

# **AUTOMATION TESTING - HC FACETS WITH UFT**

*Project report submitted in partial fulfillment of the requirement for the degree of*

## **BACHELOR OF TECHNOLOGY**

**IN**

## **ELECTRONICS AND COMMUNICATION ENGINEERING**

By

**Shachi Sharma (171049)**

**UNDER THE GUIDANCE OF**

**Sowndhariya Ravichandran (GenC Implementation, Cognizant)**



**JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT**

**May 2021**

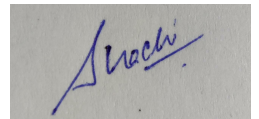
# TABLE OF CONTENTS

CAPTION	PAGE NO.
DECLARATION	i
PROJECT REPORT UNDERTAKING	ii
ACKNOWLEDGEMENT	iii
LIST OF ACRONYMS AND ABBREVIATIONS	iv
LIST OF SYMBOLS	v
LIST OF FIGURES	vi
ABSTRACT	vii
<b>CHAPTER-1: INTRODUCTION</b>	<b>1</b>
1.1 About the Company	1
1.2 Software Development Life Cycle	1
1.2.1 What is SDLC?	1
1.2.2 SDLC Phases	2
1.2.3 Software Development Life Cycle Models	3
1.4 Functional Testing	7
1.4.1 Functional Testing Techniques	8
1.4.2 Functional Testing Goals	9
1.4.3 Functional Testing Tools	10
1.4.4 Functional Testing Process	11
1.4.5 Conclusion	14
<b>CHAPTER-2: DATA SOURCE(SQL, XML, JSON)</b>	<b>15</b>
2.1 Overview	15
2.2 Types of Database	15
2.3 Importance of Datasource in Testing	17
2.4 Sequential Query Language	18
2.4.1 Relational Database	18
2.4.2 SQL Applications	19
2.4.3 Types of SQL Statements	19
2.5 Extensible Markup Language	20
2.5.1 Difference between XML and HTML	21
2.5.2 Advantages of using XML	21
2.6 JavaScript Object Notation	22

2.6.1 Advantages of using JSON	22
<b>CHAPTER-3: VBSCRIPT</b>	<b>23</b>
3.1 Overview	23
3.2 Features of VBScript	23
3.3 History and Uses	23
3.4 Basic Programming in VBScript	24
3.4.1 Running and Debugging VBScript code	24
3.4.2 Variable Declaration and Data Types	24
3.4.3 Loops and Conditions	25
3.4.4 Functions	25
3.4.5 Arrays	25
3.5 Advantages of using VBScript	26
3.6 Work Done	27
3.7 Conclusion	35
<b>CHAPTER-4: UNIFIED FUNCTIONAL TESTING</b>	<b>36</b>
4.1 Introduction	36
4.2 Software Description	36
4.2.1 Exceptional Handling	36
4.2.2 Data driven Testing	37
4.2.3 Automating custom and test objects	37
4.2.4 Extensibility	37
4.3 User Interface	38
4.3.1 Keyboard View	38
4.3.2 Expert View	38
4.4 Language	38
4.5 Work Done	38
<b>REFERENCES</b>	<b>44</b>
<b>CONCLUSION</b>	<b>45</b>

## DECLARATION

I hereby declare that the work reported in the B.Tech Project Report entitled “**Automation Testing - HC facets with UFT**” submitted at **Jaypee University of Information Technology, Waknaghat, India** is an authentic record of our work carried out under the supervision of **Sowndhariya Ravichandran(GenC Implementation, Cognizant)**. I have not submitted this work elsewhere for any other degree or diploma.



Shachi Sharma

171049

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

Sowndhariya Ravichandran

Date: 20th May, 2021.

Head of the Department/Project Coordinator

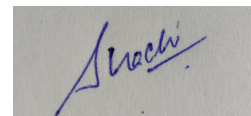
## PROJECT REPORT UNDERTAKING

I Ms. SHACHI SHARMA , Roll No.- 171049 Branch- ELECTRONICS AND COMMUNICATION 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.



Name: Shachi Sharma

Roll No.: 171049

Date: 20th May 2021

## ACKNOWLEDGEMENT

It is with profound gratitude that I express my deep indebtedness to all the employees of “COGNIZANT” without whose support and guidance it would not have been possible for this training to have materialized and taken a concrete shape.

I owe my personal thanks to my supervisor and Project Manager, **Ms. Sowndhariya Ravichandran** and **Mr. Neelamegam Regasamy** who extended full support and co-operation at every stage of my training period.

I would like to thank **Dr. Rajeev Kumar**, Head of the Department, Department of ECE, Jaypee University of Information Technology, Waknaghat for giving me this opportunity to carry out this training.

I am also indebted to my teachers, parents and friends for their constant encouragement and helping me in my endeavor. Last, but not the least, I would like to thank everyone who has contributed for the successful completion of my training.

Shachi Sharma

171049

B.Tech (ECE)

## LIST OF ACRONYMS AND ABBREVIATIONS

1. SDLC	Software Development Life Cycle
2. SRS	Specification Requirement Software
3. HLD	High Level Design
4. LLD	Low Level Design
5. QA	Quality Assurance
6. UAT	User Acceptance Testing
7. UFT	Unified Functional Testing
8. API	Application Programming Interface
9. GUI	Graphical User Interface
10. SQL	Sequential Query Language
11. XML	Extensible Markup Language
12. JSON	JavaScript Object Notation
13. DDL	Data Definition Language
14. DML	Data Manipulation Language
15. DCL	Data Control Language
16. TCL	Transactional Control Language
17. HTML	Hyper Text Markup Language
18. COM	Component Object Model
19. IE	Internet Explorer
20. WSH	Windows Scripting Host
21. IIS	Internet Information Service
22. QTP	Quick Test Professional
23. IDE	Integrated Development Environment
24. VBA	Visual Basic Applications

## LIST OF FIGURES

<b>CAPTION</b>	<b>PAGE NO.</b>
<b>Figure 1.1:</b> Phases in Software development life cycle	1
<b>Figure 1.2:</b> Phases of Waterfall Model	4
<b>Figure 1.3:</b> V-Model	5
<b>Figure 1.4:</b> Agile Model	6
<b>Figure 1.5:</b> Functional Testing Techniques	8
<b>Figure 1.6:</b> Functional Testing Process	13
<b>Figure 2.1:</b> Example of a Database	14
<b>Figure 2.2:</b> Example of Relational Database	16
<b>Figure 2.3:</b> Types of SQL Statements	18
<b>Figure 2.4:</b> Example of XML	20
<b>Figure 2.5:</b> Example of JSON in an array	20
<b>Figure 3.1:</b> Running a VBScript Code	23
<b>Figure 3.2:</b> Data Types using VbSript Code	23
<b>Figure 3.3:</b> Running a For loop in VBScript	23
<b>Figure 3.4:</b> Running a function in VBScript	24
<b>Figure 3.5:</b> Using array in VBScript	24
<b>Figure 3.6:</b> Problem Statement 1	25
<b>Figure 3.7:</b> Code 1	26
<b>Figure 3.8:</b> Output 1.1	26
<b>Figure 3.9:</b> Output 1.2	26
<b>Figure 3.10:</b> Problem Statement 2	26
<b>Figure 3.11:</b> Code 2	27
<b>Figure 3.12:</b> Output 2.1	27
<b>Figure 3.13:</b> Problem Statement 3	28
<b>Figure 3.14:</b> Code 3	29
<b>Figure 3.15:</b> Output 3.1	29
<b>Figure 3.16:</b> Output 3.2	29
<b>Figure 3.17:</b> Output 3.3	29



<b>Figure 3.18:</b> Output 3.4	29
<b>Figure 3.19:</b> Output 3.5	30
<b>Figure 3.20:</b> Output 3.6	30
<b>Figure 3.21:</b> Problem Statement 4	31
<b>Figure 3.22:</b> Code 4	32
<b>Figure 3.23:</b> Output 4.1	33
<b>Figure 3.24:</b> Output 4.2	33
<b>Figure 3.25:</b> Problem Statement 5	34
<b>Figure 3.26:</b> Code 5	34
<b>Figure 3.27:</b> Output 5	35
<b>Figure 4.1:</b> Micro Focus UFT Testing	36
<b>Figure 4.2:</b> VBScript Code in Expert View	37
<b>Figure 4.3</b> Problem Statement 1	38
<b>Figure 4.4:</b> Select Web	39
<b>Figure 4.5:</b> Record	40
<b>Figure 4.6:</b> Web Objects	40
<b>Figure 4.7:</b> Deselect Web	41
<b>Figure 4.8:</b> Record and Run	42
<b>Figure 4.9:</b> Disable mobile Settings	42
<b>Figure 4.10:</b> Object Identification	43

## **ABSTRACT**

Automation frameworks are regarded crucial in the software testing area, especially when you are interested in automation testing. We'll look at the many sorts of automation frameworks in this report. In technical terms, an automation framework is a collection of principles that includes coding standards, app code and data folder organization, test results storage, external resource access, and shared object repositories, among other things.

Standard Automation Framework Requirements:

Code should be reusable, understandable, and easy to maintain.

The automation framework must include data parameterization.

Test data should be kept separate from the code, and the code should be reusable for various input data sets.

Test results should be recorded in a separate log file and be simple to comprehend.

The automation framework must have an appropriate error handling method.

Additionally, the use of recovery scenarios should be reduced to a bare minimum, as they take additional processing time and resources.

The automation framework should be developed in such a way that if one test script fails, Unified Functional Testing will continue to run.

# CHAPTER 1

## INTRODUCTION

### 1.1 About the company

Cognizant is a leading provider of information technology, advisory, and business process outsourcing services, committed to assisting the world's most successful companies in growing their businesses. The company, headquartered in Teaneck, New Jersey, has over 100 production and distribution centres worldwide and more than 217,700 employees. It is one of the world's best-performing and fastest-growing businesses.

Cognizant serves top clients in a variety of industries, (such as banking, financial services, and insurance, healthcare, life sciences, manufacturing and logistics, retail and consumer goods, and information, media, and entertainment).

Cognizant provides creative solutions through consultancy and IT services with a vertical market emphasis.

### 1.2 Software Development Life Cycle

#### 1.2.1 What is SDLC?

Software Development Life Cycle[1] also called SDLC is a sequence of steps followed by the software industry to complete a software project. Software development life cycle describes a detailed plan about developing, maintaining, replacing, and altering or enhancing a specific software. The cycle, in its every stage, defines a method for improving software quality.

#### 1.2.2 SDLC Phases

The process of SDLC is essentially divided into the seven stages:



**Figure 1.1:** Phases in Software development life cycle

### **Phase 1: Requirement Collection and Analysis[2]:**

The first stage of the Software-Development Life-Cycle is collecting and analysing the requirements. Before any project is started, a meeting is held, with industry domain experts as well as stakeholders. Here, the quality assurance requirements are planned for, and the risks involved are identified.

### **Phase 2: Feasibility study[3]:**

After we know the customer's product requirement, we then define the software requirements which will be used throughout the production of the product. It encompasses everything that must be designed or developed till completion of the project. Five main feasibility checks include economic allowance, legal checks, feasibility of each operation, technical and schedule.

### **Phase 3: Design[4]:**

The design documents, specifically system design and software design are prepared in this phase in accordance with the specification requirement software (SRS). This contributes to the overall product. Design phase is an integral part of this life-cycle, as it serves as an input for the model's upcoming phase. During this phase, two types of designs i.e. High Level Design(HLD) and Low-Level Design ( LLD) documents are created.

### **Phase 4: Coding[5]:**

In this phase, developers begin to write code for building the system step by step in a programming language as agreed in design phase. It is here that the tasks are broken down into units or modules and assigned to different developers..

