

AUTOMATION TESTING (COGNIZANT)

Training Report in partial fulfillment of the requirement for degree of

Bachelor of Technology

In

Information Technology

By

(OJAS THAREJA (171464))

Under the supervision of

(Mr. RIZWAN UR REHMAN)

TO



Department of Computer Science & Engineering and Information

Technology

Jaypee University of Information Technology Waknaghat, Solan-

173234, Himachal Pradesh

Certificate

Candidate's Declaration

I hereby declare that the work presented in this report entitled “**Automation Testing**” in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Information Technology** submitted in the department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology, Waknaghat is an authentic record of my own work carried out over a period from February, 2021 to May, 2021 with the guidance of **Mr. Rizwan Ur Rehman**.

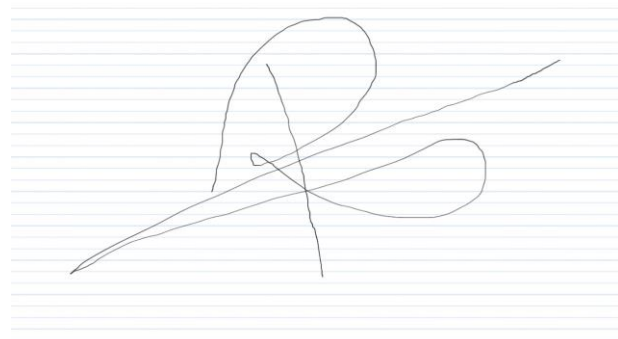
The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Ojas Thareja (171464)



(Student Signature)

This is to certify that the above statement made by the candidate is true to the best of my knowledge.



(Supervisor Signature)

ACKNOWLEDGEMENT

Many efforts have been made to complete this learning successfully. But, in order to make this learning become a reality, it has taken the kind support of many individuals and resources as well as the help of my colleagues and superiors. It would be an immense pleasure to extend my heartiest and most sincere thanks to all those who were involved in this learning directly or through me.

I am highly grateful to Mr. Rizwan Ur Rehman, keeping in mind her keen guidance and constant supervision and support throughout the process of this learning. I am also thankful to her for providing with all the necessary information and guidance which was required throughout the duration of this learning.

I would also like to convey my gratitude towards my peers from Jaypee University of Information Technology, who has been a constant support to me throughout the learning. I would also like to appreciate their kind help whenever I required it.

Project Report Undertaking

I, Mr. OJAS THAREJA

Roll No. 171464

Branch IT (CONDENSED SEMESTER)

is doing my internship with Cognizant from 27th February, 2021 to 11th June, 2021

As per procedure I have to submit my project report to the university related to my work that I have done during this internship.

I have compiled my project report. But due to COVID-19 situation my project mentor in the company is not able to sign my project report.

So I hereby declare that the project report is fully designed/developed by me and no part of the work is borrowed or purchased from any agency. And I'll produce a certificate/document of my internship completion with the company to TnP Cell whenever COVID-19 situation gets normal.

Signature:



Name: Ojas Thareja

Roll No. 171464

Date 24th May, 2021

ABSTRACT

This report includes the content studied during training at the company. It is aimed to automation testing, which is an important part of software development. It includes a prior knowledge of manual testing and good technical knowledge of SQL (Structured Query Language) and VBScript (Visual Basic Script).

Report contains a summary of topics learnt and practiced on a dedicated language-tool during training. It also includes a progress of work from week 1 to week 8. All work mentioned in Chapter 1, Chapter 2 and Chapter 3 should be considered as a prerequisite for the Chapter 4 which is the basis of overall learning.

Table Of Content

SR.NO	CONTENT	PAGE NUMBER
1	FUNCTIONAL TESTING	1
2	BASICS OF SQL	13
3	VBSCRIPT	24
4	AUTOMATION TESTING	33
5	SUMMARY	40

FUNCTIONAL TESTING

SOFTWARE

It is a non physical component of a computerized device which is instructed to perform specified type /category of operations. One can find use of software in almost every electronic device such as mobile phones, computers, laptops, LEDs, automobiles, etc. Software is generally known for its use in computers. Its development requires vast technical knowledge of computer programming languages, architectures, database systems, programming models, etc. It requires testing and maintenance to ensure that consumer does not face any operational difficulties in its use.

USES OF SOFTWARE

There are many uses of Software, such as:-

- Handling and operating computers.
- Performing business functions on computers
- Gaming
- Artificial Intelligence
- Web Applications
- Video, Audio related functionalities
- Database Management
- Application based on consoles

- Driver software

There are many more such uses

Different types of software are associated with the specific type of electronic device which is used to execute functionalities of that device.



Some of the software which are used on computers on daily basis areas follows

TYPES OF SOFTWARE



SOFTWARE TESTING

Software as a product requires vast maintenance for appropriate functioning. Developers need to keep an eye on market need and usage of software. Sometimes the customer may demand special features related to software from developers or may request for some change in existing features. Due to this reason developers need to regularly make changes and update coding of the software. In order to test whether the newly installed software functions properly or not we need to do software testing.

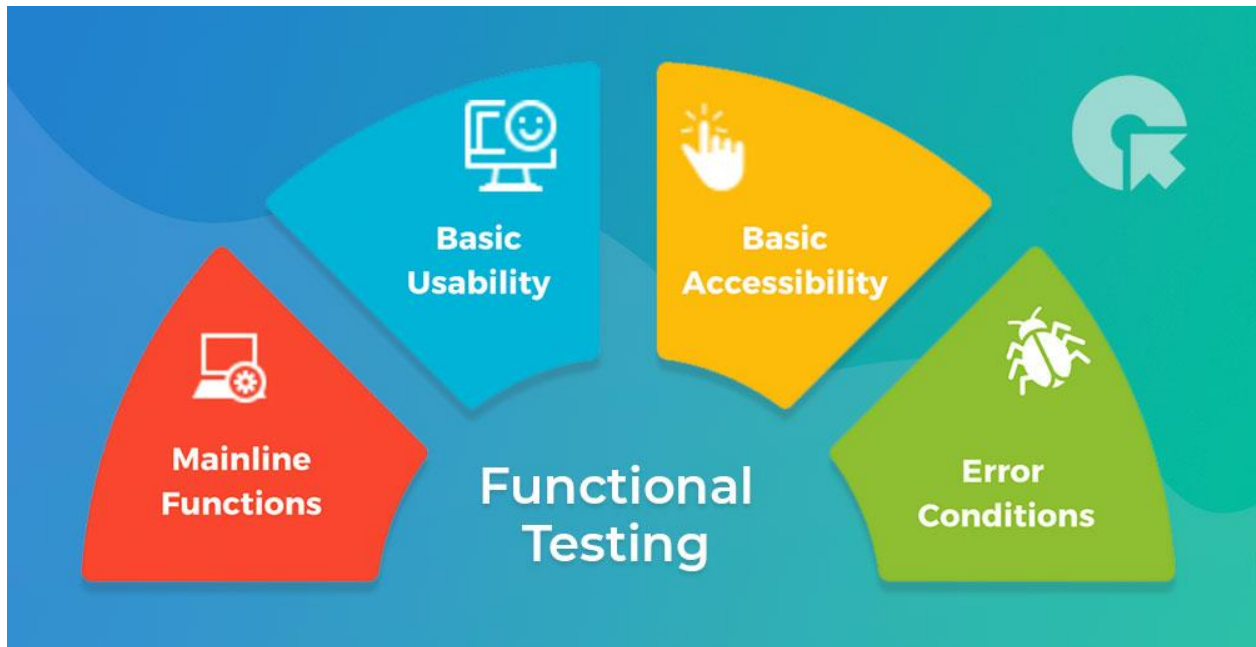
Software is also tested after design and development to verify whether the output of design is same as expectations. This is done as soon as development is complete. There are different ways by which testing team can achieve this which we will discuss in later section of this report along with necessary details.

