**ASSIGNMENT**

**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. A brief description of how tests will be performed Expected Problems Is carried out in time boxed intervals

**WHAT IS TRACEABILITY MATIX?**

* A software process should help you keeping the virtual table up-to-date.
* To protect against changes, you should be able to trace back from every system component to the original requirement that caused its presence.
* Simple technique may be quite valuable (naming convention.

• **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.

**• What is Equivalence partitioning testing?**

The numbers fall into a partition where each would have the same, or equivalent, result i.e. an Equivalence Partition (EP) or Equivalence Class.

The Valid partition is bounded by the values 1 and 100.

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)

• **What is Integration testing?**

During the process of manufacturing a ballpoint pen, the cap, the body, the tail and clip, the ink cartridge and the ballpoint are produced separately and unit tested separately. When two or more units are ready, they are assembled and Integration Testing is performed.

* **There is two types methods of Integration Testing:**
* Bing Bang Integration Testing
* Incremental Integration Testing

1. Top Down Approach
2. Bottom Up Approach

**• What determines the level of risk?**

Determining the level of risk usually involves trying to assess not only the likelihood of an identified risk from actually occurring but also the potential magnitude the consequences this risk could have on an organisation and is stakeholder should it occur.

• **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.

• **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.

It is only a kind of Black Box Testing.

• **What is component testing?**

Component Testing – The testing of individual software components.

Unit testing is the first level of testing and is performed prior to Integration Testing

Unit tests find problems early in the development cycle.

Unit testing is performed by using the White Box Testing method.

• **What is functional system testing?**

Functional Testing: Testing based on an analysis of the specification of the functionality of a component or system.

Functional testing verifies that each function of the software application operates in conformance with the requirement specification.

This testing involves checking of User Interface, APIs, Database, security, client/ server applications and functionality of the Application under Test. The testing can be done either manually or using automation.

• 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.

Non-functional testing includes, but is not limited to, performance testing, load testing, stress testing, usability testing, maintainability testing, reliability testing and portability 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

• **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 is Adhoc testing?**

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

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.

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.

Use Error Guessing to Complement Test Design Techniques.

There are different types of Adhoc testing:

1.Buddy Testing

2.Pair Testing

3.Monkey Testing

• **What is 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.

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.

⚫ Pros:

Pros: Performance bottlenecks identification before production

Improves the scalability of the system

Minimize risk related to system down time

Reduced costs of failure

Increase customer satisfaction

⚫ Cons:

Need programming knowledge to use load testing tools.

Tools can be expensive as pricing depends on the number of virtual users supported.

• **What is stress Testing?**

System is stressed beyond its specifications to check how and when it fails. Performed under heavy load like putting large number beyond storage capacity, complex database queries, continuous input to system or database load.

Stress testing is used to test the stability & reliability of the system. This test mainly determines the system on its robustness and error handling under extremely heavy load conditions. It even tests beyond the normal operating point and evaluates how the system works under those extreme conditions.

Stress Testing is done to make sure that the system would not crash under crunch situations.

Stress testing is also known as endurance testing.

Most prominent use of stress testing is to determine the limit, at which the system or software or hardware breaks.

Types of Stress Testing

⚫ Application Stress Testing

⚫ Transactional Stress Testing

⚫ Systemic Stress Testing

⚫ Exploratory Stress Testing

• **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.

Testing based upon the structure of the code

Typically undertaken at Component and Component Integration Test phases by development teams

White box testing is the detailed investigation of internal logic and structure of the code.

• **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.

What a system does, rather than HOW it does it

Typically used at System Test phase, although can be useful throughout the test lifecycle

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.

⚫ **Advantages**

Well suited and efficient for large code segments.

Code Access not required.

Clearly separates user's perspective from the developer's perspective through visibly defined roles.

Large numbers of moderately skilled testers can test the application with no knowledge of implementation, programming language or operating systems.

⚫ **Disadvantage**

Limited Coverage since only a selected number of test scenarios are actually performed.

Inefficient testing, due to the fact that the tester only has limited knowledge about an application.

Blind Coverage, since the tester cannot target specific code segments or error prone areas.

The test cases are difficult to design.

Techniques of Black Box Testing

**There are four specification-based or black-box technique:**

* Equivalence partitioning
* Boundary value analysis
* Decision tables
* State transition testing Use-case Testing
* Other Black Box Testing
* Syntax or Pattern Testing

• **Mention what are the categories of defects?**

Defect is the variance from a desired product attribute (it can be a wrong, missing or extra data).

