**Manual\_Assignment\_2**

**Submitted by Kuldeep Parmar**

1. **What is Exploratory Testing?**

**Exploratory testing** is a flexible and unscripted approach to software testing that emphasizes personal experience, intuition, and creativity rather than following predefined test cases

Testers explore the application freely to discover defects and assess its usability

It is a concurrent process and to explore the app without refering test cases

1. **What is Traceability Matrix?**

A Traceability Matrix is a document that co-relates any two-baseline documents that require a many-to-many relationship to check the completeness of the relationship

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Completion 1 | Completion 2 | ..... | Completion m |
| Requirement 1 | X |  |  | X |
| Requirement 2 |  |  | X |  |
| ..... |  | X |  |  |
| Requirement n |  |  |  | X |

1. **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 (BVA) uses the same analysis of partitions as EP and is usually used in conjunction with EP in test case design but reﬁnes equivalence partitioning

The trick is to concentrate software testing efforts at the extreme ends of the equivalence classes

e.g.,

We take a equipartition example to understand BVA

To verify the range 50 to 100

Lower bound upper bound

a-1 a a+1 b-1 b b+1

49 50 51 99 100 101

1. **What is Equivalence partitioning testing?**

Equivalence Partitioning is a software testing technique used to reduce the number of test cases by dividing input data into partitions that are expected to produce similar results

The idea is that if one condition in a partition works correctly, the others should as well

e.g.,

Verify employees IDs: Here we have devided IDs of all the employees in equal partitions for example 1 to 50

Equivelence Partition Reperesentative Status

1 to 10 6 Pass/valid

11 to 20 12 Pass

21 to 30 27 Pass

31 to 40 33 Pass

41 to 50 48 Pass

51 to 60 55 Fail (Invalid)

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

Devide your range in equivelence partitions then choose the respesentative input from each partition to test them all

If the input will pass, the whole partition will be passed

If one value ﬁnds a bug, the others probably will too

If one doesn't ﬁnd a bug, the others probably won't either

1. **What is Integration testing?**

Integration Testing is a level of the software testing process where individual units are tested combined

e.g., login process

Here username & password both are combined and needed to be checked

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

There is two types of methods of Integration Testing:

* Bing Bang Integration Testing
* Incremental Integration Testing

1. Top Down Approach
2. Bottom Up Approach
3. **What determines the level of risk?**

Likelihood is the probability of a negative outcome which generally defines the level of risk

e.g.,

Chance of the occurring of the risk, typically on a scale of 1 to 10. Here the impact of that risk is identified on a scale of 1-10

While the mitigation is the process of reducing the likelihood of that outcome

e.g.,

Via smoke and sanity testing

1. **What is Alpha testing?**

* It is the type of testing that is always performed by the developers at the software development site and sometimes by independent testing team
* It is always performed in **Virtual Environment**
* Alpha testing is not open to the market and public
* It is conducted for the software application and project
* 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

1. **What is beta testing?**

* It is the type of testing that is always performed by the customers at their own site and not performed by independent testing team
* It is performed in **Real Time Environment**
* Beta testing is always open to the market and public
* It is usually conducted for software product
* 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
* 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 also considered as the User Acceptance Testing (UAT) which is done at customers or users area
* Beta testing can be considered “**pre-release**” testing
* It is also called as **Pilot Testing** i.e the testing of the product on real world as well as collect data on the use of product in the classroom

1. **What is component testing?**

Component Testing is a level of the software testing process where individual units/components of a software/system are tested

Sometimes known as Unit Testing, Module Testing or Program Testing

Unit tests are typically written and run by software developers and it is performed by using the White Box Testing method

It is the first level of testing which is performed prior to Integration Testing

e.g.,

In Login process we try to verify login by providing username or password individually

1. **What is Functional system testing?**

Functional system testing is the testing which verifies the attributes of a component or system that are directly relate to functionalities

e.g.,

Verification of the login to a system after entering correct credentials

To verify whether the application goes into minimized mode whenever there is an incoming phone call