SOFTWARE TESTING



FUNCTIONAL TESTING

This type of testing is used by testers to verify the working of particular functionalities in our software. The functionalities should give desired output without any specific error. It should also meet the requirements mentioned by customers. It can be done either after development of software or after developers perform modifications on the software. It uses the concept of test cases. Test cases are specific conditions which mention the set of operations and input values for the software. They expect certain output from software which they match it with the original outcome and then determine whether the test has passed or failed. Design of test cases depends solely on the development and testing team.

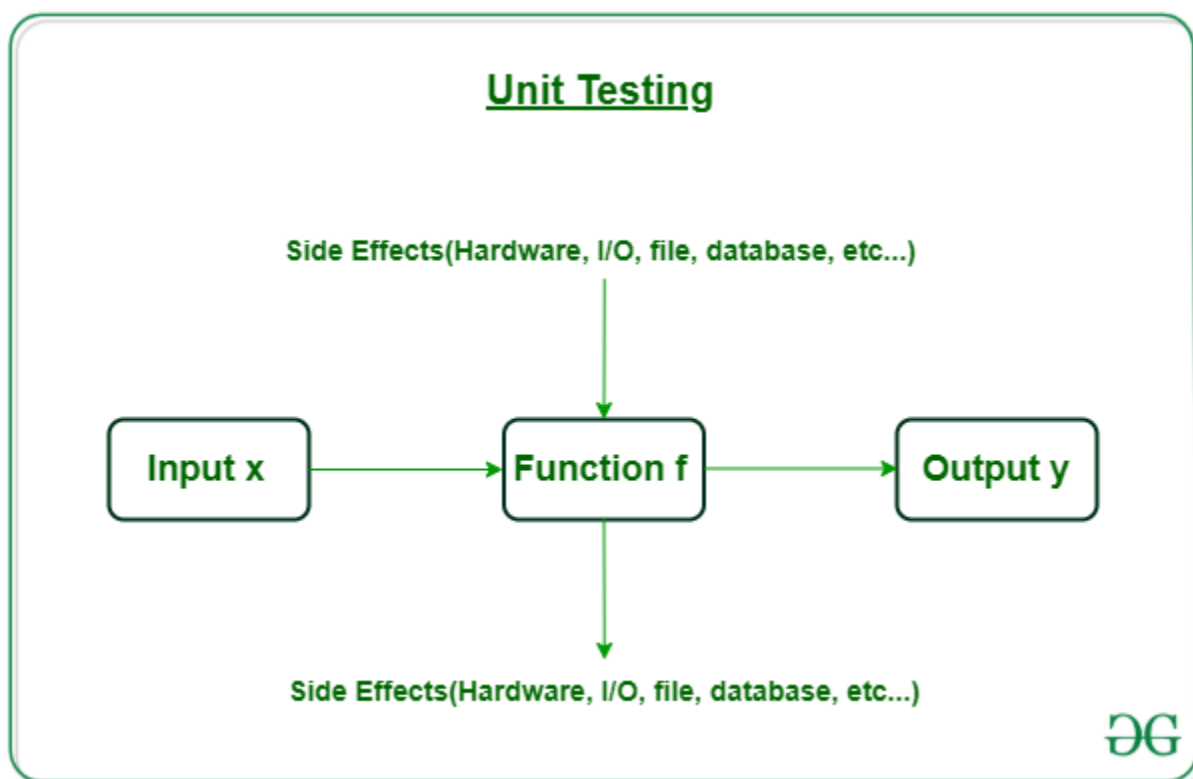


- Given above are the scenarios where functional testing is useful.
- Bugs are generated when these functionalities are not fulfilled.
- User's feedback can be included as a source along with developer's request to design specific test cases for testing.
- Functions that offer mainline performance are important.
- Bugs are noted down and are forwarded to development team for further considerations of functionalities.
- User's feedback is crucial if development method is agile.
- Type of testing is decided by situation and functionalities.
- We will discuss the situation for testing type in details.

TYPES OF FUNCTIONAL TESTING

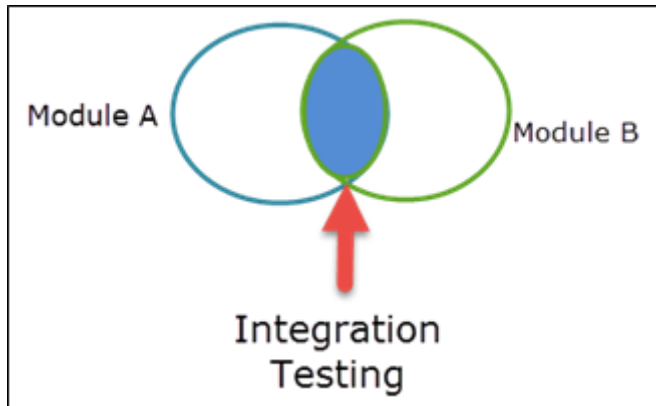
Unit Testing

- This type of testing is also known as component testing.
- In this testing each component is tested separately from rest of the software in separate environment.
- Most often it is done by developers.
- For unit testing the access to code is required.
- It helps in reducing cost of software as defects are detected earlier and one does not have to depend on the last moment to discover defects .



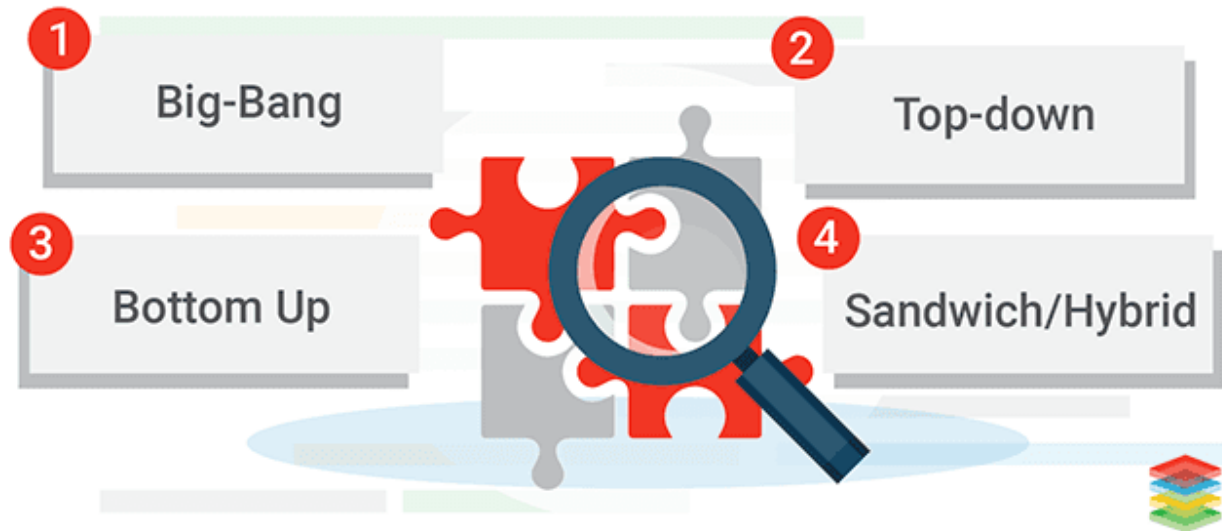
Integration Testing

- Units of software which were separately tested are merged together into a single group.
- Single group is then tested together.
- This type of testing is known as integration testing.
- For this type of testing each unit or component should be working properly.
- Integration testing has many prerequisites.



