Harsh Kacha

Module–2(Manual Testing)

(1) What is Exploratory Testing?

- 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 a “thinking” activity.
- Risk Analysis-> Charter-> ExploratorySessions-> Debriefing
- Though the current trend in testing is to push for automation, exploratory testing is a new way of thinking. Automation has its limits
- Is not random testing but it is Adhoc testing with purpose of find bugs
- Is cognitively (thinking) structured as compared to procedural structure of scripted testing. This structure comes from Charter, time boxing etc.
- Is not a technique but it is an approach. What actions you perform next is governed by what you are doing currently

(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 (e.g. system testing, from test conditions through test cases to test scripts) or it can be vertical through the layers of development documentation (e.g. from requirements to components).
- To protect against changes you should be able to trace 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.
- Types of Traceability Matrix
- (1) Forward Traceability – Mapping of Requirements to Test cases
- (2) Backward Traceability – Mapping of Test Cases to Requirements
- (3) Bi-Directional Traceability - A Good Traceability matrix is the References from test cases to basis documentation and vice versa.
  - Pros of Traceability Matrix
- Make obvious to the client that the software is being developed as per the requirements.
- To make sure that all requirements included in the test cases
- To make sure that developers are not creating features that no one has requested
- Easy to identify the missing functionalities.
- If there is a change request for a requirement, then we can easily find out which test cases need to update.
  - Cons of Traceability Matrix
- No traceability or Incomplete Traceability Results into:
- Poor or unknown test coverage, more defects found in production

- It will lead to miss some bugs in earlier test cycles which may arise in later test cycles. Then a lot of discussions arguments with other teams and managers before release.
- Difficult project planning and tracking, misunderstandings between different teams over project dependencies, delays, etc

(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
- Boundary value analysis is a method which refines equivalence partitioning.
- Boundary value analysis generates test cases that highlight errors better than equivalence partitioning.
- The trick is to concentrate software testing efforts at the extreme ends of the equivalence classes.
- At those points when input values change from valid to invalid errors are most likely to occur.
- Boundary Value Analysis (BVA) uses the same analysis of partitions as EP and is usually used in conjunction with EP in test case design

(4) What is Equivalence partitioning testing?

- Aim is to treat groups of inputs as equivalent and to select one representative input to test them all
- EP can be used for all Levels of Testing
- Equivalence partitioning is the process of defining the optimum number of tests by:
- Reviewing documents such as the Functional Design Specification and Detailed Design Specification, and identifying each input condition within a function,
- Selecting input data that is representative of all other data that would likely invoke the same process for that particular condition.
- If we want to test the following IF statement: “If value is between 1 and 100 (inclusive) (e.g value >=1 and value <=100) Then...”
- We could put a range of numbers as shown in the below figure.
- The numbers fall into a partition where each would have the same, or equivalent, result i.e. an Equivalence Partition (EP) or Equivalence Class
- EP says that by testing just one value we have tested the partition (typically a mid-point value is used). It assumes that:
- If one value finds a bug, the others probably will too
- If one doesn't find a bug, the others probably won't either
- In EP we must identify Valid Equivalence partitions and Invalid Equivalence partitions where applicable (typically in range tests)
- The Valid partition is bounded by the values 1 and 100
- Plus there are 2 Invalid partitions

(5) What is Integration testing?

- Integration Testing - Testing performed to expose defects in the interfaces and in the interactions 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.
- The purpose of this level of testing is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in Integration Testing.
- Integration testing tests integration or interfaces between components, interactions to different parts 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 test team.
- Components may be code modules, operating systems, hardware and even complete systems
- There are 2 levels of Integration Testing
- Component Integration Testing
- System Integration Testing
  - Need of Integration Testing
- A Module in general is designed by an individual software developer who understanding and programming logic may differ from other programmers. Integration testing becomes necessary to verify the software modules work in unity
- At the time of module development, there wide chances of change in requirements by the clients. These new requirements may not be unit tested and hence integration testing becomes necessary
- Interfaces of the software modules with the database could be erroneous
- External Hardware interfaces, if any, could be erroneous
- Inadequate exception handling could cause issues.
  - Component Integration Testing
- Component Integration Testing: Testing performed to expose defects in the interfaces and interaction between integrated components
- Usually formal (records of test design and execution are kept)
- All individual components should be integration tested prior to system testing
- The following testing techniques are appropriate for Integration Testing:
- Functional Testing using Black Box Testing techniques against the interfacing requirements for the component under test
- Non-functional Testing (where appropriate, for performance or reliability testing of the component interfaces, for example)
  - System Integration Testing
- It tests the interactions between different systems and may be done after system testing.
- It verifies the proper execution of software components and proper interfacing between components within the solution.
- The objective of SIT Testing is to validate that all software module dependencies are functionally correct and that data integrity is maintained between separate modules for the entire solution.
- As testing for dependencies between different components is a primary function of SIT Testing, this area is often most subject to Regression Testing.

(6) What determines the level of risk?

- The level of risk in software testing is determind by several factors including the complexity and criticality of the system, the skil and experience of the testers,availability of resources,time constraints,changes in requirements,dependencies on other systems,and regulatory compliance requirements.

(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 always performed in Virtual Environment.
- It is always performed within the organization.
- It is the form of Acceptance Testing.
- Alpha Testing is definitely performed and carried out at the developing organizations location with the involvement of developers.
- It comes under the category of both White Box Testing and Black Box Testing.

(8) What is beta testing?

- It is always performed by the customers at their own site.
- It is not performed by Independent Testing Team.
- Beta Testing is always open to the market and public.
- It is usually conducted for software product.
- It is performed in Real Time Environment.
- It is always performed outside the organization.
- It is also the form of Acceptance Testing.
- Beta Testing (field testing) is performed and carried out by users or you can say people at their own locations and site using customer data.
- Beta Testing is always performed at the time when software product and project are marketed.
- It is always performed at the user's premises in the absence of the development team.
- It is only a kind of Black Box Testing.
- It is also considered as the User Acceptance Testing (UAT) which is done at customers or users area.
- Beta testing can be considered "pre-release" testing.
- Pilot Testing is testing to product on real world as well as collect data on the use of product in the classroom.

(9) What is component testing?

