Test Automation For Software Development In Digital World
# Table Of Contents

Abstract......................................................................................................................... 3

The Background ........................................................................................................... 3

An Introduction To Test Automation ................................................................. 4

Key Value Propositions Of Test Automation .............................................. 5
  • Tangible Returns Of Test Automation................................................................. 5
  • Intangible Returns Of Test Automation............................................................... 8

A Systematic Approach To Plan & Execute Test Automation ...... 9
  • What To Automate? ............................................................................................... 9
  • When To Automate? ............................................................................................. 9
  • How Much To Automate? .................................................................................. 9
  • How To Measure Success Of Test Automation?............................................... 10

Potential Pitfalls To Be Aware ........................................................................ 11

An Online Retailer’s Success With Test Automation ............... 11

Conclusion .............................................................................................................. 11

About Softcrylic...................................................................................................... 12
Abstract

Organizations see IT as the mode for rapid, continuous, relentless experiments along with rapid, continuous, relentless delivery of new features to keep up with the Digital Competition they experience. With continuous integration/deployment and the emergence of DevOps disciplines, developers can now promise Continuous Delivery. But, for a software to be to be predictable, reliable and of high quality, testing is critical. Testers need to deliver Continuous Assurance and hence have turned towards test automation in recent times. Automation of testing process is not only desirable, but in fact a necessity given the demands of the current market trend. This white paper throws light on key value propositions and critical steps of successful Test Automation execution to help smoothen the transition process of manual to automation testing.

Background

Digital Transformation As The Driver

The digitally driven market environment calls for quick product launches without compromising over user experience and features. Customers today demand anything, anywhere, anytime, and on any device. This is the center point of the Digital Transformation spree every organization is going through. Customers correlate a software's user experience directly to the service or experience they seek to buy and this directly determines its market success. Businesses are in a digital transformation driven ‘software race’ and the goal is to get ahead and to stay ahead.

Rising User Experience

The rapid adoption of technology and changes in customer behavior are exerting intense pressures on almost all industries. The dynamics of today's market relies on customer centric software loaded with perfect user experience. As a result, the industry is in a spree to sacrifice product development time and find ways to deliver high-end user experience within a limited budget. Testing, an imperative part of the software development process is also accelerated to handle the surging user experience demands.

The Need For Speed

Digitization of businesses has increased the dependency on software for business activities. This in turn keeps organizations focused on fast-track software testing as an important component of their development process. Pioneering thoughts that wins over competitors with speed is vital to launch products that makes profit. This need for speed of the digitally driven environment is the strongest element that determines the success of businesses today.

Test Automation as the Solution

If testing process was to solely depend on manual testing there will be a bottle neck in releasing application features faster and match the industry competition. Automating software testing is the answer to increasing demands like scarcity of skilled software testers and limited time.

Every defect found late will cost extra dollars, time, and human effort. A defect found during the development phase is less expensive to fix than when it is found after the application goes live. Implementing Test Automation proves to be a cost effective solution that meets the demands of fast paced software development process and provides a platform to measure the outcomes.

A Report published by National Institute of Standards and Technology States that one-third of software errors cost could be eliminated by improved testing.
An Introduction To Test Automation

Test automation is the use of special software to control the execution of tests and the comparison of actual outcomes with predicted outcomes. The objective of automated testing is to simplify as much of the testing effort as possible with a minimum set of scripts.

Software testing was done manually ever since the beginning. In the recent past, Test Automation has transformed the way things used to be. Valued as a real-time solution to meet aggressive deadlines, this technique is designed to deliver Customer Ready Products almost at a fraction of what it costs doing it manually. This approach to tackle business challenges, though test automation is by no means an instant panacea nor a replacement for trained testers. This whole growth echoes the overall increase in global spend happening within test automation in order to fulfill the rising user expectation across niches. This paper is all about the test automation and its contribution to the digitally driven software development.

Key Value Propositions Of Test Automation

Tangible Returns Of Test Automation

The most challenging aspects of realizing the benefits of the effort from the investment point view is to spot on a quantifiable amount of time and cost it can potentially save. The following sections focus on helping to perceive the Return on Investment (ROI) of test automation better. automation and its contribution to the digitally driven software development.