- System testing is the testing where system software is tested.
- System Integration testing is the type of testing where the entire system of software is altogether tested.

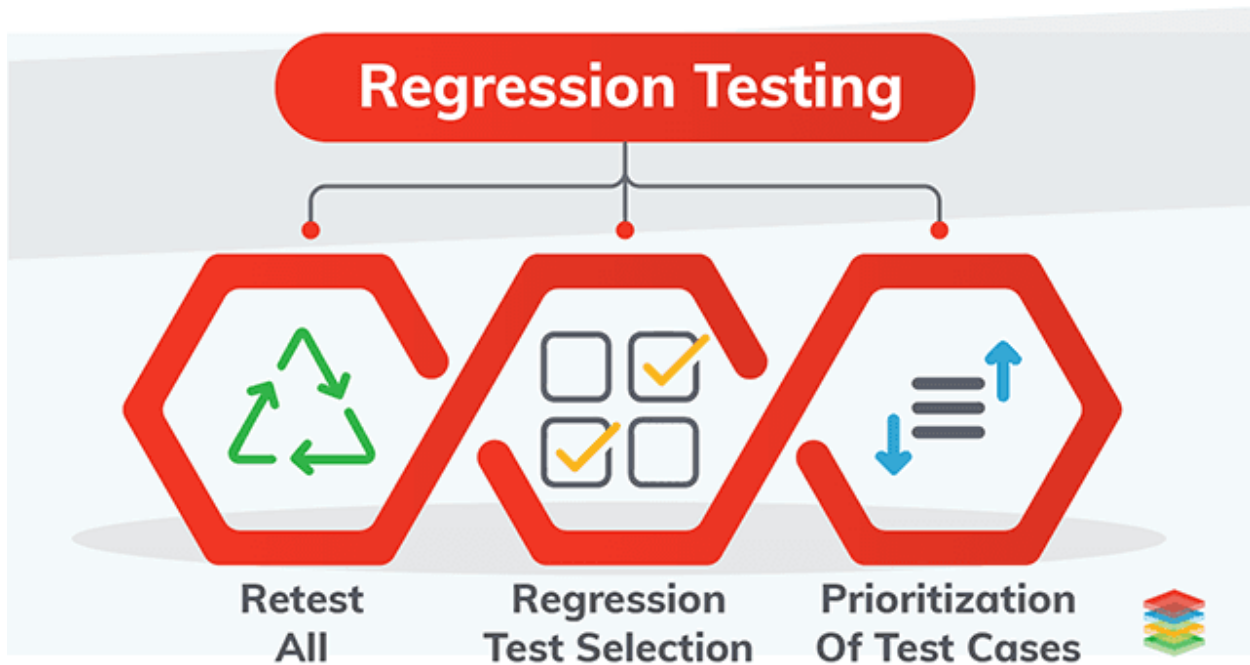
Approaches to Integration Testing



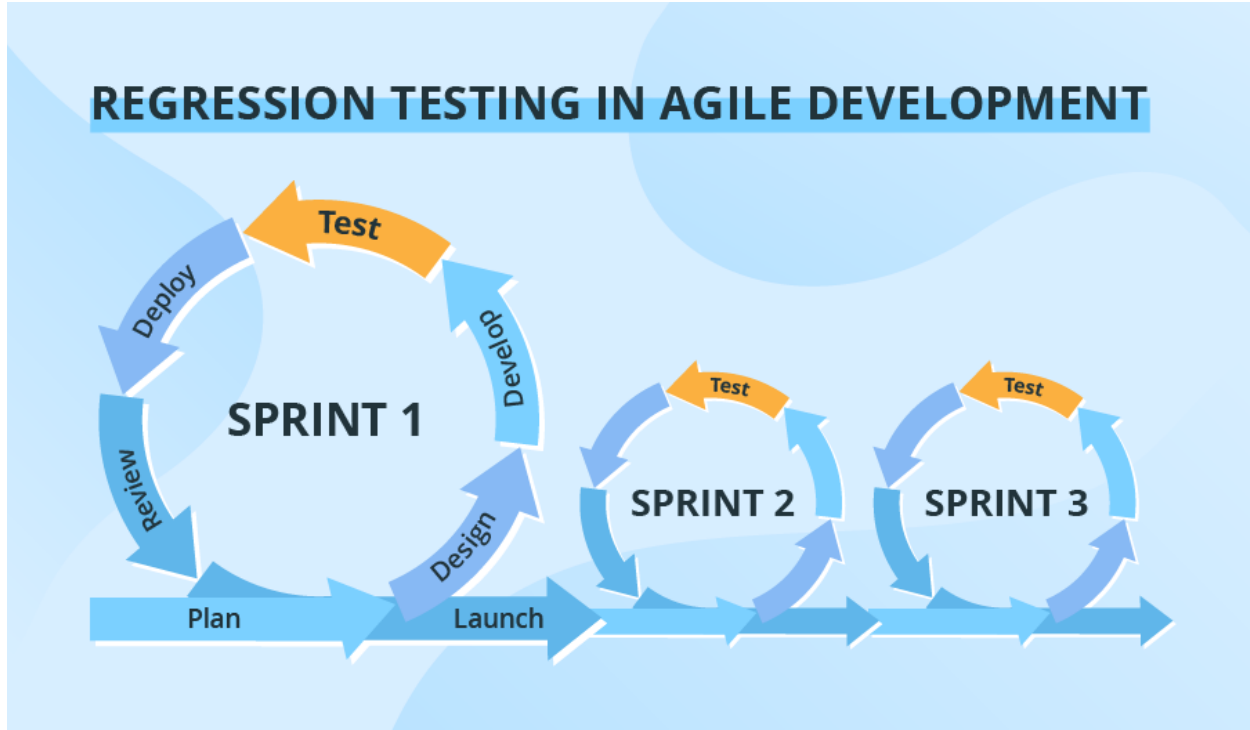
Regression Testing

- This type of testing is very useful when the modifications have been made in the software.
- It is done to verify that the functionalities that already existed with the software function properly even after the changes.
- At first we should verify the functionalities whose coding has been changed.
- Then there should be the priority of those functionalities which were either directly or indirectly related to the changed segment.
- And at last we can also verify functionalities that were independent of the changes in software.
- This can be done in consultation with the team.

Prioritization of functionalities is important before proceeding for regression testing. By which cases can be selected for testing.



- Testing all components is generally considered a bad approach for regression testing and should be avoided if possible.
- However, it depends on number of factors.
- Retesting all will be time consuming.
- Prioritization is best approach as test cases.
- Team lead can take final call on priority of tests.



- It is a diagram of regression testing in agile type of development of the software.
- Bugs are discovered in regression testing.
- Removal of bugs is important to proceed further.
- Bugs need to be removed timely.