- Component(Unit) – 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.
- Functional and Non-Functional testing
- Unit tests are typically written and run by software developers to ensure that code meets its design and behaves as intended with debugging tool.
- 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 by using the White Box Testing method.
- Test Approach :
- Test-First/Test-Driven approach – create the tests to drive the design and code
- Three steps:
- 1. Design test that defines how you think a small part of the software should behave (Incremental development).
- 2. Make the test run as easily and quickly as you can. Don't be concerned about the design of code, just get it to work!
- 3. Clean up the code. Now that the code is working correctly, take a step back and refactor to remove any duplication or any other problems that were introduced to get the test to run.
- Unit testing in Extreme Programming involves the extensive use of testing frameworks. A unit test framework is used in order to create automated unit tests. Unit testing frameworks are not unique to extreme programming, but they are essential to it.
- Tests are written before the code
- Rely heavily on testing frameworks
- All classes in the applications are tested
- Quick and easy integration is made possible

(10) What is functional system testing?

- Functional System Testing : A requirement that specifies a function that a system or system component must perform
- There is two types of Test Approach
- Requirement Based Functional Testing
- Process Based Testing
- Functional System Testing Functionality As below:
- Accuracy - Provision of right or agreed results or effects
- Interoperability - Ability to interact with specified systems
- Compliance - Adhere to applicable standards, conventions, regulations or laws
- Auditability - Ability to provide adequate and accurate audit data
- Suitability - Presence and appropriateness of functions for specified tasks

- Requirement Based Testing

- Testing against requirements and specifications
- Test procedures and cases derived from:
- detailed user requirements
- system requirements functional specification
- User documentation/instructions
- high level System design
- Starts by using the most appropriate black-box testing techniques

- Business Process Based Testing

- Test procedures and cases derived from:
- Expected user profiles
- Business scenarios
- Use cases
- Testing should reflect the business environment and processes in which the system will operate.

(11) What is Non-Functional Testing?

- Non-Functional Testing: Testing the attributes of a component or 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 Systems Testing)
- Measuring the characteristics of the system/software that can be quantified on a 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.
- The term non-functional testing describes the tests required to measure characteristics of systems and software that can be quantified on a varying scale, such as response times for performance testing.
- To address this issue, performance testing is carried out to check & fine tune system response times. The goal of performance testing is to reduce response time to an acceptable level
- Hence load testing is carried out to check systems performance at different loads i.e. number of users accessing the system
- Non - Functional Testing Examples
- Web Based Testing
- Desktop Based Testing
- Mobile Based Testing
- Game Based Testing

(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 screens with the controls like menus, buttons, icons, and all types of bars – tool bar, menu bar, dialog boxes and windows etc.
* WHAT DO YOU CHECK IN 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.
* Approach of GUI Testing
(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
- A 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.

(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 is does not create test cases altogether!
- This testing is primarily performed if the knowledge of testers in the system under test is very high.
- Testers randomly test the application without any test cases or any business requirement document.
- Adhoc Testing does not follow any structured way of testing and it is randomly done on any part of application.
- Main aim of this testing is to find defects by random checking.
- Adhoc testing can be achieved with the testing technique called Error Guessing
- Error guessing can be done by the people having enough experience on the system to "guess" the most likely source of errors.

- The Error guessing 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.
- Some people seem to be naturally good at testing and others are good testers because they have a lot of experience either as a tester or working with a particular system and so are able to find out its weaknesses.
- This is why an error guessing approach, used after more formal techniques have been applied to some extent, can be very effective.
- It also saves a lot of time because of the assumptions and guessing made by the experienced testers to find out the defects which otherwise won't be able to find.

* Types of Adhoc Testing

(1) Buddy Testing

- Two buddies mutually work on identifying defects in the same module. Mostly one buddy will be from development team and another person will be from testing team. Buddy testing helps the testers develop better test cases and development team can also make design changes early. This testing usually happens after unit testing completion.

(2)Pair testing

- Two testers are assigned modules, share ideas and work on the same machines to find defects. One person can execute the tests and another person can take notes on the findings. Roles of the persons can be a tester and scriber during testing.

(3) Monkey Testing

- Randomly test the product or application without test cases with a goal to break the system.

(14) What is load testing?

- Load testing - Its 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.
- Load testing is a 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.
- This testing usually identifies –
- The maximum operating capacity of an application
- Determine whether current infrastructure is sufficient to run the application
- Sustainability of 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.
- It is a type of non-functional testing. Load testing is commonly used for the Client/Server, Web based applications – both Intranet and Internet.

* Need For Load Testing

- Some extremely popular sites have suffered serious downtimes when they get massive traffic volumes. E-commerce websites invest heavily in advertising campaigns, but not in Load Testing to ensure optimal system performance, when that marketing brings in traffic.
- Popular toy store Toysrus.com, could not handle the increased traffic generated by their advertising campaign resulting in loss of both marketing dollars, and potential toy sales.

* Why Load Testing?
- Load testing gives confidence in the system & its reliability and performance.
- Load Testing helps identify the bottlenecks in the system under heavy user stress scenarios before they happen in a production environment.
- Load testing gives excellent protection against poor performance and accommodates complementary strategies for performance management and monitoring of a production environment.

(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 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.
- Most prominent use of stress testing is to determine the limit, at which the system or software or hardware breaks.
- It also checks whether system demonstrates effective error management under extreme conditions.
- The application under testing will be stressed when 5GB data is copied from the website and pasted in notepad.
- Notepad is under stress and gives ‘Not Responded’ error message.
- Examples of stress conditions include:
- Excessive volume in terms of either users or data; examples might include a denial of service (DoS) attack or a situation where a widely viewed news item prompts a large number of users to visit a Web site during a three-minute period.
- Resource reduction such as a disk drive failure.

(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.
- In white-box testing the tester is concentrating on how the software does it.
- For example, a structural technique may be concerned with exercising loops in the software.

- Different test cases may be derived to exercise the loop once, twice, and many times. This may be done regardless of the functionality of the software.
- Structure-based techniques are also used in system and acceptance testing, but the structures are different.
- White box testing is the detailed investigation of internal logic and structure of the code.
- White box testing is also called glass testing or open box testing. In order to perform white box testing on an application, the tester needs to possess knowledge of the internal working of the code.

- **Types of White Box Testing**

- 1. Path Testing
- 2. Loop Testing
- 3. Conditional Testing
- 4. Unit Testing
- 5. Mutation Testing
- 6. Integration Testing
- 7. Penetration Testing
- 8. Testing based on Memory Perspective
- 9. Test Performance of the Program

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

- Black-box testing: 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, not how it does it.

- Specification-based techniques are appropriate at all levels of testing (component testing through to acceptance testing) where a specification exists.

