D. ATEŞOĞULLARI

RETURN ON INVESTMENT IN TEST AUTOMATION VIA SELENIUM

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
ATILIM UNIVERSITY

DİLARA ATEŞOĞULLARI

A MASTER OF SCIENCE THESIS
IN
THE DEPARTMENT OF SOFTWARE ENGINEERING

JANUARY 2020

ATILIM UNIVERSITY 2020

RETURN ON INVESTMENT IN TEST AUTOMATION VIA SELENIUM

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
ATILIM UNIVERSITY

BY

DİLARA ATEŞOĞULLARI

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE IN
THE DEPARTMENT OF SOFTWARE ENGINEERING

JANUARY 2020

Approval of the Graduate School of Natural and Applied Sciences, Atilim University.

_____

Prof. Dr. Ali KARA
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science in Software Engineering, Atilim University.

_____

Prof. Dr. Ali YAZICI
Head of Department

This is to certify that we have read the thesis "RETURN ON INVESTMENT IN TEST AUTOMATION VIA SELENIUM" submitted by Dilara ATEŞOĞULLARI and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science- Architecture/Doctor of Philosophy.

_____

Prof. Dr. Ali YAZICI
Supervisor

Examining Committee Members:

Prof. Dr. Alok Mishra
Software Eng. Department, Atilim University _____

Prof. Dr. Ali Yazıcı
Software Eng. Department, Atilim University _____

Assoc. Prof. Dr. Ahmet Murat Özbayoğlu,
Computer Eng. Department,
TOBB Economy & Technology University _____

Date: *24 January 2020*

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name : Dilara ATEŞOĞULLARI

Signature :

ABSTRACT

RETURN ON INVESTMENT IN TEST AUTOMATION VIA SELENIUM

ATEŞOĞULLARI, Dilara

M.S. Software Engineering Department

Supervisor : Prof. Dr. Ali YAZICI

January 2020, 58 pages

There are multiple approaches to software testing. One of these approaches, the subject of test automations, differs in theory and practice. The Return on Investment (ROI) calculations should be used to determine which automation tests are the most suitable for the given project. In this study, the manual and automated test results of test runs, which are suitable for server test automation, are calculated according to a certain formula. Based on the result of this calculation, our aim is to ensure that the selected test automation tests are suitable for return of investment. In the light of all these, the main subject of this study is to perform ROI studies in terms of test automation tests.

Keywords: Return on Investment, Test Automation, Manual Tests, Selenium, ROI of Automated Tests.

ÖZ

SELENIUM ARACILIĞI İLE YAPILAN TEST OTOMASYONLARININ YATIRIM GETİRİSİ

ATEŞOĞULLARI, Dilara
YL., Yazılım Mühendisliği Bölümü
Tez Yöneticisi : Prof. Dr. Ali YAZICI

Ocak 2020, 58 sayfa

Yazılım testine yönelik birden fazla yaklaşım vardır. Bu yaklaşımlardan biri olan otomasyon testlerinin konusu teori ve pratikte farklılıklar göstermektedir. Hangi otomasyon testinin elimizdeki projeye uygun olduğunu saptamak için Yatırım Getirisi (YG) hesaplamaları kullanılmalı ve üzerine karar vermelidir. Bu çalışmada, sunucu otomasyonuna uygun test çalışmalarının manevrası ve otomatik test sonuçları belirli bir formüle göre hesaplanmıştır. Bu hesaplamanın bir sonucu olarak, amaç seçilen otomasyon testlerinin Yatırımın iadesi için uygun olduğuna emin olmaktır. Tüm bunlar ışığında, bu çalışmanın ana konusu, otomasyon testleri açısından yatırım getirisi çalışmaları yapmaktır.

Anahtar Kelimeler: Yatırım Getirisi, Otomasyon Testleri, Manuel Testler, Selenium, Otomasyon Testlerinin Yatırım Getirileri

To Ali Ulvi and Necmiye Ateşoğulları , my guardian angels

you will always be my source of strength

ACKNOWLEDGMENTS

In the two and a half adventurous years of my academic journey, I was able to study in two prestigious universities, take a lot of interesting courses, meet and collaborate with many experts and write a thesis paper in my field of interest. My sincere gratitude goes to each and every one of them. I would like to extend my gratitude to my supervisor Prof. Dr. Ali Yazıcı for taking my raw ambition and supporting me to build it into something valuable. Dr. Ali your belief and confidence in my competence was everything I could ever ask for. I would also like to thank Kadir Herkiloğlu for his support and timely guidance in making this thesis paper a reality. Prof. Dr. Alok Mishra and Assoc. Prof. Dr. Murat Özbayoğlu thank you for being valuable committee members and for the stimulating comments. I want to thank my parents and my friend Can Kürşad for making my life more meaningful and joyful. Last but not least, I would like to thank all my instructors and classmates both in Atilim University. I am also equally grateful for the support of all the people whose names I have not mentioned here.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF SYMBOLS/ABBREVIATIONS

ROI	Return on Investment
T	Run Time of Each Case
P	Priority of Test Case
CoT	Cost of Test Case
NH	Number of Human Resource
NC	Human Resource Cost
DD	Daily Duration
MD	Monthly Duration
YD	Yearly Duration
CoI	Cost of Implementation
CoM	Cost of Maintenance
UT	Used Time (in hours)

CHAPTER 1

INTRODUCTION

Recently, with the progress made in software field, software testing field is also developing. The main purpose of this development is to minimize errors, but the main goal is to improve the quality of the product with software testing. There are many different components to measure the quality of a product. In [1] Jindal stated that, Software Development Life Cycle (SDLC) is a systematic system approach. Although the SDLC tells the system to improve the quality, it cannot find the problems within the system. Therefore, software testing plays an important role.

The software tests included in Jindal's definition are an important source for the quality of the product and the stability of the product in the future. This resource contains multiple different methods and approaches. Two of the most well-known of these methods are Manual and Automation tests. According to Mahajan Prasad et. al [2], Manual testing and automation testing are the two ways of testing. Manual testing is also called static testing. It is carried out by the tester. Automation testing is also called dynamic testing.

The advantages and disadvantages of the different test methods should be taken into consideration in the decision making stages of the software testing process. According to Islam [3], in the early days of the software testing process, software testing is provided by manual tests. However, with manual tests, it created an area suitable for error or exclusion from personal factors. Test automations have been developed to eliminate this gap.

According to Asfaw [4], some of the many benefits of test automation include: being more effective, accurate and cost effective than manual tests. A more error-free testing process can be managed, which will increase the financial gain . In addition, Rafi et. al. [5] has shown different benefits as a result of his research on this subject namely, reduced testing time, reusability of tests and reduction in cost.

In test automation process, the management of the process and the decision-making process are important. First of all, unfortunately, not all projects are suitable for automation. The different dynamics of the automation work to be done in this process should be examined first. One of these dynamics is the ROI step.

The aim of this study is to provide a new perspective on the calculation of Return of Investments of test automations and include the results of multiple samples created in this context. As a result of these examples, financial comparisons of manual and automation tests will be made and the obtained results and related arguments will be analyzed.

In this research, it is planned that the data to be used for this purpose will be collected from fields belonging to different sectors. One of these areas is e-commerce and another such field to obtain data is to be social media platforms. One of these results will be included in the calculation based on the results of the manual tests and the other one will be based on the results of the automation tests. Comparison of these two results and reaching a conclusion in this context are among the objectives of this study.

The organization of the thesis is as follows: In Chapter 2 background information is provided followed by Chapter 3 where ROI in test automation is discussed. Chapter 4 and 5 give a literature review and the related work respectively. Chapter 6 defines the research problem and the methodology employed. Chapter 7 is about the implementation details and results of the experiments conducted. In Chapter 8 comparisons between the manual testing and test automation in terms of ROI. Discussions and suggestions are provided in Chapter 9. Finally, conclusions and future work are given in Chapter 10.

CHAPTER 2

BACKGROUND INFORMATION

2.1. Software Testing

Software testing generally covers the testing and quality process of the software. The main objectives of these tests are to improve the quality, to confirm that the requirements are met and to produce a product that is as accurate as possible. Jamil and et al. states that testing is a process to check whether the initial requirements of the system to be tested are met. The simplest method is a process that covers validation and verification processes [6].

Many of the varieties of tests available today are available to meet the features and requirements that exist in different areas. It is possible to divide the existing test types into functional and non-functional tests.

White-box test techniques are applied during the integration and system test stages. This technique is most often applied in unit testing phase. Because, as a result of this structure, internal granules of the code can be tested. By analyzing the internal structure of the code, test conditions and test scenarios are produced and then checked. Black box testing is a test technique performed mostly by test engineers, without the need to know the internal structure of the system. The scope of the test is limited. Testing of certain modules may not be performed and may be skipped.

Non-functional tests can be defined as follows: It is defined as a type of Software test used to control functions such as performance tests, availability tests, reliability tests, etc. For non-functional tests it is possible to run non-operational tests, such tests serve to test the readiness of the system in accordance with non-functional parameters that cannot be addressed by functional tests [7].

Some of non–functional test types are given below [8]:

1. Usability Testing is about enabling real people to interact with a website, application, or other product you create, and to observe their behavior and response to it [9].

2. Load Testing is the system's response under varying load conditions, simulating multiple users accessing the application at the same time. This test usually measures the speed and capacity of the application [10].
3. Volume Testing is a type of test performed on high volume data [11].
4. Stress Testing is a test that confirms the stability and reliability of the system. This test is used to determine the system for robustness and fault management under heavy load conditions [12].
5. Performance Testing is a type of test used to measure how software applications perform under the expected workload [13].
6. Configuration Testing is a type of software test to find the most suitable configurations in which the system can run without any defects or errors [14].
7. Compatibility Testing is a test used to check if your software is running on different hardware, operating systems, applications, network environments, or devices [15].
8. Security Testing is a kind of test used to reveal the security vulnerabilities of the system and ensure the security of its data [16].

Functional tests is a kind of black box testing that checks whether systems or applications are performing operations that meet requirements [17].