Taking more time for testers to look for game play issues, graphicsc issues, audio-visual issues etc

Types of functional testing are

* Black Box Testing
* While Box Testing
* Experience Based Testing
* Smoke Testing
* Sanity Testing
* End to End Testing

1. **What is Non-Functional Testing?**

Non-Functional system testing is the testing which verifies the attributes of a component or system that are not directly relate to functionalities

e.g.,

In website number of user/customer will increase , how the website will handled to every customer/user

To verify the response time of the application

In mobile, it will automatically switch oﬀ without any reason

Types of non-functional testing are

* Performance Testing
* Load Testing
* Volume Testing
* Stress Testing
* Security Testing
* Installation Testing
* Penetration Testing
* Compatibility Testing
* Migration Testing

1. **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

**Approach of GUI Testing:**

* Manual Based Testing
* Record And Replay
* Model Based Testing

1. **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

It is also called Error Guessing -Random checking

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

**Types of Adhoc Testing**

* **Buddy testing:** Generally performed by tester and developer pair
* **Pair testing:** Generally performed by the pair of testers
* **Monkey testing:** Tester randomly test the product or application without test cases with a goal to break the system

1. **What is load testing?**

It is a performance testing to check system behavior (stability, response time) applying load

e.g If an application will handle 1000 users at every 5 second, you have to check 1000 or <=1000 users with your application

1. **What is stress Testing?**

It is a performance testing to check system behavior (stability, response time) after applying load more than the threshold limit

e.g If an application will handle 1000 users at every 5 second, you have to check 1000 or >=1000 users with your application

1. **What is white box testing and list the types of white box testing?**

It is the type of the functional testing which is based on an analysis of the internal structure of the component or system

It is also known as glass testing or open box testing

The testers have the knowledge of how the system or component is structured inside the box

e.g.,

Debugging the code while writing the software code for web, desktop, mobile or game based testing

Hence white box testing needs the knowledge of internal structure or coding, we can state three types via coverage (i.e. test coverage measures the amount of testing performed by a set of test)

* Statement coverage
* Decision coverage
* Condition coverage

1. **What is black box testing? What are the different black box testing techniques?**

It is the type of the functional testing which is done without the knowledge of the internal structure of the component or system

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

e.g.,

Testing the perticular software without the knowledge of the source code for web, desktop, mobile or game based testing

Different types of blackbox techniques are as below:

1. Equivalence partitioning
2. Boundary value analysis
3. Decision tables
4. State transition testing
5. Use-case Testing

Other Black Box Testing e.g., Syntax or Pattern Testing

1. **Mention what bigbang testing is?**

It is the type of integration testing in which all components or modules are integrated simultaneously, after which everything is tested as a whole

Big Bang testing has the advantage that everything is finished before integration testing starts

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

1. **What is the purpose of exit criteria?**

Exit Criteria is important because it defines the items that must be completed before testing can be concluded

In an Ideal world, one can not enter the next stage until the exit criteria for the previous stage is met, but practically this is not always possible

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

1. **When should "Regression Testing" be performed?**

**Regression testing should be performed:**

* 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

**Need of Regression Testing**

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

1. **What is 7 key principles? Explain in detail?**

* Testing shows the presence of Defects

Every software has defects or errors present of any kind

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

* Exhaustive Testing is Impossible

Testing everything including all combinations of inputs

and preconditions

Sometimes testing requires enormous resources, too expensive, and takes too much time

which is practically not possible. we cannot test everything (i.e. all combinations of inputs and pre-conditions)

* 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 which are outlined in the Test Strategy

* Defect Clustering

Defects are not evenly spread in a system, they are clustered i.e. most defects found during testing are usually confined to a small number of modules

* 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

Therefore we must learn, create and use new tests based on new techniques to catch new bugs

