**Software Testing Assignment**

Module 2

(Manual Testing)

1. What is exploratory Testing?

Exploratory testing is a concurrent process where test design, execution and logging happen simultaneously.

1. What is Traceability Metrics?

To protect against changes you should be able to trace back from every System component to the original requirement that caused its presence.

1. 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.

1. What is Equivalence Portioning Testing?

Equivalence partitioning is the process of defining the optimum number of tests by Reviewing documents such as the Functional Design Specification and Detailed Design Specification, and identifying each input condition within a function Selecting input data that is representative of all other data that would likely invoke the same process for that particular condition.

1. What determines the level of risk?

When deciding on an investment strategy, one of the key factors to consider is your risk level, or how much risk you are willing to accept with your investment. Meanwhile, your risk level is the amount of financial risk that you are able to take on given your current financial situation. Whereas your risk tolerance has to do with your comfort level in taking on risk under current conditions, your risk capacity depends on how much you can afford to invest and the returns that you will need to generate to meet your goals.

1. What is Alpha testing?

Alpha Testing is performed by the Testers within the organization whereas Beta Testing is performed by the end users.

1. What is Beta testing?

Beta testing is a term that originated in software development and refers to a step in the process focused on gathering feedback from real-world users about features and products.

1. What is component testing?

 Component testing is a method of validating the single component before testing the whole application.

1. What is Functional system testing?

Functional testing is performed using the functional specification provided by the client and verifies the system against the functional requirements.

1. What is Non Functional system testing?

Non-Functional testing checks the Performance, reliability, scalability and other non-functional aspects of the software system.

1. What is GUI testing?

GUI Testing is a process of testing the application’s graphical user interface to ensure proper functionality as per the specifications.

1. What is Ad hoc testing?

**Ad hoc testing is a kind of software testing approach that involves exploring the software application in an unstructured and informal manner.**

1. What is Load testing?

Load testing is a kind of performance testing which determines a system’s performance under real-life load conditions.

1. What is Stress testing?

Stress testing is a computer simulation technique used to test the resilience of institutions and investment portfolios against possible future financial situations.

15.What is white box testing and list the types of white box testing.

Testing based on an analysis of the internal structure of the component or system.

list the types of white box testing.

### 1. Statement Coverage

### 2. Branch Coverage

### 3. Path Coverage

### 4. Decision Coverage

### 5. Condition Coverage

### 6. Multiple Condition Coverage

### 7. Finite State Machine Coverage

### 8. Control Flow Testing

### 16. 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.

### • Equivalence partitioning

### • Boundary value analysis

### • Decision tables

### • State transition testing

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

### 1. arithmetic defect

### 2.logical defect

### 3.multithreading defect

### 4.interface defect

### 5.performence defect

### 18.Mention what Big bang testing is?

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

### 19.What is the purpose of exit criteria?

### The purpose is to identify when to stop testing is exit criteria.

### 20.When should “regression testing” be performed?

 regression testing should be performed whenever your codebase has been modified or altered in any way as well as to verify any previously discovered issues marked as fixed. The more often the better: frequent partial regression testing will help your developers fix the reported defects on time, and your project to avoid any long-term pitfalls and technical debt caused by poor code quality. However, even though an occasional project might have the resources to perform the tests after the slightest changes have been introduced to the codebase, for most projects designing and maintaining such a multiplicity of regression tests may simply be infeasible. Therefore, it is important to understand when you need to start regression testing.

The most common reason to run regression tests is the introduction of new functionality. It is hard for developers to follow every thread in the code when modifying it, and there’s always a risk of compatibility issues with the existing code.

21.What is 7 key principle? Explain in detail.

## 1) Exhaustive testing is not possible

## 2) Defect Clustering

## 3) Pesticide Paradox

## 4) Testing shows a presence of defects

## 5) Absence of Error – fallacy

## 6) Early Testing

## 7) Testing is context dependent

## 1) Exhaustive testing is not possible

 Exhaustive testing is not possible. Instead, we need the optimal amount of testing based on the risk assessment of the application. and the million dollar question is, how do you determine this risk? To answer this let’s do an exercise

2) 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.

## 3) Pesticide Paradox

Repetitive use of the same pesticide mix to eradicate insects during farming will over time lead to the insects developing resistance to the pesticide Thereby ineffective of pesticides on insects. The same applies to software testing. If the same set of repetitive tests are conducted, the method will be useless for discovering new defects.

## 4) Testing shows a presence of defects

Hence, testing principle states that – Testing talks about the presence of defects and don’t talk about the absence of defects. i.e. Software 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.

But what if, you work extra hard, taking all precautions & make your software product 99% bug-free. And the software does not meet the needs & requirements of the clients.

### 5) Absence of Error – 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 is a 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.

