

Automation Testing Using UFT

Major Project report submitted in partial fulfilment of the requirement for the
degree of
Bachelor of Technology in
Computer Science and Engineering/Information Technology

By

Nikhil Sharma (171219)

Under the supervision of Dr. Pradeep Kumar Gupta
to



Department of Computer Science & Engineering and Information Technology

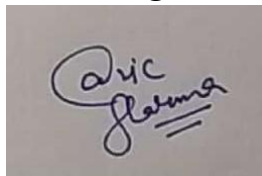
**Jaypee University of Information Technology Waknaghat, Solan- 173234,
Himachal Pradesh**

Candidate's Declaration

I hereby declare that the work presented in this report entitled “**Automation Testing Using UFT**” in partial fulfilment 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 under the supervision of **Dr. Pradeep Kumar Gupta**.

The work done embodied in the report has not been appeased for the award of any other degree or diploma.

Student signature

A handwritten signature in black ink on a light grey background. The signature appears to be 'Nikhil Sharma' written in a cursive style.

Nikhil Sharma
(171219)

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

A handwritten signature in blue ink. The signature is stylized and appears to be 'Pradeep Kumar Gupta'.

(Dr. P. K. Gupta)

Dr. Pradeep Kumar Gupta

Associate Professor

Department of Computer Science and Engineering and Information Technology

Dated:

Acknowledgement

I would like to take the opportunity to thank and express my deep sense of gratitude to my mentor and project guide **Dr. Pradeep Kumar Gupta** for his immense support and valuable guidance without which it would not have been possible to reach at this stage of our final year project.

I am also obliged to all my faculty members for their valuable support in their respective fields which helped me in reaching at this stage of my project.

Date:

Abstract

One of the most crucial aspects of software development is testing.

key goal and phases of the Software Development Life Cycle to discover issues and guarantee the software's quality Computer software Testing can be carried out manually or automatically. Manual testing is done by tester without any tool. In automation Testing is carried out using automated testing tools.

The project is based on Writing Valuable Scripts to Perform Automation Testing on various Web Applications . Different cases are made and for each test test case different smoke and regression testing is performed . The project is based on the concepts on Functional Testing Using VbScript as the Scripting language and Microfocus One as the Automation Testing Tool .

Table of Contents

1. INTRODUCTION

- 1.1 Introduction.....
- 1.2 Project overview.....
- 1.3 Objectives
- 1.4 Methodology
- 1.5 Genesis of Problem

2. LITERATURE SURVEY

- 2.1 A Comparative study of QTP and Load Runner Automation Testing Tools and Their Contribution to Software Project Scenarios.....
- 2.2 A Comparative Study of Software Testing Tools

3. SYSTEM DEVELOPMENT

- 3.1 System Requirements
- 3.1.1 Software Requirements.....
- 3.1.2 Minimum Hardware Requirements

4. IMPLEMENTATION

5.1 Tools and Techniques.....
5.2 Implementation.....

5. CONCLUSION AND FUTURE PERSPECTIVE

6.1 Conclusion.....
6.2 Future Scope.....

REFERENCES

List of Figures:

1. Figure 1: **Automated Test Life cycle methodology**
2. Figure 2: **Phases of Load Runner Testing**
3. Figure 3: **UFT Framework**
4. Figure 4: **Test Plan Tree**
5. Figure 5: **Test Resources Tree**
6. Figure 6: **Execute Test Case in Test Set**
7. Figure 7: **Run all Test Cases**
8. Figure 8: **Test Execution flow diagram**
9. Figure 9: **Defect Dashboard**

List of Tables:

1. Table 1: Manual Testing VS Automation Testing

Chapter 1: Introduction

1.1 Introduction

Computer systems have been an important aspect of our daily lives because they have the ability to affect millions of citizens in different areas of life necessitating the development of stable and dependable software. Unfortunately, humans are vulnerable to making mistakes, and the basic realities of humans' central role in software production render errors an unavoidable part of the process .

Manual and automatic software testing are also possible. Manual research does not necessitate the use of any testing tool; but, it does necessitate a lot of time and more people. Manual checking can be automated in the same way as a programmer can use a scripting language to write programmes to automate any manual operation . Automated research entails the use of software methods and/or systems in the execution of tests.

Selecting and using an acceptable testing process, as well as selecting and using the correct test automation platform and/or system, are defining factors for effective and productive software testing projects. The different techniques or techniques used to evaluate an application to ensure it performs and looks as intended are referred to as software testing methods. They cover everything from front-end to back-end monitoring, as well as unit and machine testing. This paper focuses on software test automation, specifically automatic software testing, its various tools and their categories, as well as testing mechanisms that can be used to accomplish accurate, efficient software testing. Umar gives a comprehensive description of software testing methods; however, this paper focuses on software test automation, specifically automated software testing, its various tools and their categories, as well as testing frameworks that can be used to achieve successful, efficient software testing.

