

ASSIGNMENT MODULE-2

1) What is Exploratory Testing?

- Though the current trend in testing is to push for automation. Exploratory Testing is a new way of thinking. Automation has its limits.
- It is not a random testing but it is Adhoc testing with purpose of find bugs.
- It is Structured and rigorous.
- It is cognitively (thinking) structured as compared to procedural structure of scripted testing this structure comes from charter, time boxing, etc.
- It is highly teachable and manageable.

2) What is Traceability matrix?

- ✚ To protect against changes you should be able to track back from every system component to the original requirement that caused its presence.
- ✚ A software process should help you keeping the virtual table up-to-date.

3) What is Boundary Value Testing?

- ✚ Boundary Value Analysis is a methodology for designing test cases that concentrates software testing effort on cases near the limits of valid ranges.
- ✚ It is a method which refines equivalence partitioning.
- ✚ It generates testcases that highlight errors better than eauivalence partitioning.
- ✚ The trick is to concentrate software testing efforts at the extreme ends of the equivalence classes.

4) What is Equivalence Partitioning Testing?

- ✚ Aim is to treat group of inputs as equivalent and to select one representative input to test them all.
- ✚ It can be used for all levels of Testing.
- ✚ Equivalent Partitioning says that by testing just one value we have tested the partition (typically a mid-point value is used).
- ✚ If the one value finds a bug the others probably will too.
- ✚ If one doesn't find a bug, the others probably won't either.

5) What is Integration testing?

- ✚ Integration testing is performed to expose defects in the interfaces and in the interaction 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.
- ✚ It tests integration or interfaces between components interactions to different part of the system such as an operating system, file system and hardware or interfaces between systems.
- ✚ Integration testing is done by a specific integration tester or team

There are two levels of Integration Testing

- 1) **Component Integration Testing**
- 2) **System Integration Testing**

6) What determines the level of Risk?

Risk :- A factor that could result in future negative consequences; usually expressed as impact and likelihood.

- + When testing does find defects, the Quality of the software system increases when those defects are fixed.
- + The Quality of systems can be improved through lessons learned from previous projects.
- + Analysis of root causes of defects found in other projects can lead to process Improvement.

There are two types of risk:

- 1) **Product risk**
- 2) **Project risk**

7) What is Alpha Testing?

- + It is always performed by the developers at the software development site.
- + Sometimes it is also performed by Independent Testing Team.
- + Alpha Testing is not open to the market and public
- + It is conducted for the software application and project.
- + It is the form of Acceptance Testing.
- + It is always performed within the Organization
- + It comes under the Category of both White Box testing and Black Box Testing.

8) What is Beta testing?

- + It always performed by the customers at their own site.
- + It is not performed by independent Testing Team
- + Beta Testing is always open to market and Public
- + It is usually conducted for software product
- + It is performed in **Real Time Environment**.
- + It is also the form of Acceptance Testing
- + It is only a kind of Black Box testing

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."**
- ❖ **Component Testing – The testing of individual software components.**

- ❖ **Unit Testing** is a level of the software testing process where **individual units/components of a software/system** are tested. The purpose is to validate that each unit of the software performs as designed.
- ❖ Unit testing is the first level of testing and is performed prior to Integration Testing.
Sometimes known as **Unit Testing, Module Testing or Program Testing.**
- ❖ Component can be tested in isolation – stubs/drivers may be employed
- ❖ Unit testing **frameworks, drivers, stubs and mock or fake objects** are used to assist in unit testing.
 - + A unit is the smallest testable part of an application like functions/procedures, classes, interfaces.
 - + The goal of unit testing is to isolate each part of the program and show that the individual parts are correct.
 - + Unit tests find problems early in the development cycle.
 - + Unit testing is performed using the white box testing method.

10) What is Functional System Testing?

- + Functional System testing is a requirement that specifies a function that a system or system component must perform.
- + A requirement may exist as a text document and/ or a model
 - There are two types of test Approach
 1. Requirement base functional testing
 2. Process based testing

11) What is Non-Functional Testing?

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

- May be performed at all test levels (not just Non Functional system testing)
- Measuring the Characteristics of the system/software can be quantified on a varying scale
- It is the testing of “how” the system works.

- To address this issue, performance testing is carried out to check & fine tune response times

12) What is GUI Testing?

- + Graphical user interface (GUI) testing is the process of testing the system's GUI of the system under test.
- + It involves checking the screens with the controls like menus, buttons, icons, and all type of bars.

13) what is Adhoc Testing?