**It can be of two types**

Defect from the product or a variance from customer/user expectations.

It is a flaw in the software system and has no impact until it affects the user/customer and operational system.

**Types of Defect**

⚫ Data Quality/Database Defects:

Deals with improper handling of data in the database.

**Examples:**

Values not deleted/inserted into the database properly

Improper/wrong/null values inserted in place of the actual values

⚫ Critical Functionality Defects:

The occurrence of these bugs hampers the crucial functionality of the application.

Examples: - Exceptions

⚫ Functionality Defects:

These defects affect the functionality of the application.

Examples:

All JavaScript errors

Buttons like Save, Delete Cancel not performing their intended functions

A missing functionality (or) a feature not functioning the way it is intended to

Continuous execution of loops

Types of Defect

⚫ Security Defects:

Application security defects generally involve improper handling of data sent from the user to the application. These defects are the most severe and given highest priority for a fix.

Examples: Authentication: Accepting an invalid username/password

Authorization: Accessibility to pages though permission not given

⚫ User Interface Defects: As the name suggests, the bugs deal with problems related to UI are usually considered less severe.

Examples:

* Improper error/warning/UI messages
* Spelling mistakes
* Alignment problems

**• Mention what big bang testing is?**

In Big Bang integration testing all components or modules is integrated simultaneously, after which everything is tested as a whole.

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 link to be tested could be missed easily

• What is Error, Defect, 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

• **What is the purpose of exit criteria?**

⚫ 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.

• **When should "Regression Testing" be performed?**

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

• **What is 7 key principles? Explain in detail?**

**7 key principles**

**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 can show that defects are present, but cannot prove that there are no defects.

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

1. **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.

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.

1. **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

1. **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 a small number of modules

An important consideration in test prioritisation

1. **The 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

1. **Testing is Context Dependent**

Testing is basically context dependent.

Testing is done differently in different contexts

Different kinds of sites are tested differently.

**7. Absence of Errors Fallacy**

If the system built is unusable and does not fulfil the user’s needs and expectations, then finding and fixing defects does not help

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

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

• **Difference between QA v/s QC v/s Tester**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.N** | **Quality Assurance** | **Quality Control** | **Control Testing** |
| **1** | **Activities which ensure the implementation of processes, procedures and standards in context to verification of developed software and intended requirements** | **Activities which ensure the**  **verification of developed**  **software with respect to**  **documented (or not in some**  **cases) requirements.** | **Activities which ensure**  **the identification of**  **bugs/error/defects in the**  **Software.** |
| **2** | **Focuses on processes and procedures rather than conducting actual testing on the system.** | **Focuses on actual testing by**  **executing Software with intend**  **to identify bug/defect through**  **implementation of procedures**  **and process.** | **Focuses on actual testing.** |
| **3** | **Focuses on actual testing.** | **Product oriented activities.** | **Product oriented**  **activities.** |
| **4** | **Preventive activities.** | **It is a corrective process.** | **It is a preventive process.** |
| **5** | **It is a preventive process.** | **QC can be considered as the**  **subset of Quality Assurance.** | **Testing is the subset of**  **Quality Control.** |

• **Difference between verification and Validation**

|  |  |  |
| --- | --- | --- |
| **Criteria** | **Verification** | **Validation** |
| **Validation** | **The process of evaluating**  **work-products (not the actual**  **final product) of a development**  **phase to determine whether**  **they meet the specified**  **requirements for that phase.** | **The process of evaluating**  **work-products (not the actual**  **final product) of a development**  **phase to determine whether**  **they meet the specified**  **requirements for that phase.** |
| **Objective** | **To ensure that the product is**  **being built according to**  **the requirements and**  **design specifications. In other**  **words, to ensure that work**  **products meet their**  **specified requirements.** | **To ensure that the product actually meets**  **the user’s needs, and that the**  **specifications were correct in the first place.**  **In other words, to demonstrate**  **that the product fulfils its**  **intended use**  **when placed in its intended environment.** |
| **Question** | **Are we building the product right?** | **Are we building the right product?** |
| **Evaluation**  **Items** | **Plans, Requirement Specs, Design**  **pecs, Code, Test Cases** | **The actual product/software.** |
| **Activities** | **∙ Reviews**  **∙ Walkthroughs**  **∙ Inspections** | **∙ Testing** |

• **Explain types of Performance testing.**

Types of Performance Testing

⚫ Load testing

⚫ Stress testing

⚫ Endurance testing

⚫ Spike testing

⚫ Volume testing

⚫ Scalability 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 web site under a range of loads to determine at what point the system’s response time degrades or fails.