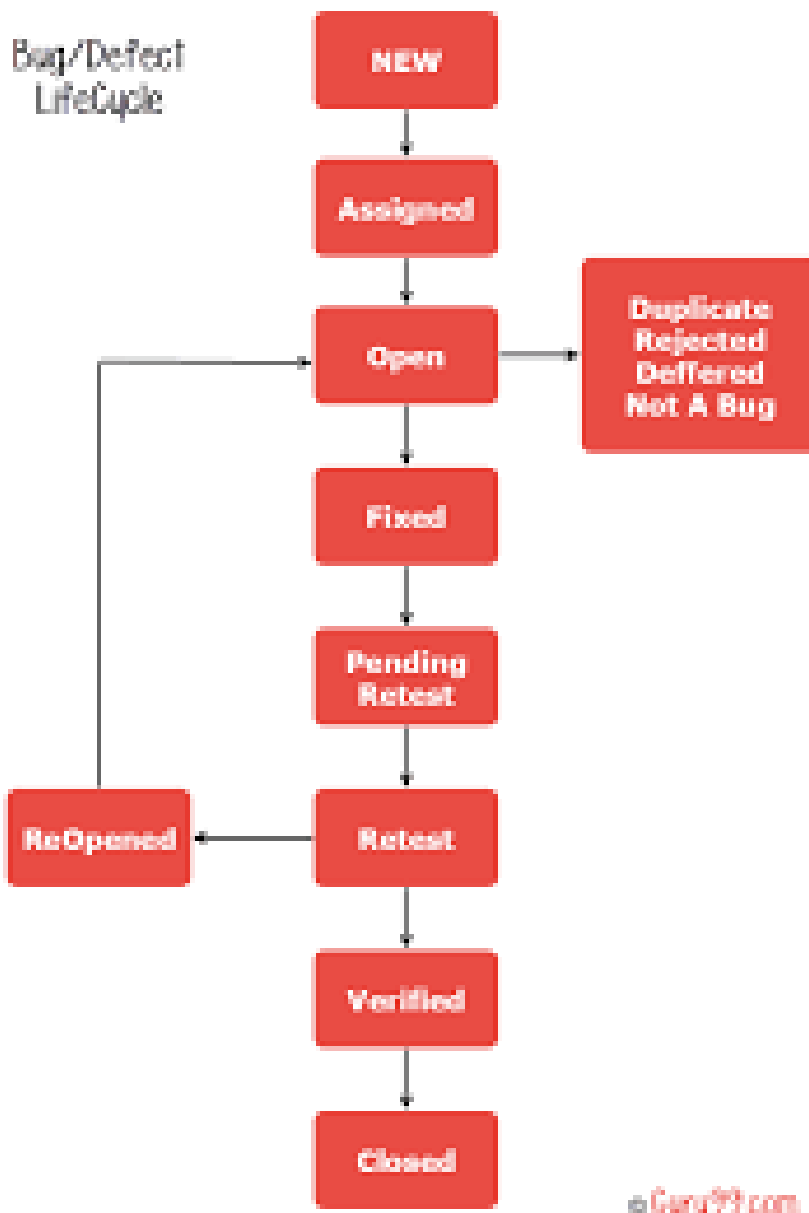
DEFECT LIFE CYCLE

- A defect is an unexpected result in functioning of software.
- Most defects are unwanted.
- It is necessary to remove bugs before delivering the software to the customers.
- Bugs are detected by testing team and are resolved by development team.

- Testing team needs to check again even after the development team makes claim to have resolved the bug.

Types of Bugs

- **New:** When a bug is detected at the first instance, its status is known as a 'New' bug
- **Open:** When a bug has been reported towards developer's team and team starts resolving it.
- **Fixed:** If the bug is resolved its status is changed to 'Fixed' by the testing team of software.
- **Retest:** The particular functionality is retested when developer makes claim of resolving it.
- **Reopen:** If the bug is not resolved the testing team can demand to reopen the bug.
- **Close:** The bug's status is changed to 'Close' once the issue has been resolved and there is no further need for its consideration.



This is the life cycle of bug according to which its status is changed periodically by the testing-team.

After verification the bug is closed.

BASICS OF SQL

SQL

- SQL stands for Structured Query Language.
- It is one of the most famous languages used for dealing with databases.
- SQL is used by many Database Management Systems.
- Some of the common software that make use of SQL are Microsoft Access, MYSQL, Oracle, etc.
- This language uses a query to which the software produces a response.

TYPES OF QUERIES

DDL

- It stands for data definition language.
- This type of language is used to define the basic structure of tables in database.

DQL

- It stands for Data Query Language.
- It is used to retrieve information.

DML

- It stands for data manipulation language.
- It is used in dealing with inserting data into database.
- It can also be used to update the existing data.

- Deletion can also be done through DML.

DCL

- It stands for data control language.
- It deals in granting permission to users.
- These queries are important to keep a check on the access to some confidential data in database.

TCL

- It stands for Transaction Control Language.
- It deals with controlling the flow of transactions in a database.

DDL

Important queries related to DDL are:

- Create Database

This query is used to create a database

Example:

```
CREATE DATABASE NAME_OF_THE_DATABASE;
```

- Create Table

It is used to create a table.

Example:

```
CREATE TABLE Employee (  
    NAME_OF_EMPLOYEE INT,  
    NAME_FIRST VARCHAR(20),
```

```
NAME_SURNAME VARCHAR(20),  
PERMANENT_ADDRESS VARCHAR(20),  
COLLEGE VARCHAR(20)
```

```
);
```

- Drop Table

This is used to drop a table from database.

Example:

```
DROP TABLE NAME_OF_THE_TABLE;
```

- Drop Database

This is used to remove a database.

Example:

```
DROP DATABASE NAME_OF_THE_DATABASE;
```

- Alter Table

This table is used to alter a table.

Example:

```
ALTER TABLE Cricket-Data  
DROP COLUMN AGE_OF_CRICKETERS;  
ADD BIRTH_DATE date;  
MODIFY COLUMN NUMBER_OF_CENTURIES int;  
ADD DESCRIPTION VARCHAR(100);  
DROP COLUMN LIST_OF_THIRTIES;
```

DQL

- Select Statement

This statement is used to retrieve data.

Examples:

```
SELECT MAX(DISCOUNT) FROM SALES;
```

```
SELECT MIN(DISCOUNT) FROM SALES;
```

```
SELECT COUNT(ID_OF_CUSTOMERS) FROM SALES;
```

```
SELECT ID_EMPLOYEE, DOB_EMP,
```

```
    SALARY, ADDRESS
```

```
    FROM DB_EMP
```

```
    WHERE ID_EMP > 50
```

```
    AND MANAGER_ID=60
```

```
    AND JOIN_YEAR=2019
```

```
    AND AGE<55;
```

DML

- Insert Into

This statement is used to add data into database.

Example:

```
INSERT INTO CRICKETERES (CRICKETER_NAME,
```

```
    COACH_NAME, CENTURIES, DESCRIPTION, PERFORMANCE_INDEX,
```

COUNT_MATCHES)

VALUES (“VIRAT KOHLI”,

“RAVI SHASTRI, 70, “One of the best cricketer”, 81

216);

- Update Statement

This is used to update existing data.

Example:

```
UPDATE CRICKETERS
```

```
SET UC_ID = 6744
```

```
WHERE COUNTRY = 'Ireland';
```

```
AND
```

```
AGE < 40
```

```
AND
```

```
FIRST_NAME = 'Kabir'
```

```
AND
```

```
LAST_NAME = 'Ali'
```

```
AND
```

```
CENTURIES=20
```

```
AND
```

```
Pi_index=30
```

```
;
```

- Delete Statement

This is used to delete certain entries from table.

Example:

```
DELETE FROM Cricketers  
  
WHERE  
  
NAME = "OJAS" AND  
  
AND LAST NAME = "THAREJA"  
  
AND AGE=23  
  
AND DOB YEAR=1998  
  
AND LOCATION="ABOHAR"  
  
AND BLOOD GROUP = "b+"  
  
;
```

JOINS

These are used to merge records in a database based on certain conditions.