### **Phase 5: Testing[6]:**

After developing the software, it is passed to the testing team where it is tested thoroughly. The overall functionality of the software is tested step by step. It is made sure that the entire application functions as per quality requirements.

If the Quality Assurance Engineer finds a bug in the application, it is immediately reported to the development team who in turn fix the bug and then send the software back to QA for one more round of testing. The testing team continues to test the software till they find it bug-free and stable, making sure that it fits the requirements as defined in its previous stages.

#### **Phase 6: Installation/Deployment[7]:**

Once testing phase is over and the system is bug-free, the deployment process begins. The final software is now released and tested for issues based on the feedback from customer, stakeholder or manager.

#### **Phase 7: Maintenance[8]:**

After the software is deployed and customers start to use the system, the maintenance of the software continues to be a part of the software development life cycle.

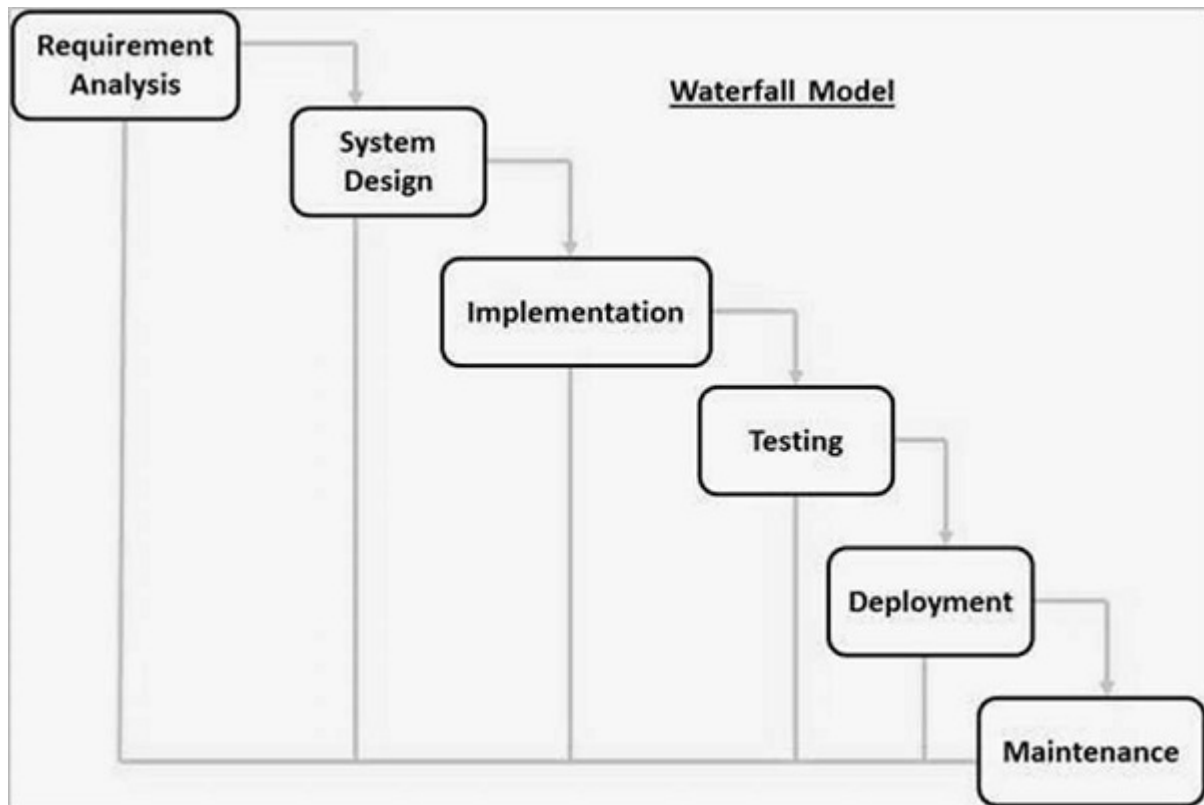
This is done as:

- Error fixing - Bugs reported by the ser
- are fixed by the development team.
- Upgrade - the software application is upgraded to a newer version every once in a while.
- Enhancement -new functionalities are added to an existing software to keep up with growing technology.

### **1.2.3 Software Development Life-Cycle Models**

#### **Waterfall Model[7]:**

Waterfall model is one of the most widely used model for Software development. It is also known as Linear Sequential Life Cycle. The entire process is divided into phases of SDLC. The outcome after completion of one phase serves as the input for the next phase. Thus the first phase needs to be completed before moving on to the next step. There is no overlap between the phases.



**Figure 1.2:** Phases of Waterfall Model

All the phases in the waterfall model are merged on top of each other and the progress is seen as a waterfall structure. It allows for departmentalisation and control. Each phase proceeds in a strict manner following a schedule for each development stage.

Following are some advantages of the Waterfall Model:

- It's simple to use and understand
- Every phase is processed and completed as scheduled..
- It works well for projects which are smaller and have a clear set of requirements.
- Because of it's rigid nature, it is not difficult to manage. Each phase has it's own deliverables.
- Every stage is clearly defined.
- Milestones are easy to achieve.

Once the developer has finished the code, and the application is ready for testing, it becomes extremely difficult to go to the previous phases and bring any change. Thus, it is not suited for

projects where requirements are likely to change or for projects which are long and complex, as not working software is produced until the late phase of life-cycle.

### V-Model in SDLC

The V-Model is named so because in this model, the execution process happen a sequential V-like structure. It is also called Verification and Validation Model. The V-model plans the testing and development in parallel. i.e. after every development phase, a testing phase is carried out which is associated with the development directly. Thus, verification phases of SDLC are carried along with validation phases. V-Model joins during the Coding phase.

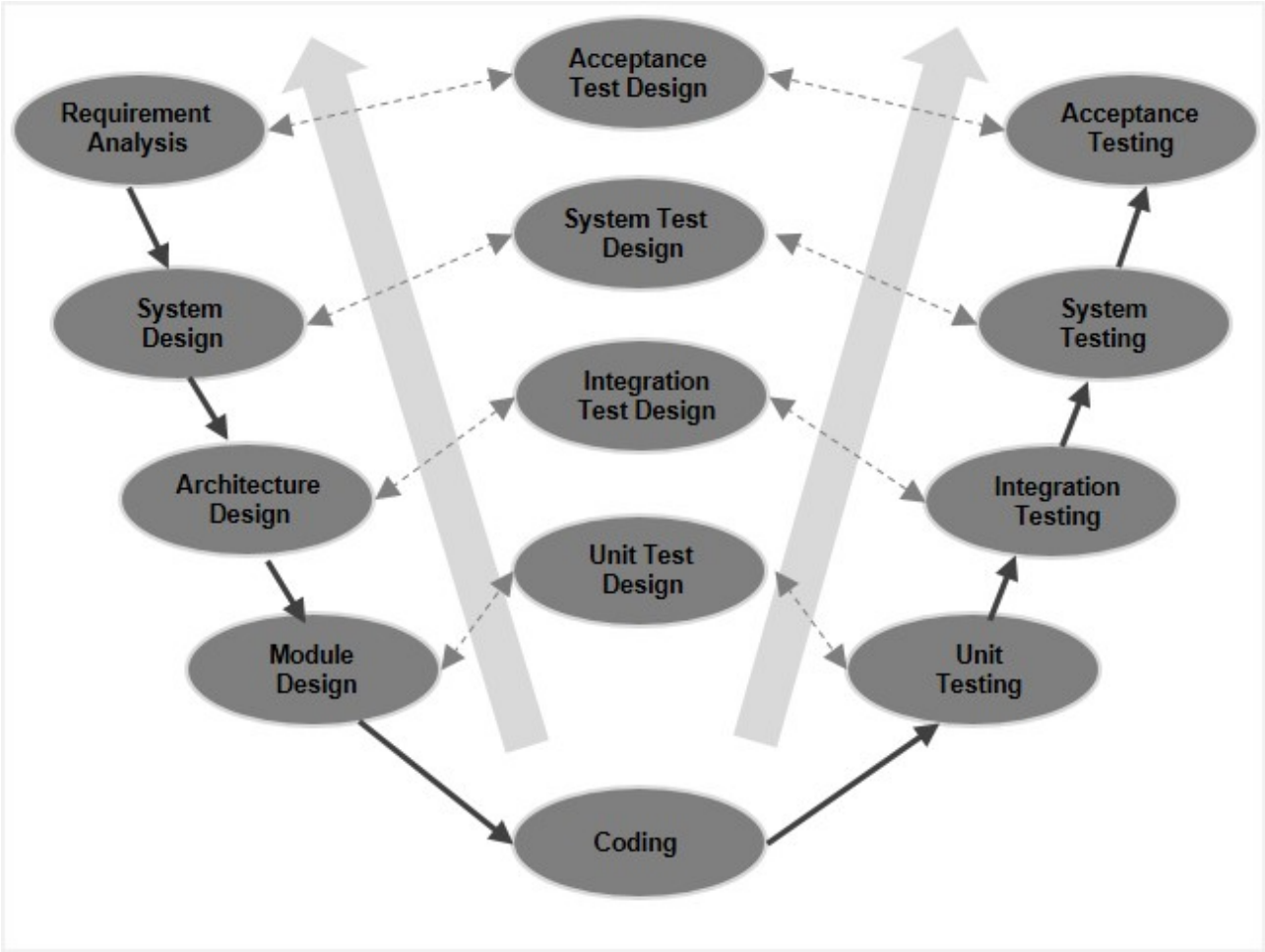


Figure 1.3: V-Model

Following are the advantages of the V-Model:-

- Phases in this model are carried out in a sequential manner.
- It works well for projects which are smaller and have a clear set of requirements.
- Since the model is rigid,, it is easy to manage.
- Every phase has its own deadlines and review process.

It is not suited for projects which do not have clear defined requirements (likely to change in the future) or for projects which are long and complex, as no prototype of the software is produced till the late phase of the product’s life cycle.

### Agile Model

Agile methodology is a practise that encourages continuous interaction between development steps and testing steps throughout the project's SDLC. This method successfully divides the project into small and incremental builds. These builds are then developed and delivered in iterations, with each iteration lasting between 1 to 3 weeks. This method is a union of iterative as well as incremental models, focusing on how adaptable the model is and satisfaction of the customer by delivery of working applications well in advance.

Agile methodology believes in handling every project differently to suit the customer requirements. The tasks here are divided into small time frames for every feature to be added to the software.

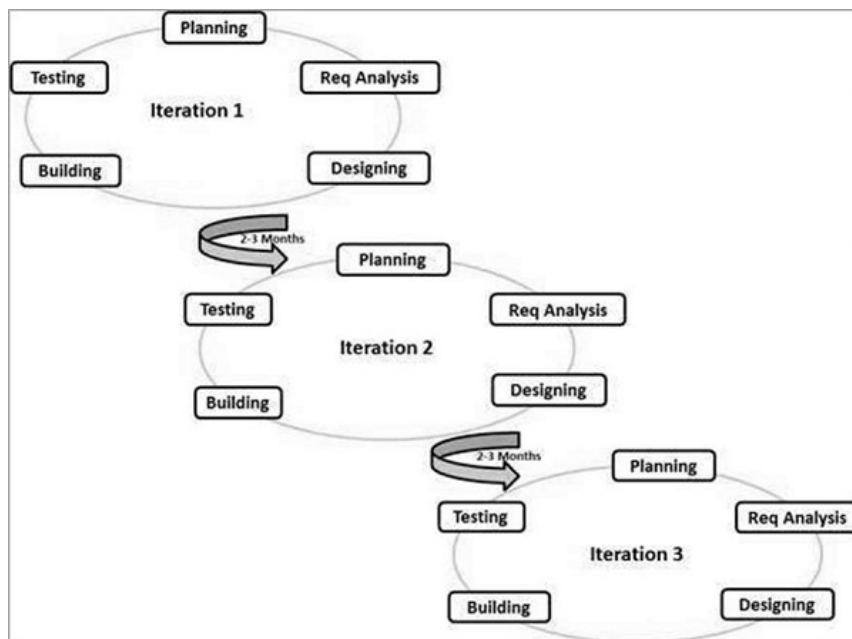


Figure 1.4: Agile Model



This model follows an iterative approach and delivers a working software after every iteration, Thus getting additional inputs from the customer. Each build has additional features and the final build holds the entire product.