This testing usually identifies

* The maximum operating capacity of an application
* Determine whether current infrastructure is sufficient to run the application
* Sustainability of 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. Pros and Cons of Load Testing

⚫ Pros:

Performance bottlenecks identification before production

Improves the scalability of the system

Minimize risk related to system down time

Reduced costs of failure

Increase customer satisfaction

⚫ Cons:

Need programming knowledge to use load testing tools.

Tools can be expensive as pricing depends on the number of virtual

users

supported.

⚫ **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.

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.

Types of Stress Testing

⚫ Application Stress Testing:

⚫ Transactional Stress

Testing:

⚫ Systemic Stress Testing:

⚫ Exploratory Stress Testing:

**Stress Testing**

**Tools**

⚫ Stress Tester

⚫ Neo Load

⚫ App Perfect

**• What is Error, Defect, Bug and failure?**

⚫ Failure: The inability of a system or component to perform its

required functions within specified performance requirements. See:

bug, crash, exception, and fault.

⚫ Bug: A fault in a program which causes the program to perform in an

unintended or unanticipated manner. See: anomaly, defect, error,

exception, and fault. Bug is terminology of Tester.

⚫ Fault: 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.

⚫ Defect: Commonly refers to several troubles with the software

products, with its external behaviour or with its internal features.

• **Difference between Priority and Severity**

|  |  |
| --- | --- |
| **Priority** | **Severity** |
| Priority determines the defect urgency of repair | Severity determines the defects effect on the application |
| **How soon we need to fix** | **How bad the defect is** |
| Priority is given by Test lead or project  manager. | Severity is given by QA testers. |
| Levels  \*P1: Fix before next build to test  \*P2: Fix before final release  \*P3: we probably won’t get to these, but we  want to track them anyway to resolve the  priority-severity divide. | Levels  • Critical: the software will not run  • High: unexpected fatal errors (includes  crashes and data corruption)  • Medium: a feature is malfunctioning  • Low: a cosmetic issue. |

• **What is Bug Life Cycle?**

A computer bug is an error, flaw, mistake, failure, or fault in a computer program that prevents it from working correctly or produces an incorrect result. Bugs arise from mistakes and errors, made by people, in either a program’s source code or its design.

The duration or time span between the first time defects is found and the time that it is closed successfully, rejected, postponed or deferred is called as ‘Defect Life Cycle’.

**⚫ 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 analysing 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”.

**• Explain the difference between Functional testing and Non Functional testing**

|  |  |
| --- | --- |
| **Functional testing** | **Non Functional testing** |
| It is performed before non-functional testing. | It is performed after the functional testing. |
| It is easy to define functional requirements. | It is difficult to define the requirements for non-functional testing. |
| Functional testing is carried out using the functional specification. | This kind of testing is carried out by performance specifications |
| Examples of Functional Testing Types   * Unit testing * Smoke testing * User Acceptance * Integration Testing * Regression testing * Localization * Globalization * Interoperability | Examples of Non-functional Testing Types   * Performance Testing * Volume Testing * Scalability * Usability Testing * Load Testing * Stress Testing * Compliance Testing * Portability Testing * Disaster Recover Testing |

**• What is the difference between the STLC (Software Testing Life Cycle) and SDLC**

**(Software Development Life Cycle)?**

|  |  |
| --- | --- |
| **SDLC** | **STLC** |
| SDLC is mainly related to software development. | STLC is mainly related to software testing. |
| Besides development other phases like testing is also included. | It focuses only on testing the software. |
| SDLC involves total six phases or steps. | STLC involves only five phases or steps. |
| In SDLC, development team makes the plans and designs based on the requirements. | In STLC, testing team(Test Lead or Test Architect) makes the plans and designs. |
| SDLC phases are completed before the STLC phases. | STLC phases are performed after SDLC phases. |
| Creation of reusable software systems is the end result of SDLC. | A tested software system is the end result of STLC. |