Some of these functional test types are given below:

1. Unit Testing is a level of software testing in which individual units / components of each software are tested. With this type of test is to test whether each unit of software works correctly [18].
2. Sanity Testing is the type of test that is performed after receiving a software compilation that includes a small code change or functionality to detect that the errors have been corrected and no other problems have arisen due to these changes [19].
3. Smoke Testing is a kind of software test that consists of a series of non-comprehensive tests aimed at ensuring that the most important functions work [20].

4. Regression Testing is a type of software test that is used to determine whether the current features of a recent program or code change are affected [21].
5. Integration Testing is a method to detect faults in interfaces and devices interactions between integrated components or systems [22].

2.2.Manual Testing

This type includes the testing of the software manually i.e. without using any automated tool or any script. In this type the tester takes over the role of an end user and test the software to identify any unexpected behavior or bug. There are different stages for testing like unit testing, integration testing, system testing and user acceptance testing. Testers use test plan, test cases or test scenarios to test the software to ensure the completeness of testing. Manual testing also includes exploratory testing as testers explore the software to identify errors in it [23].

Manual tests are generally performed to meet the functional and non-functional requirements of the system. In order to carrying out these tests, the test runs are done manually without the assistance of any tool. Such tests may have some advantages and disadvantages. For example, it may be a disadvantage if testers misinterpret the results, such minor inaccuracies can be vital for the project.

2.3. Automation Testing

Automation testing which is also known as “test automation”, is when the tester writes scripts and uses other software to test the software. Automation Testing is used to re-run the test scenarios that were performed manually, quickly and repeatedly. Apart from regression testing, automation testing is also used to test the application in terms of load, performance and stress. It increases the test coverage, improves accuracy, saves time and money in comparison to manual testing [23].

We can use multiple tools and methods to automate our tests. Since these tools and methods will vary according to the project dynamics, they serve different approaches. There are tools such as Selenium [24] and Ranorex [25] for Web Browser tests and test tools like Appium [26] are preferred for mobile tests. Many test

automation tools use a common set of software mothodologies frameworks. These software methodology frameworks have very different approaches. Some of these software frameworks are as follows:

- Linear Scripting
- The Test Library Architecture Framework
- The Data-Driven Testing Framework
- The Keyword-Driven or Table-Driven Testing Framework
- The Hybrid Test Automation Framework [27]

2.4. Test Automation Tools

2.4.1. Selenium

Selenium is a software testing framework designed to test web applications, and it is widely applicable as the tests can be coded in various popular programming languages, as well as HTML tables. Then these tests can be run directly in most of the available popular web browsers, and it can also work on multiple operating systems such as Windows, Linux or MacOS [28]. The application offers multiple test types, such as:

- Selenium Grid
- Selenium RC (Remote Control)
- Selenium IDE

2.4.1.1. Selenium Grid

Selenium-Grid can be very useful to compare performance on different settings such as combinations of different operating systems and browsers, as it allows you to run your tests in a parallel fashion, on different machines against different browsers [29].

2.4.1.2. Selenium RC (Remote Control)

WebDriver is a web automation framework that allows you to execute your tests against different browsers. WebDriver also enables you to use a programming language in creating your test scripts [30].

Selenium RC is one of the most preferred types of Selenium. Generally, it is a structure that depends on the knowledge and ability of the test automation person because it works on code basis. However, since there are multiple workspaces, GUI and backend tests can be performed easily on this platform.

2.4.1.3. Selenium IDE

Selenium Integrated Development Environment is a plug-in that is installed on a web browser, and the working principle of this plug-in is to detect user behavior and to record and automatically run multiple test runs [31]. The ease of use of this plug-in is an advantage.

The use of tools and approaches described in this section plays an important role in the test automation process. It will be an important task for the test automation of the data to be made in this process and the ROI analysis. ROI section is described in greater detail in Chapter 3.

2.4.2. Ranorex

Ranorex is one of the known test tools. There are a number of advantages and disadvantages provided by this test tool. Some of these advantages are listed as follows.

1. It can support different platforms such as, Web, Desktop and Mobile platforms.

2. Parallel test execution is provided without problems.

3. It fully supports CI / CD tools.

4. It provides in-built reporting.

In addition to these advantages, it has a number of disadvantages. For example, Ranorex is a paid tools. For this reason, it is necessary to obtain licenses and test runs in this direction. Test runs are provided with record and reply features only. This can only be a more reliable and effective method of testing automations with multiple methods, especially scrip preparing [32-34].

2.4.3. Appium

Appium is another family of software testing tool. This tool has several different advantages and disadvantages. Some of these advantages are as follows:

- It supports different platforms [35].
- It supports different software languages [35].
- With Appium native and hybrid mobile platforms can be tested [36].

In addition to its main advantages, Appium is actually a convenient testing tool for mobile device testing. Under this circumstance, Selenium will not be able to provide many different features. For example, Appium provides convenience in test runs on IOS and Android devices. However, because it provides harness via Selenium Web Driver, it has the ability to test many tools with web interface. In addition, Selenium provides manual scripting software instead of copying user behavior. This feature provides the convenience of automating many test cases and provides an effective test run [37-39].

CHAPTER 3

RETURN ON INVESTMENT IN TEST AUTOMATION

3.1.Return on Investment (ROI)

Briefly, ROI is a return on investment analysis, also called cost-benefit analysis, to estimate and compare the costs and benefits of a phenomenon [40]. ROI is generally meaningful when a transaction is greater than the return of a transaction. Some simple formulas are used to calculate this. These formulas may vary according to the place of use and metrics. According to Investopedia, to calculate ROI, the benefit (return) of an investment is divided by the cost of the investment; the result is expressed as a percentage or ratio [41]. The basic ROI formula to use within the scope of this project is given below:

$$\text{ROI} = \frac{Gain - Investment}{Investment} \quad (1)$$

When Eq.1 is applied, one of the three different values is returned. The meanings of these values shown below.

1. ROI < 1
2. ROI = 1
3. ROI > 1

These calculation results show that the ROI < 1 points are much less than what we invest on our earnings. Due to the low return, the overall test automation software will be an unnecessary and introduces an extra burden if used.

The second result is ROI = 1. The interesting point is that the expenditures are equal to the gains. The decisions to be taken in this case vary. In such cases, the project should be decided after the project dynamics are examined whether it is appropriate to transfer the project to test automation or not. In some cases, it may be desirable to

implement test automation. However, the general opinion of the test community is that test automation tests are not suitable if ROI = 1.

For these reasons, it is unfortunate that automation tests in areas where test automation are ROI < 1 and ROI = 1 do not make sense.

The result of ROI > 1 means that the expected gains from applying test automation are higher than the expenditures made. The aim is to find the most profitable solution, therefore the comparison between the ROI values of manual testing as opposed to test automation testing is the key in determining which method is more suitable for a given project.

3.2.Return on Investment (ROI) Factors

There are many parameters used during the ROI calculation.

Some of the factors with high effect rates are given below [42-44]:

- Number of Test Cases : The number of test cases to be processed during the test run.
- Number of test Cycle : The number of tests to run in a test cycle.
- Priority of Test Cases : Priority of the test scenarios written during the test (may take different metrics such as Low - Medium - High or 1 – 2 – 3 – 4 – 5).
- Complexity of Test Cases : The complexity of the test scenarios made during the test and its effect on the run process.
- Test Results Complexity : The complexity of the test results during the test and the effects on the result analysis.
- Defect Reporting is reporting of errors found and the data to be used for this reporting.
- Duration of Test Cases Maintenance is maintenance times of automated test scenarios

These parameters are capable of increasing or decreasing the calculation result. The importance and requirements of these calculation components should be calculated according to the active attributes.

There are many actions that need to be taken before the start of manual and automation tests. These actions are taken into account as an effect factor when calculating ROI. These impact factors may be sub-components of multiple calculations, or they may be presented as inputs to the next calculation.

When calculating ROI, only cost information is not used as input. Some common factors apply to both types of tests, but some apply to only one type. For this reason, all of these factors should analyze for common and common test process structuring, and the common ones should select during ROI calculation.

The different test runs include multi testing effort (Manual ve Automation based) and calculations. Metrics that will take place in these exertions and calculations should be carefully selected and calculations should be made according to the selected metrics.

The following table shows the necessary factors for manual and automation test runs. However, in addition to the different types of run methods, there are many common factors. These common factors will be defined as ineffective elements during ROI calculation. These ineffective elements will not be included in the calculation. These infective elements are shown in Table 2.1.

Table 2.1 Common Test Process and Cost Factors [42]

Test Case Implementation
Test Result Analysis
Defect Reporting
Test Result Reporting

The necessary differences for the manual test and automation test process and necessary factors are shown in the table below.

Table 2.2 Differences between Test Process and Cost Factors [42]

Manual Factors	Automation Factors
Test Case Execution(Manual)	Test Case Execution(Automation)
	Test Maintenance
	Test Case Implementation

There are multiple calculation formulas and calculation operations to be used according to the differences described above. These formulas and calculation methods are shown in the Chapter 7.

CHAPTER 4

LITERATURE REVIEW

In this part, the literature review is presented in two sections: (1) studies about automation testing, (2) comparative studies about return on investment and testing.

4.1 Automation Testing

With the development of the software world, the value and possibilities of the studies conducted in the field of testing have increased. Many types of tests have been created to meet the requirement of the software industry. The most popular of these are functional tests, load tests and performance tests.

Most of these tests are performed manually. However, due to the constraints of the developing software field, software testing automation was discovered in order to perform these tests in a shorter time and with more effective methods.

According to Graham [45], test automations have many important and effective benefits. One of these is that tests can be carried out more extensively. In other words, testing with more variations in the input and environment of the tests is not possible for manual tests. Automated tests can provide a better environment for such actions.

Another goal of software testing automation is to automate multiple different test types. Barner, Weber and Keller [46] claims that, the goal of a successful automation strategy is to combine different approaches to test automation. This combination is usually system test automation, integration automation and unit test automation. In many cases, the test code used in this approach can be reused for other test types or approaches to make the combined strategy more effective.

Since test automations address many types of tests, they are likely to reduce the test load. The biggest workload of these tests is caused by the complex test cases. Multiple limitations may be encountered during the runtime of such cases. However, automation tests enable these complex structures to be solved by more convenient and efficient methods. Hoffman stated that [42] the perspective of automation testing depends on the implementation and execution of the test and the design of the organization's exterior coating .