Types of Joins are as following:

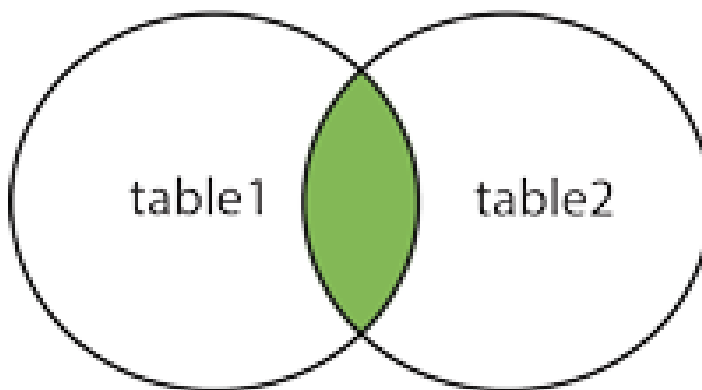
- **Inner Join**

This is used to join common values in both the mentioned tables

This is done on a particular tuple.

Basic condition is necessary.

INNER JOIN



The green section is the common section

This is present in both tables

Example:

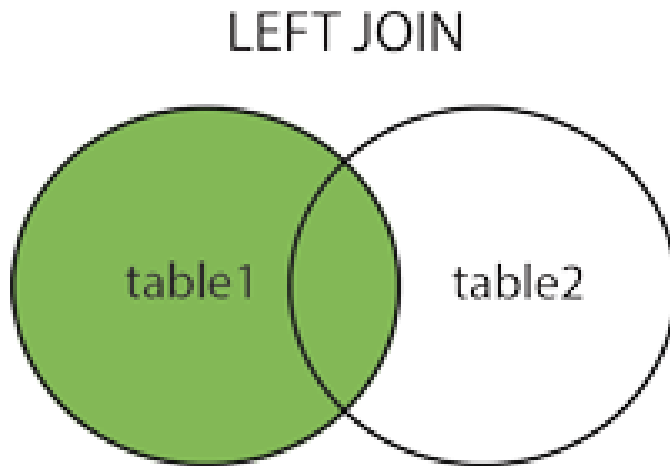
```
SELECT CRIC_NAME, CRIC_AVG
FROM CRICS
INNER JOIN
SPORTS_MANS
ON
DOB_CRIC =
DOB_SPORTS;
```

- **Left Join**

This is second type of join statement.

It contains all values of left table.

It also contains common values of both tables.



The green portion contains values from left table as well as the common values of both the tables.

Example:

Select CRIC NAME, CRIC AGE,

DOB CRIC, FIFTIES OF CRICKETER,

CENTURIES OF CRICKETER

From

CRICKETER

Left Join

SPORTS PERSON

On

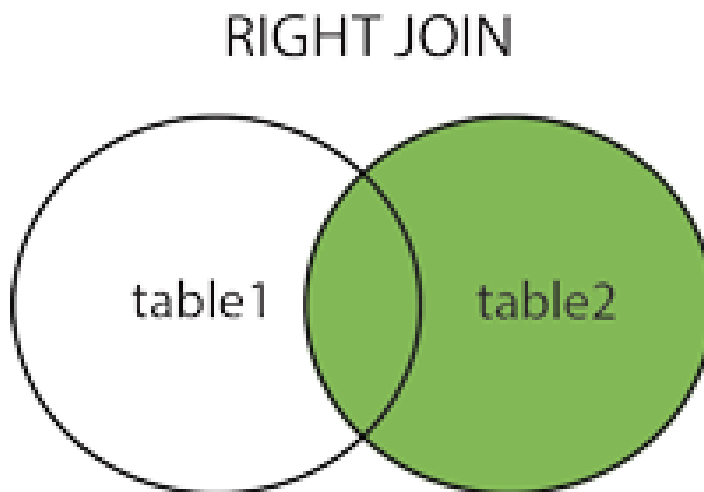
UC ID OF CRI = UC ID OF SPORTS PERSON

;

- **Right Join**

This query is used to create all entries of right table.

It also contains matching entries of both tables



Green portion contains values from right table as well as values which are common in both tables.

Example:

Select CRIC NAME, CRIC AGE,

DOB CRIC, FIFTIES OF CRICKETER,

CENTURIES OF CRICKETER

From

CRICKETER

Right Inner Join

SPORTS PERSON

On

UC ID OF CRI = UC ID OF SPORTS PERSON

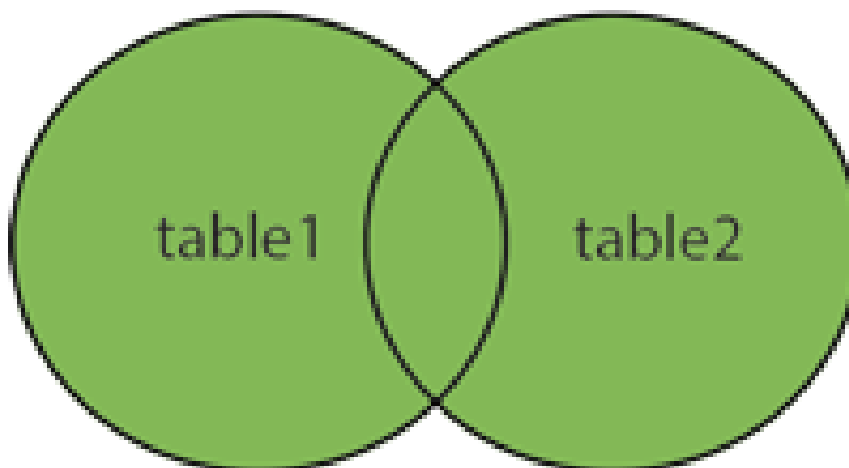
;

- **Cross Join**

It contains all values from both tables.

Overlapping values are only included once in net result.

CROSSJOIN



The green portion contains all values from both tables

Intersection is included only once.

Example:

Select CRIC NAME, CRIC AGE,

DOB CRIC, FIFTIES OF CRICKETER,

CENTURIES OF CRICKETER

From

CRICKETER

Cross Join

SPORTS PERSON

On

UC ID OF CRI = UC ID OF SPORTS PERSON

;

VBScript

- VBScript is a scripting language.
- Its full form is Visual Basic Scripting.
- It has many applications.
- It is widely used.

CONCEPTS OF VBSCRIPT COVERED IN TRAINING

Arithmetic Operations

```
1   in1=InputBox("Enter 1st Num")
2   in2=InputBox("Enter 2nd Num")
3   c=InputBox("Enter A->Addition S->Subtraction M->Multiply D->Division")
4   Dim il,i2
5   il=CInt(in1)
6   i2=CInt(in2)
7
8   If c="A" or c="a" Then
9       MsgBox il+i2
10  ElseIf c="S" or c="s" Then
11      MsgBox il-i2
12  ElseIf c="M" or c="m" Then
13      MsgBox il*i2
14  ElseIf c="D" or c="d" Then
15      If i2=0 Then
16          MsgBox "Error Division by 0"
17      Else
18          MsgBox il/i2
19      End If
20  End If
```

String Concatenation

```
Dim a : a = 5
Dim b : b = 10
Dim c

c = a+b
Document.write ("Concatenated value:1 is " &c) 'Numeric addition
Document.write ("  
</br>") 'Inserting a Line Break for readability

c = a&b
Document.write ("Concatenated value:2 is " &c) 'Concatenate two numbers
Document.write ("  
</br>") 'Inserting a Line Break for readability
```