**• What is the difference between test scenarios, test cases, and test script?**

|  |  |  |
| --- | --- | --- |
| **TEST SCENARIO** | **TEST CASES** | **TESE SCRIPT** |
| is any functionality that can be tested | Is a setoff actions executed to verify particular features or functionality | Is a set of instructions to test an app automatically |
| Is derived from test artefacts like Business Requirement Specification and Software Requirement Specification | is mostly derived from test scenarios | Is mostly derived from test cases |
| Is more focused on what to test | Is focused on what to test and how to test | Is focused on the expected result |
| The main task is to check the full functionality of a software application | The main task is to verify compliance with the applicable standards, guidelines, and customer requirement | The main task is to verify that nothing is skipped, and the results are true as the desired testing plan |
| Allows quickly assessing the testing scope | Allows detecting errors and defects | Allows carrying out an automatic execution of test cases. |

**• 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. Scheduling test analysis and design activities. Scheduling test implementation, execution and evaluation. All projects require a set of plans and strategies which define how the testing will be conducted.

• Bug categories are…

Security, Database, Functionality (Critical/General), UI.

• Advantage of Bugzilla

- Advanced search capabilities

E-mail Notifications

Modify/file Bugs by e-mail

Time tracking Strong security

Customization Localization

**• 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

• **Explain the difference between Authorization and Authentication in Web**

**testing.**

|  |  |
| --- | --- |
| Authorization | Authentication |
| Process of specifying access rights/privileges to resources related to information security | Process of confirming the truth of an attribute of a single piece of data claimed true by an entity |
| Checks a user’s privileges to access resources | Checks a person’s details to identify him |
| Verifies user’s permissions | Verifies user’s credentials |
| Occurs after authentications | Occurs before authorization |
| EX. he can access lecture slides and other learning material of the courses based on the permissions given to him | EX.A student can authenticate himself before accessing the Learning Management Systems of a University |

• **When to used Usablity Testing?**

Usability Testing identifies usability errors in the system early in development cycle and can save a product from failure.

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

**• 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 Colour of the font and warning messages is aesthetically pleasing

Check that the images have good clarity

Check that the images are properly aligned

• **To Create Scenario (Positive & Negative) Facebook Chat on Mobile**

1. Verify that the massager app install in mobile

2. Verify that the successfully login on messenger

3. Check the received messages count should be displayed on the Facebook Messages icon

4. Check the user gets all received messages in his inbox

5. Check that only message contact will display on the left hand side of the message box

6. Check the Active users display with a green got in the message box

7. Check the unread messages are highlighted so that the user can identify it

8. Check the user can send or received message from massager app

9. Check the user can search contact in the message box

10. Check the user can show the all contact profile pic in massager app

11. Check the user can send or received text, picture, documents, videos, audio so on

12. Check the user can call audio or video

13. Check the user can open more account in massager app

14. Check the user should get message after uploading an images or file of an unsupported type

15. Check the user should successfully logout or not

* **Gmail (Receiving mail)**
* 1. Check that the user can receive email are correctly displayed or not
* 2. Check that the recently received unread emails is highlighted and bold in the Inbox section 3. Check the user can get the notification of receiving mail
* 4. Check the user should open the receiving mail properly or not
* 5. Check the user can get the information ID who receiving mail
* 6. Check the attached documents of the email are download or not
* 7. Check the already read emails should not be the highlight
* 8. The number of unread email counts should be displayed beside the inbox test box
* 9. Check if the count is increased asper the number of new emails as unread
* 10. Check the name are visible to all the user whose names are present in CC and to section

**Online shopping to buy product (flip cart)**

1. Verify that user can search easily as per your choice

2. Verify that user get proper product displayed which they search on the flip cart

3. Verify that the user can all information about the product

4. Verify that the user can easily add or not in add to cart

5. Verify that the user can show the all product which they add in to the card with details

6. Verify that the user can easily buy the one product from cart

7. Verify that the user should logon successfully

8. Verify that the user get more option for payment

9. Verify that the user get product order details

10. Verify that user get product on right address

11. Verify that the user can see the review of the product

• **Write a scenario of only WhatsApp chat messages**

**Test Scenarios for WhatsApp Chat Messages**

1. Verify that the user can set a chat wallpaper.
2. Verify that the user sets privacy settings like turning on/off last seen, online status, read receipts, etc.
3. Verify that the user can update notification settings like – notification sound, on/off, and show preview for both group and individual chats.
4. Verify that the user can take the complete chat backup of his chats.
5. Verify that the user can update the phone number that is used by the WhatsApp application.
6. Verify that the user can disable/delete his WhatsApp account.
7. Verify that the user can check data usage by images, audio, video, and documents in WhatsApp chats.