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

- + It does not follow any test design techniques to create test cases
- + In fact it does not create test cases altogether.
- + This testing is primarily performed if the knowledge of testers in the system under test is very high.
- + Main aim of this testing is to find defects by random checking.
- + Adhoc testing can be achieved with the testing technique called Error guessing.
- + The error guessing technique is a technique where the experienced and good testers are encouraged to think of situations in which the software may not be able to cope.

14) 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.
- + This testing usually identifies :
 - The maximum operating capacity of an application
 - Determine whether current infrastructure is sufficient to run the application.
 - Sustainability of an application with respect to peak user load.
 - Number of concurrent users that an application can support ,and scalability to allow more users to access it.

- Load testing is commonly used for the client/server , web based applications -both Intranet and Internet.

15) 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.

- ✚ Stress testing is used to test the stability & reliability of the system. This test mainly determines the system on its robustness and error handling under extremely heavy load conditions.
- ✚ It even tests beyond the normal operating point and evaluates how the system works under those extreme conditions.
- ✚ Stress Testing is done to make sure that the system would not crash under crunch situations.
- ✚ Stress testing is also known as endurance testing.
- ✚ Under stress testing, AUT is be stressed for a short period of time to know its withstanding capacity.

16) What is Whitebox Testing and list the types of Whitebox testing?

- ✚ Testing based on analysis of an internal structure of the component or the system.
- ✚ Structure based testing technique is also known as "White Box Testing" or glass box testing.
- ✚ In white-box testing the tester is concentrating on how the software does it
For ex. A structural technique may be concerned with exercising loops in the software.

Types of White Box testing:

- 1) Branch condition Testing
- 2) Branch condition combination testing.
- 3) Modified condition decision testing
- 4) Dataflow testing
- 5) Linear code sequence and jump testing

17) What is BlackBox Testing? What are the different black box testing techniques?

Testing either functional or Non functional without reference to the internal structure of the component or system.

- 🚦 Specification based testing technique is also known as 'black-box' or input/output driven testing techniques because they view the software as a black-box with inputs and outputs.
- 🚦 The testers have no knowledge of how the system or component is structured inside the box. In black-box testing the tester is concentrating on what the software does, now how it does it.
- 🚦 The technique of testing without having any knowledge of the interior workings of the application is Black-Box testing.
- 🚦 Typically , when performing a black box test , a tester will interact with the system's user interface by providing inputs and examining outputs without knowing how and where the inputs are worked upon.

Techniques of Black Box testing:

- 1) Equivalence Partitioning
- 2) Boundary Value analysis
- 3) Decision Tables
- 4) State transition testing
- 5) Use-Case testing
- 6) Other Black Box Testing

18) Mention what are the categories of defect?

- 1) **Data Quality / Database Defects:** Deals with improper handling of data in the database
Eg. Values not deleted /inserted into the database properly
Improper /wrong /null values inserted in place of the actual values
- 2) **Critical Functionality Defects:** The occurrence of these bugs hampers the crucial functionality of the application
- 3) **Functionality Defects:** These defects affect the Functionality of the application
- 4) **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.
Examples : Authentication: Accepting an invalid Username/ Password
Authorization: Accessibility to pages though permission not given
- 5) **User Interface Defects:** As the name suggests , the bugs deal with problems related to UI are usually considered less severe.
Eg. Improper error/warning / UI messages
Spelling mistakes
Alignment problems

19) Mention What bigbang testing is?

- ✚ In Big Bang integration testing all the components or modules are 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 the components are integrated together at once, and then tested.

20) What is the purpose of exit criteria?

- ✚ 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) What should “Regression Testing” be performed?

- ✚ Testing of 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.
- ✚ The purpose of Regression testing is to confirm that a recent program or code change has not adversely affected existing features.
- ✚ Regression testing is nothing but full or partial selection of already executed test cases which are re-executed to ensure existing functionalities work fine.
- ✚ This testing is done to make sure that new code changes should not have side effects on the existing functionalities. It ensures that old code still works once the new code changes are done.

22) What are 7 key principles? explain in detail.

- 1) Testing shows presence of Defects
 - 2) Exhaustive Testing is Impossible.
 - 3) Early testing
 - 4) Defect Clustering
 - 5) The Pesticide paradox
 - 6) Testing is context Dependent
 - 7) Absence of Errors Fallacy
- 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.
 - However Testing cannot prove that there are no defects present.

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 risk 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 effort using a Risk Based Approach

3) 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
- Testing activities should start as early as possible in the development life cycle
- Remember from our definition of testing , testing doesn't start once the code has been written.

