

COMPARATIVE STUDY OF TEST AUTOMATION ROI

S.K. Muthusundar¹ Research scalar, Bharath University, Chennai.

skmuthusundar@gmail.com

D.Thara and Trilysa Anthea titus² Final year student, Kings Engineering college, Chennai

Abstract

Need of enhancing the potential benefits of a project by robust strategies that reduces the execution time of testing cycle and maintenance effort result in increased productivity by additional test cases within a given schedule this objective can be met the organist ions with the help of test Automation framework. Return on investment (ROI) can also be achieved through test automation. It reduced the maintenance effort and script development result will help increase the speed of test execution and provides maximum number of resources. In this paper we will discuss the importance of using testing automation to achieve high return on investment on the basis of time and cost benefits. The ROI formula is very normally very "Basic". Costs are fairly easy to quantify, but the value of potential benefits is far more difficult to nail down. ROI approaches often oversimplify many elements of test automation leading to inaccurate results. These inaccuracies then lead to unrealistic expectations on the part of management for test automation, and this can lead to failure (personal, team, and sometimes project).

Keywords: *Test automation*; standard automation; ROI calculation.

1.INTRODUCTION

The rapid advancement in technology in today's world s leading to the need of enhancing the potential benefits [1] of a project by robust strategies that requires to enhance the speed of test execution at a lower cost. This objective of an organization can be obtained through test automation using hybrid automation. The nature of return on investment is based on the benefits achieved. The benefits are achieved by using some ROI calculations along with test automation. The qualification of engaging test automation is where the success of automation takes place. Before the beginning of test automation, organization should:

- Examine the need for test automation of a project and Analyzing the paybacks for forecasting test automation

1.1 Need of test automation using ROI

Test automation using ROI is a vital requirement of needed by most organizations because:

1. The need for the project to focus on time investment gain for test automation ROI.
2. For improved test coverage which is done by reducing the risk factors
3. The need for reducing the monetary savings through automated test execution
4. The time investment gain of the testing lifecycle using ROI is needed

1.2 Payback

Payback period

The payback period is the time taken to break even or pay back the initial investment. Normally, the project with the shortest [10] payback period will be chosen on the basis that an organization will wish to minimize the time that project is 'in debt'

Paybacks for forecasting test automation

Instead of using traditional investments in hardware, software and staffing modern innovative ways of investing in hardware, software and staffing is performed. This approach is used enable the organization to realize the benefits of costs and increased speed to market.

Paybacks of automation

The paybacks of automation can be further enhanced using software test automation enables to reduce the maintenance effort and the script development



IBM has derived the implementation of test automation without any framework expenditure

Paybacks

Let us consider a sample size of 500 test cases per test cycle. Paybacks for automation per test cycle is given as

$$Ts = Tm - Ta$$

Where,

Ts=test automation time investment

Tm=manual testing consumption time

Ta=Time invested in automated testing.