- 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 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 – Aim is to treat groups of inputs as equivalent and to select one representative input to test them all

- Plus there are 2 Invalid partitions

2.Boundary value analysis - Boundary value analysis is a methodology for designing test cases that concentrates software testing effort on cases near the limits of valid ranges

- Boundary value analysis is a method which refines equivalence partitioning.

3. Decision tables - The techniques of equivalence partitioning and boundary value analysis are often applied to specific situations or inputs.

- The other two specification-based software testing techniques, decision tables and state transition testing are more focused on business logic or business rules.

- A decision table is a good way to deal with combinations of things (e.g. inputs).

4.State transition testing - State Diagram: A diagram that depicts the states that a component or system can assume, and shows the events or circumstances that cause and/or result from a change from one state to another. [IEEE 610]

-State Table: A grid showing the resulting transitions for each state combined with each possible event, showing both valid and invalid transitions.

-State Transition: A transition between two states of a component or system.

-State Transition Testing: A black box test design technique in which test cases are designed to execute valid and invalid state transitions. Also known as N-switch testing.

5. Use-case Testing

6.Other Black Box Testing

(18) Mention what are the categories of defects?

- Categories of Defects

- Data quality

- Database Defect

- Critical functionality Defect

- Functionality Defect

- User interface Defect

- Security Defect

(19) Mention what bigbang 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.

- Here all component are integrated together at once, and then tested.

* Advantages:

- Convenient for small systems.

* Disadvantages:

- Fault Localization is difficult.

- Given the sheer number of interfaces that need to be tested in this approach, some interfaces links to be tested could be missed easily.

- Since the integration testing can commence only after “all” the modules are designed, testing team will have less time for execution in the testing phase.

(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) When should "Regression Testing" be performed?

- 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.

- Regression Testing: 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.

- If the test is re-run and passes you cannot necessarily say the fault has been resolved because ..

- You also need to ensure that the modifications have not caused unintended side-effects elsewhere and that the modified system still meets its requirements – Regression Testing

- Regression testing should be carried out:

- 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

- It should be considered complete when agreed completion criteria for regression testing have been met

- Regression test suites evolve over time and given that they are run frequently are ideal candidates for automation

(22) What is 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.

- As we find more defects, the probability of undiscovered defects remaining in a system reduces.

- However Testing cannot prove that there are no defects present

- 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.

- For example: In an application in one screen there are 15 input fields, each having 5 possible values, then to test all the valid combinations you would need 30 517 578 125 (515) tests.

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

- So, accessing and managing risk is one of the most important activities and reason for testing in any project.

- We have learned that we cannot test everything (i.e. all combinations of inputs and pre-conditions).

- 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

- These activities should be focused on defined objectives – outlined in the Test Strategy

- Remember from our Definition of Testing, that 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’

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

- Similarly, most operational failures of a system are usually confined to a small number of modules

- An important consideration in test prioritisation!

(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.

- To overcome this “pesticide paradox”, the test cases need to be regularly reviewed and revised, and new and different tests need to be written to exercise different parts of the software or system to potentially find more defects.

- Testing identifies bugs, and programmers respond to fix them

- 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 basically context dependent.

- Testing is done differently in different contexts

- Different kinds of sites are tested differently.

- 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

- 0.01 failures per KLOC for NASA Shuttle code!

- Also different industries impose different testing standards

(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.

- If we build a system and, in doing so, find and fix defects ....

- It doesn’t make it a good system

- Even after defects have been resolved it may still be unusable and/or does not fulfil the users’ needs and expectations

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

(1) Quality Assurance

- Activities which ensure the implementation of processes, procedures and standards in context to verification of developed software and intended requirements.

- Focuses on processes and procedures rather than conducting actual testing on the system

- Process oriented activities.

- Preventive activities.

- It is a subset of Software Test Life Cycle (STLC).

Harsh Kacha

(2) Quality Control

- Activities which ensure the verification of developed software with respect to documented (or not in some cases) requirements.

- Focuses on actual testing by executing Software with intend to identify bug/defect through implementation of procedures and process.

- Product oriented activities.

- It is a corrective process.

- QC can be considered as the subset of Quality Assurance.

(3) Testing

- Activities which ensure the identification of bugs/error/defects in the Software.

- Focuses on actual testing

- Product oriented activities.

- It is a preventive process.

- Testing is the subset of Quality Control.

(24) Difference between Smoke and Sanity?

- Smoke Testing -Sanity Testing

-Smoke Testing is performed to ascertain - Sanity Testing is done to check the that the critical
functionalities of the new functionality / bugs have been fixed program is working fine

- The objective of this testing is to verify - The objective of the testing is to verify the "stability" of
the system in order to the "rationality" of the system in order proceed with
more rigorous testing to proceed with more rigorous testing

- This testing is performed by the developers - Sanity testing is usually performed by testers

or testers

-Smoke testing is usually documented -Sanity testing is usually not documented and

or scripted is unscripted

-Smoke testing is a subset of Regression testing -Sanity testing is a subset of Acceptance testing

-Smoke testing exercises the entire system - Sanity testing exercises only the
from end to end particular component of the entire system

-Smoke testing is like General Health Check Up - Sanity Testing is like specialized health check up

(25) Difference between verification and Validation

- verification

(1) 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.

(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.

(3) Are we building the product right?

(4) Plans, Requirement Specs, Design Specs, Code, Test Cases

(5) Reviews · Walkthroughs · Inspections

- Validation

(1) 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 actually meets the user's needs, and that the specifications were correct in the first place. In other words, to demonstrate that the product fulfills its intended use when placed in its intended environment.

(3) Are we building the right product?

(4) The actual product/software.

(5) Testing

(26) Explain types of Performance testing.

- Software performance testing is a means of quality assurance (QA). It involves testing software applications to ensure they will perform well under their expected workload.

- Features and Functionality supported by a software system is not the only concern. A software application's performance like its response time, do matter. The goal of performance testing is not to find bugs but to eliminate performance bottlenecks

- 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

- Types of Performance Testing

- (1) Load testing - Its 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.

- Load testing is a 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.

- (2) Stress testing - Stress Testing is done in order to check when the application fails by reducing the system resources such as RAM, HDD etc. and keeping the number of users as constant.

- 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.

- (3) Endurance testing -Endurance Testing is a type of Software Testing that is performed to observe whether an application can resist the processing load it is expected to have to endure for a long period. During endurance testing, memory consumption is considered to determine potential failures.