Find Factorial

```
1 Dim n,f
2
3 n=inputbox("enter a number")
4 f=1
5 If n<0 Then
6     MsgBox "invalid number"
7 else if n=0 or n=1 then
8     MsgBox "the factorial of given number is : "&f
9 else
10     For i=1 to n
11         f=f*i
12     Next
13
14     MsgBox "the factorial of given number is : "&f
15
16 End If
17 end if
18
```

Error Handling

```
1 Dim x, y ,z
2
3 On Error Resume Next
4
5 x = 10
6 y = 0
7 z = x / y
8 If Err.Number <> 0 Then
9     MsgBox "There's been an error #1"
10 Else
11     MsgBox z
12 End IF
13
14 z = x * y
15 If Err.Number <> 0 Then
16     MsgBox "There's been an error #2"
17 Else
18     MsgBox z
19 End If
20
21 End Sub
```


Armstrong number

```
1 Dim num As Integer = 0
2 Dim rmd As Integer = 0
3 Dim sum As Integer = 0
4
5 Dim original As Integer
6 Dim tempNum As Integer
7 Dim count As Integer = 0
8
9 num = TextBox1.Text
10 original = num
11 tempNum = num
12
13 While tempNum <> 0
14     count = count + 1
15     tempNum = Math.Floor(tempNum / 10)
16
17 End While
18
19 While num > 0
20     rmd = Math.Floor(num Mod 10)
21     num = Math.Floor(num / 10)
22     sum = sum + Math.Pow(rmd, count)
23
24 End While
25
26 If sum = original Then
27     MessageBox.Show("Armstrong")
28
29 Else
30     MessageBox.Show("Not Armstrong")
31
32 End If
33
34 End Sub
35
36
37
38
```

String Comparison

```
1 Dim sStr1, sStr2, nComp
2
3 sStr1 = "ABCD"
4 sStr2 = "abcd"
5
6 nComp = StrComp(sStr1, sStr2, 1) 'Returns 0
7
8 nComp = StrComp(sStr1, sStr2, 0) 'Returns -1
9
10 nComp = StrComp(sStr2, sStr1) 'Returns 1
11
```

File Handling

```
1 Dim oFS, d, dc, s, n
2 Set oFS = CreateObject("Scripting.FileSystemObject")
3 Set dc = oFS.Drives
4
5 For Each d in dc
6     n = ""
7     s = s & d.DriveLetter & " - "
8     If d.DriveType = 3 Then
9         n = d.ShareName
10    ElseIf d.IsReady Then
11        n = d.VolumeName
12    Else
13        n = "Drive not available"
14    End If
15    s = s & n & " : "
16 Next
17
18 document.write s
19 document.write dc.count
20
```

Extract Data from Excel Sheet

```
1 For counter = 1 to workSheetCount
2   Set currentWorkSheet = objExcel.ActiveWorkbook.Worksheets(counter)
3   REM how many columns are used in the current worksheet
4   usedColumnsCount = currentWorkSheet.UsedRange.Columns.Count
5   REM how many rows are used in the current worksheet
6   usedRowsCount = currentWorkSheet.UsedRange.Rows.Count
7
8   REM What is the topmost row in the spreadsheet that has data in it
9   top = currentWorkSheet.UsedRange.Row
10  REM What is the leftmost column in the spreadsheet that has data in it
11  left = currentWorkSheet.UsedRange.Column
12
13  Set Cells = currentWorkSheet.Cells
14  REM Loop through each row in the worksheet
15  For row = 0 to (usedRowsCount-1)
16
17    REM Loop through each column in the worksheet
18    For column = 0 to usedColumnsCount-1
19      REM only look at rows that are in the "used" range
20      curRow = row+top
21      REM only look at columns that are in the "used" range
22      curCol = column+left
23      REM get the value/word that is in the cell
24      word = Cells(curRow,curCol).Value
25      REM display the column on the screen
26      WScript.Echo (word)
27    Next
28  Next
29
30  REM We are done with the current worksheet, release the memory
31  Set currentWorkSheet = Nothing
32 Next
33 objExcel.Workbooks(1).Close
34 objExcel.Quit
35
36 Set currentWorkSheet = Nothing
37 REM We are done with the Excel object, release it from memory
38 Set objExcel = Nothing
```

Directory Creation

```
1 dim filesystem, newfolder, newfolderpath
2
3 newfolderpath = "c:\DevGuru\myfolder"
4
5 set filesystem=CreateObject("Scripting.FileSystemObject")
6 If Not filesystem.FolderExists(newfolderpath) Then
7     Set newfolder = filesystem.CreateFolder(newfolderpath)
8     Response.Write("A new folder has been created at: " & newfolderpath)
9 End If
10
```

Regular Expression

```
1 Sub VariablePartsDemo
2
3 Dim notepad
4 Dim dlgFont
5 Dim regEx
6
7 TestedApps.notepad.Run
8
9 Set notepad = Sys.Process("notepad")
10 Call notepad.Window("Notepad", "*").MainMenu.Click("Format|Font...")
11
12 Set dlgFont =notepad.Window("#32770", "Font")
13 ' Create a regular expression
14 Set regEx = New RegExp
15 ' Set a regular expression pattern that matches any sans serif typeface
16 regEx.Pattern = ".* [S|s]ans"
17 Call dlgFont.Window("ComboBox", "", 1).ClickItem(regEx)
18
19 End Sub
20
```

```

1 Sub LocaleIndependentDemo
2   Dim form1
3
4   Dim dataGridView
5
6   Dim caption
7
8   TestedApps.DataGridViewSample.Run
9
10  Set form1 = Sys.Process("DataGridViewSample").WinFormsObject("Form1")
11
12  Set dataGridView = form1.WinFormsObject("dataGridView1")
13  ' Create a regular expression
14  Set caption = New RegExp
15  ' Set a regular expression pattern that specifies a caption in multiple languages
16  caption.Pattern = "(Customer Name)|(Kundennamen)|(Nom du client)|(Nombre del cliente)"
17
18  Call dataGridView.ClickCell(3, caption)
19 End Sub
20

```

```

1 Sub ExtractDemo
2
3 Dim InStr, regEx, Match, Matches
4 InStr = "Please e-mail us at support@mycompany.com or sales@mycompany.com"
5 ' Create a regular expression
6
7 Set regEx = New RegExp
8
9 ' Set regular expression pattern that specifies an e-mail address
10 regEx.Pattern = "\w+(\.|\w-)*@([\w-]+\.)+[a-zA-Z]{2,6}"
11 regEx.Global = True ' Set global applicability
12 ' Perform search operation
13
14 Set Matches = regEx.Execute(InStr)
15 ' Iterate Matches collection
16
17 For Each Match in Matches
18   ResStr = ResStr & Match.Value & "; "
19 Next
20

```

```

1 Sub Test
2
3     Dim re, matches, match
4
5     Set re = New RegExp
6     re.Pattern = "gr[ae]y"
7     re.IgnoreCase = True
8     re.Global = True ' Find all occurrences of the pattern
9
10    Set matches = re.Execute("Do you spell it gray or grey?")
11
12    For Each match In Matches
13        Log.Message (match.Value)
14    Next
15
16 End Sub
17

