

Software Testing Assignment

Module 2

1) What is Exploratory Testing?

Through the current trend in testing is to push for automation exploratory testing is a new way of thinking.

Exploratory testing is a concurrent process where:

- Test design, execution and logging happen simultaneously
- Testing is often not recorded
- Makes use of experience, heuristics and test patterns
- Testing is based on a test charter that may include.
 - Scope of the testing (in and out)
 - The focus of exploratory testing is more on testing as “thinking” activity
 - A brief description of how tests will be performed
 - Expected problems
- Is not random testing but it is Adhoce testing with purpose of find bugs.
- Is structured and rigorous
- Is cognitively (thinking) structured as compared to procedural structural of scripted testing. This structure comes from Charter, time boxing etc.
- Is highly teachable and manageable.
- Exploratory testing is a type of testing where test cases are not created in advance but testers check system on the fly. They note down the idea about what to test before test execution.
- Exploratory testing is widely used in agile model and is all about discovery, investigation and learning. It emphasizes personal freedom and responsibilities of the individual tester.

2) 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.

- Traceability can be horizontal through all the test documentation for a given test level or it can be vertical through the layers of development documentation.
- A traceability matrix is a document that co-relates any two baseline documents that require a many to many relationship to check the completeness of the relationship.

- It is used to track the requirements and to check the current project requirements met.

3) What is Boundary Value testing?

- Boundary value analysis is a methodology for designing test cases that concentrates software testing efforts on cases near the limits of valid ranges.
- Boundary value analysis is a method which refines equivalence partitioning.
- Boundary value analysis generates test cases that highlight the error better than equivalence partitioning.
- Boundary testing is the process of testing between extreme ends or boundaries between partitions of the inputs
- So these extreme ends like start-end, Lower-Upper, Maximum-Minimum, Just inside-Just outside values are called boundary value and the testing is called "Boundary Testing".

4) What is Equivalence Partitioning Testing?

- Equivalence partitioning testing aim is to treat groups of inputs as equivalent and to select one representative input to test them all.
- Equivalence partitioning testing is the process of defining the optimum numbers of tests by:
 - Reviewing documents such as the **Functional Design Specification** and **Detail Design Specification**, and identifying each input condition within the function.
 - Selecting input data that is representative of all other data that would likely invoke the same process for that particular condition.
- Equivalence partitioning testing or Equivalence class partitioning is the type of black box testing technique which can be applied to all level of software testing like unit, integration, system etc. In this technique input data units are divided into equivalent partitions that can be used to derive test cases which reduces time required for testing because of small number of test cases.

5) What Is Integration Testing?

- Integration testing-testing performed to expose defects in the interface and in the interaction between integrated components or system.

- 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?

- Risk should be evaluated at the Business Level, Technological Level, Project Level and Testing Level.
- Risk also used to decide where to start and where more testing is needed

7) What is Alpha Testing?

- Alpha testing is definitely performed and carried out at the developing organizations location with the involvement of developers.
- It is always performed by the developers at the software development site.
- Sometimes it is also performed by independent testing team.
- It is always performed in virtual Environment.
- It comes under the category of both White box testing and Black box testing

8) What is Beta Testing?

- Beta testing is performed and carried out by users or you can say people at their own location and site using customer data
- It is always performed by the customer at their own site
- It is not performed by independent testing team
- Beta testing is always open to the market and public.
- It is performed in Real Time Environment
- It is only kind of Black box testing

9) What is Component Testing?

- Component testing- A minimal software item that can be tested in isolation. It means “A component is the smallest testable part of software.”
- Component testing – The testing of individual software components.
- Component testing is a level of 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

- Component testing is the first level of testing and is performed to integration testing.
- Sometimes also known as unit testing, module testing or program testing

10) What is Functional System Testing?

- Functional system testing – A requirement that specifies a function that a system or system component must perform.
- A requirement may exist as a text document or a model.

11) What is Non-Functional Testing?

