**ASSIGNMENT-2: MODULE-2**

**Que-1: What is Exploratory Testing?**

* **Exploratory testing is a concurrent process where:**

1. **Test design, execution and logging happen simultaneously.**
2. **Testing is often not recorded.**
3. **Makes use of experience, heuristics and test patterns.**

**Que-2: What is traceability matrix?**

* **Forward Traceability:-**
* **Mapping of Requirements to Test cases**
* **Backward Traceability:–**
* **Mapping of Test Cases to Requirements**
* **Bi-Directional Traceability –**
* **A Good Traceability matrix is the references from test cases to basis documentation and vice versa.**

**Que-3: What is Boundary value analysis (B.V.A) testing?**

* **B.V.A is a methodology for designing test cases that concentrates Software Testing effort on cases near The Limit of valid Ranges.**
* **B.V.A Testing is a method which Refines Equivalence Partitioning Testing.**

**Que-4: What is Equivalence partitioning (E.P) testing?**

* **Aim is to treat groups of inputs as equivalent and to select one representative input to test them all.**
* **E.P can be used for all levels of Testing.**
* **E.P testing says that by Testing just one value we have Tested the partition (typically a mid-point value is used).**
* **It assumes that:**

1. **If one value finds a bug, the other probably will to.**
2. **If one doesn’t find a bug, the others probably won’t either.**

**Que-5: What is Integration testing?**

* **Integration Testing 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 a group.**
* **There are two levels of Integration Testing –**

1. **Component Integration Testing**
2. **System Integration Testing**

**Que-6: What determines the level of risk?**

* **Risks are of two level of risk:-**

1. **Project Risk**
2. **Product Risk**

**Que-7: What is Alpha testing?**

* **Alpha testing is the first end-to-end testing of a product to ensure it meets the business requirements and functions correctly.**
* **It is typically performed by internal employees and conducted in a lab/stage environment.**
* **An alpha test ensures the product really works and does everything it’s supposed to do.**

**Que-8: What is beta testing?**

* **Beta testing is an opportunity for real users to use a product in a production environment to uncover any bugs or issues before a general release.**
* **Beta testing is the final round of testing before releasing a product to a wide audience.**

**Que-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.”**
* **The testing of individual software components.**
* **Sometimes known as Unit Testing, Module Testing or Program Testing.**
* **Unit testing is performed by using the White Box Testing method.**

**Que-10: What is functional system testing?**

* **Functional Testing - Testing based on an analysis of the specification of the functionality of a component or system.**
* **‘Specification’ – example- Requirements specification, Use Cases, Functional specification or maybe undocumented.**
* **‘Function’ – what the system does.**

**🡺There are Six types of Functional System Testing –**

1. **Black box testing**
2. **White box testing**
3. **Experience based testing**
4. **Smoke testing**
5. **Sanity testing**
6. **End to End testing**

**Que-11: What is Non-Functional Testing?**

* **Non-Functional Testing- Testing the attributes of a component or system that do not relate to functionality, example: reliability, efficiency, usability, interoperability, maintainability and Portability.**
* **It is the testing of “how” the system works. Non-functional testing maybe performed at all test levels.**

**🡺There are Seven types of Non-Functional Testing –**

1. **Usability Testing**
2. **Compatibility Testing**
3. **GUI Testing**
4. **Security Testing**
5. **Performance Testing**
6. **Stress Testing**
7. **Load Testing**

**Que-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.**

**Que-13: What is Ad-hoc testing?**

* **It is an informal testing type with an AIM to break the system.**
* **It does not follow any test design technique to create test cases.**
* **In fact, it is does not create test cases all together.**
* **Main AIM of this testing is to find defects by random checking.**
* **Ad-hoc testing can be achieved with the testing technique called Error Guessing.**

**🡺TYPES OF AD-HOC TESTING:**

1. **Buddy Testing**
2. **Pair Testing**
3. **Monkey Testing**

**Que-14: What is load testing?**

* **A process of testing the behaviour of the Software by applying maximum load in terms of Software accessing and manipulating large input data.**
* **It can be done at both normal and peak load conditions.**
* **This type of testing identifies the maximum capacity of Software and its behaviour at peak time**

**Que-15: What is stress Testing?**

* **Stress testing is the testing to evaluate how system behaves under unfavourable conditions.**
* **Testing is conducted at beyond limits of the specifications.**
* **It falls under the class of black box testing.**

**Que-16: What is white box testing and list the types/techniques of white box testing?**

**🡺DEFINITION:**

* **Based on code and the design of the system.**
* **The tests provide the ability to derive the extent of coverage of the whole application.**