* Testing is Context Dependent

Testing is basically context dependent

Testing is done differently in different contexts

i.e. Different kinds of sites are tested differently

* Absence of Error Fallacy

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

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

1. **Difference between QA v/s QC v/s Tester**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **QA** | **QC** | **Tester** |
| **1** | It is a subset of Software Test Life Cycle (STLC) | QC can be considered as the subset of Quality Assurance | Testing is the subset of Quality Control |
| **2** | Preventive activities | It is a corrective process | It is a preventive process |
| **3** | Process oriented activities | Product oriented activities | Product oriented activities |
| **4** | 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 |
| **5** | 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 |

1. **Difference between Smoke and Sanity?**

|  |  |
| --- | --- |
| **Smoke Testing** | **Sanity Testing** |
| Smoke Testing is performed to ascertain that the critical functionalities of the program is working fine | Sanity Testing is done to check the new functionality/bugs have been fixed |
| The objective of this testing is to verify "stability" of the system in order to with more rigorous testing | The objective of the testing is to verify the "rationality" of the system in order proceed with more rigorous testing |
| This testing is performed by the developers or testers | Sanity testing is usually performed by testers |
| Smoke testing is usually documented or scripted | Sanity testing is usually not documented and is unscripted |
| Smoke testing is a subset of Acceptance testing | Sanity testing is a subset of Regression testing |
| Smoke testing exercises the entire system from end to end | Sanity testing exercises only the particular component of the entire system |
| Smoke testing is like General Health Check Up | Sanity Testing is like Specialized Health Check Up |

1. **Difference between verification and Validation**

|  |  |
| --- | --- |
| **Verification** | **Validation** |
| Verification is a process which is performed at development level | Validation is a process which is performed at testing level |
| Verification phases are:  Business Requirement Analysis  System Design/ System Requirement  Architectural Design (Technical Specification)  Module Design (Program Specification) | Validation Phases are:  Unit Testing  Integration Testing  System Testing  Acceptance Testing |
| It is the process of evaluating product of development to check whether the specified requirements meet or not | It is the process of evaluating the product of development to check whether it satisfied business requirements or not |
| Verification can be achieved by asking “Are you building a product right?” | Validation can be achieved by asking “Are you building a right product?” |
| The evaluation of verification can be achieved by planning, Requirement Specification, Design Specification, Code Specification, and Test Cases | The evaluation of validation can be achieved as an Actual Product |
| Verification activities are Reviews and Inspections | Validation activity is Testing |

1. **Explain types of Performance testing**

Performance testing is a type of software testing that focuses on evaluating the speed, stability, and scalability of an application under varying conditions

The primary aim is to ensure that the software meets performance criteria and can handle expected and unexpected loads effectively

**Types of Performance Testing:**

### Load Testing:

It is a performance testing to check system behavior (stability, response time) applying load

### Simulates expected user load to check system behavior under normal conditions

### ****Goal** here is to i**dentify performance bottlenecks and ensure the system can handle the expected number of users

e.g If an application will handle 1000 users at every 5 second, you have to check 1000 or <=1000 users with your application

### Stress Testing:

It is a performance testing to check system behavior (stability, response time) after applying load more than the threshold limit

### Tests the system beyond its normal operational capacity to see how it behaves under extreme conditions

### Evaluates the system's limits by pushing it beyond normal capacity

**Goal here is to d**etermine the breaking point of the system and identify failure modes

e.g If an application will handle 1000 users at every 5 second, you have to check 1000 or >=1000 users with your application

### Endurance Testing:

It is a performance testing to check system behavior (stability, response time) applying load over an extended period

Assesses performance over an extended period to identify issues like memory leaks

**Goal here is to i**dentify issues like memory leaks and resource degradation during prolonged use

e.g. To check how the system will run continuously

### Spike Testing:

### It is a performance testing to check system behavior (stability, response time) applying sudden load or sharp increases in load

Tests the system's response to sudden, high traffic spikes

**Goal here is to e**nsure the system can handle unexpected traffic surges without crashing

e.g. To check extreme increment or decrement of load according to the response time

### Volume Testing:

### It is a performance testing to check system behavior (stability, response time) with a large volume of data to evaluate its performance

Analyzes how the system handles large volumes of data

**Goal here is to** dentify how the system handles large datasets and potential performance issues related to data processing

e.g. To check the capacity or volume of database

### Scalability Testing:

### It is a performance testing to check system behavior (stability, response time, scalability) by system's ability to scale up or down in response to changes in load

Checks how well the system can scale in response to increasing demands

**Goal here is to a**ssess how well the system can handle increased load by adding resources

e.g. While checking the performance of the application, continue with load until your system will be crashed

If an application will handle 1000 users at every 5 sec

1500 users: 10 sec

2000 users: 20 sec

.....