- 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**
- Measuring the characteristics of the system/software that can be quantified on varying scale-e.g. performance test scaling
- Non-functional testing includes, but is not limited to, performance testing, load testing, stress testing, usability testing, maintainability testing, reliability testing and portability testing.
- It is the testing of “how” the system works. Non – Functional testing may be performed at all test levels.

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 screen with the controls like menus, buttons, icons and all type of bars - tool bar , menu bar , dialog boxes and windows etc.
- There are some Approach of GUI
 - Manual based testing
 - Record and Replay
 - Model based testing

13) What is Adhoc Testing?

- Main aim of Adhoc testing is to find defects by random checking.
- Adhoc testing is informal testing type with an aim to break the system.
- It does not follow any test design techniques to create test case.

- In fact it does not create test case altogether.
- This testing is primarily performed if the knowledge of testers in the system under test is very high.
- Adhoc testing can be achieved with the testing technique called error guessing.

14) What is load Testing?

- Load testing – It's a performance testing to check system behavior under load. Testing an application under heavy loads, such as testing of a website under a range of loads to determine at what point the system's response time degrades or fails.
- Load testing is kind of performance testing which determines a system's performance under real life load conditions. This testing helps determine how the application behaves when multiple users access it simultaneously.

15) What is Stress Testing?

- 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 done to make sure that the system would not crash under crunch situation.
- Stress testing is also known as endurance testing.

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

- White box testing: Testing based on an analysis of the internal structure of the component or system.
- Structure based testing technique is also known as "white box" or "glass box" testing technique because here the testers require knowledge of how the software is implemented, how it works.
- White box testing is also called glass testing or open box testing. In order to perform white box on an application, the tester needs to possess knowledge of the internal working of the code..
- Types of coverage
 - Statement coverage
 - Decision coverage
 - Condition coverage

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

- The technique of testing without having any knowledge of the interior workings of the application is Black box testing.
- The tester is oblivious to the system architecture and does not have access to the source code.
- Typically when performing a black box test , a tester interact with the system's user interface by providing inputs and examining outputs without knowing how and where the inputs are worked.
- Techniques of Black box testing
 - Equivalence partitioning
 - Boundary value analysis
 - Decision Tables
 - State transition testing
 - Use case testing
 - Other black box testing
 - Syntax or pattern testing

18) Mention what are the categories of defects?

- There are some categories which are as below
 - Data quality/Database Defects
 - Critical Functionality Defects
 - Functionality Defects
 - Security Defects
 - User Interface Defects

19) Mention what big bang testing is?

- In big bang integration testing all components and modules is integrated simultaneously, after which everything is tested as a whole.
- Big bang testing has the advantages 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.

20) What is the purpose of exist criteria?

- 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 (handover from system test to UAT)

21) When should “Regression Testing “be performed?

- Change in requirements and code is modified according to the requirements
- New feature is added to the software
- Defect fixing
- Performance issue fix

22) What is 7 key principles? Explain in detail?

- Testing shows presence of Defects
- Exhaustive Testing is Impossible
- Early testing
- Defect Clustering
- The pesticide Paradox
- Testing is context dependent
- 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 reduce the probability of undiscovered defects remaining in the software but. Even if no defects are found, it is not a proof of correctness.
- As we find more defects. The probability of undiscovered defects remaining in a system reduces

2. 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 combination you would need
- That is we must Priorities our testing efforts 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.

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
- In other words, most defects found during testing are usually confined to small number of modules

5. The pesticide Paradox:

- If the same tests are repeated over and over again, eventually the same set of test case will no longer find any new defects
- To overcome this “pesticides 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.

6. Testing is context dependent:

- Testing is basically context dependent.
- Testing is done differently context
- Different kinds of sites are tested differently

7. Absence of Errors 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.
- Even after defects have been resolved it may still be unusable and/or does not fulfill the users need and expectations

23) Difference between QA v/s QC v/s Tester

- QA (Quality Assurance)
 - It is a subset of Software Test Life Cycle (STLC)
 - Preventive activities
 - Process oriented activities
 - Focus on processes and procedures rather than conducting actual testing on the system
- QC (Quality Control)
 - QC can be considered as the subset of Quality Assurance
 - It is a corrective process
 - Product oriented activities
 - Focuses on actual testing by executing software with intend to identify bug/defect through implementation of procedures and process
- Tester
 - Testing is the subset of Quality Control
 - It is a preventive process

- Product oriented activities
- Focuses on actual testing

24) Difference between smoke and sanity?