- (4)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 behavior of a software application when it receives extreme variations in traffic.

- (5)Volume testing – Volume testing is a type of non-functional testing that refers to testing the data load capabilities of a product. For instance, if we expect certain database growth, we may want to artificially grow the database to that size and test the performance of the application when using it.

- (6)Scalability testing - Scalability testing is non-functional testing that evaluates your software, application, or website performance under load. This can include load testing, performance testing, smoke testing, etc. Features like your software's response time, throughput, memory usage, and request limits are put to the test.

(27) What is Error, Defect, Bug and failure?

- Error :- a human action that produces an incorrect result. A discrepancy between a computed, observed, or measured value or condition and the true, specified, or theoretically correct value or condition. This can be a misunderstanding of the internal state of the software, an oversight in terms of memory management, confusion about the proper way to calculate a value, etc.

- Defect :- A flaw in a component or system that can cause the component or system to fail to perform its required function. Commonly refers to several troubles with the software products, with its external behavior or with its internal features.

- Failure:- Deviation of the component or system from its expected delivery, service or result. An incorrect step, process, or data definition in a computer program which causes the program to perform in an unintended or unanticipated manner. See: bug, defect, error, exception.

- 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 Severity and Priority

SL.NO	Severity	Priority
1.	Defined by the impact of a specific problem on any application’s functionality.	Defined by the impact on business.
2.	Category decided by testers.	Category decided by developers or product owners.
3.	Deals with the technical aspects of the application.	Deals with the timeframe or order to fix the defects.
4.	The value does not change with time, it’s fixed.	The priority value is subjective and may change after comparing with other defects.

(29) What is Bug Life Cycle?

- New: When a new defect is logged and posted for the first time. It is assigned a status as NEW.

- Assigned: Once the bug is posted by the tester, the lead of the tester approves the bug and assigns the bug to the developer team

- Open: The developer starts analyzing and works on the defect fix

- Fixed: When a developer makes a necessary code change and verifies the change, he or she can make bug status as “Fixed.”

- Pending retest: Once the defect is fixed the developer gives a particular code for retesting the code to the tester. Since the software testing remains pending from the testers end, the status assigned is “pending retest.”

- Retest: Tester does the retesting of the code at this stage to check whether the defect is fixed by the developer or not and changes the status to “Re-test.”

- Verified: The tester re-tests the bug after it got fixed by the developer. If there is no bug detected in the software, then the bug is fixed and the status assigned is “verified.”

- Reopen: If the bug persists even after the developer has fixed the bug, the tester changes the status to “reopened”. Once again the bug goes through the life cycle.

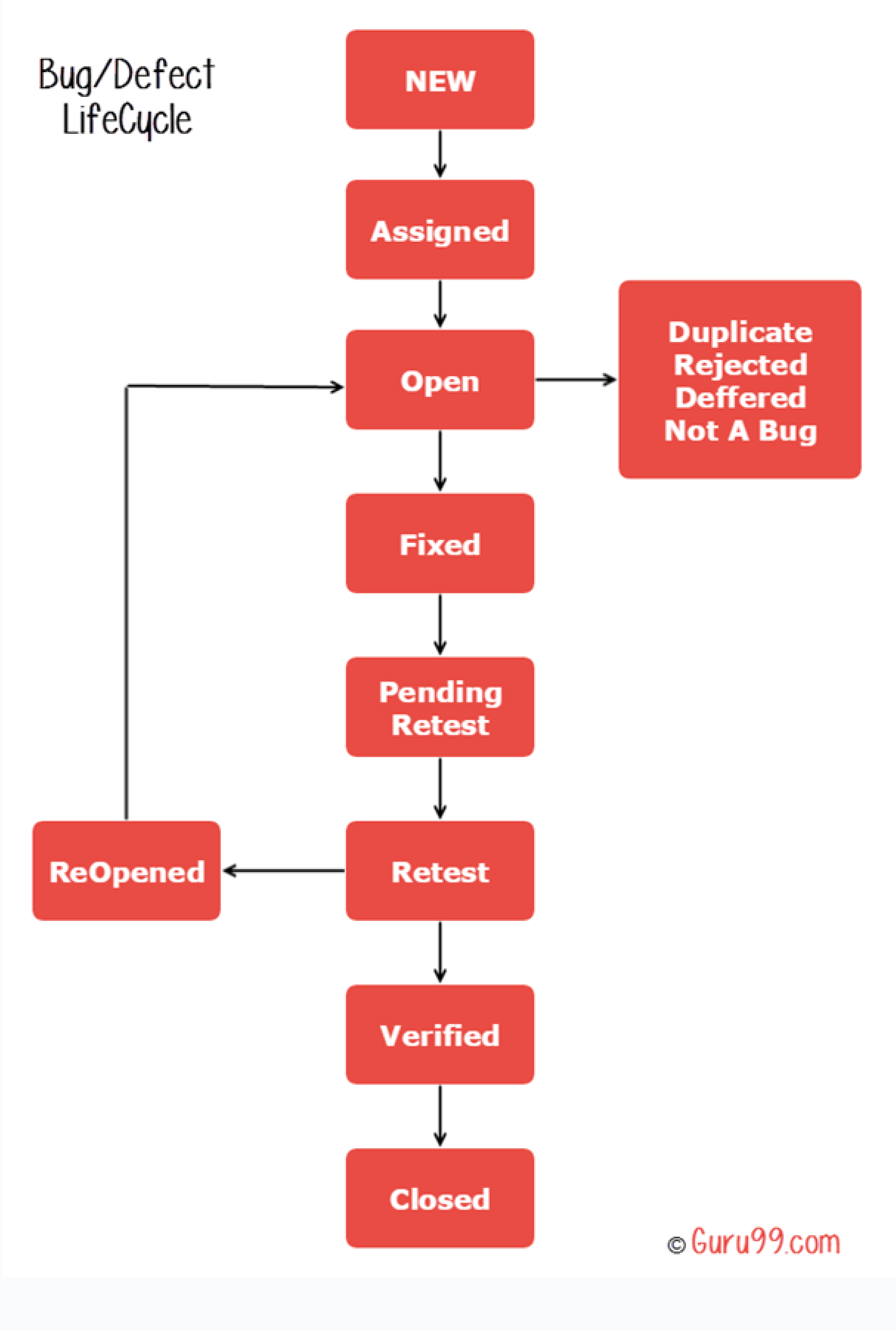
- Closed: If the bug is no longer exists then tester assigns the status “Closed.”