1.2 Objective

The aim of test automation is to provide an automation system with minimal scripts. Testing and production specifications will change in a flash nowadays. It's possible that the automation scripts would need multiple changes. This script must be flexible to adapt to the new research requirements.

As a result, its long-term relevance is obliterated

1.3 Methodology

What is Automation Testing?

Automated testing is a method of controlling the execution of experiments and comparing real results to predicted results using software different from the software under evaluation . Certain parts of manual testing are automated with automation software, but not all .Automated testing saves time because the tester can perform a huge amount of tests in a limited period of time, and it can automate critical and routine tasks as well as testing that would be impossible to do manually. Aside from saving time, automated testing also saves resources and effort, improves the efficiency of testing activities, and aids in product development.To set up the test cases and conduct the testing, a professional tester with knowledge of the automation methods and the applications being tested is needed.

Importance of Automation Testing:

Automation of the software testing process will benefit an organization's Quality Analysis (QA) departments in a number of ways, the most important of which are listed below.

- 1. Increased Test Coverage:** Automation testing improves test coverage by allowing longer tests to be completed in less time, allowing more tests to be completed within the short timeframes that product development teams face today. This is possible not only because of the speed of automation, but also because a test automation tool can run experiments without human intervention, resulting in higher time utilization.
- 2. Improved Accuracy:** And if the most dexterous tester is doing the tasks, there is a risk of human error with manual testing. With automated monitoring, all faults, no matter how minor, are absolutely avoided
- 3. Rapid Testing Process:**Since it requires the use of tools and scripts, automation testing is significantly easier than manual testing. This improved pace will give QA teams an added advantage by allowing them to run tests several times while also meeting deadlines.

4. **Reusability of Test Cases** :For automated testing, the same series of test cases can be reused to test various implementations of the app, and they can be updated if needed to include new functionality and error discoveries. Software testing can become tedious, and an automation testing tool can quickly handle the tedious tasks while still eliminating the risk of human error.
5. **Earlier Detection of Bugs** : Early in the software development life cycle, automation testing allows for the detection of errors. Although this can be insignificant, it will save a lot of time and money in the long run when it comes to repairing bugs. The checks will be run automatically if the source code is changed to find any glitches or problems. Furthermore, the simultaneous execution of test cases allows for the generation of informative reports for quick comparison.

1.4 Organization of Thesis

In Chapter 2, we have discussed the literature survey. We have looked upon the various existing approaches that have been used for social media analysis. Each one of them has its own unique features which distinguishes it from the rest

In Chapter 3 we have done feasibility study and requirement analysis.

In Chapter 4 we have thereby concluded our report and discussed the future scope of our project.

1.5 Genesis of Problem

In this modern digital age everything is working with the help of big data and artificial intelligence. Organisations are trying to increase their efficiency with the help of various development and testing techniques. So to have a bug free experience various Testing techniques have been developed over the course of time . Test would want to spend less time writing scripts for test cases and hence the need of automating testing comes into play. We will also have look on how the existing systems worked before.

Chapter 2: Literature Survey

Paper 1: Mohammad Imran, Dr. Mohamed A.Hebaishy, Dr. Abdullah Shawan Alotaibi

“A Comparative study of QTP and Load Runner Automation Testing Tools and Their Contribution to Software Project Scenarios ”

Overview:

For many development projects, automated software testing is becoming increasingly necessary in order to automatically validate core features, monitor for failure, and assist teams in running a vast number of tests in a limited period of time. The aim of this study is to assess and compare automated testing tools such as Fast Test Specialist and Load Runner in order to ascertain their usability and effectiveness.

There are a number of resources available to assist programme development teams in creating and executing automated evaluations. The most important aspect, though, is to choose the most powerful tool from among the different tool categories. The key goal is to compare the capabilities that these resources support in order to assess their usefulness and efficacy.

Manual Testing :

Software testing can be done in two ways: manually and automatically. Manual Testing (2.1) Code is manually checked in manual testing, which means that no automatic process or script is used. The method of manually checking applications for bugs is known as manual testing. It necessitates a tester taking on the task of an end user and making extensive use of all available tools.

features of the programme in order to ensure proper operation. To ensure that the research is thorough, the tester also follows a written evaluation plan that guides them through a series of key test cases.

- Manual testing necessitates more time and effort, and in certain cases, both.
- Manual research makes performance testing impossible.
- Executing the same test over and over is a time-consuming and repetitive process.
- Manual Testing makes it difficult to detect differences between GUI object sizes and colour combinations, among other things.
- Batch testing is not possible; human user intervention is needed for each and every test execution.
- The scope of the manual test is restricted.

- We are unable to reuse manual tests.
- Manual testing for a large number of users cannot cover the actual load and results. Such tasks are impossible to complete.

Automation Testing :

Test automation is the use of specific tools (not part of the software being tested) to monitor the execution of tests and the comparison of real results to expected results of software testing. [number four] Test automation can automate certain tedious but required activities in an already established formalised testing procedure, or it can implement new testing that would be difficult to do manually.