Complicated tests enable calendar jams in some cases. In this case, carrying out a process behind the planned test process can bring material and moral damages from different aspects. About this topic, Pettichord reinterpreted the automation tests with an analogy. According to Pettichord [47] the fear and helplessness of the software testers lead them to the search for some kind of silver bullet to accelerate the work and gain rational speed. It will enable the tester to analyze many challenging processes with a simple method.

The general belief in the world of testing and automation is the opinion that automation will replace manual testing. Unfortunately, such an approach is not possible. Automations are likely to reduce the use of manual tests, but it is not possible to replace them. According to Grossman [48], the presence of automated tests does not eliminate manual tests. Some functions require still the presence of manual tests.

Considering all these different perspectives, software automation tests are seen as a necessary structure for overcoming the complex structure of software tests. In addition, it is thought that it will provide convenience and simplicity both as a schedule and structure as it addresses multiple test types.

4.2 ROI and Testing

According to the latest reports of ISTQB, the subject of test automation is gaining importance day by day. Percentage of those who want to improve themselves on test automation in 2015-2016 was 58.5% [49], it increased to 64.4% in 2017-2018 [50]. In any case, it shows that the issue of test automation becomes an important and up-to-date topic.While examining this type of testing, it is observed that it offers some convenience. One of them, the biggest gains in this process is that the costs of software testing processes will decrease. On this subject, Ramler and Wolfmaier expressed that [51]testing has the largest share in software development costs. Test automation is a good and unique way to minimize the cost of this factor.

However, there are some preparations to be made during this acquisition process. The most important of these is the calculation of ROI. On the topic, Sheth and Singh stated that [52] during the test automation, test scenarios are first manually tested and then coded to be automated - and run through the tested application. This saves

more time and resources. As a result of these transactions, automated tests are developed which provide a good ROI.

The data to be used as input in this ROI study is of great importance. While this data is being calculated, more than one hidden cost may surface. Karhu, Repo, Taipale and Smolander explained this as follows that due to automated software testing, more tests can be run in a short time. This can reduce testing costs and, as a result, improve quality. However, this may result in new costs that are not taken into account during the implementation, resulting from maintenance and training [53]. In the costs that must be given as input in the calculation of ROI, implementation and maintenance costs should be included.

CHAPTER 5
RELATED WORK

There are many different research and articles in the field of software testing automation. When these articles and researches examined, there are multiple different results. Result of these studies, software test automation and ROI calculations encountered only in a few articles. A brief summary of these articles are given below.

The methods used in the test automation processes are approaches and many other factors have the potential to affect the test automation ROI. The results of the three different articles examined in the following illustrate these differences more clearly.

Article 1 [42]

The main topic of this paper is that test automations in different fields may produce different results. In this system, firstly, software automation ROI calculated through tests and build tests. In addition, GUI software automation run on the same system and ROI was calculated as a result of the test run.

Automation of build tests results:

$ROI_{automation}$(in 6 months) = 1.013 [about break even]

$ROI_{automation}$(in 18 months) = 1.786 [80% return]

Automation of GUI tests results;

$ROI_{automation}$(in 6 months) = 0.874 [small loss]

$ROI_{automation}$(in 18 months) = 1.554 [55% return]

As demonstrated by these calculated ROI results, Build tests tolerated itself during the 6-month period, but during this period GUI tests were prepared on the same system, but with little loss, they could not tolerate them.

When these tests continued for 18 months, two tests were able to tolerate. However, the build tests again tolerated more than 80% of the ROI. In addition, GIU tests have achieved a tolerance of around 55%, ensuring that the ROI remains successful.

Nevertheless, even if it is on the same system, the software automations to perform in different test areas may have different results.

Article 2 [44]

The aim of this article is to observe the same test scenarios with two different test automation techniques.

For this purpose, a test automation ROI consisting of 250 test cycles was calculated. In this context, standard automation techniques and more curator and keen automation test techniques were used. These test automation runs' ROIs calculated in 24 and 48 months periods.

First, standard test automation ROI results were -27% over a 24-month period. This means that in 24 months, the test automation was able to cover only 73%. So at this point, test automation software does not seem to be a reasonable option.

When this process increased to 48 months, the ROI value declines up to -06%. However, the test automation does not recommended because the ROI still cannot pass to the positive side.

ROI execution for 24 and 48 months with standard automation technique;

$ROI_{automation}$(in 24 months) = -%27

$ROI_{automation}$(in 48 months) = -%06

In addition, when the same test scenarios were written with more curator and keen automation test techniques, it was observed that these ROI rates changed. As a result of the work done within 24 months, the ROI calculation is 30%. Result of the 48-month period, ROI is 58%.

ROI execution for 24 and 48 months with curator and keen automation technique;

$ROI_{automation}$(in 24 months) = %30

$ROI_{automation}$(in 48 months) = %58

These results shows that the test automation software used can have different results. The major distinction is that in different testing techniques, the ROI remains in the positive or negative area and influences the decision to stop or continue test automation execution.

Article 3 [54]

The aim of this article is to observe the effect of different test automation implemented dynamics on ROI results. Different dynamics may have different results in terms of ROI.

There are multiple quality standards in the implementation of test automation. In the absence of these quality metrics, the ROI may be positive or negative. The test automations written in the first instance, regardless of the No Cycle Time Benefit, remain in the positive area of the ROI of 752%. In addition, improvements in efficiency and productivity still play an important role.

In the second part, unlike the first metric, the effectiveness field also ignored. As a result, the first calculated ROI ratio, which is 322%, decreased. As ROI, which is still in the positive area, is still a predisposing investment area for automation, but it cannot overlook in a significant decline.

In the last stage, the productivity metric extracted in the code writing standards. ROI measurements made in this stage are down to 9%. At this stage, test automation software is no longer a reasonable option. Because the ROI falls into the negative area.

Table 5.1 ROI Measurements

Element	**Impact on ROI**		
No Cycle Time Benefit	√	√	√
No Effectiveness Benefit		√	√
14 Times slower test execution			√
Project ROI	752%	322%	9%
Month to Breakeven	7 months	7 months	22 months

Different results can be observed after precise ROI measurements. It ought to be noted that the quality metrics used during test automation could be have different results.

In these articles, different dynamics and metrics in the field of test automation and ROI discussed. When these results examined in general, different test areas, metrics and techniques affect the ROI closely.

Before reaching the conclusion, the calculation of the ROI should be completed considering the dynamics found in the project and the continuity of the test automation process should be rearranged according to these values. As illustrated in the sample articles, changing project dynamics can play a major role in the transition of ROI to positive or negative area. Result of this change, it will be predicted that test automation can be successful or unsuccessful.

CHAPTER 6
RESEARCH METHODOLOGY & RESEARCH PROBLEM

6.1. Research Problem

With the latest advances in technology, software testing automation has become a well-known concept. Due to this recognition, almost all projects and companies want to apply these concepts and technologies to their own projects. However, the application of this technology is not a simple process. Multiple expenses may arise for the companies to apply this technology. Some of these expenses are:

1. Registration Training
2. Software testing automation maintenance
3. Testing tools costs
4. Resource costs to use for test continuity.

However, there are multiple aspects to be considered when performing test automation. Failure to monitor these aspects accurately may result in a test automation failure. Some of the reasons for the failure of the test automation field are listed below:

- Focusing on tools and tasks instead of people
- Failing to adequately calculate and communicate ROI
- Not setting appropriate expectations
- Automating broken processes
- Not using the right tool [55]

This study is carried out in order to provide a more comprehensive perspective to the second item in the list.

The aim of this study is to determine whether the test automation is suitable for the project, based on the calculation of the ROI. In this context, comparisons between the manual and the automated tests are integral.

As a result of this study, it is aimed to show that the data obtained from two different fields as a sample are compatible with automation and after a while the test automations have started to have a self-tolerant effect.

6.2. Research Strategy

Multiple research strategies can use for the study conducted within the scope of this thesis. However, result of examining the selected studies, the Case Study was found to be appropriate. For this reason, the case study method created by using case study strategy.

6.3. Research Method – Case Study Based Research

The case study method allows researchers to examine data more specifically under specified characteristics and conditions. In this case, it is necessary to obtain realistic data in a limited number and context. It is formed by the reality of the analyzes made on the samples collected from real life [56].

6.3.1. Case Study Design

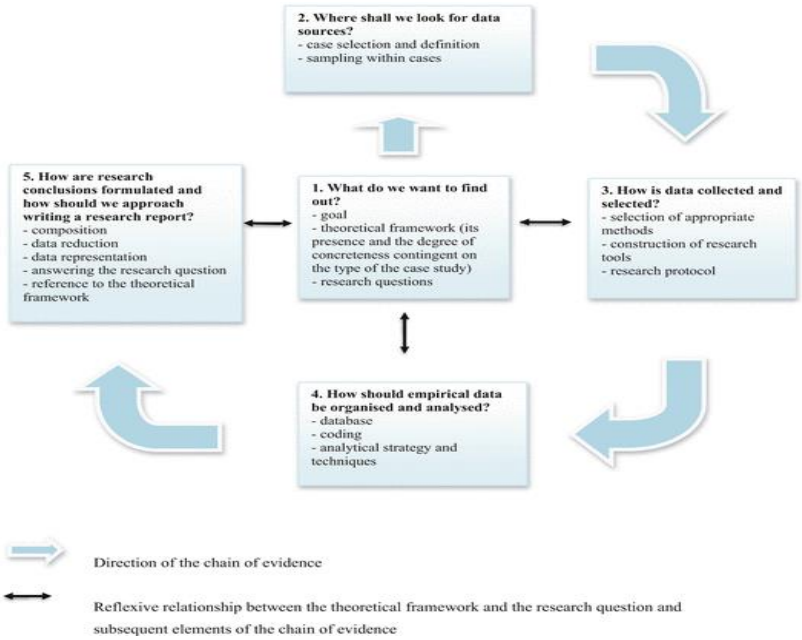

Figure 6.1 Case Study Path [57]

6.3.1.1. Objectives

The data used in this study was created by test experts. In this field, certain features of the case study method were used to create data. First of all, case study planning is required. The planning steps utilized for this research are given below.

