A Traceability Matrix is a Document that co-relates any two baseline documents that require a many to many relationship to check the completeness of the relationship.

1. **What is Boundary value testing?**

* Boundary value testing/analysis is a methodology for designing test cases that concretes software testing effort on cases near the limit of valid ranges.
* Boundary value testing/analysis is one of the widely used cases design technique for black box testing. It is used to test boundary values because input values near the boundary have higher chances of error.

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

* **Equivalence Partitioning is the process of defining the optimum number of test by reviewing documents such as the functional design specification and detailed design specification, and identifying each input condition within function, select input data that is representative of all other data that would likely invoke the same process for that particular condition.**
* **Equivalence Partitioning** or Equivalence Class Partitioning is type of black box testing technique which can be applied to all levels of software testing like unit, integration, system, etc. In this technique, input data units are divided into equivalent partitions that can be used to derive test cases which reduces time required for testing because of small number of test cases.

1. **What is integration testing?**

* Integration testing is 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 group.

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

* The importance of a risk as defined by its characteristics impact and likelihood. The level of risk can be used to determine the intensity of testing to be performed. A risk level can be expressed either qualitatively (e.g., high, medium, low) or quantitatively.

1. **What is Alpha testing?**

* Alpha Testing is always performed by the developers at the software development site Sometimes it is performed by independent testing team.
* **Alpha Testing** is a type of software testing performed to identify bugs before releasing the software product to the real users or public. It is a type of acceptance testing. The main objective of alpha testing is to refine the software product by finding and fixing the bugs that were not discovered through previous tests.

1. **What is beta testing?**

* Beta testing is a type of **User A**cceptance **T**esting among the most crucial testing, which performed before the release of the software. Beta Testing is a type of Field Test. This testing performs at the end of the **software** testing life cycle. This type of testing can be considered as external user acceptance testing. It is a type of salient testing. Real users perform this testing. This testing executed after the alpha testing. In this the new version, beta testing is released to a limited audience to check the accessibility, usability, and functionality, and more.
* Beta testing is the last phase of the testing, which is carried out at the client's or customer's site.

1. **What is component testing?**

* Component testing is the testing of individual software components.
* Component testing is used to test all the components separately as well as the usability testing; interactive valuation is also done for each specification component. It is further known as module testing or program testing or unit testing.

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

* Functional system testing is based on analysis of the specification of the functionality of a component or system.
* Functional System testing is used to verify the functionality of the software application, whether the function is working according to the requirement specification.

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

* Non Functional testing is testing the attributes of a component or system that do not relate to functionality, e.g. reliability, efficiency, usability, interoperability, maintainability and portability.

1. **What is GUI testing?**

* Graphical User Interface (GUI) testing is the process of testing the system’s GUI of the System under Test. GUI testing involves checking the screens with the controls like menus, buttons, icons, and all types of bars – tool bar, menu bar, dialog boxes and windows etc.

1. **What is Adhoc testing?**

* Adhoc testing is an informal testing type with an aim to break the system and it is not create test cases.
* Main aim of this testing is to find defects by random checking and Adhoc testing can be achieved with the testing technique called Error Guessing.

1. **What is load testing?**

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

1. **What is stress testing?**

* Stress testing - System is stressed beyond its specifications to check how and when it fails. Performed under heavy load like putting large number beyond storage capacity, complex database queries, continuous input to system or database load.
* Stress Testing is done to make sure that the system would not crash under crunch situations.
* Stress testing is also known as endurance testing.

1. **What is white box testing and list the type 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.
* **Types or Techniques of White box testing** 
  + **Structure Based(**Based on code and design of the system**)**
    - **Statement coverage**
    - **Brach coverage**
    - **Decision coverage**
  + **Experience Based(**Based on the knowledge of the tester**)**
    - **Grey Box**
    - **Error Guessing**
    - **Exploratory testing**

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

* Black-box testing is either functional or non-functional, without reference to the internal structure of the component or system.
* The technique of testing without having any knowledge of the interior workings of the application is Black Box testing.
* Black Box testing techniques
  + **Specification Based(**Based on requirements**)**
    - **Equivalence Partitioning**
    - **Boundary value analysis**
    - **Decision table**
    - **State transition testing**
    - **Use case testing**

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

* Types/Categories of Defects or Bug
  + **Data quality/Data base defect**
  + **Critical Functionality defect**
  + **Functionality defect**
  + **Security defect**
  + **User interface defect**

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

* In Big Bang integration testing all components or modules is integrated simultaneously, after which everything is tested as a whole.

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

* Purpose of exit criteria is to define when to stop testing either at the end of all testing – i.e. product go live

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

* Regression testing has been performed at 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.

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

* **7 key principles of Software Testing.**

1. **Testing shows presence of error.**

* 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.

1. **Exhaustive testing is impossible.**

* Testing everything including all combinations of inputs and preconditions is not possible.

1. **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 lifecycle.

1. **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.
* In other words, most defects found during testing are usually confined to a small number of modules similarly, most operational failures of a system are usually confined.

1. **The pesticide paradox**

* Testing is basically 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.

1. **Testing is context depended.**

* Testing is basically 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.

1. **Absence of error 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.