### 6) Early Testing

### 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. More on this principle in a later training tutorial.

## 7) 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.

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

|  |  |  |
| --- | --- | --- |
| **S.N** | **Quality Assurance** | **Quality Control** |
| **1.** | **Activity which ensures the implementation of processes, products and standards in context to verification of develop software and intended requirements.** | **Activities which ensures the verifications of developed software with respect to documented.** |
| **2.** | **Focuses on processes and products rather than conducting actual testing in the system.** | **Focuses on actual testing by executing software with intend to identify bug/defects through implementation of procedures and process.** |
| **3.** | **Process oriented activity.** | **Product oriented activity.** |
| **4.** | **Preventive activities.** | **It is a corrective process.** |
| **5.** | **It is a subset of STLC.** | **It can be considered as the subset of QA.** |

23.Difference between Smoke and Sanity.

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

24. Difference between Verification and Validation.

|  |  |  |
| --- | --- | --- |
| S.N | Verification | Validation |
| 1. | The process of evaluating work-products of a development phase to determine whether they meet the specified requirement for that phase. | The process of evaluating software during or at the end of the 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 fulfill its intended us when placed in its intended environment. |
| 3. | Are we building the product right? | Are we building the right product? |
| 4. | Places ,Requirement Specs, Design specs, Code Test cases. | The actual product/software. |
| 5. | Reviews Walkthroughs Inspections. | Testing |

25.Explian types of Performance Testing.

## Load testing

Load testing is a type of testing which involves evaluating the performance of the system under the expected workload. A typical load test includes determining the response time, throughput, error rate, etc during the course of the load test.

## Stress testing

Stress testing is a type of performance testing where we evaluate the application’s performance at a load much higher than the expected load. Another aspect of the stress testing is to determine the break-point of the application, the point at which the application fails to respond in the correct manner.

## Endurance testing

Endurance testing is also known as ‘Soak Testing’. It is done to determine if the system can sustain the continuous expected load for a long duration. Issues like memory leakage are found with endurance testing.

## Spike testing

In spike testing, we analyze the behavior of the system on suddenly increasing the number of users. It also involves checking if the application is able to recover after the sudden burst of users.

## Volume testing

The volume testing is performed by feeding the application with a high volume of data. The application can be tested with a large amount of data inserted in the database or by providing a large file to the application for processing. Using volume testing, we can identify the bottleneck in the application with a high volume of data.

26.What is error, defect, bug and failure?

Bugs:

defect accepted by development team then it is called bug.

Error:

A mistake in coding is called error.

Defect:

error found by tester is called defect

Failure:

build does not meet the requirements then it is failure.

27. Difference between Priority and severity.

|  |  |  |
| --- | --- | --- |
| S.N | Priority | Severity |
| 1. | Severity is a parameter to denote the impact of a particular defect on the software. | Priority is a parameter to decide the order in which defects should be fixed. |
| 2. | Severity means how severe defect is affecting the functionality. | Priority means how fast defect has to be fixed |
| 3. | Severity is related to the quality standard. | Priority is related to scheduling to resolve the problem. |
| 4. | Testing engineer decides the severity level of the defect. | Product manager decides the priorities of defects. |
| 5. | Its value is objective. | Its value is subjective. |
| 6. | Its value doesn’t change from time to time. | Its value changes from time to time. |
| 7. | Severity is of 5 types: Critical, Major, Moderate, Minor, and Cosmetic. | Priority is of 3 types: Low, Medium, and High |

28. What is Bug Life Cycle.

**The bug life cycle** is also known as the **Defect life cycle**. In the Software Development Process, the bug has a life cycle. The bug should go through the life cycle to be closed. The bug life cycle varies depends upon the tools used, and the process followed in the organization.

29. Difference between Functional testing and Non-functional Testing.

|  |  |  |
| --- | --- | --- |
| S.N | Functional Testing | Non Functional Testing |
| 1. | 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. |
| 2. | Functional Testing is executed first. | Non-Functional should be performed after functional testing. |
| 3. | Manual testing or automation tools can be used for functional testing. | Using tools will be effective for this testing. |
| 4. | Business requirements are the inputs to functional testing. | Performance parameters like speed scalability are inputs to non-functional testing. |
| 5. | Functional Testing is describes what the product does. | Non-Functional Testing is describes how good the product work. |
| 6. | Easy to do manual testing. | Tough to do manual testing. |
| 7. | Types of functional testing  1.Unit testing  2.Smoke testing  3.Sanity testing  4.Integration testing  5.White box testing  6.Black box testing  7.User acceptance testing  8.Regrassion testing | Types of Non-functional testing  1.Load testing  2.performance testing  3.Volume testing  4.Stress testing  5.Secuirity testing  6.Installation testing  7.Panastration testing  8.Competibility testing  9.migration testing |