- **Smoke Testing**

- Smoke testing is performed to ascertain that the critical functionalities of the program is working fine
- This testing is performed by the developers or testers
- Smoke testing is usually documented or scripted
- Smoke testing is a subset of Regression testing

- **Sanity Testing**

- Sanity testing is done to check the new functionality / bugs have been fixed
- Sanity testing is usually performed by testers
- Sanity testing is usually not documented and is unscripted
- Sanity testing is a subset of Acceptance Testing

25) Difference between verification and validation

- **Verification :**

- 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
- 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 requirement
- Are we building the product right?

- **Validation :**

- 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 actually meets the user's needs and that the specification were correct in the first place in other word , to demonstrate that the product fulfills its intended use when placed in its intended environment
- Are we building the right product?

26) Explain types of performance testing.

- Load testing
- Stress testing
- Endurance testing
- Spike testing
- Volume testing
- Scalability testing

1. Load testing:

It's a performance testing to check system behavior 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

2. Stress testing:

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 capacity queries, continuous input to system or database load.

27) What are Error, Defects, Bug and Failure?

- “A mistake in coding is called error, error found by tester is called defect, defect accepted by development team then it is called bug, build does not meet the requirements then it is failure”

28) Difference between priority and severity

- **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 mention then the developer has to fix it at the earliest. The priority status is set based on the customer requirements.
- **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 given defect has on the system

29) What is Bug Life Cycle?

- “A computer bug is an error, flow, mistake, failure or fault in a computer program that prevents it from working correctly or produces an incorrect result. Bug arise from mistake and errors made by people, in either a program’s source code or its design.”
- When bug is discovered, it goes through several states and eventually reaches one of the terminal states where it becomes inactive and closed

30) explain the difference between Functional testing and Non-functional testing

- **Functional Testing :**
 - Functional testing based on an analysis of the specification of the functionality of a component or system
 - Functional testing is based on the Functions and features – may be applied at all test levels
 - Functional testing verifies that each functions of the software application operates in conformance with the requirement specification
- **Non – Functional testing :**
 - Non – functional testing is the attributes of a component or system that do not relate to functionality e.g. reliability , efficiency , usability , interoperability , maintainability and portability
 - Measuring the characteristics of the system/software that can be quantified on varying scale
 - Non – functional testing includes, but is not limited to, performance testing, load testing , stress testing , usability testing , maintainability testing , reliability testing and portability testing.

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

❖ **Software Testing Life Cycle :**

- STLC is mainly related to software testing
- It focuses only on testing the software
- STLC involves only five phases or steps
- In STLC, less number of members are needed
- Goal of STLC is to complete successful testing of software

❖ **Software Development Life Cycle :**

- SDLC is mainly related to software development
- Beside development other phases like testing is also included
- SDLC involves total six phases or steps

- In SDLC, more number of developers are required for the whole process

33) What is the difference between test scenario, test cases and test script?

❖ **Test Scenario:**

- Is any functionality that can be tested
- Is derived from test artifacts like Business Requirements Specification (BRS) and Software Requirements Specification (SRS)
- Helps test the end to end functionality in Agile way
- Is more focused on what to test
- Test less time and fewer resources to create

❖ **Test Cases:**

- Is a set of actions executed to verify particular features or functionality
- Is mostly derived from test scenario
- Helps in exhaustive testing of an app
- Is focused on what to test and how to test
- Requires more resources and time

❖ **Test Script :**

- Is a set of instructions to test an app automatically
- Is mostly derived from test cases
- Helps to test specific things repeatedly
- Is focused on the expected result
- Requires less time for testing but more resources for scripts creating and updating

34) 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 same frame to fix the defect. If the high priority is mentioned then the developer has to fix it at earliest. The priority status to set based on the customer requirements.
- For example : If the company name is misspelled in the home page of the website then the priority is high and severity is high and severity is low to fix.