Expenditure of Automation

Expenditure availed for automation is given by

$$Ca = Chas + Cdmic$$

Where,

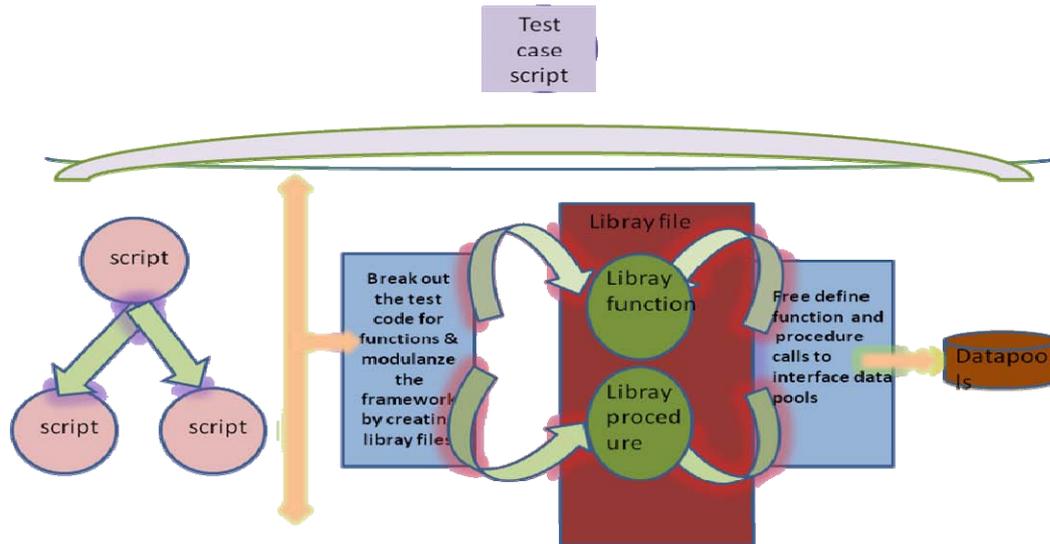
Chas=hardware and software expenditure

CRM=expenditure in developing the test automation script and maintaining it

Ct= staff training expenditure

1.3 Tangible Paybacks of automation testing

- **Speed and accuracy:** Since the execution time is reduced the software life cycle takes place faster unlike in manual testing where the process is much longer. The results achieved in automation testing are more accurate in nature
- **Easy to Access :**It allows access to all data, objects ,web applications and protocols that manual testing cannot do
- **Accumulation:** As the application matures the number of tests increases. Therefore, engineers need not test the same functionality over and over again but instead they can constantly add on to the test suite.
- **Manageability:** The artifacts can be managed through the automation tools
- **Repeatability:** This benefit helps to repeat processes that may take place on the functional and performance side of the testing process
- **Availability:** once the automation scripts are developed it can run at anytime from anywhere without the need of being attended to



1.4 Return on investment

The return on investment (ROI) is also known as the accounting rate of return (ARR). It provides a way of comparing the net profitability to the profitability to the investment required. There are some variations on the formula [3] used to calculate the return on investment but a straight ward common version is:

$$ROI = \frac{\text{average annual profit}}{\text{Investment Costs}} \times 100 \quad (\text{OR}) \quad ROI = \frac{\text{Gains} - \text{Investment Costs}}{\text{Investment Costs}}$$

(OR) $ROI = \text{Net Present Value of Investment} / \text{Initial Cost}$

Table 1.1 The payback matrix is used to represent the distribution of direct and indirect paybacks that takes place across customer/organization, development and expertise that may help in customer and organization’s satisfaction

Drivers	Direct paybacks	Indirect paybacks
customer/organization	By using test automation due to redeployment of workforce we can save on the staffing cost	Leads to increase in clients satisfaction
Development	Usage of test automation framework enables reduction in the maintenance effort and in script development	The testing process is efficient enough to handle the required test cases within the schedule
Expertise	The technology used is more reliable due to reduced test execution time in testing lifecycle	Due to reduction in the execution time the maintenance efforts needed are also reduced

2. AUTOMATION APPROACHES

Test Automation using the several [7] approaches for calculating ROI:

- (1) **Basic ROI Calculation**
- (2) **Effective ROI Calculation**
- (3) **ROI Calculation for reducing risks**

Each is detailed in this document along with a basic ROI calculation

2.1 BASIC ROI CALCULATION

The Basic ROI calculation focuses on investing on the monetary funds through automated test execution. In terms of reducing the testing expenditures this calculation may be of immense importance. The savings include automation expenditure as hardware, software licenses, training of staffs, script development [2] and maintenance, and script execution and analysis. The returns are equal to the pre-automation expenditure of executing the same test cases

2.1.1 Basic calculation

Consider working on a [3] project with an application that has several test cycles that equates to a weekly build 7Months out of the year. In addition the project has the following manual and automation aspects that will be used in calculating ROI:

General Aspects

Appendix A. 1200 test cases, 200 of which can be automated.