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No.** | **QA** | **QC** | **TESTER** |
| 1. | Activity which ensure the implementation of process, procedures and standers in context to verification of developed software intended requirements. | Activity which ensure the verification of the developed software with respect to documented requirements. | Activity which ensure the identification of bugs, error and defect in software. |
| 2. | Process oriented activity. | Product oriented activity. | Product oriented activity. |
| 3. | Preventive activity. | It’s a corrective activity. | Preventive activity. |
| 4. | It is subset of STLC. | It is subset of QA. | It’s subset of QC. |

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

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Smoke** | **Sanity** |
| 1. | Smoke testing is performed to ascertain that critical functionality of the program works fine. | Sanity testing is performed to check new functionality is working well or not. |
| 2. | This testing is performed by developer or by tester | This activity is performed by the tester. |
| 3. | Smoke testing is subset of regression testing. | Sanity testing is a subset of acceptance testing. |
| 4. | Smoke testing exercises the entire system from end to end. | Sanity testing exercises only the particular component of the system |
| 5. | Smoke testing is usually documented or scripted. | Sanity testing is not documented and is unscripted. |

1. **Difference between Verification and Validation.**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Verification** | **Validation** |
| 1. | The process of evaluating work-products 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. |
| 2. | 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. |
| 3. | It is done before coding. | It is done after coding |
| 4. | Activity of verification is Reviews,  Walkthroughs and Inspections. | Activity of validation is Testing. |

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

* **Types of Performance testing**

1. **Load testing**
2. **Stress testing**
3. **Endurance testing**
4. **Spike testing**
5. **Volume testing**
6. **Scalability testing**
   * + 1. **Load Testing:** Load testing is a performance testing to check system behavior under load. Testing an application under heavy loads, such as testing of a web site under a range of loads to determine at what point the system’s response time degrades or fails.

**2. 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.

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

Stress testing is also known as endurance testing.

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

* A mistake in coding is called error, error found by tester is called defect, defect accepted by development team then it is called bug, build does not meet the requirements then it is failure.

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 it 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 next slide.

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

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Functional** | **Non-Functional** |
| 1. | Functional testing is performed using functional specification provided by the client and verifies the system against functional requirement. | Non-functional testing performs reliability, scalability and other non-functional aspect of the system. |
| 2. | Functional testing is executed first. | Non-functional testing should be performed after functional testing. |
| 3. | Manual testing or automation tools can be used for functional testing. | Using tools can be effective for non-functional testing. |
| 4. | Business requirement are the input to the functional requirement. | Performance parameters like speed, scalability are input of the non-functional testing. |
| 5. | **Types of functional testing**   * Unit testing * Smoke testing * Integration testing * Sanity testing * White box testing * Black box testing * User acceptance testing * Regression testing | **Types of non-functional testing**   * Performance testing * Load testing * Volume testing * Stress testing * Security testing * Installation testing * Penetration testing * Compatibility testing * Migration testing |

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

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **SDLC** | **STLC** |
| 1. | It is complete process of Software development from requirement analysis to maintenance. | It is process of software testing only. |
| 2. | In this process developer, testers, customer, end users, project head are involved. | In this process usually involves QA tester. |
| 3. | In SDLC we try to understand and completes the build according to customer’s needs and expectation. | In STLC we test the build so can achieve the quality of software to be developed. |
| 4. | After the testing phase the software is resealed to end to end user | Smoke and sanity is performed after deploying the software. |
| 5. | It has following phases.   * Requirement gathering * Analysis * Design * Implementation * Testing * Maintenance | it has following phases   * Requirement analysis * Test planning * Test case development * Test environment setup * Test execution * Test cycle closer |

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No.** | **Test Scenarios** | **Test cases** | **Test scripts** |
| 1. | 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 | 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. |

1. **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.
* Test plan covered information
* Determining the scope and risks, and identifying the objectives of testing.
* 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 lifecycle activities.
* Acquisition, supply, development, operation and maintenance.
* Making decisions about what to test, what roles will perform the test activities, how the test activities should be done, and how the test results will be evaluated?
* Defining the amount, level of detail, structure and templates for the test documentation.

1. **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.

1. **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 an user is rare but the impact of application crashing is severe. So the severity is high.

1. **Bug categories are…**

* Types/Categories of Defects or Bug
* **Data quality/Data base defect**
* **Critical Functionality defect**
* **Functionality defect**
* **Security defect**
* **User interface defect**

1. **Advantage of Bugzila.**

* Open source, free bug tracking tool.
* Automatic Duplicate Bug Detection.
* Search option with advanced features.
* Move Bugs between Installs.
* Multiple Authentication Methods (LDAP, Apache server).
* Time Tracking.
* Automated bug reporting; has an API to interact with system.

1. **Difference between Priority and Severity.**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Priority** | **Severity** |
| 1. | 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. |
| 2. | Priority is categorized into three types Low, Medium, High. | Severity is categorized into five types Critical, Major, Moderate, Minor, Cosmetic. |
| 3. | Priority is associated with scheduling. | Severity is associated with functionality or standards. |
| 4. | Priority is driven by business value. | Severity is driven by functionality. |
| 5. | Priority status is based on customer requirements. | Severity status is based on the technical aspect of the product. |

1. **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.
* At the end of the iteration a working product is displayed to the customer and important stakeholders.

1. **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 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 required the login details of the user, such as username and Password etc. | It requires the user's privilege or security level. |
| 5. | 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. |

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