“Expert analysts forecasted global outsourced software testing services market growth at a CAGR of 10.25% over the period 2014-2019”

“Expert analysts forecasted global outsourced software testing services market growth at a CAGR of 10.25% over the period 2014-2019”

“The longer software defects go undetected, the more expensive they are to fix. Automated tests are fast to run, and the more tests that completed in a given of time, the more projects can be delivered on schedule.”
1. Time Saving

Test Automation frees up the time to perform more manual testing, especially on newer, high-value software features. Software development with shorter test cycle means reduced time-to-market. Let’s see how test automation proves to be Time Saving with these three hypothetical teams here.

Without test automation, time it takes to test a single build would be: 10 + (1000*10)/5 = 2010 min

2010 minutes is approximately 4 working days.

This not only proves to be costly, but also makes developers get feedbacks 4 days later.

Team A
- Number of cases to test - 1000 and growing.
- Number of minutes to setup environment for a build - 10 min.
- Number of minutes to test one scenario - 10 minutes.
- Number of testers in the team - 5.

Number of cases to test - 1000 and growing.
Number of minutes to setup environment for a build - 10 min.
Number of minutes to test one scenario - 10 minutes.
Number of testers in the team - 5.

Team B
- Everything remains as in Team A.
- 50% of 1000 test cases are automated - 500 test cases.
- Time it takes to automate 500 test cases - 3 hours.

Now, the time it takes to test one single build would be:
Task 1 (manual) - 10 + (500 * 10)/5 = 1010 min
Task 2 (automated) = 180 min

That is almost 2 working days. Though this is not the best outcome, it proved to reduce the time to test and also turned around the build one day earlier than usual. It halved the cost of testing and covered 50% of cases in 3 just hours.

Team C
- Everything remained as in Team B.
- Usage of a good hardware to run the tests faster helped automating 80% of tests (10% cannot be automated and the other 10% is new functionalities).

Now, the time it takes to test one single build would be:
Task 1 (manual) - 10 + (500 * 10)/5 = 1010 min
Task 2 (automated) = 180 min

So, 80% of tests is covered in just 30 minutes. Which is about 7 hours to turn around a build. Likewise, the probability of finding a blocking issue happens much sooner, by covering 80% of samples in 30 minutes. This means that any additional manual testing can be suspended if needed.

Representation Of Time Spent To Test A Build By Each Of The Above Scenarios
2. Cost Saving

Automation is not just to reduce test cycle time, but also to cut down the deployment cost. Cost reduction happens in two ways:

- Faster feedbacks to the developer improves overall quality of builds proficiently.
- There is a direct reduction in testing cost, as it requires lesser number of people to run the same set of tests time and again.

To recognize the true cost saving capacity of test automation, the investment cost required to automate the process needs to be determined. Now, let’s see how test automation proves to be cost effective with these three hypothetical teams.

### Team A
- Number of cases to test - 1000 and growing.
- Number of minutes to setup environment for a build - 10 min.
- Number of minutes to test one scenario - 10 minutes.
- Number of testers in the team - 5.

Assuming that there are no automated tests for this team,

**Testing cost per test cycle:** $8,300

**Total Testing Cost over a Year for manual testing is:** $415,000

### Team B
- Number of scenarios to automate - 500 (50% Test Cases)
- Number of automation developers - 2
- Cost of an automation developer - $50/hour (Engaging a vendor with offshore test automation capabilities, this cost can be reduced)
- Number of test cases a developer can develop - 5 per day

**Time to Automate:** \( \frac{500}{10} = 50 \text{ Days} = 50 \times 8 = 400 \text{ Hrs.} \)

**Automation Cost:** \( 400 \times 50 = 20,000 \)

### Team C
- Number of scenarios to automate - 800 (80% Test Cases)
- Number of offshore automation developers - 2
- Cost of an automation developer - 650 per hour
- Number of test cases a developer can develop - 5 per day

**Time to Automate:** \( \frac{800}{10} = 80 \) Days = 80 X 8 = 640 Hrs.

**Hardware Cost to run Automated Tests Faster:** \( \times = 25,000 \)

**Automation Cost:** \( 640 \times 50 = 32,000 + 25,000 = 57,000 \)

**Testing cost per test cycle:** $8,300

**Total Testing Cost over a Year for manual testing is:** $415,000

A Visual Illustration Of The Above Comparison

![A Visual Illustration Of The Above Comparison](chart.png)
The true cost saving for test automation can be recognized if automated tests are used for a longer period of time, say one year. With these numbers, 43% to 73% of cost savings is expected with test automation. The direct cost savings achieved over a year of testing is brought up below:

<table>
<thead>
<tr>
<th>Details</th>
<th>Team A</th>
<th>Team B</th>
<th>Team C</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of Test Cases</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>Manual Testing Time Per Test Case (minutes)</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>No. of Automated Test Cases</td>
<td>0</td>
<td>500</td>
<td>800</td>
</tr>
<tr>
<td>% Automation</td>
<td>0%</td>
<td>50%</td>
<td>80%</td>
</tr>
<tr>
<td>No. of Resources</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Automation Run Time (minutes)</td>
<td>-</td>
<td>180</td>
<td>30</td>
</tr>
<tr>
<td>No. of Testing Cycles In A Year *</td>
<td>50</td>
<td>50</td>
<td>50</td>
</tr>
</tbody>
</table>

### Time Per Testing Cycle

<table>
<thead>
<tr>
<th>Time Per Testing Cycle</th>
<th>Team A</th>
<th>Team B</th>
<th>Team C</th>
</tr>
</thead>
<tbody>
<tr>
<td>In Minutes (Automated + Manual Testing)</td>
<td>10,000</td>
<td>5,180</td>
<td>2,030</td>
</tr>
<tr>
<td>In Hours (Automated + Manual Testing)</td>
<td>166</td>
<td>86</td>
<td>33</td>
</tr>
<tr>
<td>In Days (With 5 Resources)</td>
<td>4</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

### Testing Cost Per Test Cycle

<table>
<thead>
<tr>
<th>Testing Cost Per Test Cycle</th>
<th>Team A</th>
<th>Team B</th>
<th>Team C</th>
</tr>
</thead>
<tbody>
<tr>
<td>$50/Hour for tester</td>
<td>$8,300</td>
<td>$4,300</td>
<td>$1,650</td>
</tr>
</tbody>
</table>

### Testing Cost Over A Year

<table>
<thead>
<tr>
<th>Testing Cost Per Test Cycle x No. of Testing Cycles In A Year (50)</th>
<th>Team A</th>
<th>Team B</th>
<th>Team C</th>
</tr>
</thead>
<tbody>
<tr>
<td>$415,000</td>
<td>$215,000</td>
<td>$82,500</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Investment in Automation</th>
<th>Team A</th>
<th>Team B</th>
<th>Team C</th>
</tr>
</thead>
<tbody>
<tr>
<td>$20,000</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Total Cost of Testing</th>
<th>Team A</th>
<th>Team B</th>
<th>Team C</th>
</tr>
</thead>
<tbody>
<tr>
<td>$415,000</td>
<td>$235,000</td>
<td>$114,500</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Savings Per Year</th>
<th>Team A</th>
<th>Team B</th>
<th>Team C</th>
</tr>
</thead>
<tbody>
<tr>
<td>$180,000</td>
<td>$300,500</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3. Return On Investment

An automated testing process will need a regular upfront cost and runs on a cheaper execution cost down the road. Judging the ROI is not to plainly validate the efforts, but is seen as an integral part of the project planning process.

While performing a return on investment analysis for your product, a guesstimate of the product’s testing cost will help in finding the right automation solution, tools and the level of skills required for the testing resources. Test Automation is attractive, essential and can get expensive only if not planned well.

Let’s calculate the ROI for investments made on automation per year below:

**Cost of Investment** = $57,000 (80% test cases automated)

**Gain from Investment** = $415,000 - $114,500 = $300,500

ROI = \( \frac{\text{Gain from Investment}}{\text{Cost of Investment}} \) = \( \frac{\$300,500}{\$57,000} \) = \( \approx 427\% \) / year
Automation is thus proven to be an amenable and good fit for a broad range of products regardless of tools, technology and platform due to liberal ROI.

Digital Transformation Has Pushed Strong Recognition Of QA And Testing

The proportion of IT spend allocated to QA and Testing is predicted to rise to 40% by 2018.

Intangible Returns Of Test Automation

Apart from the huge cost and time saving benefits, there are other values that test automation can fortify a product with. A few intangible returns not to be missed are:

1. Early and Frequent Feedbacks

Accurate and timely reviews increases the credibility of products as late feedbacks brings down build quality, and the amount of time bugs stay in the system. Every build is tested without prejudgment, and teams get a report as priority. Timely feedbacks also avoids mistakes that may become too costly to fix later. Getting quicker feedback means lesser code getting built on top of the existing buggy code.

2. Opportunity Cost

Usually, a bug leads to many and by focusing on manual scenarios, testers hardly find enough time to create new scenarios and follow up on the issues. Not only this, it is noted that by focusing on regression tests all the time, testers spend fairly less time on recent features, where there is a higher probability for bugs. By automating a substantial part, a team can facilitate testers to be more innovative and explore an application from the “human angle”, thus increasing the depth of coverage and quality.