35) What is 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.
- For example : if an application of the web page crashes when a remote link is clicked in this case clicking the remote link by an user is rare but the impact of application crashing severe. So the severity is high but priority is low.

36) Bugs categories are...

- There are some categories
 - Security
 - Database
 - Functionality (Critical/General)
 - UI

37) Advantages of Bugzilla

- Open source, free bug tracking
- Automatic Duplicate bug detection
- Search option with advanced features
- File/Modify bugs by mail
- Move bugs between installs
- Multiple authentication methods
- Time tracking
- Automated bug reporting, has an API to interact with system

38) Difference between severity and priority

❖ Severity :

- Severity is a term that denotes how severely a defect can affect the functionality of the software
- Severity is basically a parameter that denotes the total impact of a given defect on any software
- The value of severity is objective
- The value of severity changes continually from time to time
- The testing engineer basically decides a defect's severity level

❖ Priority

- Priority is a term that defines how fast we need to fix a defect
- Priority is basically a parameter that decides the order in which we should fix the defect.

- The value of priority is subjective
- The value of priority change from time to time
- The product manager basically decides a defect's priority level

39) What are the different Methodology in Agile Development Model ?

1. Scrum methodology:

Scrum is a lightweight framework of Agile Project Management, it can be adopted to conduct iterative and all types of incremental projects.

Due to specific characteristics like simplicity, sustained productivity and strength for blending several underlying approaches adapted by other agile methods, Scrum has obtained popularity over the years.

2. Kanban:

Kanban is an eminently visual workflow management approach, that can be employed for visualizing and thoroughly maintaining the making of products, it focuses on continual delivery of the product , but is not making stress to the entire software development life cycle.

Similar to scrum, kanban is the process developed for supporting collaborative teamwork more effectively.

40) Explain the difference between Authorization and Authentication in web testing. What are the common problems faced in web testing?

Authentication	Authorization
Authentication verifies who the user is.	Authorization determines what resources a user can access.
Authentication works through passwords, one-time pins, biometric information, and other information provided or entered by the user.	Authorization works through settings that are implemented and maintained by the organization.
Authentication is the first step of a good identity and access management process.	Authorization always takes place after authentication.
Authentication is visible to and partially changeable by the user.	Authorization isn't visible to or changeable by the user.

Example: By verifying their identity, employees can gain access to an HR application that includes their personal pay information, vacation time, and 401K data.	Example: Once their level of access is authorized, employees and HR managers can access different levels of data based on the permissions set by the organization.
--	--

41) When to use Usability Testing?

- Usability testing is a method of testing the functionality of a website, app or other digital product by observing real users as they attempt to complete tasks on it. The users are usually observed by researchers working for a business.
- Usability testing can and should be conducted on the current iteration of a product before beginning any new design work, after you have begun the strategy work around a brand new site or app.

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 Color 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.

43) Write agile manifesto principles?

- **Individuals and interactions, Over processes and tools** - Suppose the team finds any issue in software then they search for another process or tool to resolve the issue. But, in Agile, it is preferable to interact with client, manager or team regarding issue and make sure that the issue gets resolved.
- **Working software, Over comprehensive documentation** - Documentation is needed, but working software is much needed. Agile is not saying that documentation is not needed, but working software is much needed. For example, you have 20-page documents, but you do not have a single prototype of the software. In such a case, the client will not be happy because, in the end, the client needs a document.

- **Customer collaboration, Over contract negotiation** - Contract negotiation is important as they make the budget of software, but customer collaboration is more important than over contract negotiation. For example, if you stuck with the requirements or process, then do not go for a contract which we have negotiated. You need to interact with the customer, gather their requirements.
- **Responding to change, over following a plan** - In the waterfall model, everything is planned, i.e., at what time, each phase will be completed. Sometimes you need to implement the new requirements in the middle of the software, so you need to be versatile to make changes in the software.