Following are the advantages of Agile Model:

- The approach is practical to develop software.
- The functionality of the developed product can be demonstrated rapidly.
- Not many resources are required.
- It is also appropriate for projects which have changing requirements
- Prototypes of working solutions can also be delivered.
- It is also considered for environments that change steadily.
- It doesn;t require much planning.
- Developers are open to put new ideas to use.

## **1.4 Functional Testing[9]**

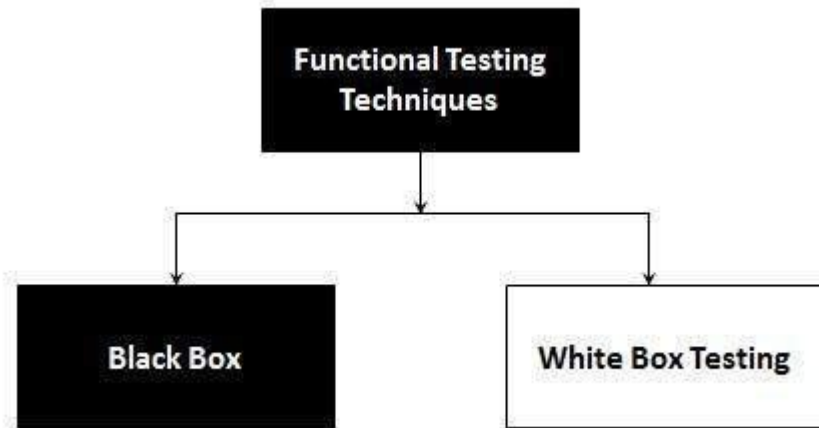
Functional Testing is an approach used for testing the features and functions of the software. It covers testing of all conditions including failure paths as well as boundary cases. Here, the tester seeks answers to questions like- if the user can successfully login to a system on entering the correct credentials? If an error message is prompted on entering an incorrect card number? And if the customer is added to the database record on clicking “add a customer”.

### **1.4.1 Functional Testing Techniques**

At the time of functional testing, testers verify the features of the application against the user specifications. Functional testing mainly checks the functionalities of the software system. It includes:-

1. Testing the main functions of the application.
2. Testing the usability of the system. It checks if the user can navigate between screens without any issues or delay.

3. It checks if the system is accessible for the user (i.e. easy to read and write).
4. It also checks for conditions that might result in error and if suitable messages are displayed for them.



**Figure 1.5:** Functional Testing Techniques

**BLACK BOX TESTING:** Testing is done on the basis of customer requirements and system functionality. Here, the software is tested without looking at an internal structure of the software.

**WHITE BOX TESTING:** White Box Testing is predicated on understanding the internal structure of the code. For this type of testing, good understanding of the software and code is required.

The other major Functional Testing techniques include:

- **UNIT OR COMPONENT TESTING:** It is the testing of a single software component or module. It is usually performed by the programmer rather than a tester because it requires understanding of design and code.
- **INTEGRATION TESTING:** It is the process of testing the integrated modules so as to make sure that the software functionality is achieved after integration. Once Integration testing is successful, the code modules are combined and concentrate on the interaction of

components and systems. It is particularly important for client-servers and systems which are distributed.

- **SYSTEM TESTING:** In this type of testing the entire system is tested in accordance with the requirements. It is similar to black-box testing but here all the requirement specifications are checked covering all system components.
- **ACCEPTANCE TESTING:** This type of testing is carried out by the client to determine if the system's end-to-end flow meets business requirements and the needs of the end-user. Once the client is satisfied with the features and functionalities, it accepts the software. After this step, the software goes into its production stage. This is also referred to as User Acceptance Testing (UAT)
- **REGRESSION TESTING:** It is the process of testing an application after changes in any functionality or code. Automation Testing Tools are usually preferred for this types of testing.
- **BETA/USABILITY TESTING:** The product is exposed to real customers in a production-like environment, and they test it. This is derived from the user's comfort, and feedback is obtained. This is comparable to User Acceptance testing.

### **1.4.2 Functional Testing Goals**

Goals in functional testing can be defined as the intended outcomes of the software testing process. The primary goal of functional testing is to determine how closely the feature adheres to the specifications. To make things easier to understand, we can divide functional testing goals into two categories. They are:

1. Validation Testing:
  - This type of testing is carried out in order to show the client whether or not the defined requirements are met.
  - If the test is successful, it demonstrates that the system functions as intended.
2. Defect Testing::
  - To identify flaws in the user interface, error messages, and text handling.

- When the functionality does not work as expected, a successful test should reveal the flaws.

### **1.4.3 Functional Testing Tools**

- Unified Functional Testing (UFT)[12] is a commercially used functional testing tool that includes a comprehensive set of features for testing APIs, web services and GUIs of Websites, Desktop and Mobile based applications across different platforms. The software includes text and image-based recognition of objects, allows reusability of test components and automated documentation.
- TestLink is a web-based management system for testing that aids in assuring the software quality. Teamtest is in charge of its development and upkeep. The platform supports test cases, test suites, test plans, test projects, and user management, in addition to a variety of reports and statistics.
- JIRA is a project management tool that is used to track issues and bugs. It is widely used for all types of testing as an issue-tracking tool. This tutorial introduces readers to the basic features, functionality, and benefits of JIRA. This tutorial will show users how to use this tool to track and report bugs in various applications.

### **1.4.4 Functional Testing Process**

Testing process refers to the steps we go through in order to test our software.

#### **1. TEST PLAN AND STRATEGY:**

A test plan is required for every project. These artefacts describe the project's testing scope:

- Identify the features that should be the project's main focus
- Methodology for testing
- Processes to be followed
- Key processes to follow
- Tools to find and correct defects
- Test environment setup
- Tackling the risks and dependencies
- Schedule

## 2. TEST ANALYSIS AND DESIGN:

After test planning, we analyze and design the test cases and test scenarios. Following are the tasks involved in Test Design:

- Analyse the test basis.
- Figure of test conditions.
- Create the tests.
- Assess the needs and system's testability.
- Plan the setup for test environment set-up and determine the infrastructure and methods that are required.

## 3. TEST IMPLEMENTATION AND EXECUTION:

Test implementation has the following major task:

- Developing test cases and creating test data for those tests.
- Create test conditions from the test cases for efficient test execution.
- Setup the test environment.

Test execution has the following major task:

- Execute test suites and individual test cases following the test procedures.
- In order to confirm a fix, re-run the tests that previously failed. Confirmation testing or re-testing is the term for this.
- To record the versions and identities of the programme under test, as well as the results of the test execution.
- To Compare actual results with projected results.
- It reports disparities as Incidents when there are disparities between actual and expected results.

## 4. EVALUATING EXIT CRITERIA AND REPORTING:

We need to create the criteria for every level of test against which we will conduct "enough testing" based on the project's risk assessment. Exit criteria are a type of criterion that varies from project to project.

When exit criteria are used, they are referred to as "exit criteria.":

- The maximum number of test cases are run with a specific pass rate.
- The bug rate drops down below threshold level.
- When the deadlines are met.

The following are the primary tasks involved in evaluating exit criteria:

- To compare logs of test to the exit criteria set forth in the planning of tests.
- Find whether additional required tests re needed or whether the given exit criteria must be altered.
- To prepare a stakeholder summary report on the test results.

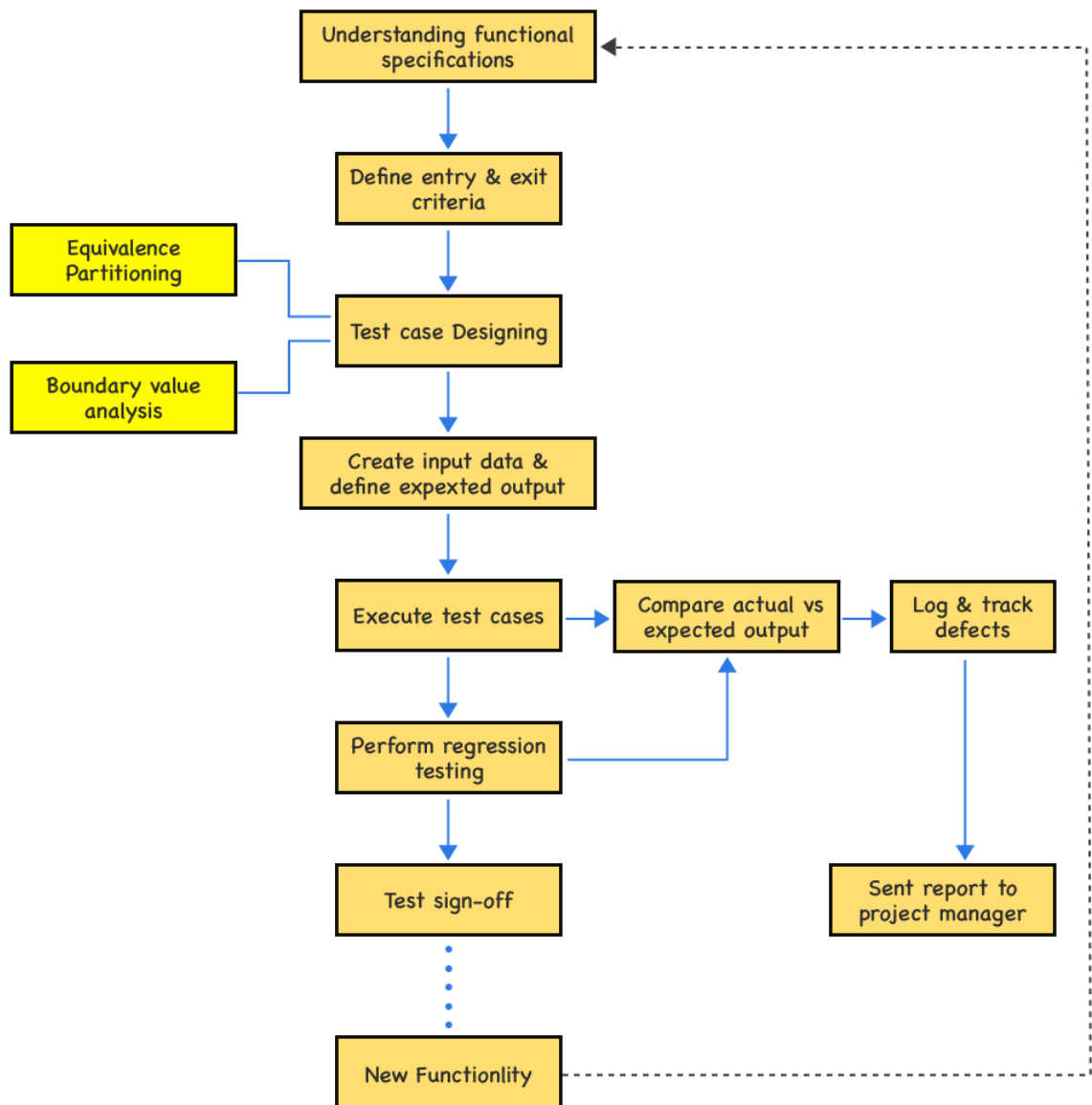
#### 5. TEST CLOSURE ACTIVITIES:

When software is delivered, test closure activities are completed. The testing might be halted for other reasons, such as:

- When you've gathered all of the information you'll need for the test.
- When a project is called off.
- When a specific goal is met.
- When there is a maintenance release or an update.

