# MODULE -2

1 What is software testing?

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

2 What is Exploratory Testing?

This may be the only type of technique used for low-risk systems, but this approach may be particularly useful under extreme time pressure – in fact this is one of the factors leading to exploratory testing.

3 What is traceability matrix?

Test conditions should be able to be linked back to their sources in the test basis, this is known as traceability. This is a software process should help you keeping the virtual table up-to-date.

4 What is Boundary value testing?

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

5 What is Equivalence partitioning testing?

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

6 What is Integration testing?

The phase in software testing in which individual software modules are combined and tested as a group. It is usually conducted by testing teams.

7 What determines the level of risk?

 The Likelihood of an adverse event and the impact of the event.

8 What is Alpha testing?

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

9 What is beta testing?

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

10 What is component testing?

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?

Type of black box testing that bases its test cases on the specifications of the software component under test. It is performed by testing teams.

12 What is Non-Functional Testing?

Testing technique which focuses on testing of a software application for its non-functional requirements. Can be conducted by the performance engineers or by manual testing teams.

13 What is GUI Testing?

The process of testing a product that uses a graphical user interface, to ensure it meets its written specifications. This is normally done by the testing teams.

14 What is Ad-hoc testing?

Testing performed without planning and documentation - the tester tries to 'break' the system by randomly trying the system's functionality. It is performed by the testing teams.

15 Mention what are the categories of defects?

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

Performance defects. ...

Usability defects. ...

Compatibility defects. ...

Security defects.

16 What is white box testing and list the types of white box testing?

Testing technique based on knowledge of the internal logic of an application’s code and includes tests like coverage of code statements, branches, paths, conditions. It is performed by software developers.

* Unit testing
* Static and dynamic analysis
* Statement coverage
* Branch coverage
* Mutation testing
* Security testing

17 What is black box testing? What are the different black box testing techniques?

A method of software testing that verifies the functionality of an application without having specific knowledge of the application's code/internal structure. Tests are based on requirements and functionality. It is performed by QA teams.

* Equivalence partitioning
* Boundary value analysis
* Decision table testing
* State transition testing
* Error guessing

18 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.

19 What is the purpose of exit criteria?

* All the high Priority bugs should be closed

Here are the exit criteria: -

* All the test cases should be executed
* If the scheduled time out is arrived
* Exit Criteria ensure that the testing of the application is completed and ready for release.
* An exit criteria is used to denote that a particular test level is completed.

20 When should "Regression Testing" be performed?

Type of software testing that seeks to uncover software errors after changes to the program (e.g. bug fixes or new functionality) have been made, by retesting the program. It is performed by the testing teams.

* when the system is stable and the system or the environment changes
* when testing bug-fix releases as part of the maintenance phase
* It should be applied at all Test Levels
* Regression test suites evolve over time and given that they are run frequently are ideal

candidates for automation.

21 What is 7 key principles? Explain in detail?

#### **1. Testing shows the presence of defects, not their absence**

We test software to discover issues, so that they can be fixed before they are deployed to live environments – this enables us to have confidence that our systems are working. However, this testing process does not confirm that any software is completely correct and completely devoid of issues. Testing helps greatly reduce the number of undiscovered defects hiding in software, but finding and resolving these issues is not itself proof that the software or system is 100% issue-free. This concept should always be accepted by teams, and effort should be made to manage client expectations.

It is important to remember however that while testing shows the presence of bugs and not their absence, thorough testing will give everyone confidence that the software will not fail.

#### **2. Exhaustive testing is impossible**

As much as we would like to believe or wish it true(!), it is absolutely impossible to test EVERYTHING – all combinations of inputs and preconditions – and you could also argue that attempting to do so is not an efficient use of time and budget. However, one of the skills of testing is assessing risks and planning your tests around these – you can then cover vast areas, while making sure you are testing the most important functions.