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) The pesticide Paradox

- If the same tests are repeated over and over again , eventually the same set of testcases 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 parts of the software or system to potentially find more defects.
- Testing identifies bugs , and programmers respond to fix them .

6) 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.
- Whilst, Testing can be 50% of development costs, in NASA's Apollo program it was 80% testing
- 3 to 10 failures per thousand lines of code (KLOC) typical for commercial software.
- 1 to 3 failures per KLOC typical for industrial software
- failures per KLOC for NASA Shuttle code!

7) Absence of Errors Fallacy

- If the system built is unusable and does not fulfil 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...
- 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

Sr no.	Quality assurance	Quality control	Testing
1	Activities which ensure the implement of process, procedure and standard 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/ errors/defects in the software.
2	Focuses on process and procedures rather than conducting actual testing on the system	Focuses on actual testing by executing software with intend to identify / bug/ defects 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 a subset of Quality control

24) Difference between Smoke V/s Sanity Testing

Sr no.	Smoke Testing	Sanity Testing
1	Check the critical functionality	Checks the new functionality
2	It is done in initial stage	It is done after 30 build
3	It checks the stability	It checks the sanity/ rationality
4	Part of acceptance testing	Part of regression testing
5	It is done by tester and developer	It is done by tester
6	Its checks the system end to end	It checks only a particular function of entire system

25) Difference between verification and Validation

Sr no.	Verification	Validation
1	The process of evaluating work products of the 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 of actually meets the user's needs, that the specification 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	Are we building the product right?	Are we building the product right?
4	Plans, Requirement specs, design specs, code, testcases	The actual product/ software.

26) Explain the types of Performance Testing?

- 1) Load testing
- 2) Stress Testing
- 3) Endurance testing
- 4) Spike testing
- 5) Volume testing
- 6) Scalability testing

The focus of performance testing is checking a software programs

Speed: Determines whether the application responds quickly

Scalability: Determines maximum user load the software application can handle

Stability: Determines if the application is stable under varying loads

27) What is Error , Bug and Failure?

- ✚ **Error:** A mistake in coding is called error.
- ✚ **Bug:** A fault in a program which causes the program to perform in an unintended or unanticipated manner. See: anomaly, defect, error, exception, and fault. Bug is terminology of Tester.
- ✚ **Failure:** The inability of a system or component to perform its required functions within specified performance requirements. See: bug, crash, exception, and fault.

28) Difference between priority and severity

Sr no	PRIORITY	SEVERITY
1	Severity is absolute and customer-focused	Priority is Relative and Business Focused
2	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.	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.
3	Severity can be critical	Priority can be Low, medium, High and Critical

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.
- ✚ The duration or time span between the first time defects is found and the time that it is closed successfully, rejected, postponed or deferred is called as 'Defect Life Cycle'
- ✚ When a bug is discovered, it goes through several states and eventually reaches one of the terminal states, where it becomes inactive and close
- ✚ The process by which the defect moves through the life cycle is depicted next slide.

30) Explain the difference between functional Testing and Non- Functional Testing?

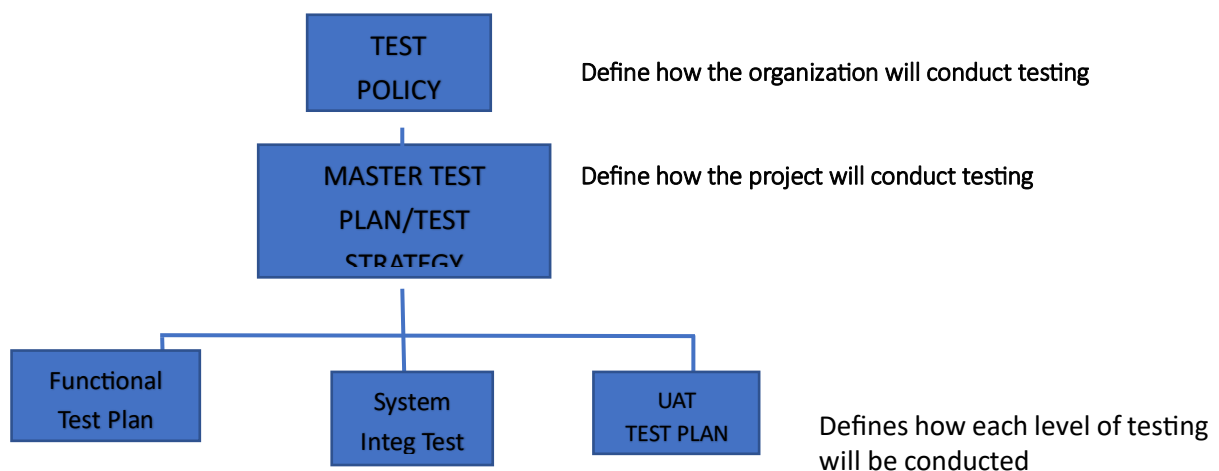
Sr no.	Functional Testing	Non Functional Testing
1	Functional testing is performed using functional specification provided by the client and verifies the system against the functional requirements	Non functional testing checks the performance reliability, scalability, and other non- functional aspects of the software system
2	Functional testing is executed first	Non functional testing should be performed after functional testing
3	Manual testing or Automation tools can be used for functional testing	Using tools will be effective for this testing
4	Business requirements are the inputs to functional testing	Performance parameters like speed, scalability are inputs to non functional testing
5	Functional testing describes what the product does	Non functional testing describes how good the product works
6	Easy to do manual testing	Tough to do manual testing

