**Software Testing Assignment**

**Module-2 (Manual Testing)**

|  |  |
| --- | --- |
| 1. | What is Exploratory Testing? |
| Ans: | Exploratory Testing is not random testing but it is adhoc testing with purpose of find bugs. |
| 2. | What is Traceability Matrix? |
| Ans: | Tracebility matrix is used to trace whether the test cases we have written or executed met the requirements or not. |
| 3. | What is boundary value testing? |
| Ans: | Boundary value testing is a methodology for designing test cases that concentrates software testing effort on cases near the limits of valid  ranges |
| 4. | What is Equivalence Partitioning Testing? |
| Ans: | It is black box testing technique.  The numbers fall into a partition where each would have the same, or  equivalent, result i.e. an Equivalence Partition (EP) |
| 5. | What is Integration Testing? |
| Ans: | Integration Testing is a level of the software testing process where individual units are combined and tested as a group**.** |
| 6. | What determines the level of risk? |
| Ans: | (1)Project risk: -Example of Project risk is Senior Team Member leaving the project abruptly.  (2)Product risk:- Example of product risks would be Flight Reservation system not installing in test environment |
| 7. | What is alpha testing? |
| Ans: | Alpha testing is a the form of Acceptance Testing. It is always performed by the developers at the software development site. It is always performed in Virtual Environment. It comes under the category of both White Box Testing and BlackBox Testing. |
| 8. | What is beta testing? |
| Ans: | Beta Testing is performed in Real Time Environment. It is always performed by the customers at their own site. It is not performed by Independent Testing Team. It is only a kind of Black Box Testing |
| 9. | What is component testing? |
| Ans: | The testing of individual software components is called Component Testing. |
| 10. | What is Functional system testing? |
| Ans: | In Functional Tetsing Each & every functionality of the system is tested by providing appropriate input, verifying the output and comparing the actual results with the expected results |
| 11. | What is Non Functional testing? |
| Ans: | Non Functional Testing is the attributes of a componentor system that do not relate to functionality, e.g. reliability,efficiency, usability, interoperability, maintainability and portability |
| 12. | What is GUI Testing? |
| Ans: | 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. |
| 13. | What is Adhoc Testing? |
| Ans: | Testers randomly test the application without any test cases orany business requirement document is called Adhoc Testing. |
| 14. | What is Load Testing? |
| Ans: | Load testing is a kind of performance testing which determines a system’s performance under real-life load conditions. This testing helps determine howthe application behaves when multiple users access it simultaneously. |
| 15. | What is Stress testing? |
| Ans: | Stress testing is to test the system behavior under extreme conditions and is carried out till the system failure. |
| 16. | What is white box testing and list the types of white box testing. |
| Ans: | White box Testing based on an analysis of the internal structure of the component or system.  There are 4 types of white box testing:   1. Web based testing:  * Analyze the logic by reading the code * Code optimization Suggest if any optimization can be done in the code which is better than the existing one * Deploy the code using different web servers which could be a case study even if the product is not supporting tother web servers  1. Desktop based testing:  * When we debug the code when we writing  1. Mobile based testing:  * The Android SDK and related plugin for Eclipse * Android devices enabled for development and debugging, with appropriate USB drivers as necessary  1. Game based testing:  * When we connect with remote device , so which device we connect will check in code * When some debug the code and play at that time game. |
| 17. | What is black box testing? What are the different black box testing technique. |
| Ans: | The technique of testing without having any knowledge of the interior workings of the application is Black Box testing.  There are 6 types of black box testing technique:   1. Equivalence partitioning 2. Boundry value analysis 3. Decision tables 4. State transition testing 5. Use case testing 6. Other blck box testing  * Syntax or pattern testing |
| 18. | Mention what are the categories of defects. |
| Ans: | 1. Data quality/Database defect:- Deals with improper handling of data in the database.   Exp:- Values not deleted/inserted into the database properly   * Improper/wrong/null values inserted in place of the actual values  1. Critical Functionality Defect:- The occurrence of these bugs hampers the crucial functionality of the application. Examples: - Exceptions 2. Functionality Defects:- This defect affect the functionality of the application. Exp:- All Java script errors. 3. Security Defects:- Application security defects generally involve improper handling of data sent from the user to the application. These defects are the most severe and given highest priority for a fix.   Exp:- Authentication: Accepting an invalid username/password   * Authorization: Accessibility to pages though permission not given  1. User interface defects:- As the name suggests, the bugs deal with problems related to UI are usually considered less severe.   Exp:- Improper error/warning/UI messages   * Spelling mistakes * Alignment problems |
| 19. | Mention what bigbang testing is? |
| Ans: | 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 are integrated together at **once**, and then tested. |
| 20. | What is the purpose of exit criteria? |
| Ans: | Successful Testing of Integrated Application.   * Executed Test Cases are documented * All High prioritized bugs fixed and closed * Technical documents to be submitted followed by release Notes |
| 21. | When should “regression testing” be performed. |
| Ans: | * Change in requirements and code is modified according to the requirement * New feature is added to the software * Defect fixing * Performance issue fix |
| 22. | What Is 7 key principles? Explain in detail. |
| Ans: | 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. * We test to find Faults * As we find more defects, the probability of undiscovered defects   remaining in a system reduces.   * However Testing cannot prove that there are no defects present  1. Exhaustive testing is impossible:  * Testing everything including all combinations of inputs and preconditions is not possible. * For example: In an application in one screen there are 15 input fields, each having 5 possible values, then to test all the valid combinations you would need 30 517 578 125 (515) tests. * So, accessing and managing risk is one of the most important activities and reason for testing in any project.  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. * Remember from our Definition of Testing, that Testing doesn’t start once the code has been written  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 to a small number of modules * An important consideration in test prioritization  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.   * Therefore we must learn, create and use new tests based on new   techniques to catch new bugs.   1. Testing is context dependent:  * 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. * If we build a system and, in doing so, find and fix defects .... * It doesn’t make it a good system * Even after defects have been resolved it may still be unusable and/or does not fulfil the users’ needs and expectations |