#### **3. Early testing saves time and money**

Testing early is fundamentally important in the software lifecycle. This could even mean testing requirements before coding has started, for example – amending issues at this stage is a lot easier and cheaper than doing so right at the end of the product’s lifecycle, by which time whole areas of functionality might need to be re-written, leading to overruns and missed deadlines. Essentially, testing early can even help you prevent defects in the first place!

#### **4. Defects cluster together**

This is the idea that certain components or modules of software usually contain the most number of issues, or are responsible for most operational failures. Testing therefore, should be focused on these areas. The Pareto principle of 80:20 can be applied – 80 percent of defects are due to 20 percent of code!

Knowing this could prove to be very valuable for your testing; if we find one defect in a particular module/area there is a strong chance of discovering many more there. Identifying the more complex components, or areas that have more dependencies or are changing the most, for example, can help you concentrate your testing on these crucial risk areas.

#### **5. Beware of the pesticide paradox**

This is based on the theory that when you use pesticide repeatedly on crops, insects will eventually build up an immunity, rendering it ineffective. Similarly, with testing, if the same tests are run continuously then – while they might confirm the software is working – eventually they will fail to find new issues. It is important to keep reviewing your tests and modifying or adding to your scenarios to help prevent the pesticide paradox from occurring – maybe using varying methods of testing techniques, methods and approaches in parallel

#### **6. Testing is context dependent**

Testing is ALL about the context. The methods and types of testing carried out can completely depend on the context of the software or systems – for example, an e-commerce website can require different types of testing and approaches to an API application, or a database reporting application. What you are testing will always affect your approach.

#### **7. Absence-of-errors is a fallacy**

If your software or system is unusable (or does not fulfil users’ wishes) then it does not matter how many defects are found and fixed – it is still unusable. So in this sense, it is irrelevant how issue- or error-free your system is; if the usability is so poor users are unable to navigate, or/and it does not match business requirements then it has failed, despite having few bugs.

22 Difference between Smoke and Sanity?

|  |  |
| --- | --- |
| **Smoke Testing** | **Sanity Testing** |
| 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 |
| The objective of this testing is to verify  the "stability" of the system in order to proceed with more rigorous testing | The objective of the testing is to verify  the "rationality" of the system in order to proceed with more rigorous testing |
| This testing is performed by the developers or testers | Sanity testing is usually performed by testers |
| Smoke testing is usually documented or scripted | Sanity testing is usually not documented and is unscripted |
| Smoke testing is a subset of Acceptance testing | Sanity testing is a subset of Regression testing |
| Smoke testing exercises the entire system from end to end | Sanity testing exercises only the particular component of the entire system |
| Smoke testing is like General Health Check Up | Sanity Testing is like specialized health check up |

**24.** Difference between verification and Validation

|  |  |
| --- | --- |
| **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 fulfils 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 evaluate. |
| Reviews  Walkthroughs  Inspections | Actual Testing is done |

25 Difference between QA v/s QC v/s Tester

|  |  |  |
| --- | --- | --- |
| **Quality Assurance** | **Quality Control** | **Tester** |
| 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. |
| 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. |
| 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. |

26 Explain types of Performance testing.?

* **Load Testing-**Load testing examines how the system behaves during normal and   high loads and determines if a system, piece of software, or computing device can handle high loads given a high demand of end-users.
* **Stress Testing**- A type of performance testing performed to find the behaviour of the application under extreme load.
* **Endurance Testing**- Endurance testing refers to tests typically done to find out whether an application can withstand the processing load it is expected to have to endure for a long period. During endurance tests, memory consumption is observed to determine potential failures.
* **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-** Volume Testing is a type of software testing that is performed to test the performance or behaviour of the system or application under the huge amount of data. Volume testing is also called flood testing and it is a type of performance testing.
* **Scalability Testing-** A scalability test is a type of load testing that measures the application's ability to scale up or down as a reaction to an increase in the number of users. In other words, it tests how the system is going to perform during a sudden spike or fall of user request loads.