The aim of automated testing is to reduce the amount of time spent testing by using a small number of scripts. Unit monitoring, for example, could be a suitable choice for automation if it occupies a significant portion of a quality assurance (QA) team's capital. Automated testing software can run tests, monitor findings, and compare results to previous test results.

Automated Test Life Cycle :

The decision to automate testing is the first step in the lifecycle. The team develops a test tool proposal and outlines the possible advantages of automation at this time.

The procurement of a test instrument is the second stage. The whole testing tool is assessed and chosen at this time. The automated testing integration process is the third stage.

This stage lays out the steps for successfully implementing automated testing in a new project. The first step in the protocol is to analyse the test process. The preparation, design, and implementation of tests is the fourth phase. This is the most critical step, in my opinion, since it involves specifying the test procedures requirements, determining the measures, and establishing the implementation standard. The number of experiments to be conducted, how the testing will be approached, and the test conditions that will be used are all part of the test design step. Finally, the test implementation phase is completed to ensure the automatic experiments are reusable, repeatable, and maintainable. The execution and supervision of tests is the fifth level. Both test plans are also carried out in accordance with the guidelines discussed previously.

Fig 1: Automated Test Life cycle methodology

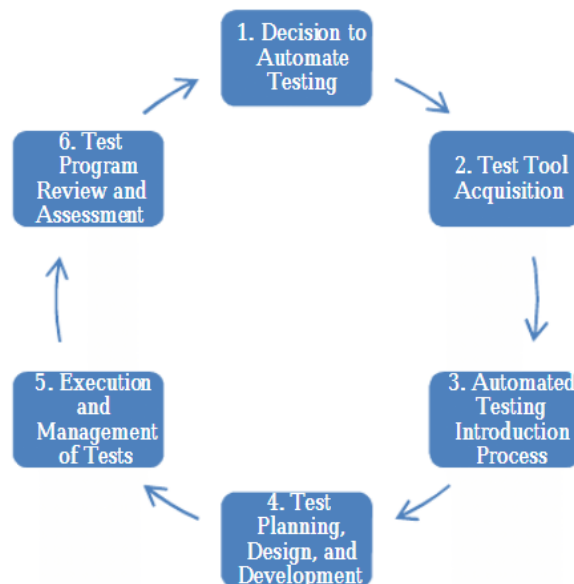
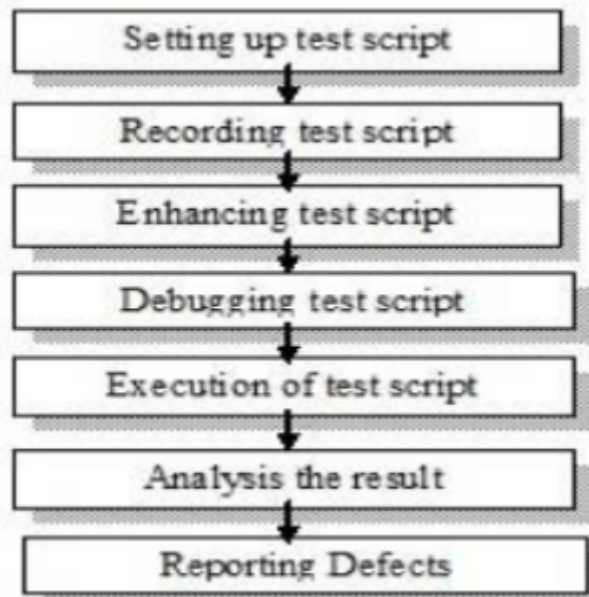


Fig 2: Phases of Load Runner Testing



Paper 2: Nazia Islam :

“A Comparative Study of Automated Software Testing Tools ”

Overview:

The Software Development Life Cycle (SDLC) process includes a phase called software testing.

Testing evaluates a software item's functionality as well as the product's efficiency. To carry out research operations, automated software testing employs a variety of methods. I've addressed the differences between automatic and manual testing in this article, as well as examined three automated software testing tools: Selenium, UFT/QTP and WATIR

In summary, There has been given a thorough analysis.

Each tool's definition focuses on its multiple feature set, performance, simplicity, and usability. I

The various facets of Selenium, UFT/QTP, and WATIR were also analysed, measured, and compared.

Finally, this study allowed me to make some clear distinctions between automated and manual processes.

testing, as well as read about and investigate different aspects of automatic testing.

Objectives of Research :

The aim of this research paper is to describe the various features and presentations of software testing tools (Selenium, WATIR, and UFT/QTP), as well as to evaluate and compare these tools in order to determine their utility. This research also provides for a simple analysis between automated and manual testing in order to demonstrate the importance of automated testing in the software industry.

- Choosing the framework that would be evaluated by both of these methods.
- Using the automatic testing software you've chosen, validate the target programme and collect the results.
- Creating a collection of comparisons to use in evaluating the methods.
- Analyzing each method and comparing them to one another based on an ideal feature set and the results

Software Testing Techniques:

Orthodox software testing techniques have a variety of approaches. The most popular strategies are listed in this section.