The following are the major tasks associated with test closure activities:

- Ensure that all incident reports have been resolved and that all planned deliverables have been delivered.
- Complete and archive testware such as scripts, test environments, and so on for future use.
- Turn over the testware to the maintenance team. They will provide software support.



**Figure 1.6:**Functional Testing Process

### 1.4.5 Conclusion

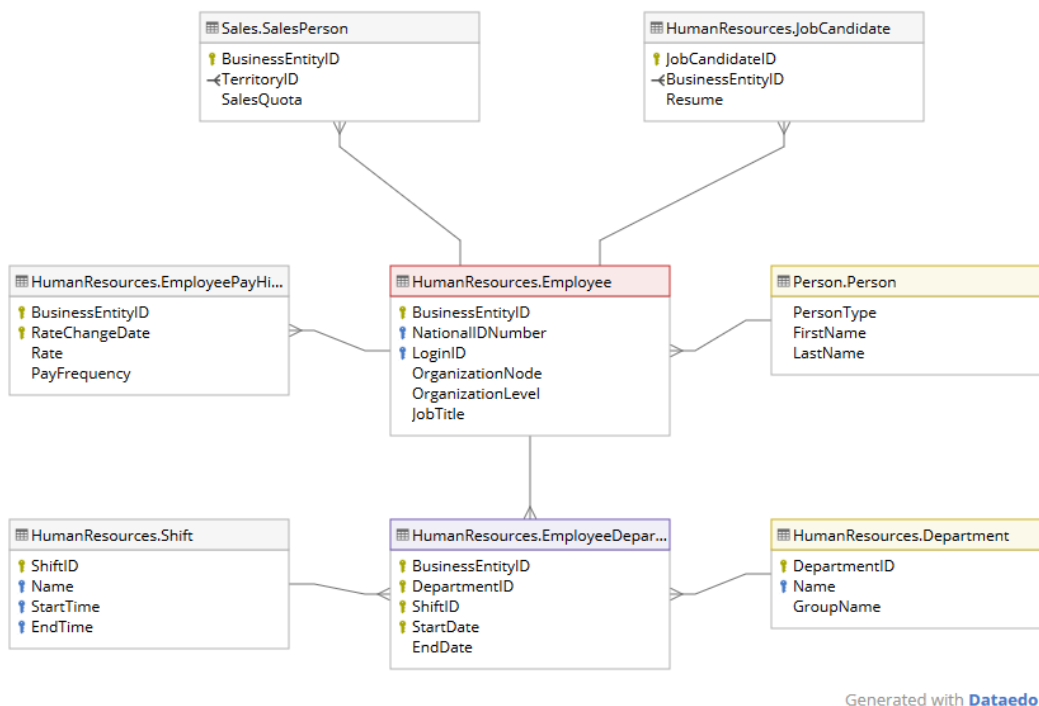
Functional testing should be at the heart of all testing methods. Functional checks are important for ensuring that a product or feature can work properly for users. Creating a robust suite of practical checks aids in ensuring that the product and features are consistently of high quality.

## CHAPTER 2

### DATASOURCE(SQL, XML, JSON)

#### 2.1 Overview

A database is a logically organised collection of interconnected data that aids in retrieving , inserting and deleting data from a database. A database organises the data in the form of tables, views, schemas, reports, and so on. A database management system is usually in charge of a database. The data or information along with the DBMS, and the applications that go with them are referred to as a database system, which is commonly abbreviated to just database.



**Figure 2.1:** Example of a Database

#### 2.2 Types of Database

Database can be divided into different types. The are:

- Relational databases:



Under this database tables are used for defining relationships in databases. Also called Relational DBMS, it is the most extensively sought-after database management system across the global marketplace. Some examples of RDBMS systems are Oracle, and Microsoft SQL Server databases.

- Object-oriented databases:

In this type of computer database, data of all kinds can be stored. The information inside it is held in the form of objects. To define what to do with the data, characteristics as well as methods of the objects stored in database are responsible.

- Centralized database:

It is a centralized site where data may be accessed by people from various backgrounds. Application processes that allow users to access data from a remote place are stored in this form of computer database.

- Cloud databases:

Cloud databases are databases that have been customized or created for specific use in a virtual surrounding. Use of a cloud database comes with a variety of benefits, including covering the bandwidth and storage cost. On-demand scalability and excellent availability are some of its other features.

- NoSQL databases:

For massive sets of distributed data, a NoSQL database is employed. There are a few issues with large data performance that can be solved by relational databases in an efficient manner. Evaluation of huge amounts of unstructured data can be done using this database.

- Hierarchical:

This form of DBMS stores data in a "parent-child" connection. It has a tree-like structure, where nodes signify records and branches signify to fields. The Windows XP registry is an example of a hierarchical database.

- Network DBMS:

Many-to-many relationships are supported by this database management system. It frequently leads to the creation of complicated database structures. The network model is implemented by RDM Server, a database management system.

- Distributed databases:

This type of database includes information gathered by local computers as well as contributions from common databases. The data in this database system isn't all in one place.

## **2.3 Importance of DataSource in Testing**

A Quality assurance tester is required have the following database skills:

- Ability to recognise various types of databases.
- The ability to work on databases via various SQL or other languages.
- Capability to write basic SQL statements as well as more complex join queries.
- The ability to decipher more complex queries.

## **2.4 Structured Query Language[10]**

Structured Query Language (SQL) is a programming language to manage databases. It was created with web apps in mind and can operate on any platform. As new and variable requirements were found with the internet, this slowly became the most chosen platform by most web developers and applications based on the web. E-commerce companies choose MySQL as they deal with a huge number of money transfers and it can be used to deal with an enormous amount of queries and handle lots of transactions. SQL's main feature is its on-demand flexibility. The SQL language is the industry standard for dealing with Relational Databases. Inserting, searching, updating, and deleting database records are all possible with SQL. SQL may do a variety of different tasks, including database optimization and maintenance.

### **2.4.1 Relational Database**

A relational database divides data into tables. It has a structure similar to that of an Excel spreadsheet. The tables in a database can be compared to individual sheets within a single Excel file, which itself is the database. Data is organised into columns and rows in each table. Each column contains a specific type of value.



**Figure 2.2:** Example of Relational Database

## 2.4.2 SQL Applications

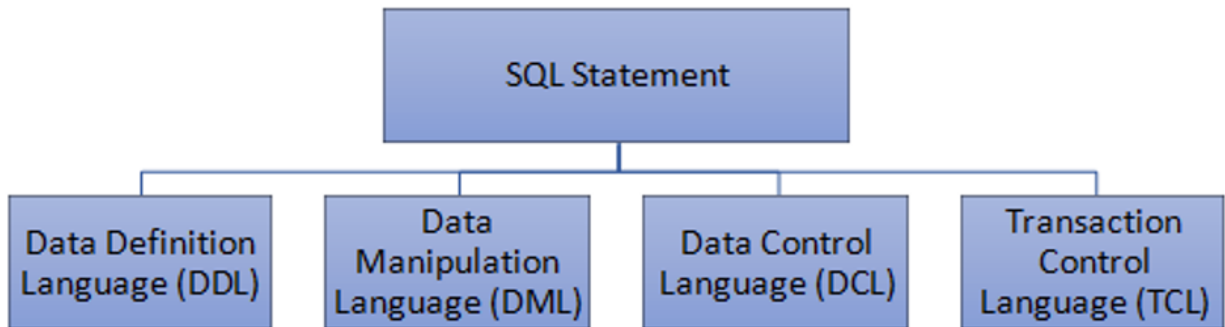
SQL is one of the most widely used query languages in databases. Some of its applications are:

- It enables users to access data in an RDBMS system.
- It enables you to describe the data.
- It enables you to define and manage data in a database.
- SQL allows you to create and delete databases and tables.
- SQL allows you to use a database function, build a view, and run a stored procedure.
- Permissions can be configured on tables, processes, and views.

## 2.3.3 Types of SQL Statements

There are four types of SQL Statements. They are:

1. Data Definition Language(DDL): It is used in database to describe how data should be stored in a database. Most commonly DDL statements are CREATE, ALTER, DELETE, TRUNCATE, COMMENT AND RENAME.
2. Data Manipulation Language(DML) deals with data manipulation like storing, modifying, retrieving, deleting and updating data in a database. The most commonly used SQL statements for this are SELECT, UPDATE, INSERT, DELETE, etc.
3. Data Control Language(DCL) defines the control over the data in the database. Commonly used SQL statements are: GRANT and REVOKE. They help to grant specific tasks and permissions to specific users.
4. Transactional Control Language(TCL) is to manage the transactions that occur in the database. Common statements for this include: COMMIT and ROLLBACK.



**Figure 2.3:** Types of SQL Statements

Other commonly used SQL statements:

INNER JOIN: It Selects the records that have the matching records in two tables.

LIKE: It is used to find if the patterns match using wildcards.

ORDER BY: It is used to sort the records in specific order.

DISTINCT: It selects different values from the defined field.

IN: It helps to find whether the value is a part of the list or not.

BETWEEN: It helps to select the values within a defined range.

GROUP BY: It is used to put together the results..=

## **2.5 Extensible Markup Language(XML)[11]**

XML refers to Extensible Markup Language. It encodes documents in such a way that they are both human and machine readable. XML's design goals are focused on simplicity, generality, and Internet usability. It is a textual data format with strong Unicode support for various human languages. XML can simplify the generation of HTML documents for huge websites by working behind the scenes .XML is a format for exchanging data between companies and systems. Databases can be offloaded and reloaded using XML.

- XML is an abbreviation for extensible Markup Language.
- XML, like HTML, is a markup language.
- XML is intended to be used to store and transport data.
- XML is intended to be self-explanatory.

### **2.5.1 Differences between XML and HTML**

- XML is intended to transport data, with emphasis on the type of data.
- HTML is intended to display data with a focus on how the data appears.
- Unlike HTML tags, XML tags are not predefined.
- HTML is a markup language, whereas XML is a framework for the definition of markup languages.
- XML is dynamic whereas HTML is static.

### **2.5.2 Advantages of using XML**

- XML can be used to accurately and unambiguously describe and identify information, allowing computers to be programmed to 'understand' your data.
- XML is extensible, which means you can construct your own self-descriptive tags, or language, to meet your needs.
- XML is a format for storing and transmitting data.