1. Developing the research question
2. Determination of units of analysis
3. Determination of the human resource
4. Collecting Data and Associating the Collected Data with Sub-Problems
5. Data Analysis and Interpretation

6.3.1.2. Research Question

Two research questions created under this scheme are given below:

1. How to calculate ROI in test automations?
2. How should the rest of the investment be calculated as a result of the data resulting from the calculation?

6.3.1.3. Case Study Method

In this research stage, the Comparative Case Studies type was used as part of the Collecting Data and Sub-Problems step. Case study method is used for two or more case studies and comparison of these studies. For the purpose of this study, a second study is carried out in order to make comparisons.The second workspace, the first selected workspace, and an area where the properties are the same or not, are selected.In order to make generalizations on a correct basis, the second working area should be selected from the first working area [58].

Multiple - Embedded was selected as the case study design in this research. In this context, the data collected were collected from 2 different fields and collected by 2 different methods for each field. The Multi-Embeded case study structure is shown in Figure 6.2 and Figure 6.3. For detailed information on data collectin see Section 6.4. in this chapter.

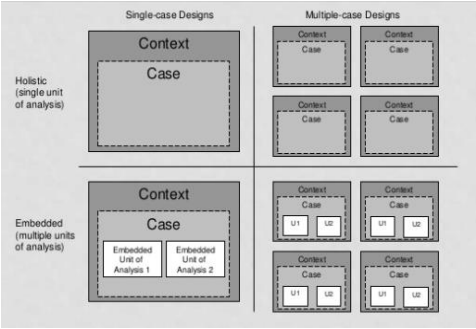

Figure 6.2 Case Study Design [57]

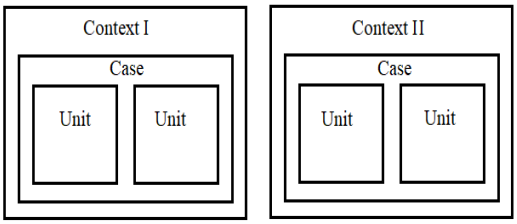

Figure 6.3 Implemented Case Study Design

6.3.2. Case Study Object and Test Environment

Under this heading, the decisions of the platforms where the test data are collected and the information of the environment used in the collection stage will be explained in detail.

6.3.2.1. Manual Testing

Special preparation is not required for environments used during manual testing. During these runs the Chrome browser was used on a computer with operating system Windows 10 operating system. Data were collected by cumulative conditioning of interconnected portions of test cases.

6.3.2.2. Automation Testing

In the process of collecting the data required for automation tests, different components have to be used. The most important of these components are the necessary automation codes written in Selenium using the Java programming language. As a result of this, the tests were run on Chrome on Windows 10.

The web applications where the data were to be retrieved were applied on a page basis. Certain operations are performed on the pages. The test cases from this situation are produced in accordance with this structure. Another effect of this was that during the implementation of the codes in Selenium, multiple design patterns were examined. However, it has been decided to use Page Object Model and Page Factory Model, which are required due to the structure of the samples.

6.4. Data Collection

Multiple methods were used in the collection of the research to the largest building block. These methods have different structures for manual and automation tests. Multiple methods were used in the collection of the research to the largest building block. These methods have different structures for manual and automation tests.

6.4.1. Manual Testing

Manual test runs can be defined as the only manual data collection phase without any auxiliary tools. Within the scope of this study, these data were collected by more than one tester and used as the average.

The steps for collecting the manual testing data were as follows:

1. Determination of the application to be tested

2. Determination of test scenarios

3.Writing the scenarios ,which are given in Appendix C, to be tested and placing them in the order of importance

4. Running test cases

5. Collecting the results of each test

6. Reach final data by calculating the results of the tested data

The results of the manual test run were calculated cumulatively and a cumulative data analysis was used.

6.4.2. Automation Testing

There is a need of using different tools for automation testing. The codes written in this study were written with the help of the Selenium test tool. In this context, the runs were collected with the help of Selenium. The data collection steps for the test automation are listed below:

The steps for collecting the automation testing data were as follows.

1. Determination of the application to be tested

2. Determination of test scenarios

3.Writing the scenarios ,which are given in Appendix C, to be tested and placing them in the order of importance

4. Selection of tools and methods necessary to automate scenarios of tests

5. Writing automation tests

6. Running written test scenarios

7. Collecting the data of the scenarios that are run

8. Reach final data by calculating the results of the tested data

Due to the nature of the operations, the continuity of the runs was ensured by the fact that the test cases were the prerequisites of each other instead of a single run. The result of the previous test case can refer to the starting point of the next test step. For this reason, test runs become faster and more efficient.

CHAPTER 7

IMPLEMENTATION & RESULTS

7.1. Return on Investment (ROI) Formulas

Firstly, there are many calculation formulas used for manual and automation tests. These calculations represented by multiple calculations. The common calculation formulas will be shown under Common ROI Formulas. Calculations to use in different and special terms in automation specific is demonstrated under the title of Discrete ROI Formulas.

7.1.1. Common ROI Formulas

There are multiple calculation approaches for ROI. However, here more than one calculation will be used. These calculations are given below in 4 (four) main formulas.

1. Calculation of each Test Case Cost
2. Calculation of Monthly Cost
3. Calculation of Yearly Cost
4. Calculation of Whole Cost

The result of these formulas will be corresponded to the *Test Case Execution* effort and will be used in both automation and manual testing.

Common Formula 1: Calculation of each Test Case Cost

In this formula, the test times used as input metrics according to their importance. A result of these inputs, we have called F_1. This result will also be used as input to the next formula. This formula calculated on a daily basis.

$$F1 = T \times P \tag{2}$$

where T is Run Time (in second) for Each Case and P is Priority (from 1 to 5 - 5 is higher 1 is lower) of Test Case.

Common Formula 2: Calculation of Monthly Cost

The purpose of this formula is to calculate the monthly test run time. As an input to this formula, the output of the previous calculation is F_1 and the annual run time will be used. The output of this formula is F_2. In addition, also use it as input for the next formula.

$$F2 = F1 \times MD \quad (3)$$

where F1 as defined in Equation (2) and MD is Monthly Duration (30 days).

Common Formula 3: Calculation of Yearly Cost

The purpose of this formula is to calculate the annual test run time. As an input to this formula, the output of the previous calculation is F_2 and the annual run time will be used. The output of this formula calls F3. Moreover, also use it as input for the next formula.

$$F3 = F2 \times YD \quad (4)$$

where F2 as defined in Equation (3) and YD is Yearly Duration (12 months).

Common Formula 4: Calculation of Whole Cost

The main purpose of this formula is to perform an individual cost analysis on an annual basis after this calculation. The cost analysis of the person reflects the cost of the test run CoT on this project.

$$CoT = F3 \times (NH \times HC) \quad (5)$$

Where F3 as defined in Equation (4), NH is Number of Human Resource and HC is Human Resource Cost.

Table 7.1 Abbreviation and Standard Values for Common ROI Formulas

Abbreviation	Description	Standard Values
T	Run Time of Each Case	-
P	Priority of Test Case	-
CoT	Cost of Test Case	-
NH	Number of Human Resource	1 person
HC	Human Resource Cost	50 TL
DD	Daily Duration	8 hour
MD	Monthly Duration	30 days
YD	Yearly Duration	12 months

The values shown in Table 7.1 are the values utilized in Equation 2, Equation 3 and Equation 4. These data keep constant throughout the research and used as constant data in the calculations.

7.1.2. Discrete ROI Formulas

In addition to the Test Case Execution metric, which will be input in ROI calculations, there are also other sub-calculation formulas used for automation. These calculation topics are as follows:

- Test Case Implementation
- Test Maintenance

Discrete Formula 1: Test Case Implementation

The main purpose of this formula is to calculate the expenses that will occur during the writing of test automation codes and to include the metrics to create for this process in the test automation process.

$$CoI = UT \times (NH \times HC) \quad (6)$$

where UT is Used Time, NH is Number of Human Resource for Implement and HC is Human Resource Cost for Implement.

Discrete Formula 2: Test Maintenance

The main purpose of this formula is to calculate the expenses that will occur in the maintenance phase of the previously written test automation codes and to include the metrics to create for this in the test automation process.

$$CoM = UT \times (NH \times HC) \quad (7)$$

where

UT is Used Time
NH is Number of Human Resource for Maintenance
HC is Human Resource Cost for Maintenance

Table 7.2 Abbreviation and Standard Values for Discrete ROI Formulas

Abbreviation	Description	Standard Values
CoI	Cost of Implementation	-
CoM	Cost of Maintenance	-
NH	Number of Human Resource	1 person
HC	Human Resource Cost	50 TL
UT	Used Time (in hours)	-

The standard values given in Table 7.2 are used in Equation 6. This data is the calculation component that will keep constant in the calculations.

7.2. Experiment Samples

In this study, samples selected from two different areas were studied. These examples come from an e-commerce application and the other one from social media. The purpose of the selection of these examples is that the human factor plays an important role in the field. In other words, both social media and e-commerce areas

due to the human factors are error-prone systems. Considering the fact that the systems that are open to this error contain many sensitive data, it is appropriate to collect samples from these areas.

7.2.1. Manual Test ROI Calculation

Metrics collected after manual test runs will use a number of specific calculations. The process of giving these calculations as input in multiple points shown below. First, data metrics of previously prepared test scenarios (see Appendix 1 and Appendix 2) are gathered.

Sample 1

Run Time for Each Case value is **Manual Run Time (in seconds)** in the See Appendix A- Sample 1 - Manual F_1 Calculation

Priority of Test Case value is **Priority of the Test Cases** in the See Appendix A- Sample - Manual 1 F_1 Calculation

Calculation of each Test Case Cost

By using Equation 2, the cost of each test case is calculated for manual testing. This data will provide as input to subsequent calculations.