Appendix B. Tester's Hourly Wages. 70 per hour

Manual Aspects

- Manual Test Execution/Analysis Time (average per test) 10 minutes

Automation Aspects

- (a) Tool and License expenditure (5 licenses). 20,000
- (b) Tool Training expenditure. 3,000
- (c) Test Machine expenditure (5 machines). 6000
- (d) Test Development/Debugging Time (average per test) 120 minutes (2 hour)
- (e) Test Execution Time. 2 minute
- (f) Test Analysis Time. 3hours for200 tests
- (g) Test Maintenance Time. 8 hours per build

Test Development Time can be converted to a dollar figure by multiplying the average *hourly* . Time taken per test (1 hour) by the number of tests (200), then by the Tester Hourly Rate (70). This equals 14,000. Test Execution Time doesn't need to be converted to a dollar figure in this example, because the tests will ideally run independently on one of the Test Machines. The Test Analysis Time can be converted to a rupees figure by multiplying the Test Analysis Time (3 hours per week given that there is a build once a week) by the time frame being used for the ROI calculation (6 months or approximately 24 weeks), then by the Tester Hourly Rate (70). The calculation is done by multiplying $3 \times 24 \times 70$ equals 5,040.

The Test Maintenance Time can be converted to a dollar figure by multiplying the Maintenance Time (8 hours per week) by the timeframe being used for the ROI calculation (6 months or approximately 24weeks), then by the Tester Hourly Rate (70). This equals 13,440.

The total investment cost can now be calculated as $20,000 + 3,000 + 6,000 + 14,000 + 5,040 + 13,440$, which equals 61,480. That's a lot of money! But before you decide to eliminate test automation from your project, let's turn our attention to the gain. The gain can be calculated in terms of the Manual Test Execution/Analysis Time that will no longer exist once the set of tests has been automated. The Manual Execution/Analysis Time can be converted to a rupees figure by multiplying the Execution/Analysis Time (10 minutes or .17 hours) by the number of tests (200), then by the time frame being used for the ROI calculation (6 months or approximately 24 weeks), and finally by the Tester Hourly Rate (70).

Note, manual test development and maintenance are not considered because these activities must take place regardless of whether or not a test is automated. The gain is therefore is $17 \times 200 \times 24 \times 70$ which equals 57,12,000! Still think 61,480 is too much?

Inserting the investment and gains into our formula:

$$\text{ROI} = \frac{\text{Gains} - \text{Investment Costs}}{\text{Investment Costs}}$$

$$\text{ROI} = \frac{57,12,000 - 61,480}{61,480} = 91.9\%$$

The ROI is calculated at 91.9%. Note that over time this ROI percentage will increase, because the tool costs eventually get replaced in the calculation by tool support costs.

2.1.2 Expenditure on automation

The most common way of carrying out an economic assessment of a proposed information system or software product is by comparing the [4] expected costs of development and operation of the system with the benefits of having it in place.

There are two types of costs that we will be focusing on. They are one time and recurring expenditure.

Fixed expenditures are needed to establish the test automation environment, it is a one-time investment expenditure. Expenditure that is incurred during test automation

(1) Expenditure of the software that is to be implemented

(2) Expenditure of the hardware that is used

(3) Expenditure on training the staffs about new software tools

Expenditure incurred while development of the automation script

During the process of the testing life cycle recurring expenditures are incurred. This type of expenditure may occur anytime and at any part of the testing process below Table 1.2

Drivers	One time expenditure	Recurring expenditure
customer/ organization	Expenditure on script development and training on modern automation tools	Expenditure spent on maintenance effort
Development	Expenditure on establishing new development	Not applicable
Expertise	Automation license expenditure Expenditure on the hardware and modern software automation tools	Maintenance expenditure of hardware and software automation tools

Table 1.2

2.1.3 Sample ROI calculation for expenditure:

Calculating the ROI for project 1, the net profit is 50,000 and the total investment is 100,000. The return on investment is therefore calculated as

$$ROI = \frac{\text{average annual profit}}{\text{Total investment}} \times 100$$

$$ROI = \frac{50,000}{100,000} \times 100 = 10\%$$

The return on investment provides a simple, easy to calculate measure of return on capital

2.2 EFFECTIVE ROI CALCULATION

This calculation is useful for projects that had already purchased an automated tool for a long enough time that there isn't much need to give much importance to its expenditure availed in the ROI calculation. This calculation is based on the effective [5] ROI calculation, but only considers the time investment benefits, for assessing testing efficiency, it does not focus on the aspect of investing monetary funds. In addition, it may be better suited for calculation by test engineers, because the aspects [6] used are easily attainable.