Black Box Testing - The basic premise of black box research is that the tester does not need to understand the internal workings of the code or the specifics of how the software is put together. In this case, the tester is only concerned with the program's interface. The tester, as the name implies, has no vision or experience of execution. The aim of this testing is to ensure that the application or software meets the required criteria or collection of inputs and produces reliable outputs. The tester does not need to be an expert programmer or understand the inner workings of the code; instead, they must understand the program's intended performance.

White Box Testing - White box testing (also known as transparent box testing) is the polar opposite of black box testing. This research necessitates some degree of programming knowledge on the part of the tester. The tester should be able to understand the system's internal configuration as well as code specifics. Since the majority of programme errors or glitches are discovered and fixed during white box testing (unit testing), it's a good idea to do so. Unit testing is a good example of white box testing, where the code built for a specific module must be checked before being integrated with the rest of the modules. As a result, unit checking is very effective at reducing overall code glitches. The white box testing methodology is used in test-driven growth (TDD).

Manual Testing- Manual testing is the most common level of testing, in which experiments are carried out by following test cases and communicating personally with the application. The tester prepares the test cases for this kind of examination. Test cases are written in plain English and depict the functions or functionalities to be evaluated as well as the predicted outcome.

- Inefficient in terms of time and labour.
- The learning curve is very linear.
- The lack of reusability is a disadvantage.
- Not an iterative process, and multiple iterations aren't always better for accuracy[20].
- Manual checks have little exposure because the tester has no understanding of how the code performs.
- The programme engineer, tester, and then the bus would repeat the tests.

Automated Testing- Testing has become much more effective as a result of the rise of research automation in the industry. The responsibility of user initiation is removed, as is the complexity of conducting different forms of testing, such as regression and load/performance testing. Complex testing activities have been much simpler than before thanks to the advancement of automatic testing, which allows for the testing of many sets of data.

Objectives of Automation:

Simplifying Regression testing: When new releases/error patches are made, regression testing is used to ensure that the newly implemented updates or bug fixes have not inserted new vulnerabilities in the code and that it is working correctly for all of the current functionality. As a result, this test must be done on a daily basis. When technology is used, it saves the organisation money, energy, and time.

Executing same tests multiple times: When test cases must be run several times with various input sets, automated testing using test scripts is very effective.

Time and budget constrain: Test automation saves the tester time and resources, but in this situation, the tester has more time to do other things. In today's world, automation is a very efficient and successful concept for businesses. Since most testers are familiar with programming, they will jump right through using the method.

Load or performance testing: At least a few virtual test user accounts must be generated and checked concurrently when checking an application's load handling ability. This role seems to be too important to depend solely on manual checking. This difficult research, on the other hand, can be done successfully with the aid of automated testing.

Frequently changing functionalities: Test automation performs well where the parameters are constantly changing and complicated test cases to be run repeatedly. In any programme where the user interface does not alter constantly but the code does, test automation can be more accurate.

Table 1 : Manual Testing VS Automation Testing

Manual testing	Automated testing
1. The process of manual testing is slow and lengthy, which takes a long time to complete the test.	1. Automated tests can be performed in a faster way than manual tests; because once the test scripts are generated, it can be executed any number of times.
2. The manual test requires vast human effort. As the testing is time consuming, it requires more testers to accomplish the task.	2. Automated tests need one-time effort to write the test scripts, as it can be executed without having human effort. So, it requires fewer investments for testers.

Chapter 3: System Development

3.1 System Requirements :

3.1.1 Software Requirements:

1. Windows 10 (64-bit) or Windows 7 (x64 or x86)
2. Node.js 8.8.2 or later.
3. Microsoft Visual C++:
4. Microsoft WSE 3.0 Runtime

3.1.2 Minimum Hardware Requirement:

- 4GB RAM or more
- CPU - Dual core with 2.6GHz
- Browser - A recent version of Internet Explorer
- Memory Required 2 GB minimum

Chapter 4: Implementations and Results

4.1 Tools and Techniques

We have used Microfocus Uft One for implementing our project.

4.1.1 Why UFT One ?

HP Unified Functional Testing is a method that can be used. To make test management easier, it comes with HP Application Lifecycle Management (aka HP QC), a test management and source code control tool.

When it comes to practical automation, it is without a doubt the most powerful commercial player on the market. It's a simple and intuitive platform that deals for both Windows and web-based apps. It's simple to integrate with ALM, the most recent version of the industry's most popular test management software. We'll go into how to combine UFT with ALM in this blog.