| 23 | Difference between QA V/S QC V/S tester. |
| Ans: | |  |  |  |  | | --- | --- | --- | --- | | Sr. | Quality Assurance | Quality Control | 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 | Testing is the subset of Quality Control. | |
| 24. | Difference between Smoke and sanity. |
| Ans: | |  |  |  | | --- | --- | --- | | Sr. | Smoke | Sanity | | 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 | | 2. | The objective of this testing is to verify "stability" of the system in order to with more rigorous testing | The objective of the testing is to verify the the "rationality" of the system in order proceed to with more rigorous testing | | 3. | This testing is performed by the developers or testers | Sanity testing is usually performed by testers | | 4. | Smoke testing is usually documented or scripted | Sanity testing is usually not documented and is unscripted | | 5. | Smoke testing is a subset of Regression testing | Sanity testing is a subset of Acceptance testing | | 6. | Smoke testing exercises the entire system from end to end | Sanity testing exercises only the particular component of the entire system | | 7. | Smoke testing is like General Health Check Up | Sanity Testing is like specialized health Check Up | |
| 25. | Difference between verification and validation |
| Ans: | |  |  | | --- | --- | | Verification | Validation | | 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. | | 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 | | Are we building the product right? | Are we building the right product | | Plans, Requirement Specs, Design Specs, Code, Test Cases | The actual product/software | | Reviews  ∙ Walkthroughs  ∙ Inspections | Testing | |
| 26. | Explain types of performace testing. |
| Ans: | 1. Load testing:- 2. Stress testing 3. Endurance testing 4. Spike testing 5. Volume testing 6. Scalability testing |
| 27. | What is error, defect, bug and failure? |
| Ans: | Error:- A mistake in coding is called error**.**   * A human action that produces an incorrect result is called Error.   Defect:- error found by tester is called defect.   * Commonly refers to several troubles with the software products, with its external behavior or with its internal features 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”. |
| 28. | Difference between priority and severity |
| Ans: | 1. **High Priority & High Severity**: An error which occurs on the basic functionality of the application and will not allow the user to use the system.   (Eg. A site maintaining the student details, on saving record if it, doesn’t allow to save the record then this is high priority and high severity bug.)   1. **High Priority & Low Severity:** The spelling mistakes that happens on the cover page or heading or title of an application. 2. **High Severity & Low Priority:** An error which occurs on the functionality of the application (for which there is no workaround) and will not allow the user to use the system but on click of link which is rarely used by the end user. 3. **Low Priority and Low Severity:** Any cosmetic or spelling issues which is within a paragraph or in the report (Not on cover page, heading, title). |
| 29. | What is Bug life Cycle? |
| Ans: | “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.”  New  Assigned  Open  Fixed  Pending Retest  Retest  Verified  Re opened  Duplicate Rejected Deferred Not a Bug Not a bug  Closed |
| 30. | Explain the difference between Functional Testing and Non Functional Testing. |
| Ans: | |  |  | | --- | --- | | Functional Testing | Non Functional Testing. | | 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 | | Functional testing is executed first | Non functional testing should be performed after functional testing | | Manual testing or automation tools can be used for functional testing | Using tools will be effective for this testing | | Business requirements are the inputs to functional testing | Performance parameters like speed , scalability are inputs to non-functional testing. | | Functional testing describes what the product does | Nonfunctional testing describes how good the product works | | Easy to do manual testing | Tough to do manual testing | | 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 | |
| 31. | To create HLR & Test case of (instagram, facebook) only first page. |
| Ans: |  |
| 32. | What is the difference between the STLC and SDLC? |
| Ans: | |  |  | | --- | --- | | STLC | SDLC | | STLC stands for Software Testing Life cycle | SDLC stands for software Development Life Cycle. | | Whereas the only objevtive of the STLC phase is Testing | The main objective of the SDLC Life Cycle is to complete successful development of the software including testing and other phases | | The QA Team analyze requirement document and create a system test plan. | In SDLC, the business analyst gathers the requirements and creates a Development plan, | | The testing team prepares the test case and executes them. | In SDLC the real code is developed and actual work takes place as per the design documents. | | STLC is part of SDLC, so it involves only the testing part like Requirement Analysis, test planning, Test Development , test environment setup, test execution & Clouser | SDLC involves requirement gathering, Design, coding, testing, Deployment and maintenance. | |
| 33. | What is the difference between test scenarios, test cases and test script? |
| Ans: | |  |  |  | | --- | --- | --- | | Test scenarios | Test cases | Test scripts | | 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 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. | | Test Scenario is ‘What to be tested’ | Test Case is ‘How to be tested’ | The Test Script can be manual or automated | |
| 34. | explain what test plan is? What is the information that should be covered. |
| Ans: | A document describing the scope, approach, resources and schedule of intended test activities.  -Determining the scope and risks, and identifying the objectives of testing.  -All projects require a set of plans and strategies which define how the testing will be conducted.  - There are number of levels at which these are defined.  - Defines how the organization will conduct testing  - Defines how the project will conduct testing  - Defines how each **level** of testing will be conducted |
| 35. | What is priority? |
| Ans: | Priority defines the order in which we should resolve a defect |
| 36. | What is severity? |
| Ans: | 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 |
| 37. | Bug categories are. |
| Ans: | Security, Database, Functionality (Critical/General), UI |
|  |  |