Software testing module 2

1. What is software testing?

Ans. Software testing is a process used to identify the correctness, completeness, and quality of developed computer software.

1. What is exploratory testing?

Ans. Exploratory testing is an approach to software testing that is often described as simultaneous learning, test design, and execution.

1. What is traceability matrix?

Ans. 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?

Ans. Software testing technique in which tests are designed to include representatives of boundary values. It is performed by the QA testing teams.

1. What is Equivalence partitioning testing?

Ans. Software testing technique that divides the input data of a software unit into partition of data from which test cases can be derived. It is usually performed by the QA teams.

1. What is integration testing?

Ans. Testing performed to expose defects in the interfaces and in the interaction between integrated components or system.

1. What determines the level of risk?

Ans. The likelihood of an adverse event and the impact of the event.

1. What is Alpha testing?

Ans. Type of testing a software product or system conducted at the developers site. Usually it is performed by the end user.

1. What is beta testing?

Ans. Final testing before releasing application for commercial purpose. It is typically done by end-users or others.

10. What is component testing?

Ans. Testing technique similar to unit testing but with a higher level of integration – testing is done in the context of the application instead of just directly testing a specific method. Can be performed by testing or development teams.

11. What is functional system testing?

Ans. A requirement that specifies a function that a system or system component must perform.

12. What is Non-functional testing?

Ans. Testing the attributes of a component or system that do not relate to functionality, e.g. reliability, efficiency, usability, interoperability, maintainability and portability.

13. What is GUI testing?

Ans. Graphical User Interface (GUI) testing is the process of testing the system GUI of the system under test. GUI testing involves checking the screens with the controls like menus, buttons, icon, and all types of bars – tool bar, menu bar, dialog boxes and windows etc.

14. What is Adhoc testing?

Ans. Adhoc testing is an informal testing type with an aim to break the system.

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

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

1. unit testing

2. static analysis

3. dynamic analysis

4. statement coverage

5. branch testing coverage

6. security testing

7. mutation testing.

16. What is black box testing? What are the different black box testing technique?

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

● 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

17. Mention what are the categories of defect?

Ans. Functional defect. Functional defects are the errors identified in case the behaviour of software is not compliant with the functional requirements.

● Performance defects

● Usability defects

● Compatibility defects

● Security defects

18. Mention what big bang testing is?

Ans. In big bang integration testing all components or modules is integrated simultaneously, after which everything is tested as a whole.

19. what is the purpose of exit criteria?

Ans. 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)

20. When should “Regression Testing” be performed.

Ans. 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.

21. what is 7 key principles? Explain in detail?

Ans. (1) Testing shows presence of defects :

● testing can show that defects are prevent, 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

(2) Exhaustive testing is impossible :

● testing everything including all combinations of inputs and preconditions is not possible.

● so, instead of doing the exhaustive testing we can use risks and priorities to focus testing efforts.

● this is very unlikely that the project timescales would allow for this number of tests.

● that is we must prioritise our testing using a risk based approach

(3) Early testing :

● testing activity should start as early as possible in the software or system development life cycle, and should be focused on defined objectives.

● testing activities should start as early as in the development life cycle.

● these activities should be focused on defined objectives – outlined in the test strategy.

(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.

● defects are not evenly spread in a system

● they are ‘clustered’

(5) 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.

● testing identifies bugs, and programmers, the software improves

● as bugs are eliminated by the programmers, the software improves

● as software improves the effectiveness of previous tests erodes

(6) Testing is 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.

● while, testing can be 50% of development costs, in NASA’s Apollo program it was 80% testing

(7) Absence of errors fallacy :

● if the system built is unusable and does not fulfil the user’s needs and expectation 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’

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

Ans.

|  |  |  |
| --- | --- | --- |
| Quality Assurance (QA) | Quality control (QC) | Tester |
| (1) Activities which ensure the implementation of processes, procedures and standards in context to verification of developed software and intended requirements. | (1) Activities which ensure the verification of developed software with respect to documented (or not in some cases) requirements. | (1) 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. | (2) Focuses on actual testing by executing software with intend to identify bug/defect through implementation of procedures and process. | (2) Focuses on actual testing. |
| (3) Process oriented activities. | (3) Product oriented activities. | (3) Product oriented activities. |
| (4) Preventive activities. | (4) It is a corrective process. | (4) It is a preventive process. |
| (5) It is a subset of software test life cycle (STLC). | (5) QC can be considered as the subset of Quality Assurance. | (5) Testing is the subset of Quality Control. |

23. difference between Smoke and Sanity?