- Duplicate: If the defect is repeated twice or the defect corresponds to the same concept of the bug, the status is changed to “duplicate.”

- Rejected: If the developer feels the defect is not a genuine defect then it changes the defect to “rejected.”

- Deferred: If the present bug is not of a prime priority and if it is expected to get fixed in the next release, then status “Deferred” is assigned to such bugs

- Not a bug:If it does not affect the functionality of the application then the status assigned to a bug is “Not a bug”.


Bug/Defect LifeCycle
NEW
Assigned
Open
Duplicate
Rejected
Deffered
Not A Bug
Fixed
Pending Retest
Retest
ReOpened
Verified
Closed
© Guru99.com

Harsh Kacha

(30) Explain the difference between Functional testing and NonFunctional testing

(1) Functional Testing

- Functional testing is executed first
- Functional testing is performed using functional specification provided by the client and verifies the system against the functional requirements
- Manual testing or automation tools can be used for functional testing
- Business requirements are the inputs to functional testing
- Functional testing describes what the product does
- Easy 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

(2) Non-Functional Testing

- Non functional testing should be performed after functional testing

- Non-Functional testingchecksthe Performance, reliability, scalability and other non-functional aspects of the software system.

- Using tools will be effective for this testing

- Performance parameters like speed , scalability are inputs to non-functional testing.

- Nonfunctional testing describes how good the product works

- Tough to do manual testing

- Types of Nonfunctional testing are

- Performance Testing

- Load Testing

- Volume Testing ·

- Stress Testing

- Security Testing

- Installation Testing

- Penetration Testing

- Compatibility Testing

- Migration Testing

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

- SDLC (Software Development Life Cycle)

- SDLC is a structure imposed on the development of a software product that defines the process for planning, implementation, testing, documentation, deployment, and ongoing maintenance and support. There are a number of different development models.

- The methodology within the SDLC process can vary across industries and organizations, but standards such as ISO/IEC 12207 represent processes that establish a lifecycle for software, and provide a mode for the development, acquisition, and configuration of software systems.

* SDLC Phases

1.Requirements Collection/Gathering - Establish Customer Needs

2.Analysis - Model And Specify the requirements- “What”

3.Design - Model And Specify a Solution – “Why”

4.Implementation - Construct a Solution In Software

5.Testing - Validate the solution against the requirements Maintenance Repair defects and adapt the solution to the new requirements

6.Maintenance - Repair defects and adapt the solution to the new requirements

STLC (Software Testing Life Cycle)

* STLC Phases

1. Requirement Analysis - also known as Requirement Analysis in which test team studies the requirements from a testing point of view to identify testable requirements and the QA team may interact with various stakeholders to understand requirements in detail

2. Test Planning - STLC is a phase in which a Senior QA manager determines the test plan strategy along with efforts and cost estimates for the project

3. Test case development - The Test Case Development Phase involves the creation, verification and rework of test cases & test scripts after the test plan is ready.

4. Test Environment setup - Test Environment Setup decides the software and hardware conditions under which a work product is tested.

5. Test Execution - Test Execution Phase is carried out by the testers in which testing of the software build is done based on test plans and test cases prepared.

6. Test Cycle closure - Test Cycle Closure phase is completion of test execution which involves several activities like test completion reporting, collection of test completion matrices and test results.

(32) What is the difference between test scenarios, test cases, and test script?

(1) Test Scenarios

- A Scenario is any functionality that can be tested. It is also called Test Condition, or Test Possibility.

- Test Scenario is ‘What to be tested’

- Test scenario is nothing but test procedure.

- Test Scenario represents a series of actions that are associated together.

(2) Test Case

- Test cases involve the set of steps, conditions and inputs which can be used while performing the testing tasks.

- Test Case is ‘How to be tested’

- Test case consist of set of input values, execution precondition, expected Results and executed post-condition developed to cover certain test Condition.

- Test cases are derived (or written) from test scenario.

- Test Case represents a single (low level) action by the user.

- Test cases are set of input and output given to the System

(3) test script

- A set of sequential instruction that detail how to execute a core business function

- One script is written to explain how to simulate each business scenario

- Written to a level of detail for which someone else (other than the script writer) would be able to easily execute

- Identifies the test condition that is being satisfied for each step, if applicable

- Should demonstrate how the system can support the HCA warehouse business processes

- 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.

(33) 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

- Determining the scope and risks, and identifying the objectives of testing.

- Defining the overall approach of testing (the test strategy), including the definition of the test levels and entry and exit criteria.

- Integrating and coordinating the testing activities into the software life cycle activities: acquisition, supply, development, operation and maintenance.

- Defining the amount, level of detail, structure and templates for the test documentation.

- Test Policy - Defines how the organisation will conduct testing

- Master Test Plan/Test Strategy - Defines how the project will conduct testing

- Functional Test Plan - Defines how each level of testing will be conducted

- System Integ Test Plan - Defines how each level of testing will be conducted

- UAT Test Plan - Defines how each level of testing will be conducted

(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 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.

- Priority can be of following types:

- Low: The defect is an irritant which should be repaired, but repair can be deferred until after more serious defect has been fixed.

- Medium: The defect should be resolved in the normal course of development activities. It can wait until a new build or version is created.

- High: The defect must be resolved as soon as possible because the defect is affecting the application or the product severely. The system cannot be used until the repair has been done.

- Critical: Extremely urgent, resolve immediately

(35) 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.

- For example: If an application or 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 is severe. So the severity is high but priority is low.

- Severity can be of following types:

- Critical: The defect that results in the termination of the complete system or one or more component of the system and causes extensive corruption of the data. The failed function is unusable and there is no acceptable alternative method to achieve the required results then the severity will be stated as critical.

- Major (High): The defect that results in the termination of the complete system or one or more component of the system and causes extensive corruption of the data. The failed function is unusable but there exists an acceptable alternative method to achieve the required results then the severity will be stated as major.

- Moderate (Medium): The defect that does not result in the termination, but causes the system to produce incorrect, incomplete or inconsistent results then the severity will be stated as moderate.

- Minor (Low): The defect that does not result in the termination and does not damage the usability of the system and the desired results can be easily obtained by working around the defects then the severity is stated as minor.