**🡺TECHNIQUES/TYPES:**

1. **STATEMENT COVERAGE**
2. **DECISION/BRANCH COVERAGE**
3. **CONDITION COVERAGE**

**Que-17: What is black box testing? What are the different black box testing techniques?**

**🡺DEFINITION:**

* **Based on requirement.**
* **From the requirements, tests are created.**
* **Specification Models can be used for systematic test case design.**

**🡺 TECHNIQUES:**

1. **EQUIVALENCE PARTITIONING TESTING**
2. **BOUNDARY VALUE ANALYSIS TESTING**
3. **DECISION TABLES TESTING**
4. **STATE TRANSITION TESTING**
5. **USE CASE TESTING**

**Que-18: Mention what are the categories of defects?**

* **Database Defects**
* **Critical Functionality Defects**
* **Functionality Defects**
* **Security Defects**
* **User Interface Defects**

**Que-19: 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.**
* **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 is integrated together at once, and then tested.**

**🡺 ADVANTAGES:**

* **Convenient for small systems.**

**🡺 DISADVANTAGES:**

* **Fault Localization is difficult.**
* **Since the integration testing can commence only after “all” the modules are designed, testing team will have less time for execution in the testing phase.**

**Que-20: What is the purpose of exit criteria?**

* **Exit criteria is used to determine whether a given test activity has been completed or NOT.**
* **Exit criteria can be defined for all the test activities right from planning, specification and execution.**
* **Exit criteria should be part of test plan and decided in the planning stage.**

**Que-21: When should "Regression Testing" be performed?**

* **Regression Testing: 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.**

**Que-22: What is 7 key principles? Explain in detail?**

* **7 KEY PRINCIPLES ARE-**

1. **Testing shows presence of Defects.**
2. **Exhaustive Testing is impossible!**
3. **Early Testing.**
4. **Defect Clustering.**
5. **Pesticide Paradox.**
6. **Testing is Context Dependent.**
7. **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.**

🡺 **Exhaustive Testing is impossible! –**

* **Test everything including all combination of inputs and pre condition is not possible.**
* **Why do not test everything?**

**Exhaustive testing of complex software application --**

1. **Requires enormous resources**
2. **Is too expensive**
3. **Takes too long**

**It is therefore impractical.**

**Need an alternative that is pragmatic, affordable, timely and provide results.**

* **Example-**

**System has 20 screens**

**Avg. 4 menus/screen**

**Avg. 3 options/menus**

**Avg. 10 fields/screen**

**2 types of input per field**

**Around 100 possible values.**

**Approximate TOTAL = 20\*4\*3\*10\*2\*100**

**= 480000 tests.**

**Test length = 1 sec, Duration = 17.7 days**

**Test length = 10 sec, Duration = 34 days**

**Test length = 1 min, Duration = 4 years**

**Test length = 10 min, Duration = 40 years**

🡺 **Early Testing–**

* **Testing activities should start as early as possible in the development life cycle.**
* **Remember from our definition of testing, that testing doesn’t start once the code has been written.**

**🡺 Defect clustering—**

* **Defects are not evenly spread in a system.**
* **They are “Clustered”.**
* **In other word, most defects found during testing are usually confined to a small no. of modules.**

**🡺 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 exercised different parts of the software or system to potentially find more defects.**

**🡺 Testing in context dependent—**

* **Testing is basically context dependent.**
* **Testing is done differently in different contents.**
* **3-10 failures per KLOC for Commercial Software.**
* **1-3 failures per KLOC for Industrial Software.**
* **0.01 failures per KLOC for NASA Shuttle Code.**

**🡺 Absence of Error Fallacy—**

* **Even after defects have been resolved it may still be usable and/or does not fulfil the users “needs and expectations”.**

**Que-23: Difference between QA v/s QC v/s Tester?**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr no.** | **QA** | **QC** | **TESTER** |
| **1.** | **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.** |
| **2.** | **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.** |
| **3.** | **Process oriented activities.** | **Product oriented activities.** | **Product oriented activities.** |
| **4.** | **Preventive activities.** | **It is a corrective process.** | **It is a preventive process.** |
| **5.** | **It is a subset of Software Test Life Cycle (STLC).** | **QC can be considered as the subset of Quality Assurance (QA).** | **Testing is the subset of Quality Control (QC).** |

**Que-24: Difference between Smoke and Sanity?**

|  |  |  |
| --- | --- | --- |
| **Sr no.** | **Smoke** | **Sanity** |
| **1.** | **Checks the Critical Functionality.** | **Checks the New Functionality.** |
| **2.** | **It is done in initial stage.** | **It is done after 30 builds.** |
| **3.** | **It checks the stability.** | **It checks the Sanity/Rationality.** |
| **4.** | **Part of Acceptance Testing.** | **Part of Regression Testing.** |
| **5.** | **Done by tester & developer.** | **Done by Tester.** |
| **6.** | **General health check-up.** | **Advance health check-up.** |
| **7.** | **It checks the system end-to-end.** | **It checks only a particular function of entire system.** |