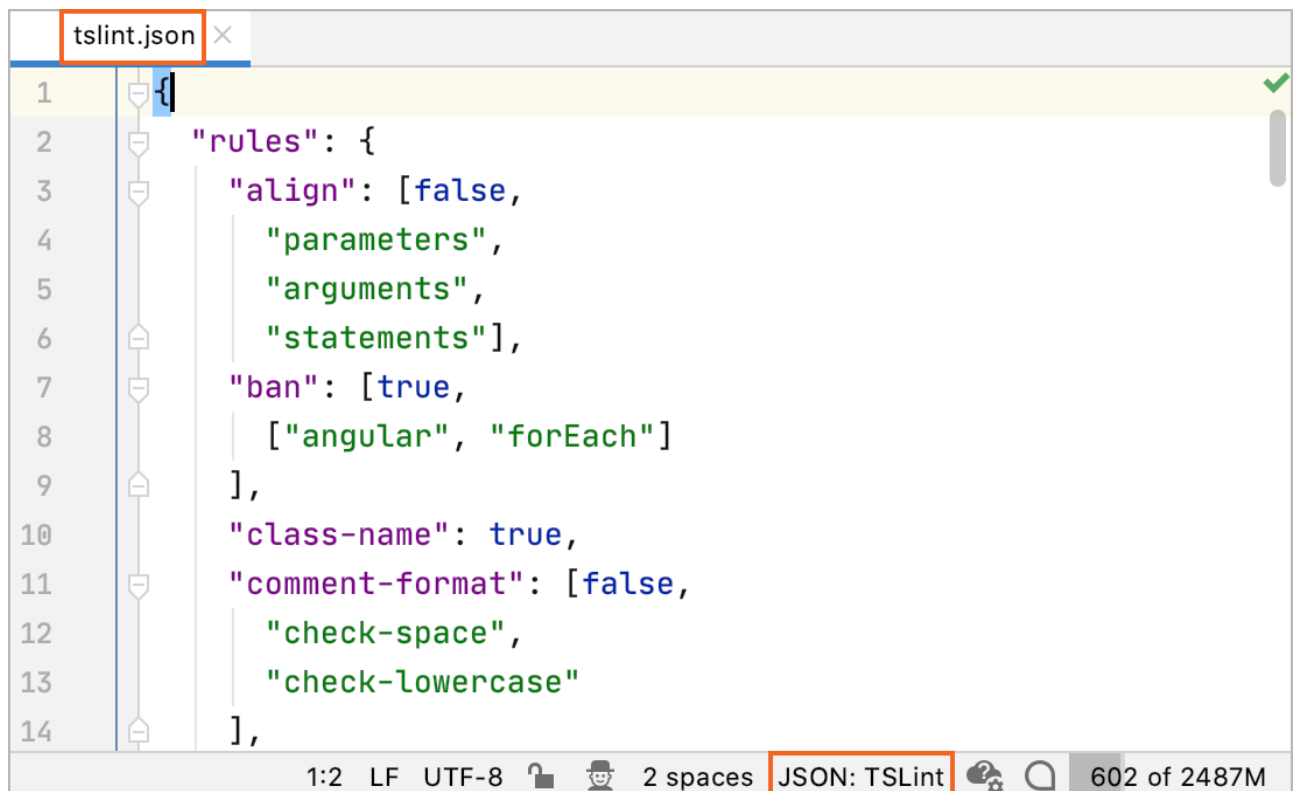
- XML provides a common syntax for systems that allows information to be exchanged between different applications.
- XML is a publicly available standard that was developed by the World Wide Web Consortium and is available as an open standard.

```
<note>
  <to>Raj</to>
  <from>Ravi</from>
  <heading>Reminder</heading>
  <body>Meeting at 8am.</body>
</note>
```

Figure 2.4: Example of XML

## 2.6 JavaScript Object Notation[12] (JSON)

JSON, or JavaScript Object Notation, is a data structuring format. Web applications use this data format to communicate with one another. JSON is only a text format. It is used to send information to and from a server. It can be used by any programming language.



```
tslint.json
1 {
2   "rules": {
3     "align": [false,
4       "parameters",
5       "arguments",
6       "statements"],
7     "ban": [true,
8       ["angular", "forEach"]
9     ],
10    "class-name": true,
11    "comment-format": [false,
12      "check-space",
13      "check-lowercase"
14  ],
15  }
```

**Figure 2.5:** Example of JSON in an array

### **2.6.1 Advantages of using JSON**

- It's utilized in the development of JavaScript-based applications, such as browser extensions and webpages.
- JSON is a serialization and transmission format for structured data over a network connection.
- It's mostly used to send data from a server to web apps.
- JSON is the format used by web services and APIs to expose public data.
- It's compatible with today's programming languages.

# CHAPTER 3

## VBSRIPT

### 3.1 Overview

VBScript stands for Visual Basic Script Language. It is a scripting language, VBScript is a subset of Visual Basic for Applications (VBA) by Microsoft. It is a combination of Visual Basic and JavaScript, even though it is much lighter than Visual Basic. It is used to develop dynamic pages and software testing using UFT.

### 3.2 Features of VBScript

- VBScript is a simple and efficient scripting language.
- VBScript is case insensitive. It is easy to implement and has the syntax is also easy and simple to learn and use.
- VBScript is an object-based scripting language, not an Object-Oriented Programming language, like C++ or Java.
- So as to access the elements of the environment in which it is running, it employs the Component Object Model.
- Only in a Host Environment, such as Internet Information Services (IIS), Internet explorer or Windows Scripting Host, can VBScript be successfully executed (WSH).

### 3.3 History and Uses

Microsoft released the first version of VBScript in 1996, and it was version 1.0. Now the recent stable version is 5.8, that is included with Internet Explorer 8 and Windows 7. The places where VBScript can be used are numerous and are not limited to the ones listed below.

- Quick Test Professional, abbreviated as QTP, is a popular automation testing application that uses VBScript as a scripting language.
- Windows Scripting Host is a tool for automating the Windows desktop that is mostly used by Windows System Administrators.



- Active Server Pages (ASP) is a server-side scripting environment that employs VBScript or Java Script to create dynamic webpages.
- In Microsoft Internet Explorer, VBScript is utilized for client-side scripting.
- Microsoft Outlook Forms usually runs on VBScript; however, the application-level programming relies on VBA.

### 3.4. Basic programming in VBScript[13]

When used on the client-side in Internet Explorer, it functions similarly to JavaScript. The code is included in HTML pages, thus explicitly communicating with the DOM to do things that HTML alone cannot. This happens on the front-end. In terms of backend, it processes web-pages on the server-side, mostly with active server pages. To run VBScript, the ASP form library and the calls .dll files. These sections are embedded within percent and percent > that switch the contexts on an ASP list.[14]

#### 3.4.1 Running and Debugging VBScript code

VbScript files are saved with .vbs extension. Windows employs cscript.exe, the command-line variant of Windows Script Host, to execute all .vbs files by simply passing the path to the vbs file. To debug vbscript code, use the /x flag while using cscript, and ensure that the Just-In-Time debugger, TypeScript 3.9 SDK, and JavaScript and TypeScript are installed.

```
C:\Users\John>cscript hello.vbs
Microsoft (R) Windows Script Host Version 5.812
Copyright (C) Microsoft Corporation. All rights reserved.

Hello World!
```

Figure 3.1:Running a VBScript Code

#### 3.4.2 Variable Declaration and Data Types

```

Dim varInteger, varString, varDouble

varInteger = 20
varString = "30"
varDouble = 40.1

' Display Data Type of a variable
Wscript.Echo varType(varInteger) & ": " &
typeName(varInteger)
Wscript.Echo varType(varString) & ": " &
typeName(varString)
Wscript.Echo varType(varDouble) & ": " &
typeName(varDouble)

' Output:
' 2: Integer
' 8: String
' 5: Double

```

**Figure 3.2:** Data Types using VbSript Code

### 3.4.3 Loops and Conditions

```

For i = 0 To 10 Step 2
    Wscript.Echo i
    If i = 6 Then
        Wscript.Echo "I am done, Bye!"
        Exit For
    End If
Next

' Output:
' 0
' 2
' 4
' 6
' I am done, Bye!

```

**Figure 3.3:**Running For loop in VBScript

### 3.4.4 Functions

```

Function congratsSomeone(name)
    ' Specify a return value:
    congratsSomeone = "Congratulations " & name & "!"
End Function

Wscript.Echo congratsSomeone("Bill")
Wscript.Echo congratsSomeone("Gates")

' Output:
' Congratulations Bill!
' Congratulations Gates!

```

**Figure 3.4:**Running a function in VBScript Code

### 3.4.5 Arrays

```

Dim arr(2) ' Actually with the size of 3 items.

arr(0) = "John"
arr(1) = 15
arr(2) = True

' It's easy to concatenate Strings, Integers and even
Booleans!
Wscript.Echo arr(0) & arr(1) & " Is " & arr(2)

' Output: John15 Is True

```

**Figure 3.1:**Using array in VBScript Code

## 3.5 Advantages of Using VBScript

- It's used to automate administrative tasks in Windows (but today, it is mostly replaced by Windows PowerShell).
- It is the scripting language used by Quick Test Professional (QTP), a popular test automation tool.
- It is the scripting language used internally for embedded applications.
- VBScript is also supported by DBMS InterSystem Cache.
- It provides extensive support for error management, subroutines, and so on.
- It is used as a macro language in Outlook 97 instead of VBA.

### 3.6 Work Done:

Example 1:

#### Problem Statement:

Write a procedure, which will accept an integer value as input and displays the factorial for the given number.

**Figure 3.6** Problem statement 1

Solution1: CODE:

```
option explicit

dim n,f,m
n= InputBox("Enter the number")
m=n
f=1

Do while n>0
f=f*n
n=n-1
Loop
msgbox "Factorial of" & m & " is "& f
```

**Figure 3.7** Code 1

OUTPUT:



**Figure 3.8** Output 1.1

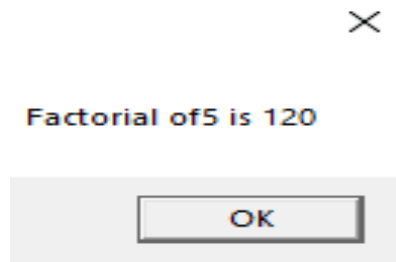


Figure3.9 Output1.2

Example 2:

Problem Statement:

Write a procedure which will accept a number from 1 to 12 as input parameter and output should be the month name. For example if the input is 1, then the output should be "January", if the input is 2 then the output should be "February" and so on.

Figure3.10 Problem statement 2

Solution 2: CODE

```
dim month
month=InputBox("Enter month number")
Select case month
case 1
MsgBox "January"
case 2
MsgBox "February"
case 3
MsgBox "March"
case 4
MsgBox "April"
case 5
MsgBox "May"
case 6
MsgBox "June"
case 7
MsgBox "July"
case 8
MsgBox "August"
case 9
MsgBox "September"
case 10
MsgBox "October"
case 11
MsgBox "November"
case 12
MsgBox "December"
End select
```

Figure 3.11 Code 2

OUTPUT :



Figure 3.12 Output 2.1

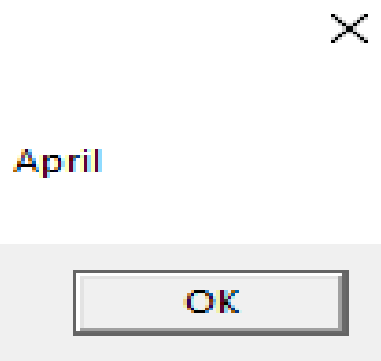


Figure3.13 Output 2.2

Example 3:

**Problem Statement:**

Write a program to handle Array out of Index error using error handling techniques

Figure 3.9 Problem statement 3

Solution 3: CODE

```

a=Array("a","b","c","d","e")
for each element in a
  msgbox element
Next
On Error Resume Next
MsgBox a(5)="f"
Err.Raise 101
MsgBox "Error " & err.number
err.clear
msgbox "Error " & Err.Number & " END "

```

Figure 3.13 Code 3

OUTPUT:

×

a

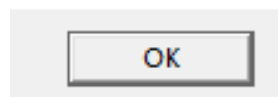


Figure 3.14 Output 3.1

×

b

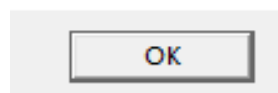


Figure 3.15 Output 3.2

×

c

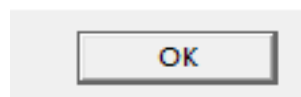
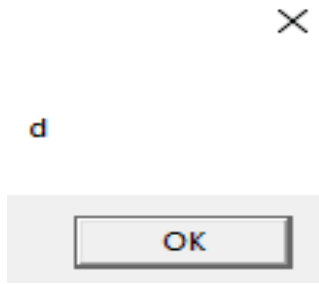
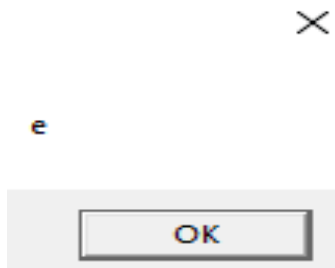