```

```

1 Sub main()
2
3     Dim notepad, window, dialog, control
4     Call WshShell.Run("notepad.exe", SW_NORMAL)
5     Set notepad = Sys.Process("notepad")
6
7     ' Get Notepad's main window by using a regular expression
8     Set window = notepad.Find("WndCaption", "regexp:. *Notepad", 5)
9
10    window.MainMenu.Click("Format|Font...")
11    Set dialog = notepad.Find("WndCaption", "Font")
12
13    ' Specify words that must be in the specified property
14    ' of the desired object by using a regular expression
15    Set control = dialog.FindChild("wItemList", "regexp:. *Western.*Arabic.*", 5)
16
17    ' Specify the item to be clicked using a regular expression
18    control.ClickItem("regexp:. *European$")
19 End Sub
20

```

AUTOMATION TESTING

INTRODUCTION

- In this testing, software is used to for executing test cases.
- Programmed coding is required to perform such execution.
- The software is designed to perform every operation from opening test source to generating report of testing.
- Software can be executed in iterations.
- Time consumption is the biggest reason which makes it a more effective method than manual testing
- Each and every iteration can be supplemented with minor changes in the input values to the software.
- Maximum projects in IT industry prefer to use an automated tool to perform testing for software.
- Software can manage to pick input values from external file and export output of execution to an external file.
- For maintenance purpose automation is very crucial. It provides easy maintenance of the old software.
- The best advantage of automation is that it makes use of coding language without requiring any prior excess to internal coding and design of software.
- It can analyze the internal working of software by a single execution of basic operations on software.

- Automation can be used at regular interval as operations performed on software does not require much time for execution.
- Code Complexity is also less in automation testing.

Some of the common Automation Testing are:

- ✚ Selenium
- ✚ Test Complete
- ✚ Micro Focus Unified Functional Testing
- ✚ Katalon Studio
- ✚ Ranorex
- ✚ Cumumber
- ✚ Watir
- ✚ Test Studio
- ✚ HP Win Runner Tricentis Tosca
- ✚ Parasoft
- ✚ Parasoft virtualize
- ✚ Capybara
- ✚ Ascential Test
- ✚ QF – Test
- ✚ Test Partner
- ✚ Twist
- ✚ Rational Performance
- ✚ Jtest

In training I have worked on Micro Focus UFT

UFT Micro Focus

- It is one of the easiest and one of the most common automation tools used in IT market by professionals.
- Here are some important features of UFT:
- It allows users to add all kinds of adds on.
- Adds on are added every time UFT is opened.
- It is very useful when security is concerned.
- It is one of the fastest automation tools.
- It provides different kind of features such as:
 1. Record
 2. Object Spy
 3. Automatic Script Generation
 4. Class hierarchy
 5. Use case diagrams
- It allows different kind of applications to be tested on UFT.
- It allows user to export data to local as well as external file.

UFT LEARNING

ADD IN

Micro Focus Unified Functional Testing - Add-in Manager

License: DEMO

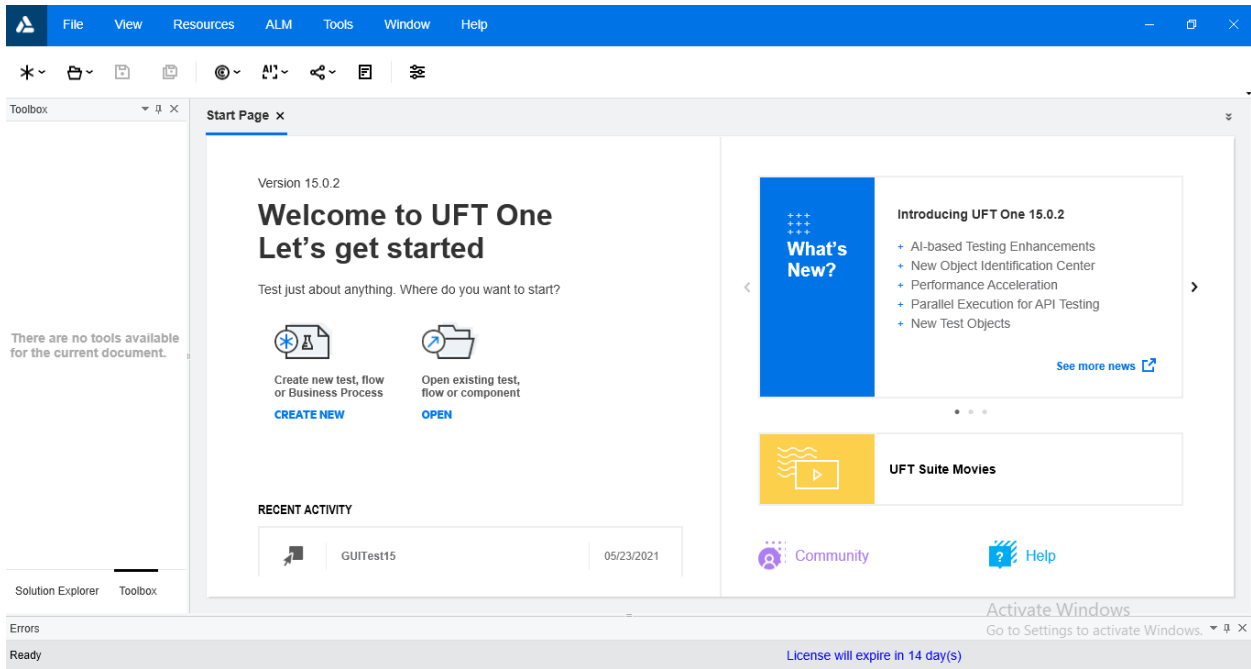
SELECT ADD-INS TO LOAD ⓘ

<input checked="" type="checkbox"/>	ActiveX	Licensed
<input checked="" type="checkbox"/>	Mobile	Licensed
<input checked="" type="checkbox"/>	UI Automation	Licensed
<input checked="" type="checkbox"/>	Visual Basic	Licensed
<input checked="" type="checkbox"/>	Web	Licensed
<input checked="" type="checkbox"/>	AgGrid	Licensed
<input checked="" type="checkbox"/>	UIGrid	Licensed