4.1.2 UFT Script Components:

- **Driver Script** - The script that controls the whole process. It takes care of the prerequisites and initial configurations for the execution. This script would have checkpoints for each test stage as well as calls to other components (Functions).
- **Functional Library** - The Function Library is made up of Associated Functions. It typically includes basic services, page operations, an excel controller, and a reporter, among other things. The.vbs or.qfl extension refers to a mutual feature register.
- **Object Repository** - Both test objects are saved in the Object Repository. The Object Spy will be used by the tester to classify page objects and link them to the Object Repository. Then, in our key driver scripts or shared feature files, we'll use specified objects. It typically has a tsr extension and is used as a shared object repository.
- **Test Data**- Both parameters and DataTables can be used to separate our test data, but DataTables are more common because they can handle different sets of data. A DataTable, similar to Microsoft Excel, aids testers in the development of data-driven test cases that can be used to repeat an Action.
- **Recovery Scenario**- We should use the "ApplicationCrash" recovery scenario, which re-runs our test in the event that our application (such as Chrome or Internet Explorer) crashes.

Fig 3 -UFT Framework

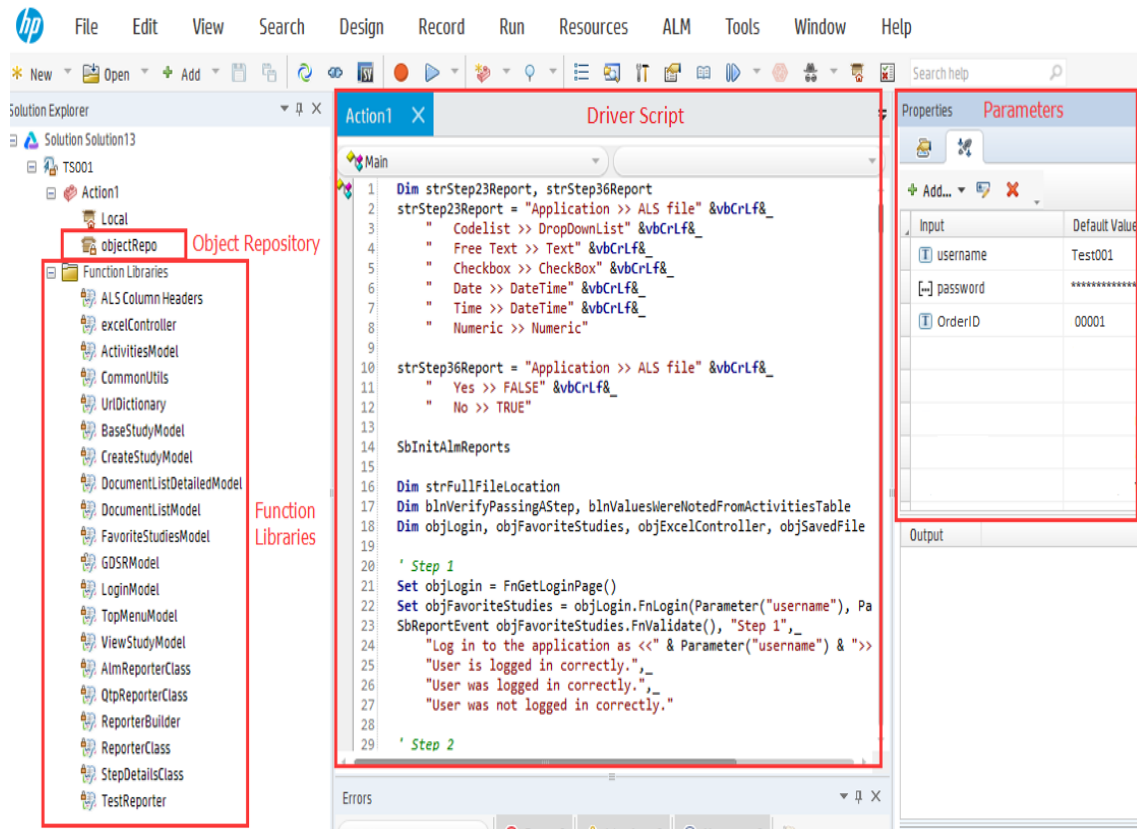


Fig 4 -Test Plan Tree

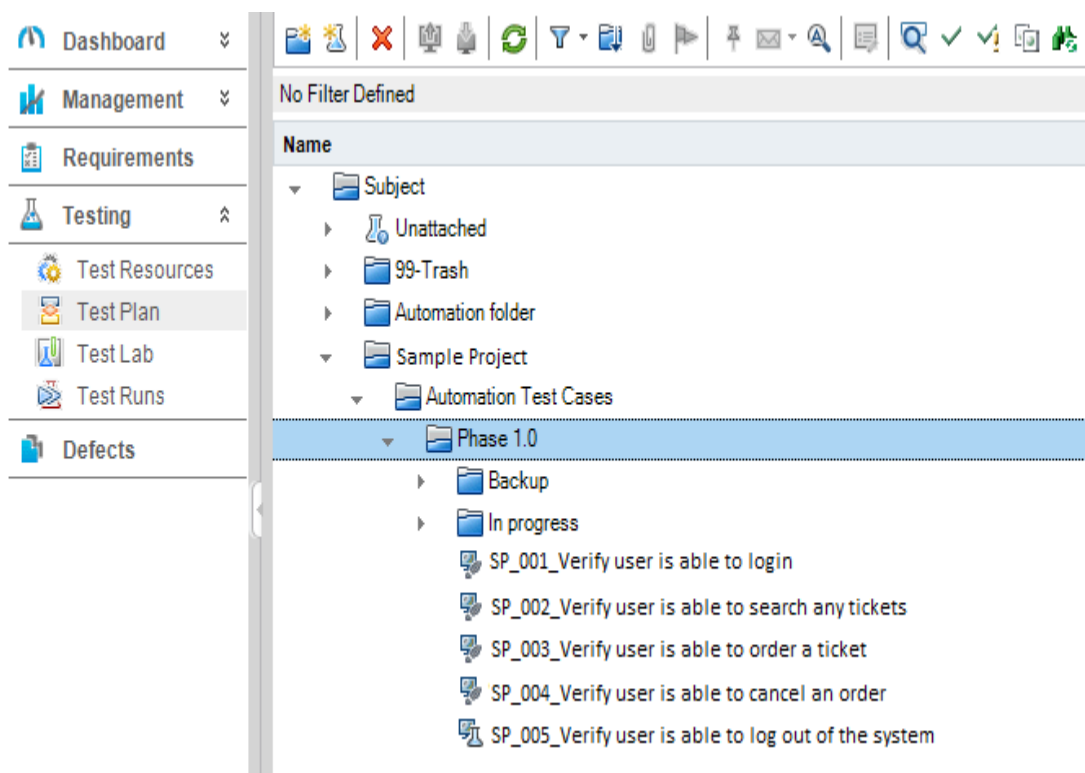
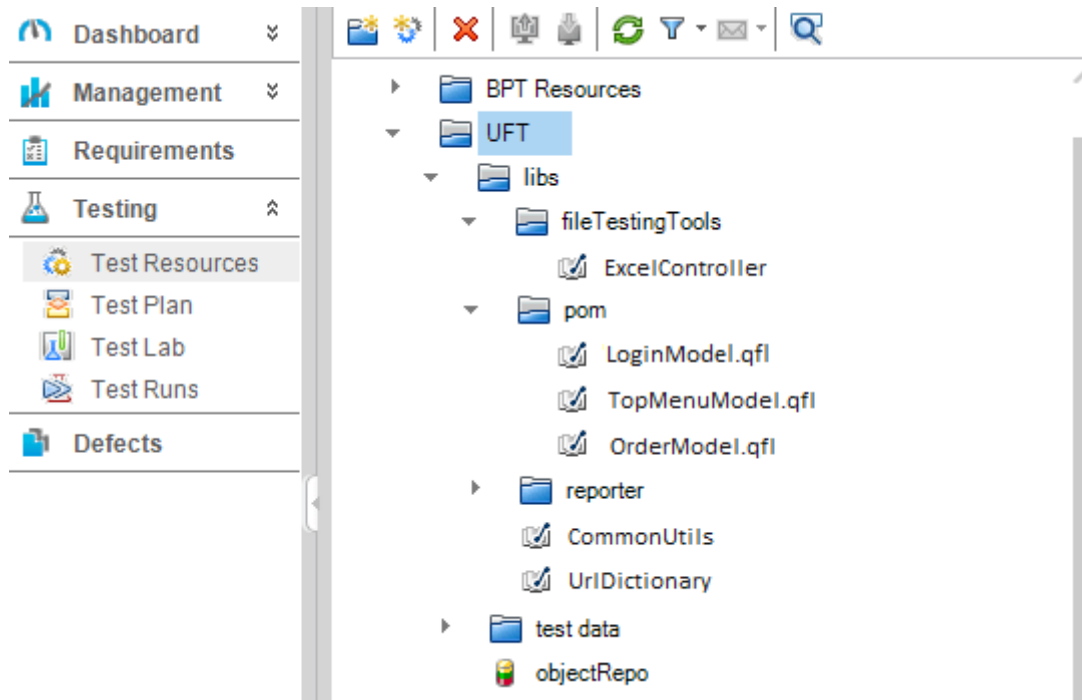