See Appendix A- Sample 1 – Manual F_1 Calculation

F_1 = 10255 seconds

F_1 = 10255 seconds = 170 minutes 55 seconds = 2 hours 51 minutes per day

Calculation of Monthly Cost

By using Equation 3, monthly cost is calculated using Equation 2 calculation result for manual testing. This data will provide as input to subsequent calculations.

F_2 = 2h 51m * 30

F_2 = 85 hour 30 minutes per month = 10 days 5 hour 30 minutes per month (8 hours for each day)

Calculation of Yearly Cost

By using Equation 4, yearly cost is calculated using Equation 3 calculation result for manual testing. This data will provide as input to subsequent calculations.

F_3 = 85 hour 30 minutes * 12 = 1026 hours per year = 128 days 2 hours per year (8 hours for each day)

Calculation of Whole Cost

By using Equation 5, whole cost calculated with Equation 4 calculation result for manual testing. This data will provide as input to subsequent calculations.

CoT= 1026 * (1*50)

CoT= 51300 TL per year.

Sample 2

Run Time for Each Case value is **Manual Run Time (in second)** in the See Appendix A- Sample 2 - Manual F_1 Calculation

Priority of Test Case value is **Priority of the Test Cases** in the See Appendix A - Sample 2 - Manual F_1 Calculation

Calculation of each Test Case;

By using Equation 2, the cost of each test case calculated for manual testing. This data will provide as input to subsequent calculations.

See Appendix A- Sample 2 - Manual F_{F1} Calculation

F_1 = 9775 seconds

F_1 = 9775 seconds =162 minutes 55 seconds = 2 hours 43 minutes per day

Calculation of Monthly Cost

Via using Equation 3, monthly cost calculated with Equation 2 calculation result for manual testing. This data will provide as input to subsequent calculations.

F_2 = 2h 43m * 30

F_2 = 81 hour 30 minutes per month = 10 days 1 hour 30 minutes per month (8 hours for each day)

Calculation of Yearly Cost

By using Equation 4, yearly cost calculated with Equation 3 calculation result for manual testing. This data will provide as input to subsequent calculations.

F_3 = 81 hour 30 minutes * 12 = 978 hours per year = 122 days 2 hours per year (8 hours for each day)

Cost of Whole Test Cases

By using Equation 5, whole cost calculated with Equation 4 calculation result for manual testing. This data will provide as input to subsequent calculations.

CoT= 978 * (1*50)

CoT= 48900 TL per year.

7.2.2. Automated Test ROI Calculation

Metrics collected after automated test runs will use a number of specific calculations. The process of giving these calculations as input in multiple points shown below. First, data metrics of previously prepared test scenarios (see Appendix A) are gathered.

7.2.2.1. Test Automation Run Calculation

Sample 1

Run Time for Each Case value is **Manual Run Time (in second)** in the See Appendix A- Sample 1 – Automation F_1 Calculation

Priority of Test Case value is **Priority of the Test Cases** in the See Appendix A- Sample 1 – Automation F_1 Calculation

Calculation of each Test Case Cost

By using Equation 2, the cost of each test case calculated for automation testing. This data will provide as input to subsequent calculations.

See Appendix A- Sample 1 -Automation F_1 Calculation

F_1 = 6411 seconds

F_1 = 6411 seconds = 106 minutes 51 seconds = 1 hours 47 minutes per day

Calculation of Monthly Cost

Via using Equation 3, monthly cost calculated with Equation 2 calculation result for automation testing. This data will provide as input to subsequent calculations.

F_2 = 1h 47m * 30

F_2 = 53 hour 30 minutes per month = 6 days 5 hour 30 minutes per month (8 hours for each day)

Calculation of Yearly Cost

By using Equation 4, monthly cost calculated with Equation 3 calculation result for automation testing. This data will provide as input to subsequent calculations.

F_2 = 53 hour 30 minutes * 12 = 642 hours per year = 80 days 2 hours per year (8 hours for each day)

Calculation of Whole Cost

By using Equation 5, monthly cost calculated with Equation 4 calculation result for automation testing. This data will provide as input to subsequent calculations.

CoT= 642 * (1*50)

CoT= 32100 TL per year.

Sample 2

Run Time for Each Case value is **Manual Run Time (in second)** in the See Appendix A- Sample 2 – Automation F_1 Calculation

Priority of Test Case value is **Priority of the Test Cases** in the See Appendix A- Sample 2 - Automation F_1 Calculation

Calculation of each Test Case;

By using Equation 2, the cost of each test case calculated for automation testing. This data will provide as input to subsequent calculations.

See Appendix A- Sample 2- Automation F_1 Calculation

F_1 = 4132 seconds

F_1 = 4132 seconds =68 minutes 52 seconds = 1 hours 09 minutes per day

Calculation of Monthly Cost

Via using Equation 3, monthly cost calculated with Equation 2 calculation result for automation testing. This data will provide as input to subsequent calculations.

F_2 = 1h 09m * 30

F_2 = 34 hour 30 minutes per month = 4 days 2 hour 30 minutes per month (8 hours for each day)

Calculation of Yearly Cost

By using Equation 4, monthly cost calculated with Equation 3 calculation result for automation testing. This data will provide as input to subsequent calculations.

F_3 = 34 hour 30 minutes * 12 = 414 hours per year = 51 days 6 hours per year (8 hours for each day)

Cost of Whole Test Cases

By using Equation 5, monthly cost calculated with Equation 4 calculation result for automation testing. This data will provide as input to subsequent calculations.

CoT= 414 * (1*50)

CoT= 20700 TL per year.

7.2.2.2. Automated Test Implementation and Maintenance Calculation

In In this section, in addition to the cost of the run, test automation codes implementation and maintenance costs calculated. However, when these 3 costs are collected, the actual cost of test automation will be calculated.

Case-based coding times of the codes written in this process given in **Sample 1 Automation - Implementation and Maintenance** and **Sample 2 Automation - Implementation and Maintenance** table in Appendix B. In this table, the calculation of the writing time cost using the data to this is as follows.

Common Formula: Implementation and Maintenance Time

This formula will used to calculate the total implementation and maintenance time in hours. In this process, the cost of the implantation and maintenance process will calculated based on hours.

Hour Calculation;

$$F4 = \frac{Test\ Case\ 1\ Implementation\ or\ Maintenance\ Time + \cdots + Test\ Case\ n\ Implementation\ or\ Maintenance\ Time}{60} \quad (8)$$

Cost Calculation;

$$F5 = \left(\frac{Test\ Case\ 1\ Implementation\ or\ Maintenance\ Time + \cdots + Test\ Case\ n\ Implementation\ or\ Maintenance\ Time}{60} \right) \times HC \quad (9)$$

Where HC is Human Resource Cost for Implement or Maintenance.

Sample 1

In this area, the necessary calculations for sample 1 made. The necessary information provided in the **Sample 1 Automation - Implementation and Maintenance** tables when performing these calculations.

Hour Calculation for Implementation;

$$F4 = \frac{Sum\ of\ Implementation\ Time}{60} \quad (10)$$

Cost Calculation for Implementation;

$$F5 = Sum\ of\ Implementation\ Time\ (hour\ based) \times HC \quad (11)$$

where HC is Human Resource Cost for Implement.

Hour Calculation for Maintenance;

$$F4 = \frac{Sum\ of\ Maintenance\ Time}{60} \quad (12)$$

Cost Calculation for Maintenance;

$$F5 = Sum\ of\ Maintenance\ Time\ (hour\ based) \times HC \quad (13)$$

Where HC is Human Resource Cost for Maintenance.

Implementation Time Calculation

The Implementation Time calculated for Sample 1 using Equation 10 and Equation 11.

F_4 = 15448 / 60 = 257 hours 28 mins

F_5 = 257 hours 28 mins * 50 = 12875 TL

Maintenance Time Calculation

The Implementation Time calculated for Sample 1 using the Equation 12 and Equation 13.

F_4 = 10222 / 60 = 170 hours 22 mins

F_5 = 170 hours 22 mins * 50 = 8537 TL

Sample 2

In this area, the necessary calculations for sample 2 make. The necessary information provide in the **Sample 2 Automation - Implementation and Maintenance** tables when performing these calculations.

Implementation Time Calculation

The Implementation Time calculated for Sample 2 using the Equation 10 and Equation 11.

F_4 = 11896 / 60 = 198 hours 16 mins

F_5 = 198 hours 16 mins * 50 = 9913 TL

Maintenance Time Calculation

The Implementation Time calculated for Sample 2 using the Equation 12 and Equation 13 formulas.

F_4 = 6469 / 60 = 107 hours 49 mins

F_5 = 107 hours 49 mins * 50 = 5362 TL

7.2.2.3. Automated Test Total Cost Calculation

As a result, of the calculation of test automation data, total calculation is required annually.

The data to use by filling in for this calculation are as follows;

1. Cost of Run

2. Cost of Implementation

3. Cost of Maintenance

Total cost calculated by adding these data.

Total Cost of Automation = Cost of Run + Cost of Implementation + Cost of Maintenance

Total test automation calculation according to this formula given in the table below.

Table 7.3 Total Cost of Automation

	Sample 1	**Sample 2**
Cost of Run	32100 TL	20700 TL
Cost of Implementation	12875 TL	9913 TL
Cost of Maintenance	8537 TL	5362 TL
Total Cost (Yearly)	53512 TL	35975 TL

CHAPTER 8

COMPARISON OF THE RESULTS

In this section, comparative calculations of manual and test automation test runs and planning results will make. The table below summarizes the calculations of manual and automation tests. The existing results in this table will use as input.

Table 8.1 Summary of Calculation

	Manual Run		**Automation Run**	
	Sample 1	**Sample 2**	**Sample 1**	**Sample 2**
Cost of Manual Run	51300 TL	48900 TL		
Cost of Automation Run			32100 TL	20700 TL
Cost of Implementation			12875 TL	9913 TL
Cost of Maintenance			8537 TL	5362 TL

Test automation process should be start with a manual test run. After these processes, the automation of the cases to automate should implement. After this process, the implemented test automation maintain and the results should obtain. In addition, maintenance should carry out where necessary. All of these processes need to extend. The calculation for this given below.

Two different formulas used to calculate the Return on Investment process in software test automations. The first one is the formula in which income and expenses given as input in the first year. In this formula, Implementation data given as input to the expenditure section.