27 What is Error, Defect, Bug and failure?

A Mistake in coding is called **Error**. Error found by the tester is called **Defect**. Defect accepted by the development team then it is called **Bug**, Build does not meet the requirement then it is called **Failure.**

28 Difference between Priority and Severity.?

* **Severity** is how seriously the bug is affecting the application. The severity type is defined by the tester based on the written test cases and functionality.
* **Priority** is the order in which developer has to fix the bug. 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.

28. What is Bug Life Cycle?

* 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

            - Rejected

            - Fixed

            - Tested

            - Reopen

**-** Verified

**-** Closed

29. Explain the difference between Functional testing and Non-functional testing

|  |  |
| --- | --- |
| **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 | Non-functional 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 Non-functional testing are  ∙ Performance Testing  ∙ Load Testing  ∙ Volume Testing  ∙ Stress Testing  ∙ Security Testing  ∙ Installation Testing  ∙ Penetration Testing  ∙ Compatibility Testing  ∙ Migration Testing |

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

* SDLC defines all the standard phases which are involved during the software development process, whereas the STLC process defines various activities to improve the quality of the product.
* SDLC is a Development Life Cycle whereas STLC is a Testing Life Cycle.
* In SDLC, the development team creates the high and low-level design plans while In STLC, the test analyst creates the System, Integration Test Plan
* In SDLC, real code is developed, and actual work takes place as per the design documents, whereas in STLC testing team prepares the test environment and executes test cases.
* The SDLC life cycle helps a team to complete successful development of the software while STLC phases only cover software testing.

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

* **Test Scenario-**It gives the idea of what we have to test or testable part of an application is called Test Scenario.
* **Test Cases-** It is a set of conditions under which tester determines whether an application/software is working correctly or not.
* **Test Script-** 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.

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

* A Test Plan is a detailed document that describes the test strategy, objectives, schedule, estimation, deliverables, and resources required to perform testing for a software product. Test Plan helps us determine the effort needed to validate the quality of the application under test.

* **Creating a Test Plan involves the following steps:**
* Product Analysis.
* Designing Test Strategy.
* Defining Objectives.
* Establish Test Criteria.
* Planning Resource Allocation.
* Planning Setup of Test Environment.
* Determine test schedule and estimation.
* Establish Test Deliverables.

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

* There are various methodologies present in agile testing and those are listed below:

            ∙ **Scrum**

            ∙ **eXtreme Programming**

            ∙ Below listed methodologies are used less frequently

            ∙ **Dynamic System Development Method (DSDM)**

            ∙ This is an Iterative and incremental approach that emphasizes on the

          continuous user involvement.

       ∙ **Test Driven Development (TDD)**

       ∙ This is a technique which has short iterations where new test cases covering

          the desired improvement or new functionality are written first.

       ∙ **Feature Driven Development**

       ∙ This is an iterative and incremental software development process and this

          can aim depends on the features.

       ∙ **XBreed**

       ∙ Agile enterprise previously known as Xbreed. It is agile way of managing,

          architecting and monitoring the enterprise.

       ∙ **Crystal**

          Crystal is an adaptive technique mainly used for software development methodologies.

34. Explain the difference between Authorization and Authentication in Web testing**.**

* **Authentication**

           The process of establishing the identity of the user. Authentication can take many

           forms including but not limited to: passwords, biometrics, radio frequency

           identification, etc.

* **Authorization**

           The process of determining that a requester is allowed to receive a service or perform               an operation. Access control is an example of authorization.

35. What are the common problems faced in Web testing?

* Below are five web application testing challenges faced by web developers during the development process.
* Integration. Integration testing exposes problems with interfaces among different program components before deployment. ...
* Interoperability. ...
* Performance. ...
* Quality Testing, Exceptional Services.
* Security. ...
* Usability. ...