**Module–2(Manual Testing)**

**1) What is Exploratory Testing?**

* Exploratory testing is concurrent process, test design, execution, logging happen simultaneously.

**2) What is component testing?**

* Testing of individual software component in iscolation.

**3) What is Integration testing?**

* testing performed to expose defect in interface and interaction between integrated software component.

**4) What is adhoc testing?**

* Informal testing type with an aim to breck the system.

**5) What is functional system testing?**

* Testing is based on analysis of a specification of the functionality of a component or system.

**6) What is Non-Functional Testing?**

* Testing attributed of a component or system to do not relate to functionality.

**7) What is Error, Defect, Bug and failure?**

* A mistake in coding its called error, Error found by tester its called defect, defect accepted by development team it’s called bug, build does not meet the requirement then its failure.

**8) What is black box testing? What are the different black box testing techniques?**

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

**9 ) What is white box testing and list the types of white box testing?**

* It use to test functional and non functional test without knowing its internal structure the types are boundery value anlisis , state tranction ,etc.

**10) 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.

**11) What is Boundary value testing?**

* Boundary value analysis is a methodology for designing test cases that concentrates software testing efforts on cases near the limits of valid ranges.

**12) What is Equivalence partitioning testing?**

* Aim is to treat groups of inputs as equivalent and to select one representative input to test them all.

**13) What determines the level of risk?**

* A factor could result in future negative consequences, usually expressed as impact and likelihood.
* 2 types of risk

1. Project risk
2. Product risks

**14) What is Alpha testing?**

* It is always performed by the developers at the software development site.

**15) What is beta testing?**

* Beta Testing is always performed at the time when software product and project are marketed.

**16) 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.

**17) What is 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.

**18) 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 load.

**19) Mention what are the categories of defects?**

1. Data quality /Database Defects
2. Critical Functionality Defects
3. Functionality Defects
4. Security Defects
5. User Interface Defects

**20) 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.

**21) 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)

**22) When should "Regression Testing" be performed?**

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

**23) What is 7 key principles? Explain in detail**

1. **Testing shows presence of defects**

* Testing can show that defects are present, but cannot prove that there are not defects.

1. **Exhaustive testing is impossible**

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

1. **Early testing**

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

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.

1. **The 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.

1. **Testing is context dependent**

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

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.

**24) Difference between QA v/s QC v/s Tester**

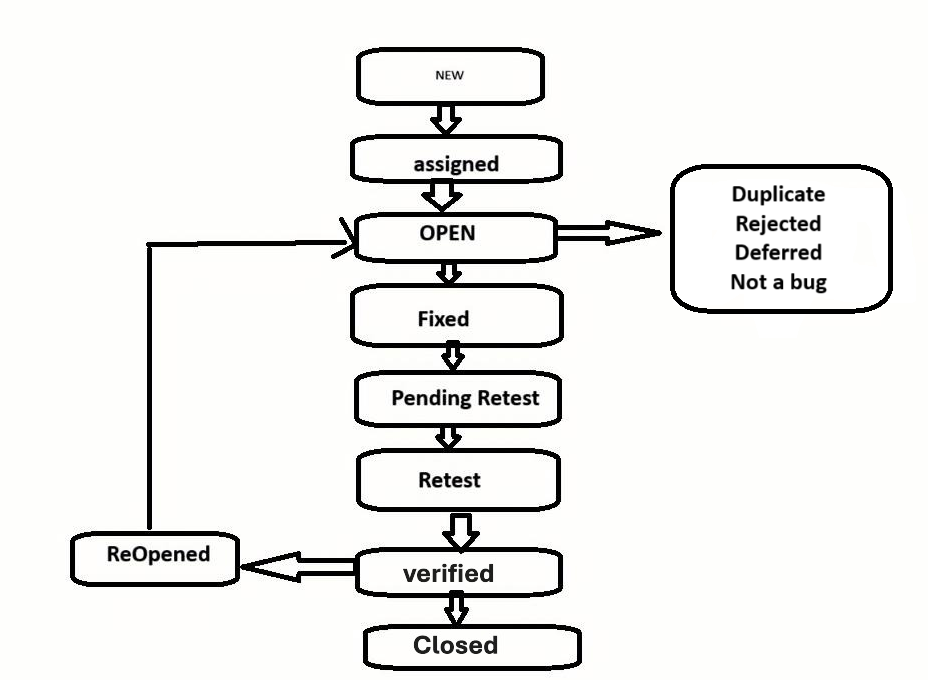
|  |  |  |  |
| --- | --- | --- | --- |
| **Sr.no** | **QA** | **QC** | **Tester** |
| 1 | QA is a subset of STLC | QC is a subset of QA | Testing is a subset of QC |
| 2 | Process oriented activities. | Product oriented activities. | Product oriented activities. |
| 3 | Preventive activities | It is a corrective process | It is a preventive process |

**25) Difference between Smoke and Sanity?**

|  |  |  |
| --- | --- | --- |
| **Sr.no** | **Smoke** | **Sanity** |
| 1 | Smoke testing performed after the software build to ascertain that the critical functionalities of the program working fine | Sanity testing is performed after receive build, the minor changes in code or functionalities. Sanity testing is ascertain that bugs have been fixed and not further issue introduces due to these changes. |
| 2 | Smoke testing usually scripted or documented | Sanity testing is unscripted or undocumented |
| 3 | The objective of this testing is to verify stability of the system | The objective of this testing is to verify the rationality of the system |
| 4 | Smoke testing is like general health checkup | Sanity testing is like specialized heath checkup |