1,00,000 users: 1000 sec... system crashed

1. **What is Error, Defect, Bug and failure?**

**Error:** A mistake in coding is called error

A discrepancy between a computed, observed, or measured value or condition and the true, specified, or theoretically correct value or condition

e.g., 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:** Error found by tester is called defect

Commonly refers to several troubles with the software products, with its external behavior or with its internal features

**Bug:** Defect accepted by development team then it is called bug

A fault in a program which causes the program to perform in an unintended or unanticipated manner See: anomaly, defect, error, exception, and fault Bug is terminology of tester

**Failure:** Build does not meet the requirements then it is failure

The inability of a system or component to perform its required functions within specified performance requirements. See: bug, crash, exception, and fault

1. **Explain the difference between Functional testing and Non Functional testing**

|  |  |
| --- | --- |
| **Functional Testing** | **Non-Functional Testing** |
| Functional testing is performed using the functional specification provided by the client and verifies the system against the functional requirements | Non-Functional testing checks the performance, reliability, scalability and other non-functional aspects of the software system |
| Functional testing is executed first | Non functional testing should be performed after functional testing |
| Manual testing or automation tools can be  used for functional testing | Using tools will be effective for this testing |
| Business requirements are the inputs to functional testing | Performance parameters like speed, scalability are inputs to non-functional testing |
| Functional testing describes what the product does | Nonfunctional testing describes how good the product works |
| Easy to do manual testing | Tough to do manual testing |
| Types of Functional testing are  Unit Testing  Smoke Testing  Sanity Testing  Integration Testing  White box testing  Black Box testing  User Acceptance testing  Regression Testing | Types of Non-functional testing are  Performance Testing  Load Testing  Volume Testing  Stress Testing  Security Testing  Installation Testing  Penetration Testing  Compatibility Testing  Migration Testing |

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

|  |  |
| --- | --- |
| **SDLC** | **STLC** |
| Focused on software development | Focused on software testing |
| Helps to develop good quality software | Helps to make software defects free |
| SDLC phases are completed before the STLC phases | STLC phases are performed after SDLC phases |
| Coders create a well-organized development plan | QA team defines the test plan |
| Developers create the actual software | Tester designs test cases, set up the environment & work out the RTM |

1. **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

Different Methodologies in Agile Development Model Agile Methodologies are:

* Scrum
* Kanban
* XP

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

|  |  |  |
| --- | --- | --- |
| **Test Scenario** | **Test Case** | **Test Script** |
| Any functionality that can be tested | It involve the set of steps, conditions and inputs which can be used while performing the testing tasks | A set of sequential instruction that detail how to execute a core business function |
| May include multiple test cases | Includes preconditions, steps, and expected results | Written in a programming/scripting language |
| e.g., "Verify user login functionality." | e.g., TC001: Verify login with valid credentials | e.g.,Selenium script for automated login test |
| The scenarios are derived from use cases | Test cases are derived from test scenario | Set of instructions for executing a test case i.e. derived from test case |

1. **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 the testing
* Defining the overall approach of testing including test entry and exit criteria
* Integrating and coordinate the testing activities into software life cycle
* Scheduling test analysis, design, implementation, execution and evaluation activities

Test Plan & Strategy, Test Planning Factors, Test Planning Activities, **Entry Criteria, Exit Criteria should be covered in** Test plan

1. **When to used Usablity Testing?**

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

There are many software applications / websites, which miserably fail, once launched, due to following reasons:

Where do I click next?