1st Year ROI Formula;

$$ROIAutomation(1st\ Year) = \left(\frac{(Cost\ of\ Manuel\ Run - Cost\ of\ Automated\ Run) - (Cost\ of\ Implementation + Cost\ of\ Manuel\ Run)}{Cost\ of\ Implentation + Cost\ of\ Manuel\ Run} \right) \quad (14)$$

The second formula includes the calculation for the years after the first year. The most important area in this section is to add the cumulative earnings of previous years to the earnings section. This other difference is the implementation costs that added to the calculation formula of the first year. These expenses are not included in this section. However, Maintenance expenses provided as expense input to this formula instead.

Nnd Year ROI Formula;

$$ROIAutomation(Nst\ Year) = \left(\frac{Gain\ of\ Previos\ Year + \left(\begin{matrix}(Cost\ of\ Manuel\ Run - Cost\ of\ Automated\ Run) - \\ (Cost\ of\ Implementation + Cost\ of\ Manuel\ Run)\end{matrix}\right)}{Cost\ of\ Implentation + Cost\ of\ Manuel\ Run}\right) \quad (15)$$

These two calculation formulas applied Sample 1 and Sample 2 in below.

Sample 1

1st Year Automation Calculation;

1st year automation value of Sample 1 calculated using Equation 14.

$ROI_{Automation(1st\ year)}$ = (((51300-32100)- (51300+12875))/ (51300+12875))*100

$ROI_{Automation(1st\ year)}$ = -70% Lost

2nd Year;

2nd year automation value of Sample 1 calculated using Equation 15.

$ROI_{Automation(2nd\ year)}$ = (((19200 + (51300-32100)) - (51300 + 8537)) / (51300 + 8537))*100

$ROI_{Automation(2nd\ year)}$ = -36% Lost

3rd Year;

3rd year automation value of Sample 1 calculated using Equation 15.

$ROI_{Automation(3rd\ year)}$ = (((38400+ (51300-32100)) - (51300 + 8537))/(51300 + 8537))*100

$ROI_{Automation(3rd\ year)}$ = -4% Lost

4th Year;

4th year automation value of Sample 1 calculated using Equation 15.

$ROI_{Automation(4th\ year)}$ = (((57600+ (51300-32100)) - (51300 + 8537))/(51300 + 8537))*100

$ROI_{Automation(4th\ year)}$ = 28% Gain

Sample 2

1st Year;

1st year automation value of Sample 2 calculated using Equation 14.

$ROI_{Automation(1st\ year)}$ = (((48900 -20700) – (48900 + 9913))/ (48900 + 9913))*100

$ROI_{Automation(1st\ year)}$ = -52% Lost

2nd Year;

2nd year automation value of Sample 2 calculated using Equation 15.

$ROI_{Automation(2nd\ year)}$ = (((28200 +(48900 -20700)) – (48900 + 9913)) / (48900 + 9913))*100

$ROI_{Automation(2nd\ year)}$ = -4% Lost

3rd Year;

3rd year automation value of Sample 2 calculated using Equation 15.

$ROI_{Automation(3th\ year)}$ = (((56400+(48900 -20700)) – (48900 + 9913))/ (48900 + 9913))*100

$ROI_{Automation(3th\ year)}$ = 44% Gain

4th Year;

4th year automation value of Sample 2 calculated using Equation 15.

$ROI_{Automation(4th\ year)}$ = (((84600+(48900 -20700)) – (48900 + 9913))/ (48900 + 9913))*100

$ROI_{Automation(4th\ year)}$ = 92% Gain

Result of test automations for Sample 1, there is a financial loss of -70% of the 1st year. This material loss declined to around -36% in the 2nd year. As the difference between the test automation process has started to tolerate itself. At the end of the 3rd year, the difference decreased to near 0 (zero). At the end of the 3rd year, the result is -4%. Result of these calculations, the test automation process has managed to tolerate itself for almost 3 years. As seen in the results of the 4th year, the test automation process has fully tolerated itself and has shifted to the positive area, i.e. the gain area. It proven that after this time it will bring harm, not gain, as long as the existing test automation remains constant.

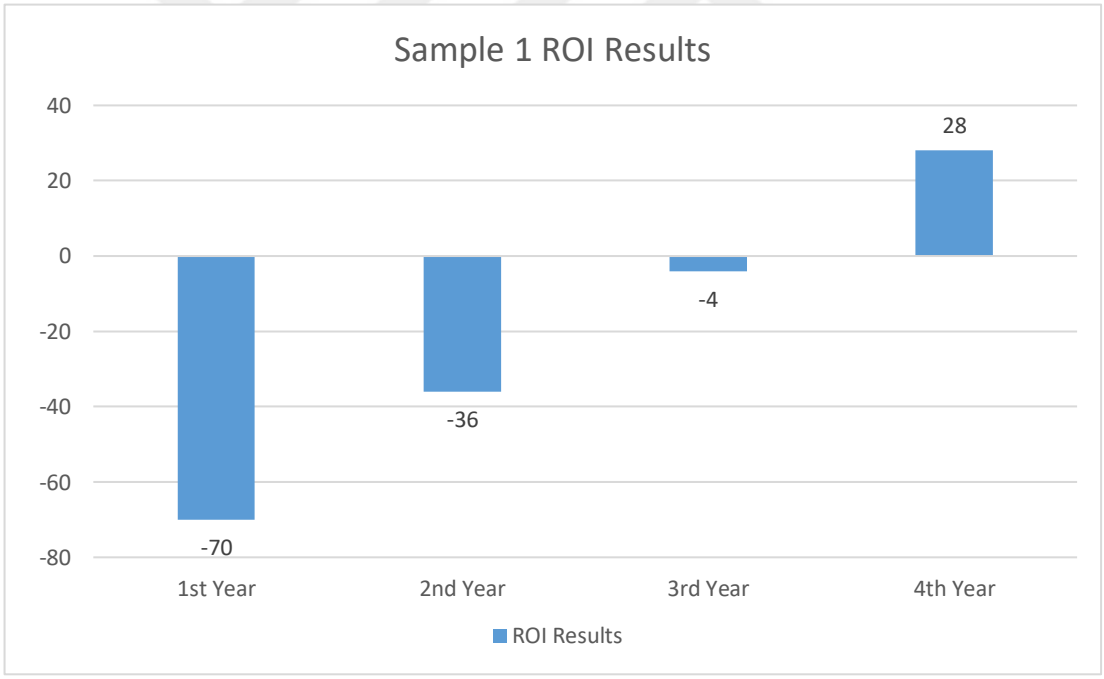

Figure 8.1 Sample 1 - Return on Investment for Automation

The results of the test automation study for Sample 2 are as follows. Financial data for the first year indicate - 52%. This means that the test automation work has caused damage. However, in the second year studies, this value has decreased to -4%.

This value, which approaches almost 0 (zero) value, indicates that test automation expenditures are beginning to tolerate blood. In the third year of the test automation result, there is a transition from the loss side to the gain side. The 3rd year test automation's results mean that with 44% it is now profitable. At the same time, the test automation operations performed in the fourth year show the profit of test automation with 92% result.

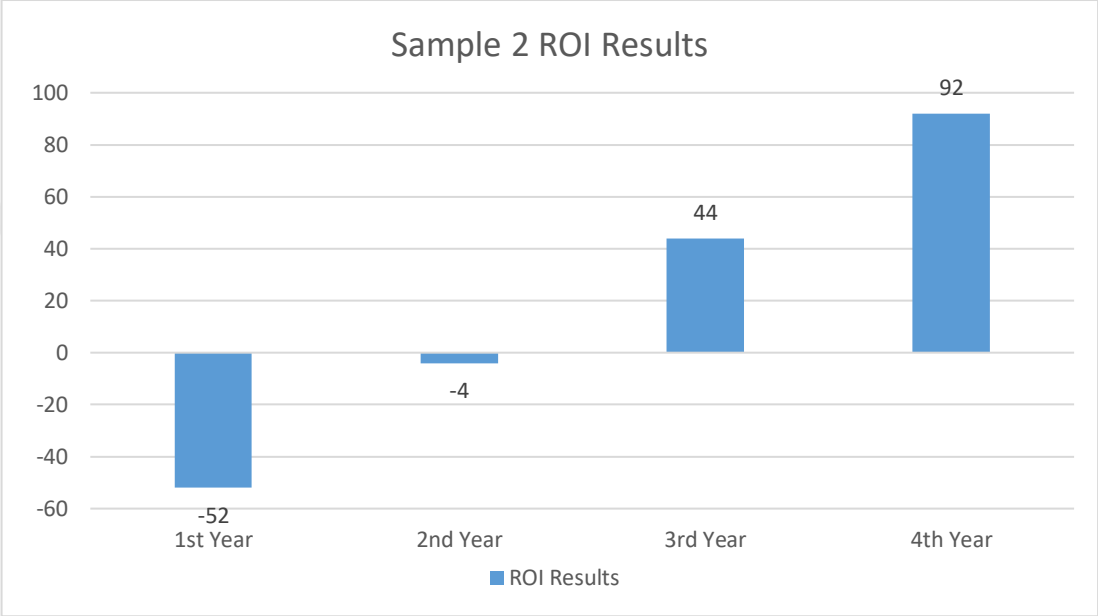

Figure 8.2 Sample 2 - Return on Investment for Automation

CHAPTER 9

DISCUSSIONS AND SUGGESTIONS

In this thesis, many test tools and software test automation examples are examined. There are many software areas where these approaches can be applied. Some of these areas are social media, banking and finance, games and e - commerce. Considering the accessibility of these areas, games and social media come to the fore. However, when security is another criterion, banking and e-commerce fields are at the forefront. Social media and e-commerce are at the forefront in terms of sustainability, which is a third criterion. Within the scope of this subject, the choice of study area was determined as social media and e-commerce, as a result of these three criteria.