2.3 ROI CALCULATION FOR REDUCING RISKS

The Risk Reduction ROI calculation seeks to address ROI concerns left by other calculations by looking at automation benefits independently of manual testing. Test automation saves time in test execution, which provides testing resources with more time for increased analysis, test design, development and execution of new tests. It also provides more time for ado networks and exploratory testing to take place. Using risk reduction ROI calculation the test coverage in the testing process is increased which reduces the risk of production failures. By assessing the risk of not performing automation (relative to potential production failures), and calculating the cost to the project if the risk turns into a loss, this calculation addresses the ROI relative to an increased quality of testing. Given the Need for tester's rates, risk analysis, and cost calculations this formula may be best suited for calculation by upper level management when there is a desire to see organizational test automation benefits.

The tabular column given below depicts risk analysis process based on the Importance and likelihood of the risk that are incurred

Risk	importance	likelihood
Software never completed or delivered	HIGH	---
Project cancelled after design stage	HIGH	---
Software delivered rate	MEDIUM	MEDIUM
Development budget exceeded \leq 20%	LOW	MEDIUM
Development budget exceeded $>$ 20%	MEDIUM	MEDIUM
Maintenance costs higher than estimated	LOW	LOW
Response time targets not met	LOW	HIGH

Table 1.3

A case study:

We would now take glance a brief study on the investment of time execution for test automation. ROI calculations are performed on the basis of paybacks and the expenditure incurred using test automation.

2.4 ROI for Standard Automation Advancement

ROI for the advancement in standard automation is accomplished by the graphical representation of a graph. Certain data containing sample size of 250 test cycles the ROI frequencies of standard automation is depicted in the graph using the following frequencies:

ROI execution for 24 and 48 test cycles for 2 years using standard automation technique:

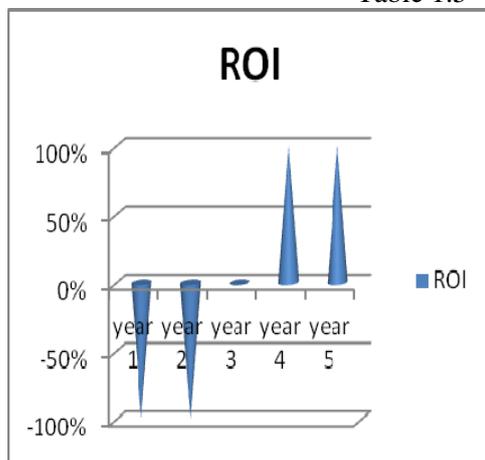
Elements	Two years	Four years
Total number of cycles	24	48
Total investment gained on efforts within 2 years	1125	2247.84
Total investment in expenditure due to automation	337500	506250
Expenditure on automation	875	500
Development efforts	1050	900
Average script maintenance	1925	1400
Total Effort spent	385000	280000
Total expenditure	40000	40000
Total hardware/software expenditure	7500	186250
Net profit	87500	23750
ROI	-27%	-06%

Table 1.4

ROI execution for 24 and 48 test cycle for 2 years using curator and keene automation technique:

Elements	Two years	Four years
Total number of cycles	24	48
Total investment gained on efforts within 2 years	1125	2247.84
Total investment in expenditure due to automation	337500	506250
Expenditure on automation	500	500
Development efforts	600	900
Average script maintenance	1100	1400
Total Effort spent	220000	280000
Total expenditure	40000	40000
Total hardware/software expenditure	7500	186250
Net profit	77500	186250
ROI	30 %	58%

Table 1.5



Thus from the above graph we understand that the ROI value of the test execution will increase with increasing test cycle frequencies

3. WAYS TO IMPROVE ROI

There are several common factors that affect each of the ROI calculations, so improvements (reductions) to any of those factors will certainly improve ROI, regardless of how it's calculated. These factors are as follows:

Automated:

- [1] Test Development/Debugging Time (average per test)
- [2] Test Execution Time
- [3] Test Analysis Time
- [4] Test Maintenance Time

Test Development/Debugging can be decreased by increasing the initial time investment in creating a framework and process in which the automated test development will be conducted. This will have a big eventual payoff, because it imposes standards that will provide for faster automation.