(36) Bug categories are… /Severity/Priority

- (1)Database bug

- (2)GUI bug

- (3)Functionality bug

- (4)Security bug

(37) Advantage of Bugzila

- This open bug-tracker enables users to stay connected with their clients or employees, to communicate about problems effectively throughout the datamanagement chain

- Key features of Bugzilla includes

- Advanced search capabilities

- E-mail Notifications

- Modify/file Bugs by e-mail

- Time tracking

- Strong security

- Customization

- Localization

(38) Difference between priority and severity

Severity	**Priority**
Defect Severity is specified as the degree of impact that a defect has on the operation of the product.	Defect Priority has specified the order in which the developer should fix a defect.
Severity means the seriousness of the defect in the product functionality.	Priority means how soon the bug should be fixed.
The test engineer determines the severity level of the defect.	Priority of defects is decided in discussion with the manager/client.
It is driven by functionality.	It is driven by business value.
Severity status is established on the technical aspect of the product.	Priority status is established on customer requirements.

(39) What are the different Methodologies in Agile Development Model?

- The Agile methodology is a way to manage a project by breaking it up into several phases. It involves constant collaboration with stakeholders and continuous improvement at every stage. Once the work begins, teams cycle through a process of planning, executing, and evaluating.

- Agile is a philosophy, i.e., a set of values and principles to make a decision for developing software.

- Agile is based on the iterative-incremental model. In an incremental model, we create the system in increments, where each increment is developed and tested individually.

(1) 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. Scrum believes in empowering the development team and advocates working in small teams (say- 7 to 9 members).

(2) Kanban

- Kanban is a very popular framework for development in the agile software development methodology.

- It provides a transparent way of visualizing the tasks and work capacity of a team.

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

- Authorization : The authorization process verifies whether you have permission to access the data you want from the server. Not all data is available for everyone inside a company, so there lies the solution as Authorization. With the authorization, the server first checks whether the data you are asking for can be shown to you. If it can be, you get the desired response.

- Authentication: Accepting an invalid username/password, Authentication is the process of identifying someone's identity by assuring that the person is the same as what he is claiming for. Authentication does not ensure what tasks under a process one person can do, what files he can view, read, or update. It mostly identifies who the person or system is actually.

- Authentication and authorization are two vital information security processes that administrators use to protect systems and information. Authentication verifies the identity of a user or service, and authorization determines their access rights. Although the two terms sound alike, they play separate but equally essential roles in securing applications and data. Understanding the difference is crucial. Combined, they determine the security of a system. You cannot have a secure solution unless you have configured both authentication and authorization correctly.

(41) Write a scenario of only Whatsapp chat messages

positive:

- Check whether the user can create a new one or not.

- Check the user can add multiple contacts from the contact list.

- Verify the user can insert the group name and select an image for DP.

- Check the user can add and remove contacts from the group.

- Check the user can delete a group.

- Check the user can send and receive text messages in the group.

- Check the user can send and receive documents in the group chat box.

- Check the user can send and receive photos in the group chat box.

- Check the user can send and receive videos in the group chat box.

- Check the user can send and receive audio in the group chat box.

- The user can send and receive emotions icons in the group chat box.

- Check that the user can send and receive contacts in the group chat box.

- Check the user can send and receive Location in the group chat box.

- Check the user can send and receive GIFs in the group chat box.

- Check the user can send and receive Stickers in the group chat box.

-Users can delete text, video, audio, locations, and documents in the group chat box.

- Check the user can send recorded voice mail in the group chat box.

- Check the user can make multiple video calls in the group chat box.

- Verify the user can see the group contact information from Group Info in the group chat box.

- Check the user can share images, links, and documents from Group Media in the group chat box.

-The user can search specific chat history using the search option in the group chat box.

- Check the user can mute the group in the group chat box.

- Check the users have options like Report, Block, Clear Chat, Export Chat, and Add Shortcut.

Negative:

- Try to send more than 10 images at a time. (as it allow only 10 images at a time)
- Try to send video having size > than the limited size.
- Copy paste thousands of msgs at a time and observe the behavior.
- send multiple message to multiple users and groups.
- Try to send all emoticals at a time in one msg

(42) Write a Scenario of Pen

positive:

- Verify the type of pen, whether it is a ballpoint pen, ink pen, or gel pen.
- Verify that the user is able to write clearly over different types of papers.
- Check the weight of the pen. It should be as per the specifications. In case not mentioned in the specifications, the weight should not be too heavy to impact its smooth operation.
- Verify if the pen is with a cap or without a cap.
- Verify the color of the ink on the pen.
- Check the odor of the pen’s ink on writing over a surface.
- Verify the surfaces over which the pen is able to write smoothly apart from paper e.g. cardboard, rubber surface, etc.
- Verify that the text written by the pen should have consistent ink flow without leaving any blob.

- Check that the pen's ink should not leak in case it is tilted upside down.
- Verify if the pen's ink should not leak at higher altitudes.
- Verify if the text written by the pen is erasable or not.
- Check the functioning of the pen by applying normal pressure during writing.
- verify the strength of the pen's outer body. It should not be easily breakable.
- Verify that text written by pen should not get faded before a certain time as mentioned in the specification.
- Check if the text written by the pen is waterproof or not.
- Verify that the user is able to write normally by tilting the pen at a certain angle instead of keeping it straight while writing.
- Check the grip of the pen, and whether it provides adequate friction for the user to comfortably grip the pen.
- Verify if the pen can support multiple refills or not.
- In the case of an ink pen, verify that the user is able to refill the pen with all the supported ink types.
- For ink pens, verify that the mechanism to refill the pen is easy to operate.
- In the case of a ballpoint pen, verify the size of the tip.
- In the case of a ball and gel pen, verify that the user can change the refill of the pen easily.

Negative:

- Verify the functioning of a pen at extreme temperatures – much higher and lower than room temperature.
- Verify the functioning of a pen at extreme altitude.
- Check the functioning of a pen at zero gravity.
- Verify the functioning of the pen by applying extreme pressure.
- Verify the effect of oil and other liquids on the text written with a pen.
- Check if the user is able to write with a pen when used against gravity i.e. upside down.
- Verify the functioning of a pen when a user tries to write on unsupported surfaces like glass, plastic, wood, etc.
- Verify if the pen works normally or not when used after immersing in water or any other liquid for some period of time.

(43) Write a Scenario of Pen Stand

Positive:

- verify pen stand is resusable
- verify the different material of pen stand
- verify that for different types of pen
- verify the usability of pen stand in any weather
- verify the transportability pen of pen stand
- verify the it can stand on any type of surface
- verify that pen stand have different compartment
- verify that it can contain other things small size dairy.