In software test automations, selection of area and scenario is of great importance. As a matter of fact, every detail existing in the system should be tested thoroughly. However, unfortunately, not every detail can be automated. In such cases, determining the test scenarios that need to be automated would be beneficial both for the operation of the automation processor and for software test automation ROI. The intended scenarios might have a positive contribution to ROI as well as a negative outcome. When calculating ROI, it is necessary to make a calculation that is objective and free from manipulation, otherwise it can lead to unsuccessful scenarios. Some assumptions related to expenses are made during the ROI calculation. These assumptions must be realistic and acceptable in order to get an accurate ROI calculation.

As mention before in this study, it is not feasible to implement test automation where ROI < 1. When ROI = 1 the literature suggests that software test automation process may or may not be implemented depending on your project dynamics. In my opinion, in this situation, implementing test automation is a more rational approach. In the long term, project team can eliminate human based factors (increasing wages, employee’s performance related issues) with test automation.

CHAPTER 10

CONCLUSIONS AND FUTURE WORK

In this thesis, we obtained the results of manual and automation tests and return on investment calculations for two different examples. Our aim was to deduct whether the selected projects are feasible for test automation or not, based on their ROI calculation. The field of these projects can also have an effect on the ROI results. Within the scope of this study, the test cases of the identified projects are prepared. These cases are applied to manual and test automation tests respectively. The results of these tests are used as input for the return on investment calculation.

In the context of this study, different projects ROI's are calculated. As a result of this ROI calculation, we obtained different financial feedbacks from each projects. These feedbacks have shown us that when ROI is higher than 1 financial returns are accumulated over a long period of time.

As future work, this work can be improved in two ways. Firstly, the number of test samples obtained from existing projects can be increased in order to get more accurate data. As a second way, a new project can be selected from a different area. In this case, a new field and project analysis can be made. Since different results will be obtained, it will be easier to reach a general opinion in terms of test automation.

REFERENCES

[1] Jindal , T. (2016). Importance of Testing in SDLC. *International Journal of Engineering and Applied Computer Science (IJEACS)*, *1*(2), 54–56. Retrieved from www.ijeacs.com

[2] Mahajan Prasad, et al. “Automation Testing In Software Organization. ” International Journal of Computer Applications Technology and Research, vol. 5, no. 4, Jan. 2016, pp. 198–201., doi:10.7753/ijcatr0504.1004.

[3] Islam, Nazia, "A Comparative Study of Automated Software Testing Tools" (2016). Culminating Projects in Computer Science and Information Technology. 12. https://repository.stcloudstate.edu/csit_etds/12

[4] Asfaw, Daniel L. “Benefits of Automated Testing Over Manual Testing. ” International Journal of Innovative Research in Information Security (IJIRIS) , vol. 1, no. 2, Jan. 2015, pp. 5–13. ISSN: 2349-7017, www.ijiris.com.

[5] Rafi, Dudekula Mohammad, et al. “Benefits and Limitations of Automated Software Testing: Systematic Literature Review and Practitioner Survey.” 2012 7th International Workshop on Automation of Software Test (AST), 2012, doi:10.1109/iwast.2012.6228988.

[6] Jamil, Muhammad Abid, et al. “Software Testing Techniques: A Literature Review.” 6th International Conference on Information and Communication Technology for The Muslim World, 2016, pp. 177–181. 978-1-5090-4521-1, doi:DOI 10.1109/ICT4M.2016.40.

[7] “What Is Non Functional Testing? Types with Example.” Guru99, https://www.guru99.com/non-functional-testing.html.

[8] Mahajan, Prasad. “Different Types of Testing in Software Testing.” International Research Journal of Engineering and Technology (IRJET) , vol. 03, no. 04, Apr. 2016, pp. 1661–1664. e-ISSN: 2395 -0056.

[9] “Usability Testing 101: What It Is & What It's NOT (2020 Guide).” Usability Testing 101: What It Is & What It's NOT (2020 Guide), https://www.hotjar.com/usability-testing.

[10] “Load Testing Complete Guide for Beginners.” Software Testing Help, 10 Nov. 2019, https://www.softwaretestinghelp.com/load-testing/.

[11] “What Is Volume Testing? Definition of Volume Testing, Volume Testing Meaning.” The Economic Times, https://economictimes.indiatimes.com/definition/volume-testing.

[12] “What Is STRESS Testing in Software Testing? Tools, Types, Examples. ” Guru99, https://www.guru99.com/stress-testing-tutorial.html.

[13] “Performance Testing Tutorial: What Is, Types, Metrics & Example.” Guru99, https://www.guru99.com/performance-testing.html.

[14] “What Is Configuration Testing? Example Test Cases.” Guru99, https://www.guru99.com/configuration-testing.html.

[15] “What Is Compatibility Testing? Forward & Backward Testing (Example). ” Guru99, https://www.guru99.com/compatibility-testing.html.

[16] “Security Testing.” Software Testing Fundamentals, 13 July 2018, http://softwaretestingfundamentals.com/security-testing/.

[17] Gandhi, Dev, et al. “Functional Testing: A Complete Guide with Types and Example.” Software Testing Help, 10 Nov. 2019, https://www.softwaretestinghelp.com/guide-to-functional-testing/.

[18] “Unit Testing.” Software Testing Fundamentals, 3 Mar. 2018, http://softwaretestingfundamentals.com/unit-testing/.

[19] “Sanity Testing Vs Smoke Testing: Introduction & Differences.” Guru99, https://www.guru99.com/smoke-sanity-testing.html.

[20] “Smoke Testing.” Software Testing Fundamentals, 13 July 2018, http://softwaretestingfundamentals.com/smoke-testing/.

[21] “What Is Regression Testing? Definition, Test Cases (Example).” Guru99, https://www.guru99.com/regression-testing.html.

[22] “Integration Testing.” Software Testing Fundamentals, 3 Mar. 2018, http://softwaretestingfundamentals.com/integration-testing/.

[23] SOFTWARE TESTING TUTORIAL. (n.d.). Retrieved October 20, 2019, from http://moodle.nccu.edu.tw/pluginfile.php/77731/mod_resource/content/1/software_testing(1).pdf.

[24] Browser Automation. (n.d.). Retrieved October 15, 2019, from https://www.seleniumhq.org/.

[25] “Test Automation Tools: Ranorex Automated Software Testing.” Ranorex, https://www.ranorex.com/test-automation-tools/.

[26] Appium. (n.d.). Retrieved October 15, 2019, from http://appium.io/.

[27] Test Automation Frameworks - Stuff you must Know! (n.d.). Retrieved October 20, 2019, from https://www.guru99.com/test-automation-framework.html.

[28] Selenium - A brief overview. (2017, April 20). Retrieved from https://www.360logica.com/blog/selenium-a-brief-overview/.

[29] Browser Automation. (n.d.). Retrieved from https://www.seleniumhq.org/docs/07_selenium_grid.jsp.

[30] What is Selenium WebDriver? Difference with RC. (n.d.). Retrieved October 21, 2019, from https://www.guru99.com/introduction-webdriver-comparison-selenium-rc.html.

[31] Selenium IDE · Open source record and playback test automation for the web. (n.d.). Retrieved October 21, 2019, from https://www.seleniumhq.org/selenium-ide/.

[32] "SELENIUM Vs RANOREX." QodeStack, https://qodestack.com/selenium-vs-ranorex/.

[33] "Comparing Automated Testing Tools: Selenium, TestComplete, Ranorex, and More." AltexSoft, https://www.altexsoft.com/blog/engineering/comparing-automated-testing-tools-selenium-testcomplete-ranorex-and-more/.

[34] "The Good and the Bad of Ranorex GUI Test Automation Tool." AltexSoft, https://www.altexsoft.com/blog/engineering/the-good-and-the-bad-of-ranorex-gui-test-automation-tool/.

[35] "Appium vs Selenium: What Are the Differences?" StackShare, https://www.stackshare.io/stackups/appium-vs-selenium.

[36] Solutions, Mindfire. "Selenium vs Appium: Difference Between Appium and Selenium: Mindfire." Blogs @ Mindfire Solutions, 31 May 2018, http://www.mindfiresolutions.com/blog/2018/05/selenium-vs-appium/.

[37] "Appium vs Selenium: Which One Is Better (With Infographics)." EDUCBA, 20 Sept. 2019, https://www.educba.com/appium-vs-selenium/.

[38] Manasa, Aliya. "Selenium vs Appium." Medium, Medium, 15 Apr. 2019, https://medium.com/@aliyamanasa23/selenium-vs-appium-4706e0a9fbb8.

[39] "What Is the Difference between Selenium and Appium Testing?" Quora, https://www.quora.com/What-is-the-difference-between-Selenium-and-Appium-testing.

[40] Matthews, Joseph R. "Library Leadership and Managment." Library Leadership and Management, 1 Feb. 2011, https://www.researchgate.net/publication/49116122_What's_the_Return_on_ROI_The_Benefits_and_Challenges_of_Calculating_Your_Library's_Return_on_Investment.

[41] Chen, James. "Return on Investment (ROI)." Investopedia, Investopedia, 3 Dec. 2019, https://www.investopedia.com/terms/r/returnoninvestment.asp.

[42] Hoffman, D. (1999). Cost Benefits Analysis of Test Automation. *Cost Benefits Analysis of Test Automation*.

[43] Münch, Stefan, et al. "When Calculating Return of Investment, Only Cost Information Is Not Used as Input. in Addition, Multiple Factors Affect This Issue. Some Factors with High Effect Rates Are given below." *Pharmageutical Engineering, The Offical Maganize of ISPE*, vol. 32, ser. 4, 2012. 4.

[44] Muthusundar S. K., et al. "COMPARATIVE STUDY OF TEST AUTOMATION ROI." *Indian Journal of Computer Science and Engineering (IJCSE)*, vol. 2, no. 4, 2011, pp. 615–623. *ISSN : 0976-5166*.

[45] Graham, Dorothy. “ROI of Test Automation: Benefit and Cost.” *ROI of Test Automation: Benefit and Cost*, Nov. 2010.

[46] Berner, S., Weber, R., & Keller, R. (may 15-21 ,2005). Observations and lessons learned from automated testing. *Proceedings. 27th International Conference on Software Engineering, 2005. ICSE 2005.*doi:10.1109/icse.2005.1553603

[47] Pettichord, B. (2001). Success with Test Automation. *Success with Test Automation*.