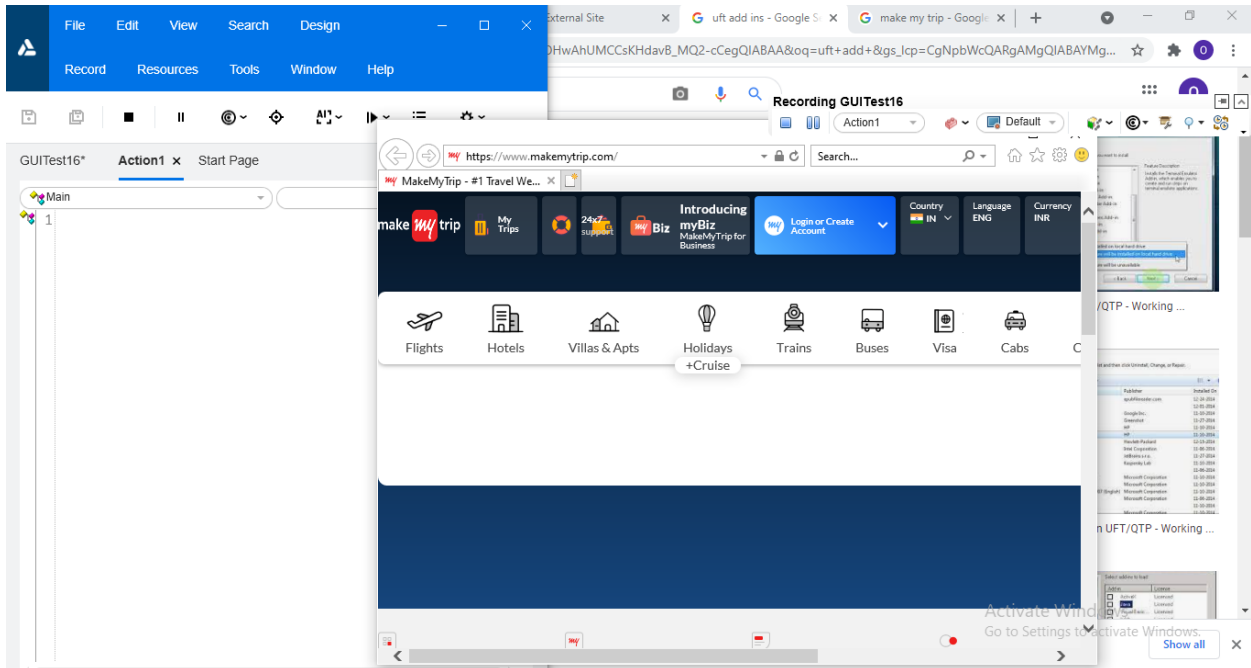
Show on startup

OK

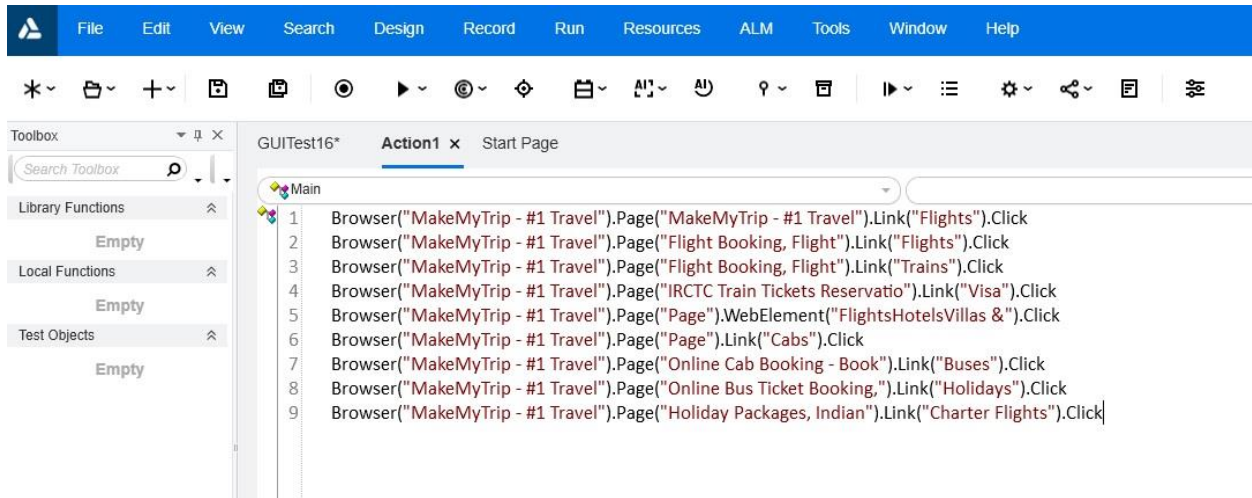
Home Page



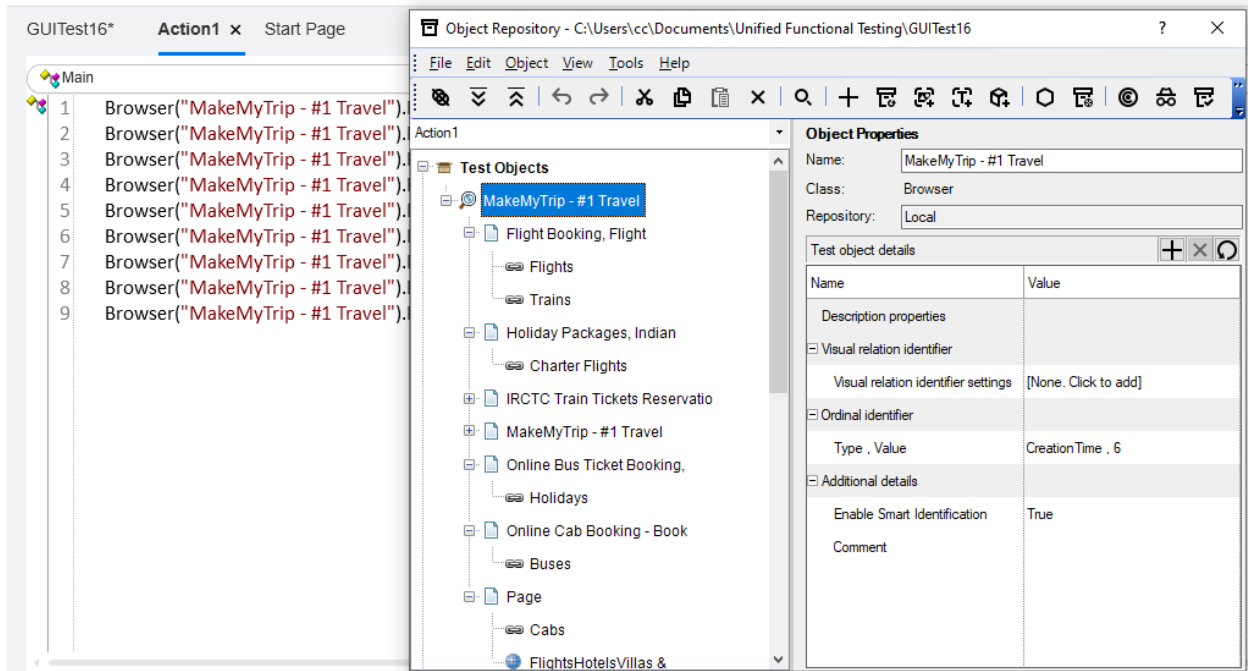
Record



Script



Object repository



Report

UFT Report : GUITest16 - Res1 x GUITest16* Action1 Start Page

GUITest16 - Res1

Passed	10 (100%)
Warnings	0 (0%)
Failed	0 (0%)

Test Data

Execution Time 2021-05-24 14:54:36 Duration 00:01:49 Tool Name Micro Focus Unified Functional Testing 15.0.2

Show More ▶

Errors list **Test flow** Search... Collapse All Expand All

- Test Iteration: Row 1
 - Action1
 - MakeMyTrip - #1 Travel
 - Flights.Click

Test Iteration Details

Test Iteration
Iteration: Row 1

Execution Time
2021-05-24 14:54:42

Activate Windows

SUMMARY

- **WEEK 1**

Functional Testing

Basics of Agile

Introduction to Functional Testing

Testing Types

Testing Environment

Test Scenarios and Test Cases

Details of Manual Testing

Manual Test Report

Basics of Automation Testing

- **WEEK 2**

Data Source

Basics about Data

Introduction to MySQL

DDL, DQL and DML Queries

Joins and Sub-queries

SQL Hands On

Database Fundamentals

Quiz on Database

XML basics

XML hands on

JSON introduction

JSON hands on

Aggregate and Scalar Functions

Functions Hands On

- **WEEK 3**

VB Script

Basics of VB Script

Conditional Statements

Iterative Statements

Reusable Functions

Drives, Folders and File System Object

Excel Object Model

Regular Expression

Error Handling

Core Objects

Inspection of variables

VB Script Hands On

Quiz on VB Script

- **WEEK 4,5,6,7**

Automation Testing - UFT

Introduction to Automation Testing

Recording Web Pages

Dealing with Objects

Using HP UFT

Script Generation using VB Script

Automation of Several Websites

Descriptive Programming

Check Points Addition