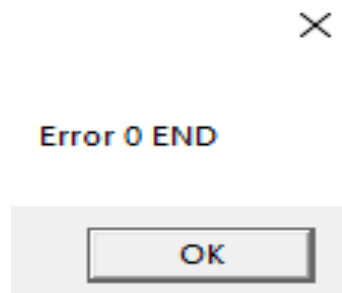
Figure 3.16 Output 3.3



**Figure 3.17** Output 3.4



**Figure 3.18** Output 3.5



**Figure 3.19** Output 3.6



**Figure 3.20** Output 3.7



Example 4:

**Problem Statement:**

Write a Program to replace all the numbers with „\*” in a given string using regular expression Examples:

Input String : " Hello 123 , How are you 1" Output String : " Hello \*\*\* , How are you \*"

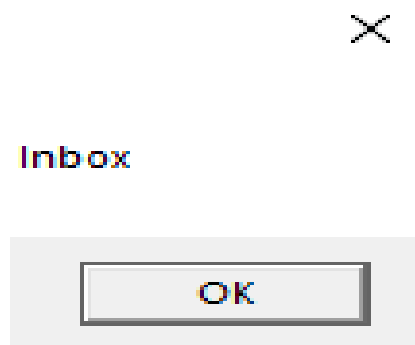
**Figure 3.21** Problem statement 4

Solution 4: CODE

```
Dim oRE, oMatches
Set oRE = New RegExp
oRE.Pattern = "Inbox\([0-9]\)"
oRE.Global = True
oRE.IgnoreCase = True
Set oMatches = oRE.Execute("two times Inbox(0) three equals Inbox(3) three times two")
For Each oMatch In oMatches
    If (oMatch="Inbox(0)") Then
        MsgBox("Inbox")
    Else
        MsgBox "Match: " & oMatch.Value & " At: " & CStr(oMatch.FirstIndex + 1)
    End If
Next
```

**Figure 3.22** Code 4

OUTPUT:



**Figure 3.23** Output 4.1



Match: Inbox(3) At: 33



Figure 3.24 Output 4.2

Example 5:

Figure 3.25 Problem statement 5

**Problem Statement:**

Write a function which will create a directory in a particular path if the directory does not exist.

Eg: C:\test.

Program should check whether "Test" exists and if it does not exist it should create a folder.

Solution 5: CODE

```
Dim ofso, startfolder
startfolder="D:\abcd"
set ofso=CreateObject("Scripting.FileSystemObject")
if ofso.FolderExists(startfolder) then
msgbox startfolder &"already exists",0,"Result."
else
ofso.CreateFolder(startfolder)
MsgBox strfolder &"created.",1,"Status"
end if
```

Figure 3.26 Code 5

OUTPUT:

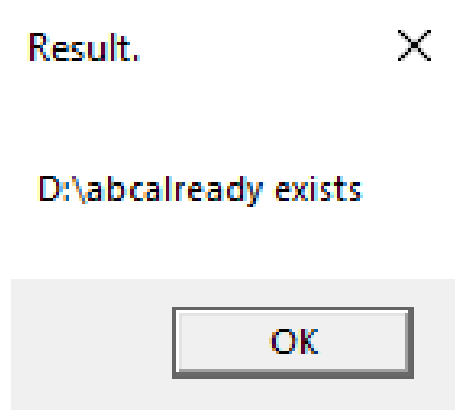


Figure 3.27 Output 5

Example 6:

Figure 3.28 Problem statement 6

**Problem Statement:**

Write a function to generate a unique number. The number should be the combination of : YYYYMMDDHHMMSS

Solution 6: CODE

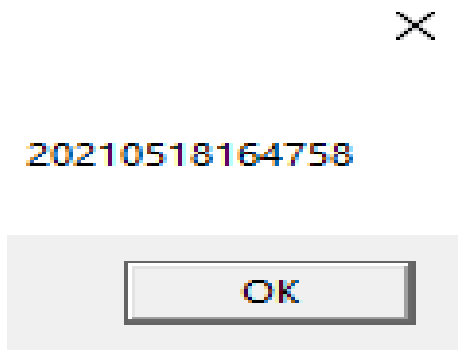
```
option Explicit
dim a,curr_date,curr_time,y,d,m,hr,min,sec
a=Unique

Function Unique()
curr_date=Date
curr_time=Time
y=Year(curr_date)
m=Right("00"&Month(curr_date),2)
d=Right("00"&Day(curr_date),2)
hr=Right("0"&Hour(curr_time),2)
min=Right("0"&Minute(curr_time),2)
sec=Right("0"&Second(curr_time),2)
MsgBox y&m&d&hr&min&sec

End Function
```

Figure 3.29 Code 6

OUTPUT:



**Figure 3.30** Output

### **3.5 Conclusion**

Although VBScript's success as a client-side scripting language has been limited, it remains one of the three most widely used scripting languages (the other two being JavaScript and Perl). With the introduction of Microsoft IIS, it became the primary scripting language for developing ASP (Active Server Pages). Even for Windows Script Host (WSH), the batch language for Windows, it is only the primary option for scripting WSHs.

## **CHAPTER 4**

### **UNIFIED FUNCTIONAL TESTING (UFT)**

#### **4.1 Introduction**

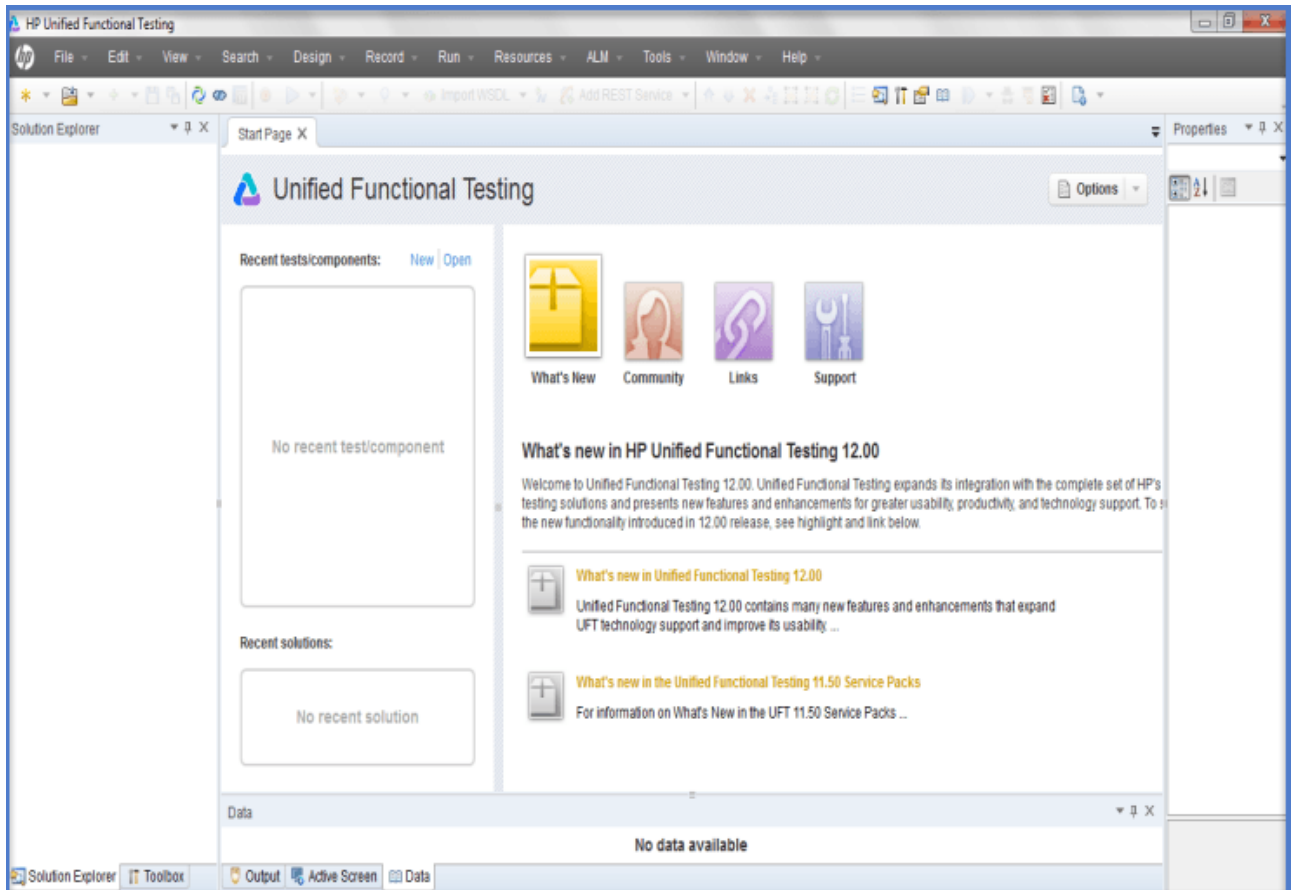
Micro Focus Unified Functional Testing[15] (UFT) is a software application that automates functional and regression testing for software applications and environments. It was previously called Quick Test Professional (QTP).

UFT has a graphical user interface and it supports both the scripting interfaces and the keyword. It specifies a testing protocol while manipulating the object. It controls the testing application using a scripting language called VBScript, developed by Microsoft. Developers can test program's operation using a single console.

#### **4.2 Software Description[16]**

Micro Focus UFT is a collection of automation testing tools that can be used on a wide range of products, environments and software applications. It conducts testing such as regression and functional through the use of a web interface or GUI which is user interface. It recognises objects in an application's web page or user interface and performs desired operations such as typing, clicking or moving cursor).The VBScript scripting language is used by HPE Unified Functional Testing to describe the protocols of test and modify the application artefacts and control. Users must update and manipulate the VBScript in order to accomplish more complicated behaviours.

While UFT is best known for automating "UI-based" test cases, it can also automate "non-UI" cases of test like: database testing and Web services.



**Figure 4.1:** Micro Focus UFT Testing

#### **4.2.1 Exceptional Handling[17]**

Focus on the little details UFT handles exceptions with recovery situations, with the intention of continuing to run tests even though anything goes wrong. Since UFT looks into the applications memory space that is being evaluated, there might be some cases where HPE Unified Functional Testing gets terminated..

#### **4.2.2 Data-driven testing**

While working with UFT, one must pay attention to the specifics. Here, data-driven research is encouraged. We can use the output of a table elsewhere, for example. A Microsoft Excel workbook that can be viewed using UFT is used for data-driven research. Global data sheets and action (local) data sheets are the two types of data tables used by UFT. The test stages will read data from such data tables and in the application, insert variable data, verifying that the required outcome is achieved.

#### **4.2.3 Automating custom and complex objects**

UFT may miss customised user interface objects and other complicated things. Users can classify these items as insight or virtual objects.

#### 4.2.4 Extensibility[18]

Add-ons for a range of environments that aren't supported can be added to Micro Focus UFT.Web,.NET, Java, and Delphi are all supported by UFT add-ins. HP Functional Testing software includes HPQuickTestProfessional and the HPQuickTestProfessional add-ins.

### 4.3 User Interface

Focus on the little details A test script can be viewed and modified in two ways in UFT: Expert View and Keyword view. The above mentioned views helps UFT to function as a test's Integrated Development Environment (IDE), also UFT provides several features, for example breakpoints, which allow you to stop the test at required points.

#### 4.3.1 KeyBoard View[19]

Keyword View allows users to create and view test measures in a tabular format. Each table row indicates a row that can be changed. The mentioned columns can be included in the Keyword View: Operation, Assignment, Item, Documentation, Comment and Value. UFT displays a similar script line for each move in the Keyword View, centred on the values of column and row. Users can edit, delete, add, or alter the work at any moment.

In Keyword View, users may see properties for objects like checkpoints, performance values, and behaviour, as well as implement expressions, conditions and add breakpoints to help debug a test.