Negative:

-verify the pen stand height and weight there are proper

- verify the pen stand color are proper

- verify the pen stand size are proper

- verify the pen stand company name are proper

-verify the pen stand usability any weather are proper

(44) Write a Scenario of Door

Positive:
-Verify if the door is single door or bi-folded door.
-Verify that the dimension of the doors are as per the specifications.

- Verify that the material used in the door body and its parts is as per the specifications.

-Verify that color of the door is as specified.

- Verify if the door is sliding door or rotating door.

- Check the position, quality and strength of hinges.

- Check the type of locks in the door.

- Check the number of locks in the door interior side or exterior side.

- Verify if the door is having peek-hole or not.

- Verify if the door is having stopper or not.

- Verify if the door closes automatically or not – spring mechanism.

- Verify if the door makes noise when opened or closed.

- Check the door condition when used extensively with water.

- Check the door condition in different climatic conditions- temperature, humidity etc.

Negative:
- Verify if the door is single door or bi-folded door
- Check if the door opens inwards or outwards
- Verify that the dimension of the doors is as per the specifications
- Verify that the material used in the door body and its parts is as per the specifications
- Verify that color of the door is as specified
- Verify if the door is sliding door or rotating door
- Check the position, quality and strength of hinges

(45)Write a Scenario of ATM

Positive:

- Verify the type of ATM machine, if it has a touch screen, both keypad buttons only, or both.

Harsh Kacha

- Verify that on properly inserting a valid card different banking options appear on the screen.

- Check that no option to continue and enter credentials is displayed to the user when the card is inserted incorrectly.

- Verify that the touch of the ATM screen is smooth and operational.

- Verify that the user is presented with the option to choose a language for further operations.

- Check that the user is asked to enter a pin number before displaying any card/bank account detail.

- Verify that there is a limited number of attempts up to which the user is allowed to enter the pin code.

- Verify that if the total number of incorrect pin attempts gets surpassed then the user is not allowed to continue further. And operations like temporary blocking of the card, etc get initiated.

- Check that the pin is displayed in masked form when entered.

- Verify that the user is presented with different account type options like- saving, current, etc.

- Verify that the user is allowed to get account details like available balance.

- Check that the correct amount of money gets withdrawn as entered by the user for cash withdrawal.

- Verify that the user is only allowed to enter the amount in multiple denominations as per the specifications.

- Verify that the user is prompted to enter the amount again in case the amount entered is less than the minimum amount configured.

- Check that the user cannot withdraw more amount than the total available balance and a proper message should be displayed.

- Verify that the user is provided the option to get the transaction details in printed form.

Negative:

- Verify that all the labels and controls including text boxes, buttons, images, and links are present on the screen.

- Check the informative text written displayed on the screen is clearly visible and legible.

- Verify that the size, color, and UI of the different objects are as per the specifications.

- Verify that the application's UI is responsive i.e. it should adjust to different screen resolutions of ATM machines.

(46) When to used Usablity Testing?

- Usability Testing also known as User Experience (UX) Testing, is a testing method for measuring how easy and user-friendly a software application is. A small set of target end-users, use software application to expose usability defects. Usability testing mainly focuses on user's ease of using application, flexibility of application to handle controls and ability of application to meet its objectives.
This testing is recommended during the initial design phase of SDLC, which gives more visibility on the expectations of the users.

(47) What is the procedure for 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 types of bars – tool bar, menu bar, dialog boxes and windows etc.

- 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

(48) Write a scenario of Microwave Owen

Positive:

- Verify that the dimensions of the oven are as per the specification - provided.
- Verify that the oven's material is optimal for its use as an oven and as per the specification.
- Verify that the oven heats the food at the desired temperature properly.
- Verify that the oven heats food at the desired temperature within a specified time duration.
- Verify the ovens functioning with the maximum attainable temperature.
- Verify the ovens functioning with minimum attainable temperature.
- Verify that the oven's plate rotation speed is optimal and not too high to spill the food kept over it.
- Verify that the oven's door gets closed properly.
- Verify that the oven's door opens smoothly.
- Verify the battery requirement of the microwave oven and check that it function's smoothly at that power.
- Verify that the text written over the oven's body is clearly readable.
- Verify that the digital display is clearly visible and functions correctly.
- Verify that the temperature regulator is smooth to operate.

Negative:

- Verify that the structure and dimensions of the microwave and see if it confirms to the specified dimensions mentioned in the user manual.
- Verify that Test if the door of the oven fits squarely and securely and opens and closes smoothly.
- Verify that the door hinges are in proper condition and the surface of the door is not damaged.
- Verify that there is no corrosion is evident on the door, the door hinges or the oven interior.
- Verify that power terminal and see if the LED glows. And test if the oven is actually getting power supply. Also test the current received and maximum allowed.
- Verify that Test if the microwaves generated in the microwave oven cease to exist once the electrical power

(49) Write a scenario of Coffee vending Machine

Positive:

- Verify that the dimension of the coffee machine is as per the specification.
- Verify that outer body, as well as inner part’s material, is as per the specification.
- Verify that the machine’s body color as well brand is correctly visible and as per specification.
- Verify the input mechanism for coffee ingredients-milk, water, coffee beans/powder, etc.
- Verify that the quantity of hot water, milk, coffee powder per serving is correct.
- Verify the power/voltage requirements of the machine.
- Verify the effect of suddenly switching off the machine or cutting the power. The machine should stop in that situation and in power resumption, the remaining coffee should not get come out of the nozzle.
- Verify that coffee should not leak when not in operation.
- Verify the amount of coffee served in single-serving is as per specification.
- Verify that the digital display displays correct information.

Negative:

- verify whether user is able to get coffee even after making cancel
- To verify whether user is able to get coffee even after there is no powder in machine
- To verify whether user is able to get coffee even after not making right choice
- To verify whether user is able to get coffee even there is no water in machine
- To verify whether user is unable to get coffee even after making a right choice

(50) Write a scenario of chair

- Verify that the chair is stable enough to take an average human load.
- Check the material used in making the chair-wood, plastic etc.

- Check if the chair's leg are level to the floor.
- Check the usability of the chair as an office chair, normal household chair.
- Check if there is back support in the chair.
- Check if there is support for hands in the chair.
- Verify the paint's type and color.
- Verify if the chair's material is brittle or not.
- Check if cushion is provided with chair or not.
- Check the condition when washed with water or effect of water on chair.