Which page needs to be navigated?

Which Icon or Jargon represents what?

Error messages are not consistent or effectively displayed

Session time not sufficient

Usability Testing identifies usability errors in the

system early in development cycle and can save a product from failure

e.g.,

In Web Based Testing , Desktop Based , Mobile based & Game based Testing, if fields on a page (Text box, radio options, drop-down lists) aren’t aligned properly, not accessible by keyboard shortcuts then we need usability testing in those cases

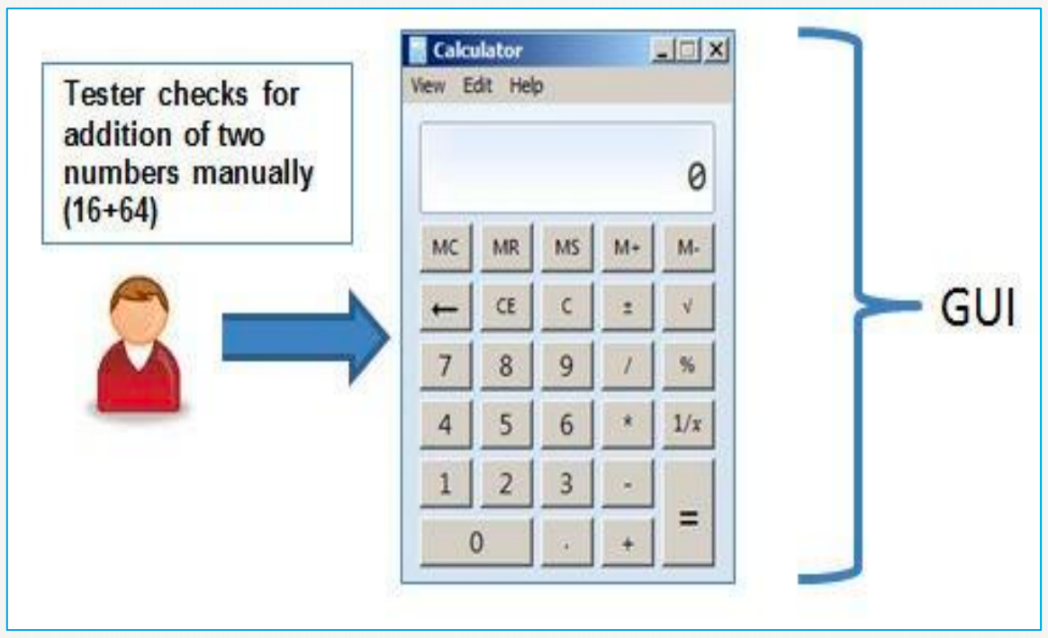
In short, effectiveness of the system, efficiency, eccuracy, user friendliness are goals in need of usability testing