Test Execution Time can be decreased by better [9] exception handling routines. Failed test steps and failed tests drastically prolong test execution, because the automated test tool spends time looking for what doesn't exist, and spends extra time

verifying that it really doesn't exist. In addition, automated test failures often compound upon one another, so one failure may cause another, although subsequent failures aren't related to defective application functionality.

Test Analysis Time can be decreased by building better reporting mechanisms into the tests. This may be accomplished by adding special logging statements to the automated tests that will print to the report in the event of a failure. In addition, you may have the test trigger a screen capture in the event of a failure, so that it is easy to know what state the application was in at the time of failure.

Finally, Test Maintenance may also be decreased by increasing the initial time investment in creating a framework and process in which the automated test development will be conducted. In addition, increasing modularity and adding comments to automated tests is another to make maintenance easier and faster.

4. TESTING TOOLS USED BY VARIOUS VENDORS

- (1) **IBM/rational software** its uses functional tester as the functional test product and performance tester as the performance test product
- (2) **Hewlett Packard** uses Win Runner, quick test pro as functional products and uses load runner for performance products
- (3) **Microsoft** uses tools such as team System: time edition as performance products.
- (4) **Review** uses Weft and web load for functional and performance products
- (5) **Empire** uses e-tester and e-load for functional and performance products
- (6) **Ponte** uses IT guru as a product for security testing process

4.1 Managing the resources, scope and result using test automation

The scope, resources and time to market should be done effectively by treating testing as a project. A graph is depicted by taking “time to market” on the x-axis and resources on the y-axis in given below

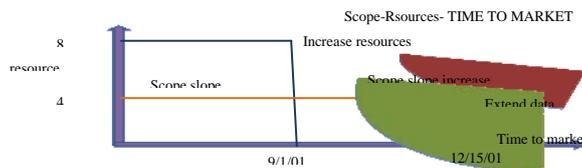


diagram:

If the scope of the efforts or work needs to be increased then this is done by either increasing the resources or extending the time frame.

5. CONCLUSION

Projects must be evaluated on strategic, technical and economic grounds. Using ROI calculation (on the basis of execution time and cost investment benefits) in Test Automation frameworks will reduce the [9] maintenance effort and script development result and will enable enhancing the speed of test execution. Test automation helps to ensure the reliability of the software, increase in the test coverage and ability to increase the resource efficiency.

Future work:

Framework supports automation testing for an organization. Shared resources (such as image files, dynamic files, reference document) are encapsulated in the framework that is also a hierarchical directory. The test automation frameworks [8] some discrete business keywords are used which are reusable, has greater productivity and optimum reusability of tools and Return on investment is high (framework are used to reduce execution time but without any framework costs)

REFERENCES

- [1] D. Hoffman, “Cost Benefits Analysis of Test Automation”, STAR West 1999, Oct 999.
- [2] N. Jayachandran, “Compiled Test Scripts, Test Cases and Defect Reports From The Experiment Conducted For This Thesis”, 2005.
- [3] R. Rachlin, “Return On Investment Manual: Tools and Applications for Managing Financial Results”, Sharpe Professional, 1997.
- [4] J. Barnes, “Creating an ROI assessment for implementing IBM Rational solutions”, Rational DeveloperWorks, 15 Oct. 2004,
- [5] Mentora Group, “ROI of an Oracle Apps Repeatable Automated Test Suite”. IBM Rational, “The Rational Unified Process (RUP)”.

- [6] M. Fewster, D. Graham, "Software Test Automation: Effective Use of Test Execution Tools", Addison-Wesely, 1999.
- [7] J. Varghese, "Test Automation – An ROI based Approach", 3rd Annual
- [8] International Software Testing Conference, Bangalore, India, 2001.
- [9] M. Kelly, "Choosing a test automation framework", IBM DeveloperWorks, 18 Jul 2003.
- [10] V. Sikka, "Maximizing ROI on Software Development", Auerbach Publications, 2005.
- [11] Bob Hughes & Mike Cotterell, "Software Project Management" Tata McGraw-Hill Forth Edition, 2006.