3. Reliability

Test automation does reduce possible human error of manual test execution. Consistency of test results in manual testing is absolutely reliant on the tester and can’t assure 100% accurate test results. However, automation reduces this threat by putting a process in place and ensures accuracy.

4. Increased Test Coverage

Depth and scope of tests are pushed to the next level with automation. Lengthy tests that are run as a routine can be executed unattended to avoid mistakes that happens due to monotonous manual testing. Expect to see best coverage as it is used to execute several different complex test cases during every test run. Automated testing sees through your product’s memory contents, data tables, file contents, and internal program conditions to decide if it is performing as per expectation.

Illustration Of Test Coverage Trend Change
A Systematic Approach To Plan & Execute Test Automation

What To Automate?

As a product grows and evolves, the size of regression tests grows along. The need for ensuring best coverage in short time calls for test automation in most business requirements. Understanding the product's life span plan and a high-level overview of future changes in the product is essential for successful test automation. The trick to get the best out of automation is to identify what to automate. Just like how it is impossible to test 100%, it is impossible to automate 100% as well, let's get to know some of the ideal candidates for test automation.

1. Sanity Tests

Basic functionalities of a product seldom undergo modifications, this quality makes sanity testing ideal for automation. This is for the most part beneficial while holding several releases on a monthly basis.

2. Regression Tests

By utilizing automation for regressing tests, cost and effort of testers can solely be used to focus on new functionality verification. Handing over the repetitive tasks to automation will eliminate errors happen due to human tedium while testing the same functionalities endlessly.

3. Unit Tests

These gives the highest ROI as they are totally inexpensive to write and maintain. The team get benefited for multiple times per day and thus providing best overall value.

4. Performance Tests

It is quite hard to perform controlled web application tests manually with several of users. But with automation, it is possible to simulate numerous virtual users to check the product's load capacity and thus making is perfect for automation testing process.

When To Automate?

Test Automation tools are meant to help teams test faster during tight deadlines and when there is a need to free up test engineers so they can focus on tests that require manual attention and unique human skills. Here is a quick list of considerations about when to automate testing:

1. Automate testing procedure when there is a lot of regression work.
2. Automation is considered only when the development code is stable.
3. Automate load testing work for creating virtual users to check load capacity of application.
4. Automate testing when GUI is almost frozen but for the regular functional changes.

How Much To Automate?

It can be argued that almost anything can be automated but it is problematic and arduous to use automation for the whole bunch in practice. So where to draw the line?
Identifying areas that cannot be tested with automation like an application area that are still under structuring, unstable, and mostly in flux can all be handled by seasoned testers. Not all automatable tests must be automated without considering the actual need for it.

Prioritizing automation test efforts on grounds of its outcome can give maximized benefits. The baseline for measuring the implementation progress can be drawn when the Percent Automatable or Automation Index is defined. This can be defined as the percentage of a set of given test cases that can be automated.

How To Measure Success Of Test Automation?

Proper usage of metrics can aid in quantifying test automation outcomes and thus improves the process by tracking its status. Based on the outcome various metrics, defects remaining to be fixed in a testing cycle can be assessed, schedules can be adjusted or goals can be reduced. Here is a brief explanation on how to derive them.

1. Automation Progress

This is to find out the percentage of automatable test cases at a given time. Basically, this metric will highlight and track the growth during the various stages of automated testing development. This metric can be derived with the equation here.

2. Percent Of Automated Testing

Another automated software metric that must be considered is the Percent of Automated Testing Test Coverage. This metric focuses on how much of the product's functionality is being covered, it can be said that it is a metric to measure its' dimension. This metric can be derived with the equation here.

3. Defect Removal Efficiency

A widely known metric for tracking through testing is the Defect Removal Efficiency (DRE), not specific to automation, but a useful metric that is used in combination with automation efforts. Usually, DRE is calculated in percentage. The higher the percentage, the higher is the positive impact of the product quality. This metric can be derived with the equation here.

4. Requirement Coverage

The quality of a finished piece of software is demarcated by its ability to meet the requirements. Requirement coverage looks at the cross section between the business requirements and actual processes or workflows. Instead of only focusing on each atomic requirement at the configuration and execution level, using a workflow model can define stronger and more robust test coverage, with an emphasis on the workflows users actually follow. This metric can be derived with the equation here.
Potential Pitfalls To Be Aware

The first hitch to the system’s failure will be from the lack of mental preparedness of the management and team members. There is a tendency of support fading off among the group after not witnessing instant results during the initial days, this particularly happens when testers are required to test the current release.