**Que-25: Difference between verification and Validation?**

|  |  |  |
| --- | --- | --- |
| **Sr no.** | **Verification** | **Validation** |
| **1.** | **Verification addresses the concern: "Are you building it right?"** | **Validation addresses the concern: "Are you building the right thing?"** |
| **2.** | **Ensures that the software system meets all the functionality.** | **Ensures that the functionalities meet the intended behaviour.** |
| **3.** | **Verification takes place first and includes the checking for documentation, code, etc.** | **Validation occurs after verification and mainly involves the checking of the overall product.** |
| **4.** | **Done by developers.** | **Done by testers.** |
| **5.** | **It has static activities, as it includes collecting reviews, walkthroughs, and inspections to verify a software.** | **It has dynamic activities, as it includes executing the software against the requirements.** |

**Que-26: Explain types of Performance testing.**

* **Performance testing is a non-functional software testing technique that determines how the stability, speed, scalability, and responsiveness of an application holds up under a given workload.**
* **TYPES OF PERFORMANCE TESTING: -**

1. **LOAD TESTING: -**

**It is type of testing which involves evaluating the performance of the system under the expected workload.**

1. **STRESS TESTING: -**

**It is a type of performance testing where we evaluate the application’s performance at load much higher than the expected load.**

1. **ENDURANCE TESTING: -**

**It is also known as “Soak Testing”.**

**It is done to make sure the software can handle the expected load over a long period of time.**

1. **SPIKE TESTING: -**

**In spike testing, we analyse the behaviour of the system on suddenly increasing the no. of users.**

1. **VOLUME TESTING: -**

**The Volume Testing is performed by feeding the application with a high volume of data.**

1. **SCALABILITY TESTING: -**

**The objectives 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.**

**Que-27: What is Error, Defect, Bug and failure?**

1. **ERROR:**

* **A mistake in coding is called ERROR.**
* **A human action that produces an incorrect result.**

1. **DEFECT:**

* **ERROR found by tester is called DEFECT.**
* **A flaw in a component or system that can cause the components or system to fail to perform its required function.**

1. **BUG:**

* **DEFECT accepted by development team then it is called BUG.**

1. **FAILURE:**

* **Build does not meet the requirements then it is FAILURE.**

**Que-28: Difference between Priority and Severity?**

* **Severity: -**
* **Severity is absolute and Customer-Focused. It is the extent to which the defect can affect the software. In other word it defines the impact that a given defect has on the system.**

* **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**

**Que-29: 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.”**

**Que-30: Explain the difference between Functional testing and Non-functional testing.**

|  |  |  |
| --- | --- | --- |
| **Sr no.** | **Functional testing (F.T)** | **Non-functional testing (NON - F.T)** |
| **1.** | **It is executed first.** | **It should be performed after F.T.** |
| **2.** | **Manual testing or Automation tools can be used for F.T.** | **Using tools will be effective for this Testing.** |
| **3.** | **Business Requirements are the inputs to F.T.** | **Performance parameters like speed, scalability are inputs to Non – FT.** |
| **4.** | **F.T describes what the product does.** | **Non-FT describes how good the product works.** |
| **5.** | **Easy to do Manual Testing.** | **Tough to do Manual Testing.** |

**Que 31: - What is the difference between the STLC (Software Testing Life Cycle) and SDLC (Software Development Life Cycle)?**

* **STLC:-**

**In an Ideal world, you will not enter the next stage until the exit criteria for the previous stage is met. But practically this is not always possible.**

* **SDLC:-**

**Testing is a process that’s take place throughout the Software Development Life Cycle (SDLC).**

**Que 32: - What is the difference between test scenarios, test cases, and test script?**

* **Test Scenarios:**
* **Test scenarios are high-level descriptions of a feature or functionality to be tested.**
* **Test scenarios are less detailed than test cases and do not go into specific steps or expected results.**
* **They focus on the big picture of what needs to be tested.**
* **Test scenarios may cover multiple test cases or even multiple features.**
* **Test scenarios are created during the requirements analysis phase of the software development life cycle.**
* **Test scenarios are useful for communicating testing requirements to stakeholders who may not be technical.**
* **Test Cases:**
* **Test cases are detailed descriptions of a specific condition or action that verifies a requirement or functionality.**
* **Test cases are more detailed than test scenarios and include specific input data, expected output, and steps to be performed during testing.**
* **Test cases are typically focused on a single requirement or functionality.**
* **Test cases are created during the testing phase of the software development life cycle.**
* **Test cases are primarily used by testers to execute and validate the system.**
* **Test Scripts:**
* **Test scripts are code that automates the execution of test cases or sets of test cases.**
* **Test scripts are even more detailed than test cases and include specific code to perform actions and verify expected results.**
* **Test scripts can be used to execute a single test case or a series of test cases.**
* **Test scripts are created during the testing phase of the software development life cycle.**
* **Test scripts are primarily used by testers or developers to automate the testing process.**