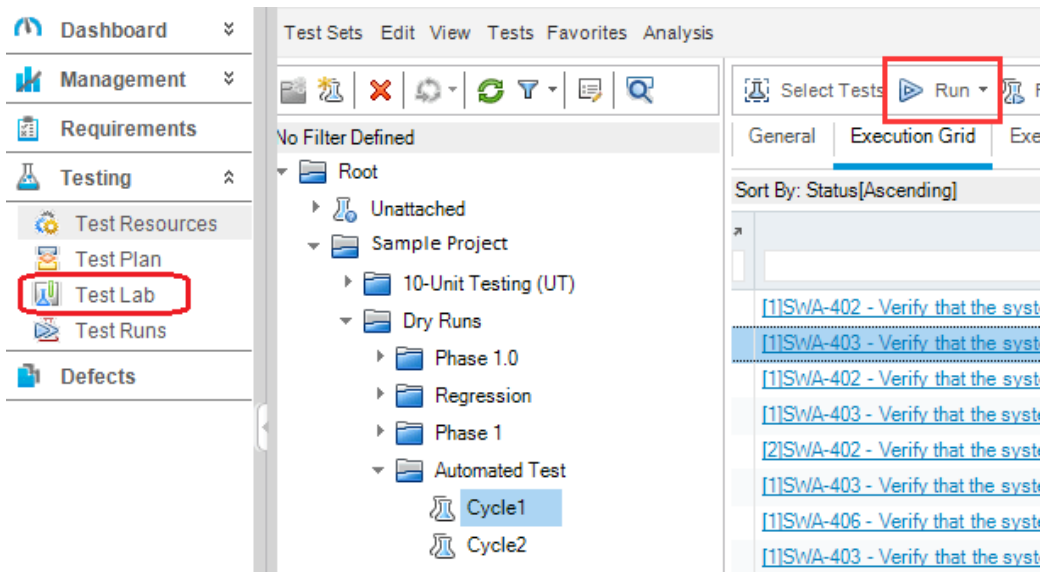
Fig 5 -Test Resources Tree



4.2 Implementation:

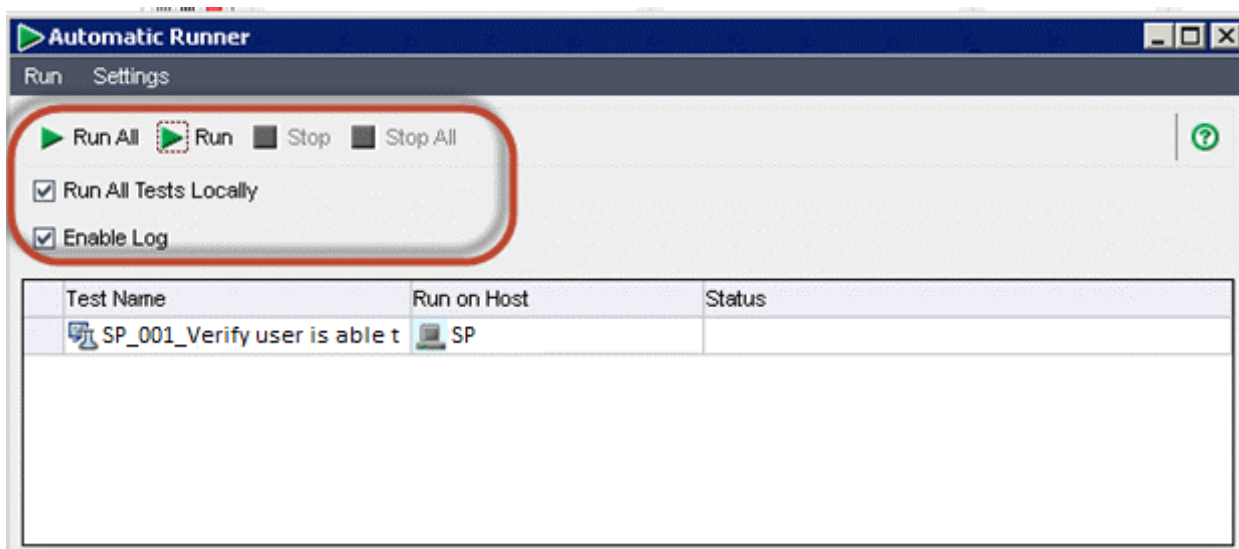
4.2.1 Executing Scripts in ALM : In the “Test Lab” Module, a test range is generated for implementation. The test instances can be added from the test plan tree after the test set has been developed.

Fig 6 -Execute Test Case in Test Set



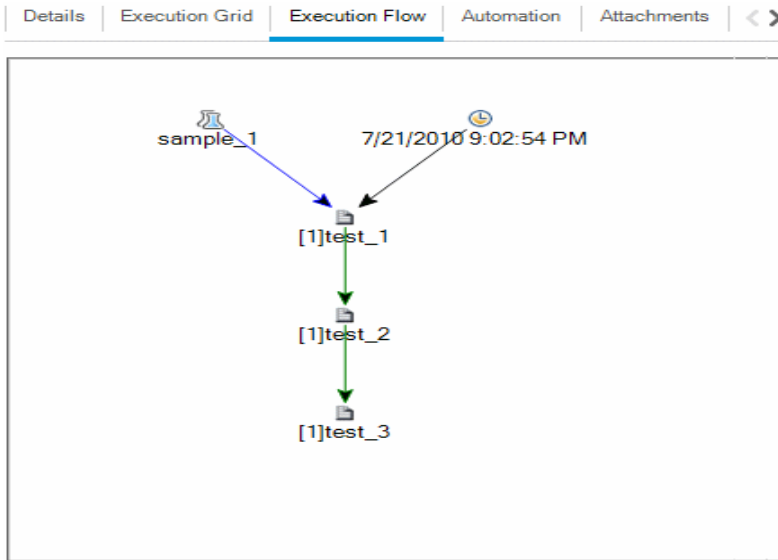
4.2.2 Running Scripts - To start the exam, click "Run." By pressing Run all, we can run all test cases from a selected test range. In addition, we can run only the selected test instance. While the script is being run, UFT will be released in the background.