• **Write a Scenario of Pen**

**Test Scenarios of Pen**

1.Verify the type of pen, whether it is a ballpoint pen, ink pen, or gel pen.

2.Verify that the user is able to write clearly over different types of papers.

3.Verify if the pen is with a cap or without a cap.

4.Verify the colour of the ink on the pen.

5.Verify if the pen’s ink should not leak at higher altitudes.

6.Verify if the text written by the pen is erasable or not

7.Check the functioning of the pen by applying normal pressure during writing

8.Verify the strength of the pen’s outer body. It should not be easily breakable.

9.Check if the text written by the pen is waterproof or not.

10.Verify if the pen can support multiple refills or not.

• **Write a Scenario of Door**

1.Verify if the door is single door or bi-folded door

2.Check if the door opens inwards or outwards

3.Verify that colour of the door is as specified

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

5.Check the type of locks in the door

6.Verify if the door closes automatically or not

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

• **Write a Scenario of ATM**

1.Verify the ATM machine accepts card and PIN details

2.Verify the error message by inserting a card incorrectly

3.Verify the error message by inserting an expired card

4. Verify the error message by entering an incorrect PIN

5. Verify that the user is asked to enter the PIN after inserting a valid ATM Card

6. Verify the user is allowed to do only one cash withdrawal transaction per PIN request

7. Verify the machine logs out of the user session immediately after successful withdrawal

8. Verify the message when there is no money in the ATM

9. Verify the language selection functionality

10. Verify the cash withdrawal functionality by entering some valid amount

11. Verify the cash withdrawal functionality by entering an amount greater than the total available balance in the account.

12. Verify the ATM machine successfully takes out the money.

13. Verify the ATM machine takes out the balance printout after the withdrawal

14. Verify the font of the text displayed in ATM screen

15. Verify the text on the screen buttons visible clearly.

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

17. Verify the user is allowed to choose different account types like Savings, Current etc.

18. Verify the functionality of the receipt printer

19. Verify whether the printed data is correct or not in the receipt

20. Verify how much time the system takes to log out.

• **When to used Usability Testing?**

Usability Testing identifies usability errors in the system early in

Development cycle and can save a product from failure.

⚫ Pros

* It helps uncover usability issues before the product is marketed.
* It helps improve end user satisfaction
* It makes your system highly effective and efficient
* It helps gather true feedback from your target audience who actually

use your system during usability test. You do not need to rely on “opinions” from random people.

⚫ Cons

* Cost is a major consideration in usability testing. It takes lots of resources to set up

a Usability Test Lab. Recruiting and management of usability testers can also be

expensive

* However, these costs pay themselves up in form of higher customer satisfaction,

retention and repeat business. Usability testing is therefore highly

recommended.

• **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.

**⚫ MANUAL BASED TESTING**

Under this approach, graphical screens are checked manually by testers

in conformance with the requirements stated in business requirements

document.

**⚫ 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.

**⚫ MODEL BASED TESTING**

A model is a graphical description of system’s behaviour. It helps us to

understand and predict the system behaviour. Models help in a generation of

efficient test cases using the system requirements.

• **Write a scenario of Microwave Owen**

1. Verify that the dimensions of the oven are as per the specification provided.
2. Verify that the oven’s material is optimal for its use as an oven and as per the specification.
3. Verify that the dimensions of the oven are as per the specification provided.
4. Verify that oven heats food at the desired temperature within a specified time duration.
5. Verify that oven heats food at the desired temperature within a specified time duration.
6. Verify that oven heats food at the desired temperature within a specified time duration.
7. Verify that oven heats food at the desired temperature within a specified time duration.
8. Verify that oven heats food at the desired temperature within a specified time duration.
9. Verify the battery requirement of the microwave oven and check that it function’s smoothly at that power**.**
10. Verify that the temperature regulator is smooth to operate.
11. Verify that the temperature regulator is smooth to operate.
12. Verify that the temperature regulator is smooth to operate.
13. Verify that the temperature regulator is smooth to operate.