[48] Grossman, Paul. *Automated Testing ROI: Fact or Fiction?* HP, 2009, *Automated Testing ROI: Fact or Fiction?*, https://deepgnosis.me/wp-content/uploads/Automation-ROI.pdf.

[49] Bazzana, Gualtiero, et al. ISTQB® Worldwide Software Testing Practices Report 2015 - 2016. Edited by Geoff Thompson and Patricia McQuaid, International Software Testing Qualifications Board, 2016, ISTQB® Worldwide Software Testing Practices Report 2015 - 2016, https://www.istqb.org/documents/ISTQB_Worldwide_Software_Testing_Practices_Report.pdf.

[50] Baron, Yan, et al. ISTQB® Worldwide Software Testing Practices REPORT 2017-18. International Software Testing Qualifications Board, 2018, ISTQB® Worldwide Software Testing Practices REPORT 2017-18, http://www.turkishtestingboard.org/files/ISTQB-Worldwide-Software-Testing-Practices-Report-2017-18.pdf.

[51] Ramler, R., & Wolfmaier, K. (2006). Economic perspectives in test automation. *Proceedings of the 2006 International Workshop on Automation of Software Test - AST 06*,85-91. doi:10.1145/1138929.1138946

[52] Sheth, T., & Singh, S. K. (august 2015). Software Test Automation- Approach on evaluating test automation tools. *Approach on Evaluating Test Automation Tools,5*(2).

[53] K. Karhu, T. Repo, O. Taipale, and K. Smolander, “Empirical Observations on Software Testing Automation,” in 2009 International Conference on Software Testing Verification and Validation. IEEE, Apr. 2009, pp. 201–209.

[54] *Whats My Testing ROI? - Robertvbinder.com*. robertvbinder.com/wp-content/uploads/sites/4/2011/06/Whats-My-Testing-ROI1.pdf.

[55] Nizri, Gabby. “5 Most Common Reasons IT Process Automation Projects Fail.” Ayehu, Gabby Nizri, Co-Founder, CEO of Ayehu , 3 Mar. 2016, https://ayehu.com/5-most-common-reasons-it-process-automation-projects-fail/.

[56] Zainal, Z. (2007). Case study as a research method . *Jurnal Kemanusiaan*, *5*(1), 1–6. Retrieved from https://jurnalkemanusiaan.utm.my/index.php/kemanusiaan/issue/view/44

[57] Strumińska-Kutra M., Koładkiewicz I. (2018) Case Study. In: Ciesielska M., Jemielniak D. (eds) Qualitative Methodologies in Organization Studies. Palgrave Macmillan, Cham doi: https://doi.org/10.1007/978-3-319-65442-3_1

[58] AYTAÇLI, B. (2012). Durum Çalışmasına Ayrıntılı Bir Bakış. *Journal of Educational Sciences,Adnan Menderes Üniversitesi Eğitim Fakültesi Eğitim Bilimleri Dergisi*, *3*(1), 1–9. Retrieved from http://dergi.adu.edu.tr/egitimbilimleri/

APPENDIX A

A. Sample 1 Manual – F_{F1} Calculation

Test Cases Number	Priority of the Test Cases	Manual Run Time (in second)	Run Time for Each Case * Priority of Test Case - E_{F1} Calculation
Test Case 1	3	30	90
Test Case 2	2	45	90
Test Case 3	2	51	102
Test Case 4	1	52	52
Test Case 5	4	57	228
Test Case 6	5	67	335
Test Case 7	1	73	73
Test Case 8	3	75	225
Test Case 9	2	78	156
Test Case 10	2	80	160
Test Case 11	3	84	252
Test Case 12	4	89	356
Test Case 13	4	93	372
Test Case 14	5	97	485
Test Case 15	1	101	101
Test Case 16	3	106	318
Test Case 17	4	109	436
Test Case 18	4	114	456
Test Case 19	5	116	580
Test Case 20	2	119	238
Test Case 21	3	123	369
Test Case 22	3	126	378
Test Case 23	2	128	256
Test Case 24	4	133	532
Test Case 25	5	136	680
Test Case 26	2	139	278
Test Case 27	4	142	568
Test Case 28	5	144	720
Test Case 29	4	146	584
Test Case 30	1	153	153
Test Case 31	2	157	314
Test Case 32	2	159	318

B. Sample 2 Manual -F_{F1} Calculation

Test Cases Number	Priority of the Test Cases	Manual Run Time (in second)	Run Time for Each Case * Priority of Test Case - E_{F1} Calculation
Test Case 1	2	60	120
Test Case 2	3	92	276
Test Case 3	4	94	376
Test Case 4	1	105	105
Test Case 5	5	123	615
Test Case 6	5	125	625
Test Case 7	4	127	508
Test Case 8	2	130	260
Test Case 9	3	133	399
Test Case 10	1	140	140
Test Case 11	2	141	282
Test Case 12	4	145	580
Test Case 13	5	147	735
Test Case 14	3	150	450
Test Case 15	4	152	608
Test Case 16	1	156	156
Test Case 17	2	157	314
Test Case 18	2	159	318
Test Case 19	3	160	480
Test Case 20	4	164	656
Test Case 21	5	168	840
Test Case 22	4	173	692
Test Case 23	2	60	120

C. Sample 1 Automation -F_{F1} Calculation

Test Cases Number	Priority of the Test Cases	Automation Run Time (in second)	Run Time for Each Case * Priority of Test Case - E_{F1} Calculation
Test Case 1	3	7	21
Test Case 2	2	11	22
Test Case 3	2	13	26
Test Case 4	1	19	19
Test Case 5	4	27	108
Test Case 6	5	31	155
Test Case 7	1	36	36
Test Case 8	3	39	117
Test Case 9	2	41	82
Test Case 10	2	47	94
Test Case 11	3	51	153
Test Case 12	4	53	212
Test Case 13	4	56	224
Test Case 14	5	59	295
Test Case 15	1	61	61
Test Case 16	3	63	189
Test Case 17	4	67	268
Test Case 18	4	71	284
Test Case 19	5	75	375
Test Case 20	2	79	158
Test Case 21	3	81	243
Test Case 22	3	84	252
Test Case 23	2	86	172
Test Case 24	4	89	356
Test Case 25	5	91	455
Test Case 26	2	94	188
Test Case 27	4	97	388
Test Case 28	5	100	500
Test Case 29	4	103	412
Test Case 30	1	105	105
Test Case 31	2	108	216
Test Case 32	2	111	222

D. Sample 2 Automation -F_{F1} Calculation

Test Cases Number	Priority of the Test Cases	Automation Run Time (in second)	Run Time for Each Case * Priority of Test Case - E_{F1} Calculation
Test Case 1	2	12	24
Test Case 2	3	15	45
Test Case 3	4	19	36
Test Case 4	1	23	23
Test Case 5	5	28	140
Test Case 6	5	35	175
Test Case 7	4	39	156
Test Case 8	2	47	94
Test Case 9	3	51	153
Test Case 10	1	56	56
Test Case 11	2	63	126
Test Case 12	4	69	276
Test Case 13	5	77	385
Test Case 14	3	82	246
Test Case 15	4	87	348
Test Case 16	1	93	93
Test Case 17	2	98	196
Test Case 18	2	102	204
Test Case 19	3	106	318
Test Case 20	4	111	444
Test Case 21	5	114	570
Test Case 22	4	119	476
Test Case 23	2	121	142

APPENDIX B

A. Sample 1 Automation – Implementation and Maintenance

Test Cases Number	Implementation Time (in minute)	Maintenance Time (in minute)
Test Case 1	529	337
Test Case 2	534	329
Test Case 3	451	249
Test Case 4	443	282
Test Case 5	448	351
Test Case 6	456	336
Test Case 7	371	266
Test Case 8	436	339
Test Case 9	462	241
Test Case 10	575	367
Test Case 11	526	341
Test Case 12	448	363
Test Case 13	454	376
Test Case 14	435	359
Test Case 15	471	254
Test Case 16	463	383
Test Case 17	494	262
Test Case 18	557	346
Test Case 19	485	358
Test Case 20	464	332
Test Case 21	538	349
Test Case 22	453	228
Test Case 23	442	303
Test Case 24	484	257
Test Case 25	405	278
Test Case 26	492	312
Test Case 27	484	346
Test Case 28	475	337
Test Case 29	564	293
Test Case 30	581	326
Test Case 31	502	358
Test Case 32	526	364

B. Sample 2 Automation – Implementation and Maintenance

Test Cases Number	Implementation Time (in minute)	Maintenance Time (in minute)
Test Case 1	539	292
Test Case 2	554	223
Test Case 3	547	301
Test Case 4	552	228
Test Case 5	551	223
Test Case 6	481	238
Test Case 7	556	323
Test Case 8	468	246
Test Case 9	549	306
Test Case 10	464	345
Test Case 11	471	267
Test Case 12	482	347
Test Case 13	597	267
Test Case 14	556	334
Test Case 15	464	254
Test Case 16	451	332
Test Case 17	592	232
Test Case 18	487	281
Test Case 19	473	345
Test Case 20	504	226
Test Case 21	425	301
Test Case 22	574	236
Test Case 23	559	322

APPENDIX C

A. Example of Social Media Test Case

ID:
Test Case 3

Summary:
Verify That Click to “What do you think?” area.

Priority:
Medium

Preconditions:
Test Case 2

Steps:

1. Open browser
2. Enter www.facebook.com
3. Enter the web site login page.
4. Enter valid username
5. Enter valid password
6. Click to login page
7. Observe main page
8. Observe “What do you thing?” area
9. Click it.

Expected Result:
What do you thing popup will open clearly.

B. Example of E-Commerce Test Case

ID:
Test Case 5

Summary:
Verify that go to second page on the search page

Preconditions:
Test Case 4

Steps:
1. Open website
2. Login website with valid user
3. Observe main page
4. Find Search bar
5. Enter “Samsung” to serach bar
6. Find search button
7. Click the button
8. Observe the search page
9. Find the Samsung keyword from search page
10. Observe pagination bar
11. Click to second page from there
12. Observe second page on the search page

Expected Result:
Pagination should be working clearly. Second page should be see on the page.