Ans.

|  |  |
| --- | --- |
| Smoke Testing | Sanity Testing |
| (1) Smoke Testing is performed to ascertain that the critical functionalities of the program is working fine. | (1) Sanity testing is done to check the new functionality / bugs have been fixed. |
| (2) The objective of this testing is to verify the “stability” of the system in order to proceed with more rigorous testing. | (2) The objective of the testing is to verify the “rationality” of the system in order to proceed with more rigorous testing. |
| (3) This testing is performed by the developers or testers. | (3) Sanity testing is usually performed by testers. |
| (4) Smoke testing is usually documented or scripted. | (4) Sanity testing is usually not documented and is unscripted. |
| (5) Smoke testing is a subset of regression testing. | (5) Sanity testing is a subset of Acceptance testing. |
| (6) Smoke testing exercises the entire system from end to end. | (6) Sanity testing exercises only the particular component of the entire system. |
| (7) Smoke testing is like general health check up. | (7) Sanity testing is like specialized health check up. |

24. difference between verification and validation.

Ans.

|  |  |
| --- | --- |
| Verification | Validation |
| (1) Definition: The process of evaluating work-product (not the actual final product) of a development phase to determine whether they meet the specified requirements for that phase. | (1) Definition: The process of evaluating software during or at the end of the development process to determine whether it satisfies specified business requirements. |
| (2) Objective: To ensure that the product is being built according to the requirements and design specification. In other words, to ensure that work products meet their specified requirements. | (2) Objective: 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 fulfils its intended use when placed in its intended environment. |
| (3) Question: Are we building the product right? | (3) Question: Are we building the right product? |
| (4) Evaluation Item: Plans, Requirement specs, Design specs, Code, Test, cases | (4) Evaluation items: The actual product/software. |
| (5) Activities: ● Reviews  ●Walkthroughs  ●Inspections | (5) Activities: Testing |

25. Explain types of performance testing.

Ans. ● Load testing

● Stress testing

● Endurance testing

● Spike testing

● Volume testing

● Scalability testing

● Load testing: Testing technique that puts demand on a system or device and measures its response. It is usually conducted by the performance engineers.

● Stress testing: Testing technique which evaluates a system or component at or beyond the limits of its specified requirements. It is usually conducted by the performance engineer.

● Endurance testing: Type of testing which checks for money leaks or other problems that may occur with prolonged execution. It is usually performed by performance engineers.

● Spike testing: Spike testing is a type of performance testing in which an application receives a sudden and extreme increase or decrease in load. The goal of spike testing is to determine the behaviour of a software application when it receives extreme variations in traffic.

● Volume testing: Testing which confirms that any values that may become large over time (such as accumulated counts, logs, and data files), can be accommodated by the program and will not cause the program to stop working or degrade its operation in any manner. It is usually conducted by the performance engineer.

● Scalability Testing: Part of the battery of non-functional tests which tests a software application for measuring its capability to scale up – be it the user load supported, the number of transactions, the data volume etc. It is conducted by the performance engineer.

26. What is Error, Defect, Bug and failure?

Ans. Error: A mistake in coding is called error.

Defect: 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.

27. Difference between Priority and Severity.

Ans.

|  |  |
| --- | --- |
| Priority | Severity |
| Priority is associated with scheduling | Severity is associated with functionality |
| ● It indicate how soon the bug should be fixed | ● It indicate the seriousness of defect |
| ● Priority of defect is consultation with the client | ● QA engineer determine the severity level |
| ● Priority is driven by business level | ● Severity is driven by functionality |
| ● Priority levels are: Critical, high, medium, low | ● Severity levels are: Critical, major, minor, moderate and cosmetic |

28. What is Bug Life Cycle?

Ans. Bug life cycle is nothing but the various phases a bug under goes after it is raised or reported.

