**MODULE 2 ASSIGNMENT**

**Q.1What is exploratory testing?**

Exploratory testing is a concurrent process where test design ,execution ,& login happen simultaneously testing is often not recorded Makes use of experience, heuristics and test patterns

Testing is based on a test charter that may include

Scope of the testing (in and out)

The focus of exploratory testing is more on testing as a “thinking”

activity.

A brief description of how tests will be performed

Expected problems.

Is carried out in time boxed intervals.

More structure than error gussing.It is highly teachable & manageable.

**Q.2 What is traceability matrix?**

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

Types of traceability matrix

1.Forward traceability

2. Backward traceability

3.Bidirectional traceability

**Q.3 What is Boundary value testing?**

Boundary value analysis is methodology for designing test cases that concentrates software testing effort on cases near the limit of valid ranges boundary value analysis is a method which refines equivalence petitioning boundary value analysis generates test cases that highlight errors better than equivalence partying.

The trick is concentrate software testing effort at the extreme end of the equivalence.

**Q 4 What is Equivalence partitioning testing?**

Aim is to treat groups of inputs and to select one represented input to test them all.

EP can be used for all level of testing.

Epis 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. If we want to test the following IF statement: “If value is between 1 and 100 (inclusive) (e.g value >=1 and value <=100)

**Q,5 What is Integration testing?**

Testing perform to expose defects in the interfaces and in the interactions between integrated components or system.

The purpose of this testing is to expose faults in the interaction between integrated units .

Test drivers and test stubs are used to assist in integration testing.

Integration testing test the integration or interfaces between component,interections,to different parts of the system such as an operating system .

**Q.6. What is Component testing?**

The testing of individual software component.it is level of software testing process where individual unit of a software are tested .The purpose is to validate that each unit of the software perform as designed .

Unit testing is the first level of testing and is performed prior to

Integration Testing.

Sometimes known as Unit Testing, Module Testing or Program Testing.

Component can be tested in isolation – stubs/drivers may be employed.

Unit testing frameworks, drivers, stubs and mock or fake objects are used

to assist in unit testing.

Functional and Non-Functional testing.

Unit tests are typically written and run by software developers to

ensure that code meets its design and behaves as intended with debugging

tool.

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

Testing based on an analysis of the specification of the functionality of a component. Functional testing verifies that each function of the software. Application operates in conformance with the requirement specification.

,eg requirements specification ,use cases ,functional specification or may be undocumented

**Q 8 .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. May be performed at all Test levels (not just Non Functional Systems Testing) ‘.Measuring the characteristics of the system/software that can be quantified on a varying scale- e.g. performance test scaling Non-functional testing includes, but is not limited to, performance testing, load testing, stress testing, usability testing, maintainability testing, reliability testing and portability testing.

**Eg.1 web Based testing**

**2.Desktop based testing**

**3.Mobile based testing**

**4.Game based testing**

**9. What is GUI Testing?**

GUI is the abbreviation of “Graphical User Interface”. It contains several visual elements, such as buttons, text boxes, menus, icons and all types of bars – tool bar, menu bar and windows etc. GUI testing refers to the validating UI functions or features of an application that are visible to the users, and they should comply with business requirement.

**10.What is Adhoc testing?**

Adhoc testing is an informal testing type with an aim to break the system. Main aim of this testing is to find the defects by random checking. Adhoc testing can be achieved with the testing technique called Error Guessing.

**11. What is 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.

**12.What is 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 l

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

White box testing is a software testing technique that involves testing the internal structure and workings of software application. The tester has access to source code and uses this knowledge to design test cases that can verify the correctness the software at the code level.

TECHNIQUES OF WHITE BOX TESTING:

* Branch Condition testing
* Modified Condition Decision testing
* Dataflow testing
* Linear Code Sequence
* Branch Condition Combination testing

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

**BLACK BOX TESTING**: Testing, either functional or non-functional, without reference to the internal structure of the component or system. The testers have no knowledge of how the system or component is structured inside the box.

**TECHNIQUES OF BLACK BOX TESTING: Techniques of Black Box Testing**

**There are four specification-based or black-box**

**technique:**

**Equivalence partitioning**

**Boundary value analysis**

**Decision tables**

**State transition testing**

Use-case Testing

Other Black Box Testing

Syntax or Pattern Testing

TECHNIQUES OF WHITE BOX TESTING:

-Branch Condition testing

-Modified Condition Decision testing

-Dataflow testing

-Linear Code Sequence

-Branch Condition Combination testing

**Q15.Mention what big bang testing ?**

In big bang integration testing all component 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.

The major disadvantage is that in general it is time consuming and

difficult to trace the cause of failures because of this late integration.

Here all component are integrated together at once, and then

Tested.

16.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 and End of phase of testing.

**17.When should "Regression Testing" be performed?**

Regression testing should be carried out when the system is stable and the system or the environment changes, when testing bug-fix releases as part of the maintenance phase.