It is essential to understand the formalities to be followed in automation for making the system fall in place. Being prepared and giving it the time it needs to stay around long enough to show results will prove to be beneficial. The tendency to ‘give up’ when things are still budding will only ruin a good investment made. Knowing the total test automation investment will help the team and the organization to anticipate and plan accordingly.

Another main reason for test automation failure is a preconception that it is a solution that needs less customization and one condition is enough to suit all. Every changing parameter needs to be addressed through continuous improvements if the automation needs to run efficiently.

An Online Business’s Success With Test Automation

A large direct marketing company that markets merchandise and magazine subscriptions with sweepstakes, prize-based game, search, and lottery websites had to improve the quality of their digital campaigns. The need was to deliver unique high quality user experiences and accelerate their campaign launches without breaking their already tight budget.

The client chose Softcrylic to establish, run and streamline their QA process from a holistic perspective. Softcrylic custom designed a system to recognize business scenarios and delivered a test automation solution with advanced data processing capabilities. The process ensured flawless digital campaigns, improved functionality, creative features and healthy customer feedback that in turn enhanced the client’s brand image several fold.

Conclusion

Automation is a tactic and not a ritual. It is a proven methodology to save several untold hours and more of a change in the mindset of the management, the testers, and the developers.

Embracing test automation will pave way to extensive testing as nothing really stops you from performing the same test several times with diversified test data in various environments. Getting to increase confidence in the system and its quality when more comprehensive tests are performed makes it a good choice. There will be lesser errors left in the system as developers get to see the consequences of a code change almost immediately.

There is a change in the way test automation is weighed off late. While reduction in testing cycle time is seen as the most important benefit of automation, the highest scoring benefit of automation is to achieve better detection of defects.

The Faster You Automate,
The Faster Your Product Goes To Market!
About Softcrylic

Softcrylic, LLC., is an IT Solutions and Services company dedicated to delivering effective solutions to clients since 2000. Softcrylic provides a full range of solutions and services in a variety of technical disciplines including Software Development, Test Engineering and Digital Analytics. For more information, go to [http://www.softcrylic.com](http://www.softcrylic.com) to learn more.

Softcrylic has developed a hybrid test automation tool, Automate-On®, designed to significantly reduce the pitfalls and overcome the challenges, and to deliver the advantages and benefits of intelligent test automation.

Getting Started In Test Automation:

Automate-On® is built on top of popular open source tools such as Selenium WebDriver and Appium. Automate-On® completely eliminates the initial investment to build a test framework and other efforts needed to implement open source tools for test automation needs. It helps Developers and QA engineers with programming language experience to rapidly develop and maintain test scripts. It abstracts all of the complexities involved in open sources tools and significantly reduces that steep learning curve.

Standardizing Test Scripts Development:

Automate-On® is based on a hybrid test automation approach. It is the most comprehensive and flexible functional testing tool available. Automate-On® provides high usability, re-usability and test flow coverage and offers built-in consistency and severity validation. Automate-On® enforces standards in script development and significantly reduces maintenance costs. It will satisfy an organization’s most wide-ranging automation requirements covering multiple applications, platforms and environments, and it provides strong support for distributed testing teams.

Test Cases:

This problem is not solved by Automate-On®, but it is a general practice to optimize the volume of test cases that are chosen for automation. Softcrylic LLC has successfully used test optimizations techniques, including mind map tools, with experienced testers. In one project, the number of test cases selected for automation was reduced from 5714 test cases to 1683 test cases for automation; a 61% reduction in test actions and a 5% reduction in validations.

In-House Lab Versus Cloud Services:

Automate-On® directly resolves the challenges, and delivers the advantages, of using cloud-based testing labs. It is integrated with two of the top cloud testing infrastructure providers, Sauce Labs and Browser Stack, right “out of the box.” Automate-On® provides tight integration with these services, so users can switch from one provider to another without changing any code. If the use of cloud providers is not an option, Automate-On® can be used to execute tests on in-house test infrastructures including real mobile devices.

Test Automation Is Not Necessarily The Tester’s Job:

Automate-On® provides an easy-to-use programming model based on an object-oriented approach for development and maintenance of automated test scripts. It provides all the tools and constructs for anyone with entry-level programming language experience to start developing test scripts.
References

5. Implementing Automated Software Testing: By Elfriede Dustin, Thom Garrett, Bernie Gauf