The different phases of bug life cycle are,

- New or Opened

- Assigned

- Fixed

- Tested

- Closed

29. 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 |
| ● 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 |
| ●Business requirements are the inputs to functional testing | ● Performance parameters like speed, scalability are inputs to non-functional 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 |

30. What is difference between the STLC(Software Testing Life Cycle) and SDLC(Software Development Life Cycle)?

Ans.

|  |  |
| --- | --- |
| STLC | SDLC |
| Focuses on testing a product. | Focuses on building a product. |
| A child of SDLC process. | A parent process. |
| Ensuring the product is working as expected. | Building a product as user requirement. |
| STLC phases start after SDLC phases are completed. | SDLC phases are completed before testing. |
| End goal is to finding and fixing the bugs/defects. | End goal is to deploy a high quality product to user. |

31. What is the difference between test scenarios and test cases?

Ans.

|  |  |
| --- | --- |
| Test Scenario | Test Case |
| A test scenario contains high-level documentation which describes an end to end functionality to be tested. | Test cases contain definite test steps, data, expected results for testing all the features of an application. |
| It focuses on more “what to test” than “how to test”. | A complete emphasis on “what to test” and “how to test.” |
| Test scenario are a one-liner. So, there is always the possibility of ambiguity during the testing. | Test cases have defined a step, pre-requisites, expected result, etc. Therefore, there is no ambiguity in this process. |
| Test scenario are derived from test artifacts like BRS, SRS, etc. | Test case is mostly derived from test scenarios. Multiple Test case can be derived from a single Test Scenario. |
| It helps in an agile way of testing the end to end functionality | It helps in exhaustive testing of an application |
| Test scenarios are high-level actions. | Test cases are low-level actions. |
| Comparatively less time and resources are required for creating and testing using scenarios. | More resources are needed for documentation and execution of test cases. |

32. Explain what Test Plan is? What is the information that should be covered.

Ans. It is a high level document in which how to perform testing is described. The Test Plan document is usually prepared by the Test Lead or Test Manager and the focus of the document is to describe what to test, how to test, when to test and who will do what test.

● Master test plan: A test plan that typically addresses multiple test levels.

● Phase test plan: A test plan that typically addresses one test phase.

33. What are the different Methodology in Agile Development Model?

Ans. Scrum: Scrum is an agile development method which concentrates particularly on how to manage tasks within a team based development environment. Basically, Scrum is derived from activity that occurs during rugby match.”

● Process Flow of Scrum

● Each iteration of a scrum is known as Sprint

● Product backlog is a list where all details are entered to get end product

● During each Sprint, top items of Product backlog are selected and turned into Sprint backlog.

● Team works on the defined sprint backlog

● Team checks for the daily work

● At the end of sprint, team delivers product functionality.

Extreme Programming

● This is a light weight agile testing methodology in which development and testing happen in parallel. Business requirements are gathered in terms of stories.

● All those stories are stored in a place called parking lot.

● In this type of methodology, releases are based on the shorter cycles called iteration with span of 14 days’ time period.

● Each iteration include phases like coding, unit testing and system testing where at each phase some minor or major functionality will be built in the application.

34. Explain the difference between Authorization and Authentication in web testing.

Ans.

|  |  |
| --- | --- |
| Authorization | Authentication |
| While in authorization process, a the person’s or user’s authorities are checked for accessing the resources. | In the authentication process, the identity of users are checked for providing the access to the system. |
| While in this process, users or persons are validated. | In the authentication process, users or persons are verified. |
| While this process is done after the authentication process | It is done before the authorization process. |
| While it needs the user’s privilege or security levels. | It needs usually the user’s login details. |
| While it determines What permission does the user have? | Authentication determines whether the person is user or not. |

35. What are the common problem faced in web testing?

Ans. Integration. Integration testing exposes problems with interfaces among different program components before deployment.

● Interoperability

● Security

● Performance

● Usability

●Quality Testing, Exceptional Services.