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

* **A test plan is a document that outlines the entire software testing process, including the testing objectives, scope, approach, resources, schedule, risks, and acceptance criteria.**
* **It serves as a roadmap for the testing team and ensures that all aspects of the testing process are covered**

**Que 34: - What is priority?**

* **Priority is the measure of the importance or urgency assigned to a software issue, indicating how critical it is to the functioning of the software and how quickly it needs to be addressed.**

**Que 35: - What is severity?**

* **Severity is the measure of the impact or seriousness of a software defect on the software's functionality, performance, or user experience.**

**Que 36: - Bug categories are….**

* **Functional defects**
* **Performance defects**
* **Usability defects**
* **Compatibility defects**
* **Security defects**
* **Installation/Configuration defects**
* **Documentation defects.**

**Que 37: - Advantage of Bug-Zila.**

* **Bugzilla is a web-based bug tracking tool that helps software development teams to manage and track bugs efficiently.**
* **Some advantages of Bugzilla include easy bug reporting, efficient bug tracking and management, collaboration and communication between team members, customizable-workflows, and integration with other tools.**

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

* **There are several methodologies in Agile Development Model, including:**

1. **Scrum: An iterative and incremental framework for managing and completing complex projects.**
2. **Kanban: A visual method for managing work that emphasizes just-in-time delivery and continuous improvement.**
3. **Lean: A methodology that focuses on maximizing customer value while minimizing waste and improving efficiency.**
4. **Extreme Programming (XP): A set of software development practices that emphasizes frequent releases, testing, and customer involvement.**
5. **Crystal: A flexible methodology that adapts to the needs of the project and the team.**
6. **Feature-Driven Development (FDD): A methodology that breaks down the project into small, feature-sized chunks and emphasizes iterative development and testing.**
7. **Dynamic Systems Development Method (DSDM): A framework for delivering projects on time and within budget while maintaining quality.**
8. **Adaptive Software Development (ASD): A methodology that emphasizes collaboration, communication, and feedback to adapt to changing requirements and circumstances.**

**Que 40: - Explain the difference between Authorization and Authentication in Web testing. What are the common problems faced in Web testing?**

|  |  |
| --- | --- |
| **Authorization** | **Authentication** |
| **Authorization determines what resources a user can access after authentication** | **Authentication is the process of verifying the identity of a user or system.** |
| **It is the process of verifying whether a user has permission to access the requested resource or not.** | **It is the process of confirming whether a user is genuine or not.** |
| **It involves validating the user’s role or privilege and checking if it matches the resource requirements.** | **It involves the use of credentials like username and password, biometric information, or digital certificates.** |
| **It is mainly concerned with granting or denying user access to specific resources based on their role or privilege level.** | **It is the first line of defense against unauthorized access to a system or resource.** |
| **Authorization ensures that only authorized users can access sensitive data and functionalities.** | **It is the first line of defense against unauthorized access to a system or resource.** |

* **Common problems faced in Web Testing:**
* **Cross-browser compatibility issues: Web applications need to be tested across different browsers, devices, and operating systems, which can result in compatibility issues that affect the user experience.**
* **Security vulnerabilities: Web applications are susceptible to attacks such as cross-site scripting (XSS), SQL injection, and cross-site request forgery (CSRF), which can compromise user data and application functionality.**
* **Performance issues: Web applications need to perform well under heavy traffic and load, and may suffer from slow page loading, slow server response times, and other performance issues.**
* **Functional defects: Web applications may have defects related to user interface, functionality, and data processing that affect the user experience and application usability.**
* **Integration issues: Web applications often need to integrate with other systems and technologies, such as payment gateways, third-party APIs, and databases, which can result in integration issues that need to be tested and resolved.**

**Que 41: - When to used Usability Testing?**

* **Before Any Design Decisions Are Made.**
* **When It's Time to Evaluate and Iterate.**
* **After Launch.**
* **In High-Risk, Low-Certainty Situations.**

**Que 42: - What is the procedure for GUI Testing?**

* **Check all the GUI elements for size, position, width, length and acceptance of characters or numbers.**
* **For instance, you must be able to provide inputs to the input fields.**
* **Check you can execute the intended functionality of the application using the GUI.**
* **Check Error Messages are displayed correctly.**
* **Check for Clear demarcation of different sections on screen.**
* **Check Font used in application is readable.**
* **Check the alignment of the text is proper Check the Colour of the font and warning messages is aesthetically pleasing.**
* **Check that the images have good clarity Check that the images are properly aligned**
* **Check the positioning of GUI elements for different screen resolution.**