#### 4.3.2 Expert View[20]

```
For i = 1 to 10000
  'VERIFY + AND * OPERATIONS ON THE CALCULATOR
  Window("Calculator").WinEdit("Edit").Set(i)
  Window("Calculator").WinButton("+").Click
  Window("Calculator").WinEdit("Edit").Set(i)
  Window("Calculator").WinButton("=").Click
  intResult_1 = Window("Calculator").WinEdit("Edit").GetROProperty("text")

  Window("Calculator").WinEdit("Edit").Set(i)
  Window("Calculator").WinButton("*").Click
  Window("Calculator").WinEdit("Edit").Set("2")
  Window("Calculator").WinButton("=").Click
  intResult_2 = Window("Calculator").WinEdit("Edit").GetROProperty("text")

  If intResult_1 <> intResult_2 Then
    Reporter.ReportEvent micFail, "VERIFY", "RESULT INCONSISTENT FOR DATA !" & i
  End If
Next
```

## Figure 4.2: VBScript Code in Expert View

UFT's Expert View allows users to view and edit the source code of a test using VBScript. Users are allowed to modify most of the test actions. Adjustments are made using the Keyword View.

### 4.4 Language

VBScript is the scripting language used by Micro Focus UFT. It encourages classes, but does not support polymorphism. Unlike VBA, VBScript does not have the skill to use specific keywords of Visual Basic.

### 4.5 Work Done

Example 1:

**Problem Statement:**

Verify availability of add-ins. How to select and de-select an add-in. Try to unselect Web add-in and identify web-based objects. Compare the observations between Web add-in is selected and not selected while identifying objects in web page.

Also Use Generate script to understand configuration of UFT

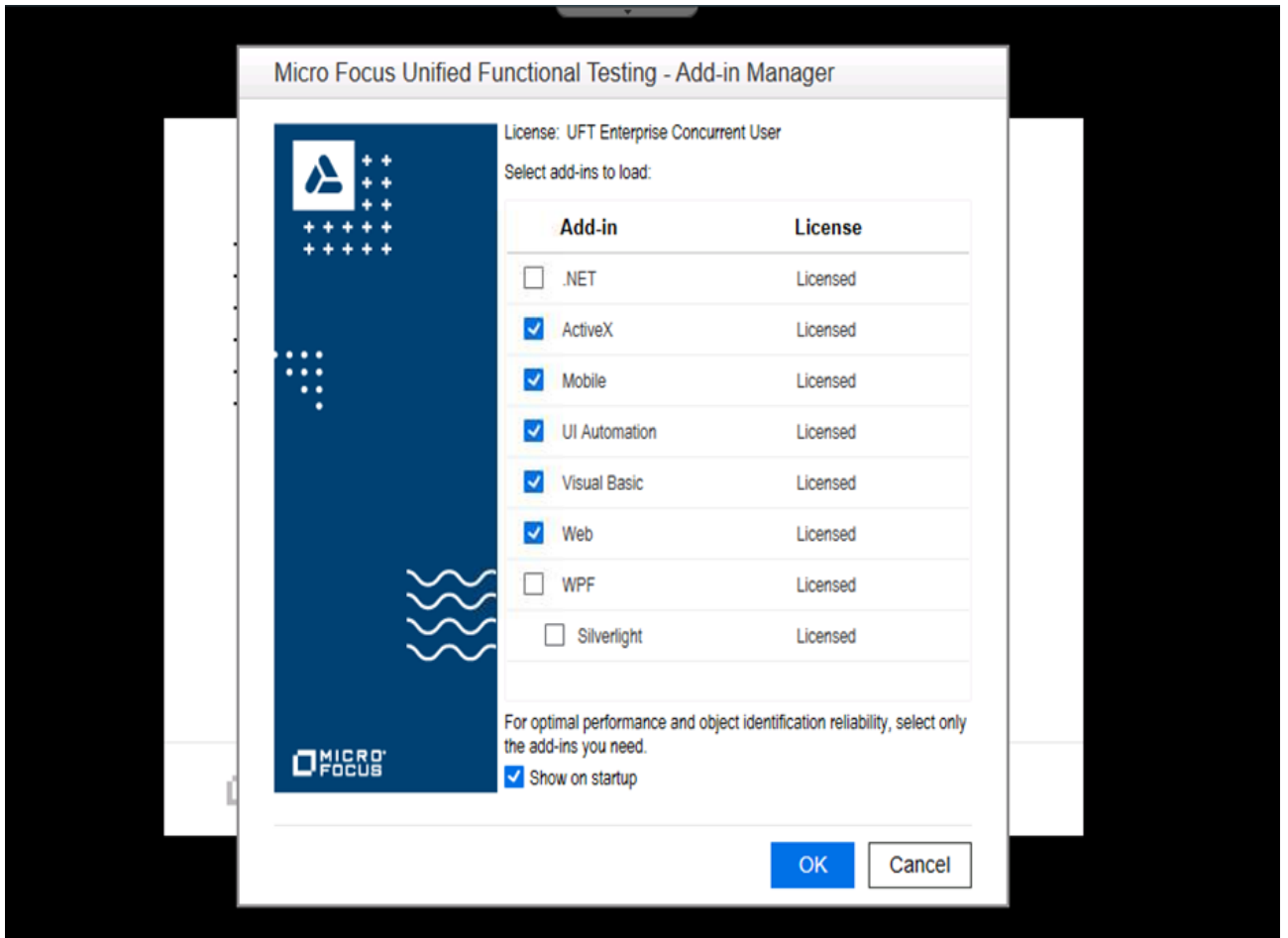
### Figure 4.2 Problem statement 1

Solution 1:

1. Start UFT one



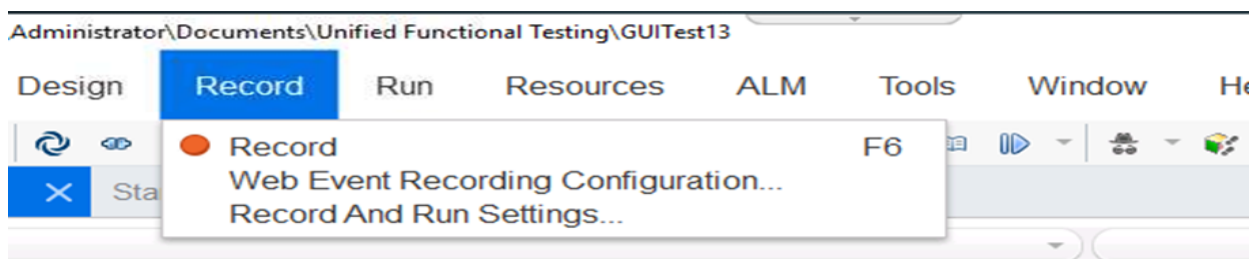
2. The Unified Functional Testing Add-in Manager Dialog Box opens.