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

**General Testing Principles**

1. **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.
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 small number of modules contain most of the defects discovered during pre-release testing, or are responsible for the most operational failures.
5. **The Pesticide Paradox:** - 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:** - Testing is basically context dependent. Testing is done differently in different contexts and Different kinds of sites are tested differently.

**Absence of Errors Fallacy:** - Even after defects have been resolved it may still be unusable and/or does not fulfil the users’ needs and

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

|  |  |  |
| --- | --- | --- |
| Quality Assurance (QA) | Quality Control (QC) | Tester |
| 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. |
|  | QC can be considered as the subset of Quality Assurance. | Testing is the subset of Quality Control. |
| It is a subset of Software Test Life Cycle (STLC). |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Smoke Testing** | **Sanity Testing** |  |  |
| 1. 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 |  |  |
| 1. Smoke testing is usually documented or scripted | Sanity testing is usually not documented and  is unscripted |  |  |
| 1. Smoke testing is a subset of Regression testing | Sanity testing is a subset of Acceptance testing |  |  |
| 1. Smoke testing performed by developers or testers | Sanity testing performed by testers |  |  |
| 1. Smoke testing is may be stable/unstable | Sanity testing is always stable |  |  |
| 1. Smoke testing’s main goal is to verify “stability” | Sanity testing’s main goal is to verify “rationality” |  |  |
| 1. Smoke testing is like General Health Check Up | Sanity Testing is like specialized health check up |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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

|  |  |  |  |
| --- | --- | --- | --- |
| **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 |  |  |
| 1. Functional testing is executed first | Non-functional testing should be performed after functional testing |  |  |
| 1. Manual testing or automation tools can be used for functional testing | Using tools will be effective for this testing |  |  |
| 1. Business requirements are the inputs to functional testing | Performance parameters like speed, scalability are inputs to non-functional testing. |  |  |
| 1. Functional testing describes what the product does | Nonfunctional testing describes how good the product works |  |  |
| 1. Easy to do manual testing | Tough to do manual testing |  |  |
| 1. Types of Functional testing are   ∙ Unit Testing  ∙ Smoke Testing  ∙ Sanity Testing  ∙ Integration Testing  ∙ White box testing  ∙ Black Box testing  ∙ User Acceptance testing  ∙ Regression Testing | Types of Nonfunctional testing are  ∙ Performance Testing  ∙ Load Testing  ∙ Volume Testing  ∙ Stress Testing  ∙ Security Testing  ∙ Installation Testing  ∙ Penetration Testing  ∙ Compatibility Testing  ∙ Migration Testing |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Difference between verification and Validation**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | **Criteria** | **Verification** | **Validation** | | Definitions | 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. | | Objectives | 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. | | Questions | 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. | |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| |  |  |  | | --- | --- | --- | |  | **SDLC** | **STLC** | | Full form | Software Development Life Cycle | Software Testing Life Cycle | | Objectives | 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. | | Requirement Gathering | In SDLC the business analyst gathers the requirements and create Development Plan | In STLC, the QA team analyze requirement documents like functional and non-functional documents and create System Test Plan | | Design | In SDLC, the development team creates the high and low-level design plans | In STLC, the test analyst creates the Integration Test Plan | | Coding | The real code is developed, and actual work takes place as per the design documents. | The testing team prepares the test environment and executes them | | Maintenance | SDLC phase also includes post-deployment supports and updates. | Testers, execute regression suits, usually automation scripts to check maintenance code deployed | |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

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

* Test Script: - A set of sequential instruction that detail how to execute a core business function
* Test Scenario: - A Scenario is any functionality that can be tested. It is also called Test Condition, or Test Possibility.

Test Case: - Test cases involve the set of steps, conditions and inputs which can be used while performing the testing tasks

**Key features of Bugzilla includes:**

* Advanced search capabilities
* E-mail Notifications
* Modify/file Bugs by e-mail
* Time tracking
* Strong security
* Customization
* Localization
* Bugzilla is an open-source issue/bug tracking system that allows developers effectively to keep track of outstanding problems with their product. It is written in Perl and uses MYSQL database.

**Performance testing: -** Software performance testing is a means of quality assurance (QA). It involves testing software applications to ensure they will perform well under their expected workload.

**Types of performance testing:** -

1. **Stress testing: -** System is stressed beyond its specifications to check how and when it fails. Performed under heavy load like putting large number beyond storage capacity, complex database queries, continuous input to system or database load.

* **Stress testing tools**

Stress Tester

Neo Load

App Perfect

1. **Load testing: –** It’s 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.

* **Load testing tools**

Load runner

Web Load

Astra Load Test

Review’s Web Load

Studio, Rational Site Load

Silk Performer

1. **Endurance testing**
2. **Spike testing**
3. **Volume testing**
4. **Scalability testing**
5. **What is Error, Defect, Bug and failure?**

* ERROR: - A mistake in coding is called error
* DEFECTS: - 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

**Difference between Priority and Severity?**

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

Bug life cycle is the specific set of stages that defects or bugs goes through in its entire life. The purpose of defects life cycle is to easily coordinate or communicate current status of defects which change to various assignee and make the defects fixing process systematic and efficient.