**26) 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.”



**27) 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** | Plans, Requirement Specs, Design Specs, Code, Test Cases | The actual product/software. |
| **4** | Reviews ∙ Walkthroughs ∙ Inspections | Testing |

**28)** **Explain types of Performance testing.**

* **Load testing-** checks the application’s ability to perform under anticipated user loads. The objective is to identify performance bottlenecks before the software application goes live.
* **Stress testing-** involves testing an application under extreme workloads to see how it handles high traffic or data processing. The objective is to identify the breaking point of an application.
* **Endurance testing-** is done to make sure the software can handle the expected load over a long period of time.
* **Spike testing-** tests the software’s reaction to sudden large spikes in the load generated by users
* **Volume testing-** Under Volume Testing large no. of. Data is populated in a database, and the overall software system’s behaviour is monitored. The objective is to check software application’s performance under varying database volumes
* **Scalability testing-** The objective of scalability testing is to determine the software application’s effectiveness in “scaling up” to support an increase in user load. It helps plan capacity addition to your software system.

**29) Difference between Priority and Severity**

|  |  |  |
| --- | --- | --- |
| **Sr.no** | **Priority** | **Severity** |
| **1** | How urgent is to fix the defect | How serious the impact of a defect is on functionality or performance |
| **2** | Business urgency or importance | Technical impact of the bug |
| **3** | Development or release timeline | Stability, functionality, or performance |
| **4** | Typo on homepage of a major site high priority | May not be fixed immediately if priority is low |

**30) Explain the difference between Functional testing and Non-Functional testing**

|  |  |  |
| --- | --- | --- |
| **Sr.no** | **Functional testing** | **Non-Functional testing** |
| **1** | Testing is based on analysis of a specification of the functionality of a component or system. | Testing attributed of a component or system to do not relate to functionality. |
| **2** | Functional testing is executed first | Non-functional testing should be performed after functional testing |
| **3** | Easy to do manual testing | Tough to do manual testing |

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

|  |  |  |
| --- | --- | --- |
| **Sr.no** | **SDLC** | **STLC** |
| **1** | SDLC is a structure imposed on the development of a software product that defines the process for planning, implementation, testing, documentation, deployment, and ongoing maintenance and support | **STLC** is a **structured process** that defines a series of **specific steps and phases** carried out during the **testing of software**. It ensures that the software meets the required quality standards and works as expected, by systematically planning, designing, executing, and closing test activities. |
| **2** | Beside development other phases like testing is also included | It focuses only on testing the software |
| **3** | In SDLC more number of members are required for the whole process | In STLC less number of members are needed |
| **4** | It helps in developing good quality software | It helps in making the software defects free |

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr.no** | **Test case** | **Test scenarios** | **Test script** |
| **1** | Test cases involve the set of steps, conditions and inputs which can be used while performing the testing tasks. | A Scenario is any functionality that can be tested | A set of sequential instruction that detail how to execute a core business function |
| **2** | Focused on what to test and how to test | Focused on what to test | To automate or document the execution of a test. |
| **3** | Low-level actions | High-level actions | Low-level actions |
| **4** | Verifies the test situation by applying steps | Addresses end-to-end functionality of the software program | Very detailed, including code or execution commands |

**33) Explain what Test Plan is? What is the information that should be covered.**

* A document describing scope, approach, resource, and schedule intended test activities.
* Define **what will be tested** and **how it will be tested.**
* Provide a clear **roadmap** for testing activities.
* Align testing objectives with **project goals.**
* Identify **risks, resources**, and **responsibilities.**
* Ensure **stakeholder visibility** into the testing process.

**34) What is priority?**

* **Priority in testing** refers to the **level of urgency assigned to a defect or test case**, indicating **how quickly it should be fixed or executed** based on its impact on the business or project timelines.

**35) What is severity?**

* **Severity** is a measure of **how badly a defect affects the system's functionality or performance,** as determined by the tester or developer.

**36) Bug categories are…**

* Bug Category is Security, Database, Functionality (Critical/General), UI

**37) Advantage of Bugzilla.**

* 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.

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

* **Scrum-** SCRUM is an agile development method which concentrates particularly on how to manage tasks within a team-based development environment.
* **Kanban-** Kanban is a very popular framework for development in the agile software development methodology.

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

|  |  |  |
| --- | --- | --- |
| **Sr.no** | **Authorization** | **Authentication** |
| **1** | The process of verifying the identity of a user | The process of checking user permissions |
| **2** | o ensure only **valid users** can log in | To ensure users **only access permitted resources** |
| **3** | Happens **before authorization** | Happens **after successful authentication** |
| **4** | User enters **username and password** to log in | Logged-in user tries to **access admin panel;** access is restricted based on role |
| **5** | Test **login pages, session handling, password policies** | Test **role-based access, URL restrictions, permission boundaries** |

**Common problems: -**

1. Security Issues
2. **Compatibility Issues**
3. Functionality Issues
4. Performance Issues
5. Usability Issues
6. Data Integrity Issues
7. Navigation Issues