3. Select the web add in as shown below

**Figure 4.3** Select web

4. Now go to record and run settings



**Figure 4.4 Record**

5. It can be seen that the web objects are shown in the image below

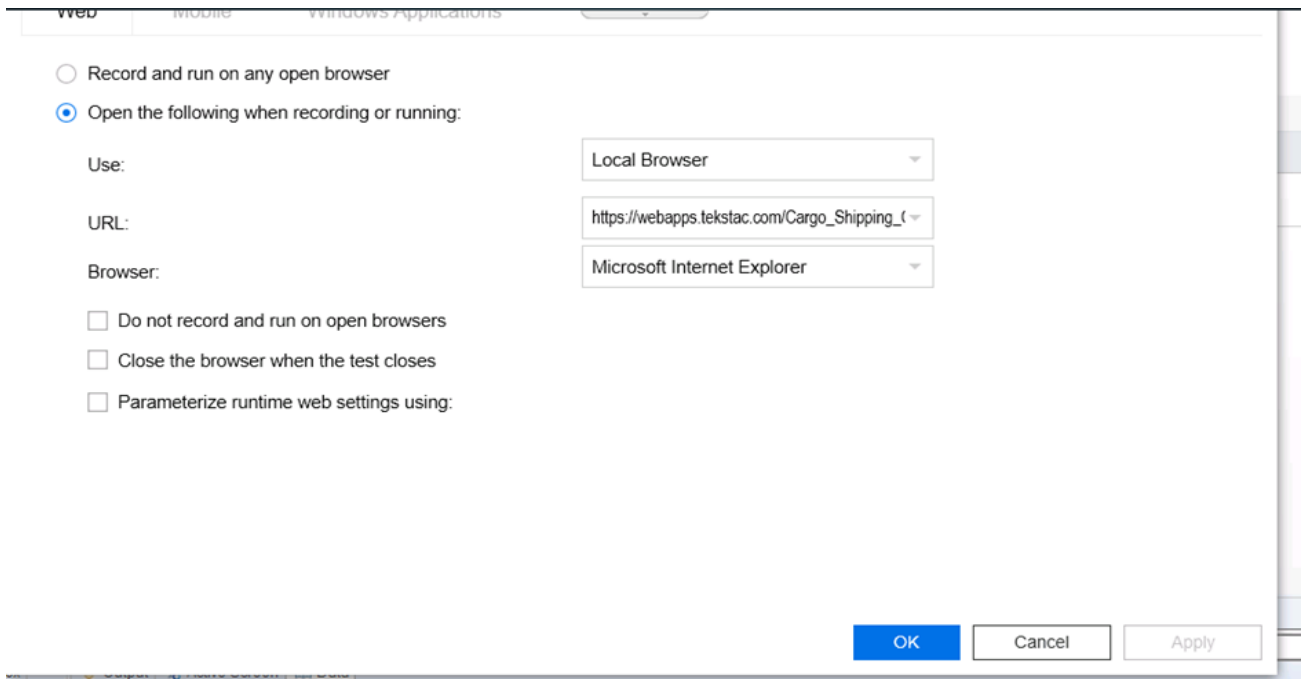


Figure 4.5 Web objects

6. Now close the application and open it again and deselect the web add in as shown below

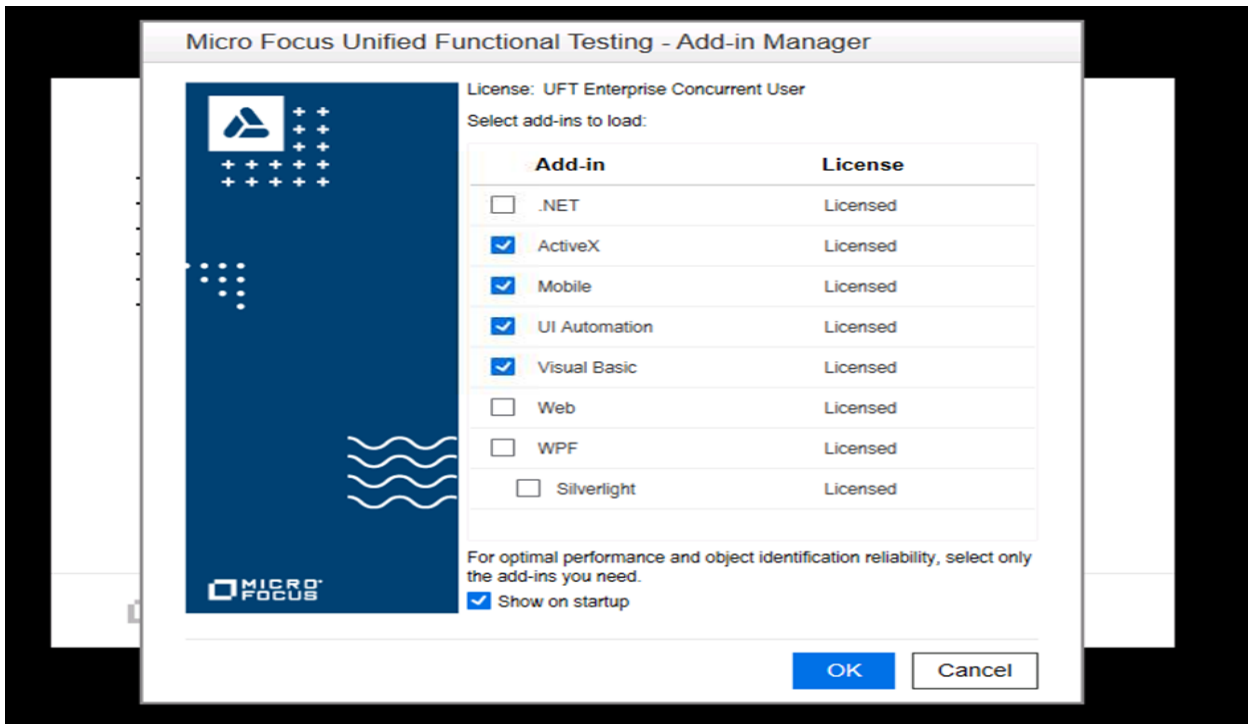


Figure 4.6 Deselect Web

7. Go to record and run settings and we can see that the web objects are not present

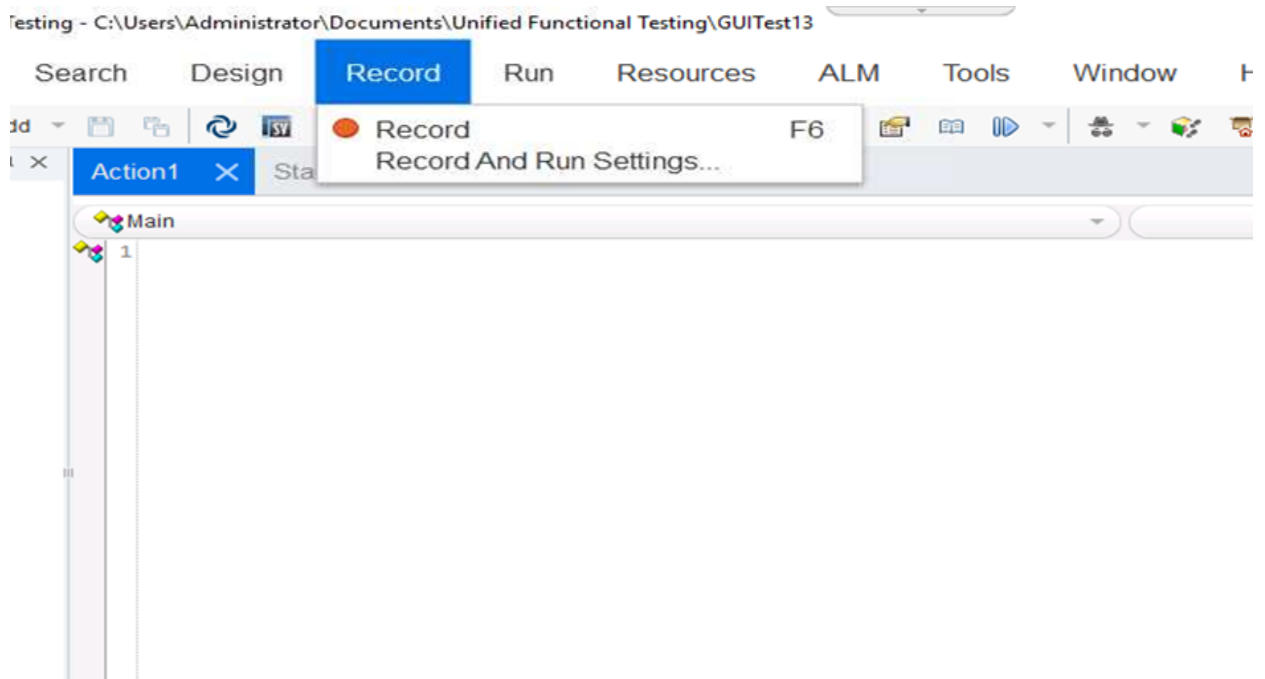


Figure 4.7 Record and run

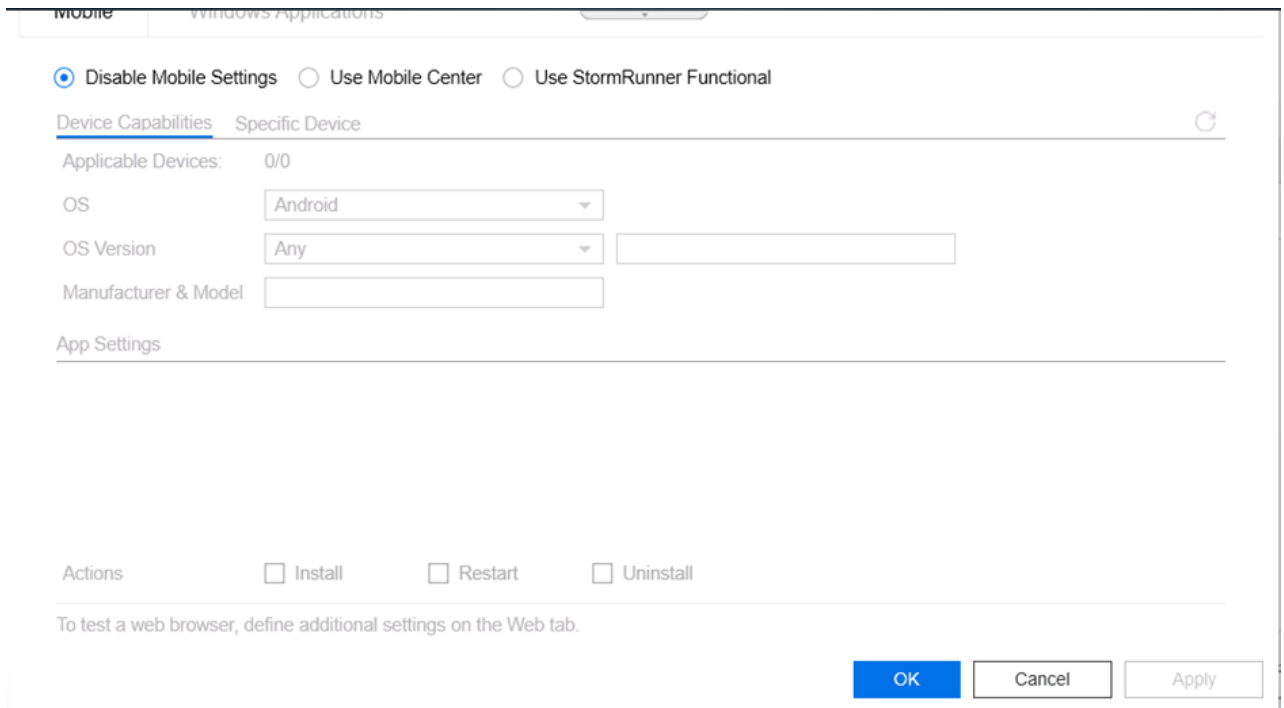


Figure 4.8 Disable mobile settings

8. For object identification go to tools> object identification

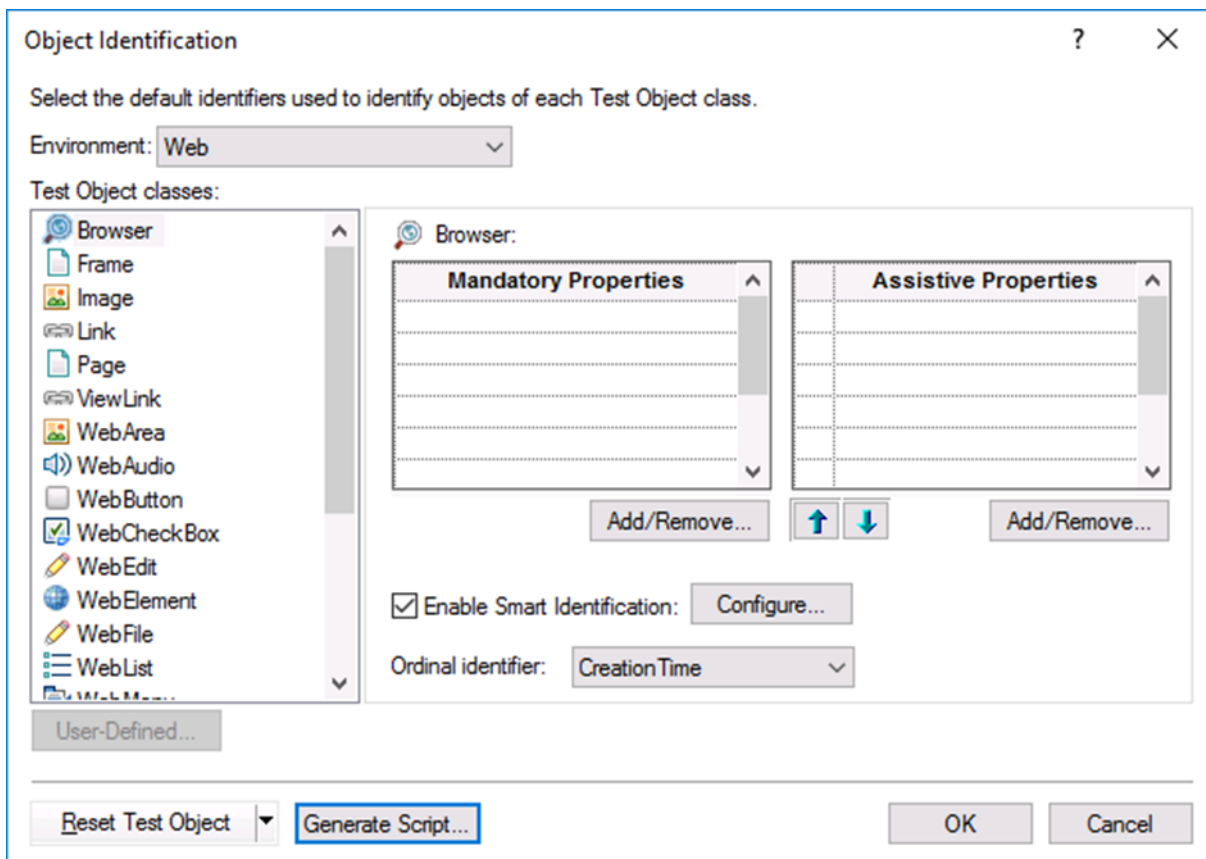


Figure 4.9 Object identification

## REFERENCES

[1]: [https://www.tutorialspoint.com/sdlc/sdlc\\_overview.htm](https://www.tutorialspoint.com/sdlc/sdlc_overview.htm)

[2],[3],[4],[5],[6],[7],[8]: <https://www.guru99.com/software-development-life-cycle-tutorial.html>

[9]: <https://www.guru99.com/functional-testing.html>

[10]: <https://www.tutorialspoint.com/sql/sql-overview.htm>

[11]: <https://en.wikipedia.org/wiki/XML>

[12]: <https://en.wikipedia.org/wiki/JSON>

[13],[14]: <https://www.geeksforgeeks.org/vbscript-introduction/>

[15],[16],[17],[18],[19],[20]:

[https://en.wikipedia.org/wiki/Micro\\_Focus\\_Unified\\_Functional\\_Testing](https://en.wikipedia.org/wiki/Micro_Focus_Unified_Functional_Testing)

## CONCLUSION

This learning experience has been very enlightening as I got to learn a lot from the industrial experience. I learned technologies like Functional Testing, Agile Methodologies, Database- SQL, XML and JSON, VBScript and UFT. Getting industrial experience on all these topics has been a totally different experience for me.

I understood the work of a QA Testing Engineer and how important it is for every product to be tested before releasing it in the market. I got to learn about a commercially used software-UFT.

UFT fundamentals have been made simple via QA Eversity HP Unified Functional Testing (UFT) does break the bank however it is nonetheless a marketplace chief within the automation checking out business. Nowadays, many open supply and coffee-value automation equipment got here into play however QTP and UFT are nonetheless widespread for their complex options and controls.