31) What is the difference between test scenarios , test cases and testscript?

Sr no.	Test scenarios	Test cases	Testscript
1	A scenario is any functionality that can be tested. It is also called test condition or test possibility	Test case involve the set of steps , conditions and inputs which can be used while performing the testing tasks.	A set of sequential instruction that detail how to execute a core business function
2	Test scenario is 'what to be tested'	Test case is how to be tested	One script is written to explain how to simulate each business scenario
3	Test scenario is nothing but the test procedure	Test case consist of set of input values, execution precondition, expected results and executed post-condition developed to cover certain test condition	Identifies the test condition that is being satisfied for each step , if applicable
4	The scenarios are derived from the usecases	Testcases are derived from test scenario	Identified the input /test data that should be entered for each transaction

32) Explain what Test plan is? What is the information that should be covered?

- ✚ A document describing the scope , approach, resources, and schedule of intended test activities
- ✚ All project require a set of plan and strategies which define how the testing will be conducted.
- ✚ There are number of levels at which these are defined:



33) What is 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

34) What is severity?

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.

35) Bug Categories are...

- ✚ There are different categories of BUG
 - 1) Performance Bugs
 - 2) Security Bugs
 - 3) Unit level Bugs
 - 4) Functional Bugs
 - 5) Usability Bug
 - 6) Low priority bugs
 - 7) High priority bugs
 - 8) Critical bugs

36) Advantage of Bugzilla

- ✚ Open source, free bug tracking tool
- ✚ Automatic Duplicate Bug detection
- ✚ Search option with advanced features
- ✚ File/ modify / bugs by email
- ✚ Move bugs between installs
- ✚ Time tracking

37) Difference between priority and severity

Sr no	Priority	Severity
1	Priority is Relative and Business Focused	Severity is absolute and customer focused
2	Priority can be Low, Medium, High, critical	Severity can be Critical, moderate, high, cosmetic
3	Priority defines the order in which we should resolve a defect	It is extent to which the defect can affect software

38) What are the different methodologies of Agile Development model?

- ✚ Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product
- ✚ Agile Methods break the product into small incremental builds.

- ✚ These builds are provided in iterations.
- ✚ Each iteration typically lasts from about one to three weeks.
- ✚ Every iteration involves cross functional teams working simultaneously on various areas like planning, requirements analysis, design, coding, unit testing, and acceptance testing.
- ✚ At the end of the iteration a working product is displayed to the customer and important stakeholders.

39) Explain the difference between authorization and Authentication in web testing . what are common problems faced in web testing?

Authentication: Accepting an invalid username/password

Authorization: Accessibility to pages though permission not given

The common problems faced in web testing are:

- 1) The bugs deal with problems related to UI are usually considered less severe
For eg. Improper error/ warning/UI messages
Spelling mistakes
Alignment problem

40) When to used usability testing?

Asthetics and design are important .how well a product looks usually determines how well it works

There are many software applications /websites which miserably fail once launched

- 1) Before any design decisions are made
- 2) When its time to evaluate and iterate
- 3) After launch
- 4) In high risk and low certainty situations

41) What is the procedure of 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 type of bars- tool bar, menubar, dialog boxes and windows etc.

1) MANUAL BASED TESTING:

Under this approach, graphical screens are checked manually by testers in conformance with the requirements stated in business requirements document.

2) RECORD AND REPLAY:

GUI testing can be done using automation tools. This is done in 2 parts. During Record , test steps are captured into the automation tool. During playback, the recorded test steps are executed on the Application under Test. Example of such tools - QTP

3) MODEL BASED TESTING:

Model is a graphical description of system's behavior. It helps us to understand and predict the system behavior. Models help in a generation of efficient test cases using the system requirements.