Negative:

- To verify the balance of the chair with one arm.
- To verify the balance of the chair with three legs.
- To verify Chair stress testing by dropping the Chair down from the practical height.
- To verify that nothing is breaking, no damage to the Chair and Chair is performed without any issues.
- To verify that how Chair is working at different climate environmental conditions.

(51) Write a Scenario of Gmail (receiving mail)

Positive:

- Check that the recently received unread email is highlighted and bold in the Inbox section.
- Check if all the elements of the received email are correctly displayed or not.
- Check whether the user clicks on the new email; it redirects the user to the email content.
- Check if the email content is displaying in the proper format or not.
- Check the attached documents of the email are downloadable.
- Check the already-read emails should not be the highlight.
- The number of unread email counts should be displayed beside the inbox text box.
- Check if the count is increased as per the number of new emails you are received.
- Check the count is increased when you mark an email as unread.
- Check after opening or make as read an email. The count should be decreased.
- Check the names are visible to all the users whose names are present in CC & To section.
- Check those names or emails are present in the BCC section and should not display to others.
- Check that Gmail open properly
- Check that it shows proper mail home page

Negative:

- Check the receiving mail functionality without internet
- Check the movement functionality of mail on same page
- Check that it shows option of search data from sender
- Check the receiving mail notification are received the pc or phone

- Check the gmail it mail are show proper in the page
- Check mail all functionality are proper
- Check mail all messgae are proper sent

(52) Online shopping to buy product (flipkart)

Positive:

- Check that open website propely
- Check that product is available on flipkart
- Check that it shows proper page of product
- Check that it shows price of product properly
- Check the available offers option
- Check the terms & condition option of offers
- Check that shows proper produt details
- Check that it shows product specifications properly
- Check the product image option is working properly
- Check buy now option of product is working properly
- Check that it shows pincode for delivery option of product
- Check that on the product page, and a user can select the desired attribute of the product, e.g., size, color, etc.
- Check that user can add to the cart one or more products.
- Check that user can add products to the wish list.
- Check that users can buy products added to the cart after signing in to the application (or as per the website's functionality).

Negative:

- Check the buy products functionality without internet
- Check that product price is available in different currecy
- Check that all offers can apply at a same time
- Check that it can verify multi pincode at a same time
- Check the availability of delivery out of country option
- Check that proper send complete order

(53) Write a Scenario of Wrist Watch

Positive:

- Verify the type of watch – analog or digital.
- In the case of an analog watch, check the correctness time displayed by the second, minute, and hour hand of the watch.
- In the case of a digital watch, check the digital display for hours, minutes, and seconds is correctly displayed.
- Verify the material of the watch and its strap.
- Check if the shape of the dial is as per specification.
- Verify the dimension of the watch is as per the specification.
- Verify the weight of the watch.

- Check if the watch is waterproof or not.
- Verify that the numbers in the dial are clearly visible or not.
- Check if the watch is having a date and day display or not.
- Verify the color of the text displayed in the watch – time, day, date, and other information.
- Verify that clock's time can be corrected using the key in case of an analog clock and buttons in case of a digital clock.

Negative:

- To verify that Watch is working without power or power cell or not.
- To verify that the Watch stress testing by dropping the Watch down from the desired height.
- To verify that the Watch is working at different climate environmental conditions.
- To verify that watch can use GPS when no Sim or internet connection.
- verify that watch hours,minutes and seconds are working properly or not

(54) Write a Scenario of Lift(Elevator)

Positive:

- Verify the dimensions of the lift.

- Verify the type of door of the lift is as per the specification.

- Verify the type of metal used in the lift interior and exterior.

- Verify the capacity of the lift in terms of the total weight.

- Verify the buttons in the lift to close and open the door and numbers as per the number of floors.

- Verify that the lift moves to the particular floor as the button of the floor is clicked.

- Verify that the lift stops when the up/down buttons on a particular floor are pressed.

- Verify if there is an emergency button to contact officials in case of any mishap.

- Verify the performance of the floor – the time taken to go to a floor.

- Verify that in case of power failure, the lift doesn't free-fall and gets halted on the particular floor.

Negative:

- Verify if the lift goes to the specific floor upon pressing the floor button.
- Verify that the lift stops when a specific floor's up/down buttons are clicked.
- Verify if there is an emergency button to contact associated person in case of any emergency
-Verify the performance of the lift – the time it takes to reach a floor.
-Verify that the lift does not free-fall and gets halted on the specific floor in case of electric failure.
-Verify that the lift does not work if the user presses the button to open the door before reaching the specified floor.

(55)Write a Scenario of whatsapp Group (generate group)

Positive:

Check whether the user can create a new one or not.
Check the user can add multiple contacts from the contact list.
Verify the user can insert the group name and select an image for DP.
Check the user can add and remove contacts from the group.
Check that whatsapp setting and generate a new group
Check that it shows new group option
Verify the available of contact list
Check that it shows search contact option
Check that choose and add contact in new group
Check its multi selection functionality of member
Check functionality like set group name, group icon
Verify the disappearing message functionality
Check new group generate functionality without internet
Verify the maximum member add limit in new group
Verify functionality of adding unsaved member in new group
Verify that delete all data in group admin

Negative:

- Check how many users are active

- Check how many users can log into the group chat.

- Check how many users can be limited to the channel or groups.

- Check all the members can view the message.

- Check if all the messages are accessible by the moderator or admin.

- Check if the channel can connect with the available integration apps.

- Check if the user can leave the chat at any time.

- Check if the user can mute the conversation.

- Check if the valid username and avatar are visible to the user.

(56) Write a Scenario of Whatsapp payment

Positive:

- verify that user is able to see Scan code screen on mobile phone or not
- verify that QR code is scan from whatsapp pay

Harsh Kacha

- verify that user can get message for payment
- Check that it shows payment option
- Verify that it shows add bank account functionality
- Check the availability of bank list
- Check the search bank functionality in payment option
- Verify that we can send money to any valid UPI id
- Check that payment is securely done through UPI pin
- Check payment functionality without internet
- Verify the maximum transferrable limit of payment
- Check the multi payment functionality at a time
- Verify the availability of payment option in different currency

Negative:

-Verify the availablity of payment option show perfectly or not
- Check the availability of bank list
- Verify that it shows add bank account functionality or not
- verify that QR code is scan from whatsapp pay

- Verify that we can send money to any valid UPI id

- Verify the maximum transferrable limit of payment or not

- Check that payment is securely done through UPI pin
- Check that all whatsapp pay all funcation are proper show