1. **What is the procedure for GUI Testing?**

**Procedure for GUI testing involves:**

* **MANUAL BASED TESTING**

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



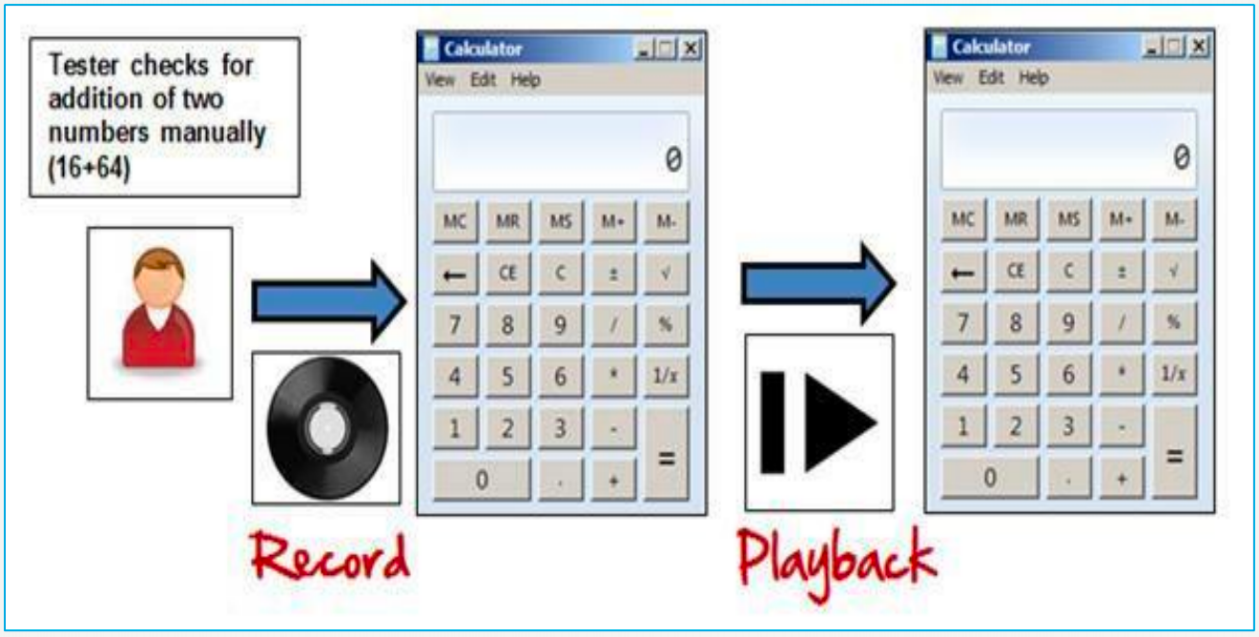
* **RECORD AND REPLAY**

GUI testing can be done using automation tools.

This is done in two 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 behavior. It helps us to understand and predict the system behavior. Models help in a generation of efficient test cases using the system requirements (e.g. Charts, Decision tables etc)

e.g. Build a model

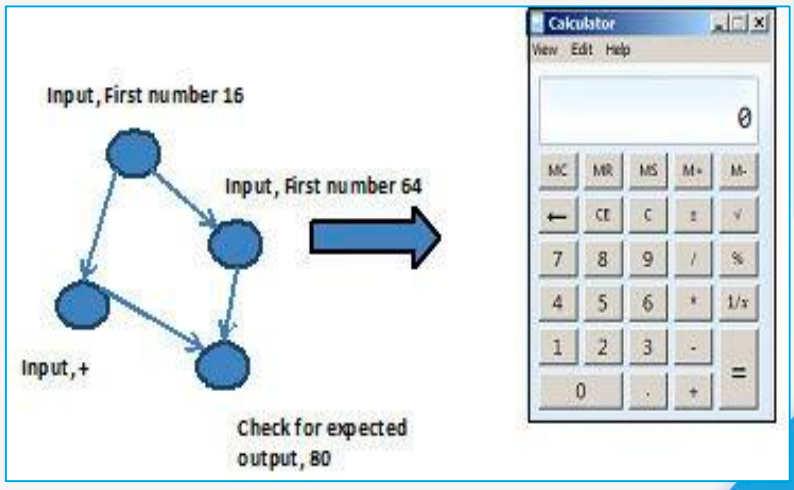
Determine Inputs for the model

Run the tests

Compare the actual output with the

Expected output

Decision on further action on the model



1. **Write test scenarios for real objects (Pen, Pen Stand, Door, ATM, Microwave Owen, Coffee Vending Machine, Chair, Wrist Watch, Lift (Elevator), Water Bottle, Fan)**

****

1. **Write test scenarios for apps (Gmail (Receiving mail, Online shopping to buy product (Flipkart), Only Whatsapp chat messages, Whatsapp Group (generate group, Whatsapp payment)**

****

1. **To create HLR, Test Scenario and Test cases on**

Instagram (web) Login Page: <https://www.instagram.com/accounts/login/?hl=en>



Facebook (web) Login Page: <https://www.facebook.com/>



WhatsApp (web): <https://web.whatsapp.com/>



ArtOfTesting Contact us page: https://artoftesting.com/