• **Write a scenario of Coffee Vending Machine**

1. Verify that the dimension of the coffee machine is as per the specification
2. Verify that outer body, as well as inner part’s material, is as per the specification
3. Verify the input mechanism for coffee ingredients-milk, water, coffee beans/powder, etc.
4. Verify that the quantity of hot water, milk, coffee powder per serving is correct
5. Verify the power/voltage requirements of the machine
6. Verify that coffee should not leak when not in operation
7. Verify the amount of coffee served in single-serving is as per specification
8. Verify that the coffee served has the same and correct temperature each time it is served by the machine
9. Verify that the coffee served has the same and correct temperature each time it is served by the machine
10. Verify that the coffee served has the same and correct temperature each time it is served by the machine
11. Verify that the coffee served has the same and correct temperature each time it is served by the machine
12. Verify that the coffee served has the same and correct temperature each time it is served by the machine

• **Write a scenario of chair**

1. Verify that the chair is stable enough to take an average human load

2. Check the material used in making the chair-wood, plastic etc

3. Check if the chair’s leg are level to the floor

4. Check the usability of the chair as an office chair, normal household chair

5. Check there is back support in the chair

6. Check if there is back support for hands in the chair

7. Verify the paint’s type and colour

8. Verify the chair’s material is brittle or not

9. Check that cushion is provided with chair or not

10. Check that the condition when washed with water or effect of water on chair

11. Verify that the dimension of chai is as per the specification

12. Verify that the weight of the chair is as per the specifications

**• Write a Scenario of WhatsApp Group (generate group)**

1. Check whether the user can create a new one or note

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

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

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

5. Check the user is able to delete group

6. Check the user can send and receive text message in the group

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

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

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

10. Check the user can send and receive emotion icons in the group chat box

11. Check the user can send and receive contact

12. Check the user can delete text, video, audio, location and documents

13. Check the user is able to invite or add multiple video call

14. Check the users have options like Report, block, clear chat, export chat and add shortcut

15. Check the how many users get authority of add other person

• **Write a Scenario of WhatsApp payment**

1. Verify the WhatsApp open properly or not

2. Verify the payment option should be open or when we go to right side on WhatsApp and tick on the

three dots

3. Verify the show the payment page

4. Verify the security protect functionality should be work properly or not

5. Verify the add payment method functionality should open

6. Check the all bank name show into the page

7. Verify that the we can select the multiple bank in a WhatsApp payment

8. Check the verification should properly or we get the otp in verify the mobile number

9. Verify that the we can see the transaction history

10. Verify the QR code generate properly or not

11. Verify that how many time get for transaction payment

12. Verify that the internet speed is affect on payment time

• **Write a Scenario of Wrist Watch**

1. Verify that the which types of Wrist Watch

2. Verify that the model of wrist Watch

3. Verify that the company name or logo show the properly or not

4. Verify that the strip of watch is fitted properly or not

5. Check is all accessories of the Watch are fitted correctly

6. Check if the dial of the watch is fitted properly

7. Check if the watch fits properly on the wrist or not

8. Check if is the watch making any sound when it is moving

9. Check if the watch making any vibration when it’s in movement

10. Check can you change the watch’s strap with the other one

11. Check if we can watch different power cell with this watch

12. Check the colour of the watch is as per the requirement

13. Check the length, breath, and weight as per the requirement

14. Check the material used to manufacture the watch

15. Check whether the glass of the watch is transparent or not

16. Check if the watch is waterproof or not

• **Write a Scenario of Lift(Elevator)**

1. Verify that the types of the lift

2. Verify that the elevator is capable of moving up and down

3. Check the is stop at each floor or not

4. Check the it moves precisely to that floor when the corresponding floor no is pressed

5. Check it moves up when called from upward and down when the call downward

6. Check it wait until the close button is not pressed

7. Check the all button should working properly or not

8. Check the sensor are properly working or not

9. Check the capacity of the elevator lift

10. Verify that the presence of display where the floor number appears

11. Verify that the floor number is being announced on each floor

12. Verify that the light or music should work or not

13. Verify that the emergency button properly work or not

• **Write a Scenario of Instagram (video call with chat)**

1. Verify that the Instagram application is install or not

2. Verify that the camera should be available on the mobile phone

3. Verify that the Internet connection should be on both person

4. Check the voice celerity

5. Verify that the user can send the massage when video call is on

6. Verify that the picture celerity during the video call

7. Verify that the front camera and top camera available or not

8. Verify that the video call with microphone is working or not

9. Verify that the group video call is available or not

11. Verify that the user can send video during the video call