Fig 7 -Run all Test Cases



4.2.3 Test Execution Flow - We may also monitor how test instances in a test set are executed. You may define a date and time, as well as set conditions for executing a test case, using the Execution Flow tab of the Test Lab module.

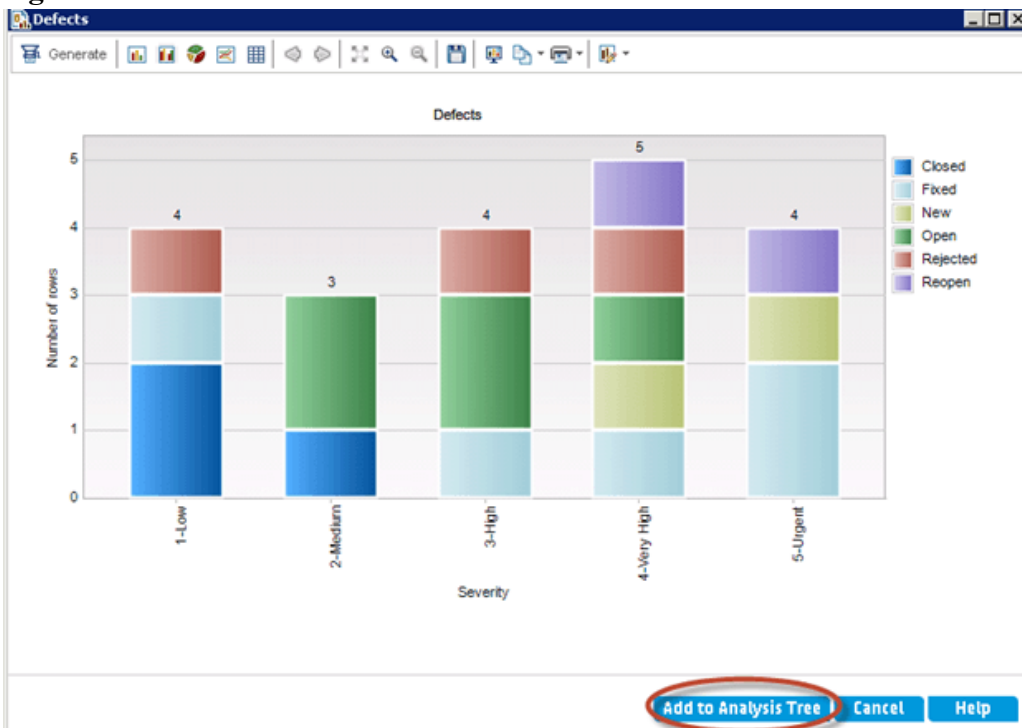
Fig 8 -Test Execution flow diagram



4.2.4 Test Results Analysis - The Test Lab stores all completed test results, and ALM allows users to create reports and graphs based on the Test Lab's results.

Developers can create, monitor, and display analysis objects such as diagrams, progress reports, and Excel reports using the analysis view module.

Fig 9 -Defect Dashboard



Chapter 5: Conclusion and Future Scope

5.1 Conclusion

Test automation has been a critical component of successful software testing. Test automation, according to the new World Quality Report 2018–2019, is the greatest bottleneck in delivering “Quality at Speed,” as it saves time, money, boosts performance, and improves accuracy.

As a result, without the correct automation tools and framework, accurate and efficient test automation is impossible. This article includes a concise overview of various automated test toolkits, as well as perspectives into some of the most relevant aspects to consider when choosing an automated tool and structure.

What we have done till now?

We have gone through literature and information from Eudemy Courses. And we have done sufficient hands on exercises to boost our skills with necessary tools and technologies. We have participated in various code challenges as well.

5.2 Future Scope:

Since technology has advanced at such a rapid rate, there has never been a greater need for projects to be completed as quickly as possible. To finish tasks quickly, all processes used during the programme life cycle must be accelerated as well.

Automation may be used to save money and resources in the field of software testing, but only in time-consuming programmes. Automation testing is the way to go when it comes to doing regression testing on a wide scale. It could be a good option.

Test automation has a range of important benefits, including improving product efficiency, reducing manual software testing operations and eliminating unnecessary testing practises, creating more standardised repeatable software evaluations, reducing repetitive work and generating more reliable testing results, and increasing accuracy.

References

1. Mohammad Imran, Dr. Mohamed A.Hebaishy, Dr. Abdullah Shawan Alotaibi: “A Comparative study of QTP and Load Runner Automation Testing Tools and Their Contribution to Software Project Scenarios ”
2. Nazia Islam : “A Comparative Study of Automated Software Testing Tools ”

Project Report Undertaking

I Mr. /Ms. Nikhil Sharma Roll
No. 171219 Branch CSE is doing
my internship with Cognizant Technology Solutions from Feb 2021 to
May 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 Nikhil Sharma

Roll No. 171219

Date 24-05-2021