



Consultants
an ALTEN company

Uncommon (but essential) Metrics for Measuring Software Test Automation ROI

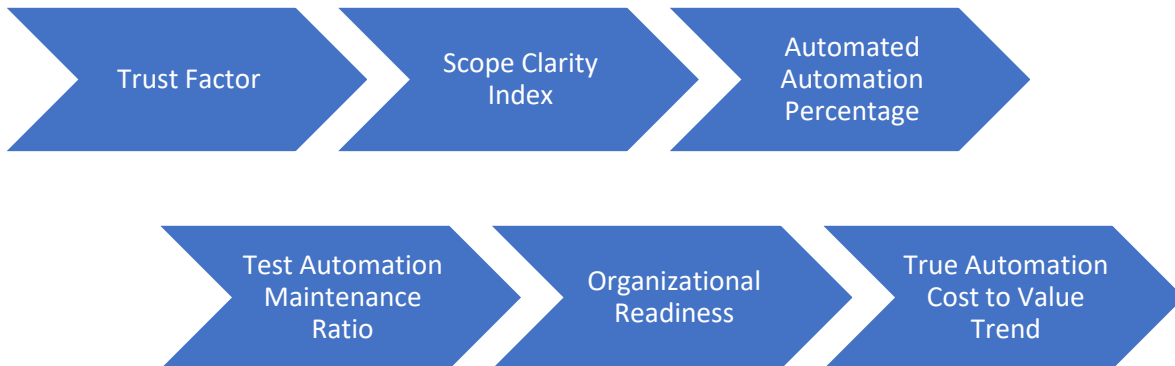
qaconsultants.com

Introduction

Test automation is often seen as an “Easy Button” that is solved with a moderate investment in tools. However, the reality is often quite different and it’s usually a significant investment for a company. To measure the return on investment on your software test automation, it is essential to have the right metrics. Having clear metrics allows stakeholders to understand the value that test automation is providing to the organization. Without these metrics, it’s very challenging to determine whether the investment in test automation was worth the cost. Clear metrics can help stakeholders make informed decisions about how to continue investing in test automation to ensure it provides powerful value.

Analyzing common metrics such as the number of tests run, test failure rates, and the time spent executing tests, stakeholders can identify areas for improvement and optimize the test automation process. While these are important to monitor, they are not designed to drive the right behavior of the overall software development organization. Quality is a team sport where all players must be on the field and attacking the same goal to make it work.

In our experience helping hundreds of customers with test automation projects, there are some uncommon metrics usually missing that would improve ROI for these programs. The top uncommon metrics that we look to implement on all test automation projects, especially those that need refactoring, improvement, and maturity are:



Uncommon metrics and why should they be measured

Collecting metrics for IT projects can be a challenging process for companies due to a variety of reasons. One of the main challenges is obtaining accurate data and measuring progress consistently, especially when dealing with complex and dynamic projects. Lack of stakeholder buy-in and insufficient resources for tracking and analyzing metrics can also hinder the collection process. Furthermore, companies may struggle to effectively communicate and use the metrics they collect to drive decision-making and improve project outcomes.

In many years of providing quality engineering services to our customers, we have realized there are several important aspects of software test automation to measure such as:

1 TRUST: The lack of trust in software testing automation.

This is due to concerns about the accuracy of the results generated by automated testing, which can lead to redundant testing being performed. As a result, the business team may be duplicating testing efforts to compensate for this lack of trust in automation. Overall, there is a need for increased confidence in automated testing to reduce redundancy and streamline the testing process.

2 SCOPE: Issues of not having a clear scope for test automation.

Several issues arise due to the absence of a clear scope, including the lack of hygiene in test automation assets, infrequent reviews of the assets, and no clear understanding of automation efficiency and associated costs. The whole team should understand the need for a more structured and systematic approach to test automation, with clear guidelines for maintaining and evaluating automation assets.

3 AUTONOMY: The elimination of the need for human interaction to run automated tests.

Most companies, test automation still requires manual intervention to run, which can slow down the testing process and introduce errors. Likewise, a poor or non-implementation of DevOps is lacking, which limits the effectiveness of automation. Investing in test automation also requires investments in automating as much of the testing process as possible including DevOps practices to streamline the process and improve efficiency.

4 MAINTENANCE: A large effort dedicated to test automation maintenance.

Often, the majority of the team's time is spent on modifying and updating existing automated tests, which can be a time-consuming and resource-intensive process. There is no proper tracking of changes from a business workflow level, which can make it difficult to assess the impacts of changes on existing automated tests and lead to inefficiencies and errors in the testing process. For a more effective, streamlined and systematic approach to test automation maintenance, it is crucial to have clear guidelines for tracking and assessing changes and their impacts on existing automated tests.

5 READINESS: Challenges related to manual dependencies and pre-and-post-conditions in test automation.

The deployment process is complex and infrequent, and not fully automated, which can lead to inefficiencies and errors in the testing process. The test environment setup and updates are usually not automated, which can further slow down the testing process. Likewise, test data creation and updates are (totally or partially) manually performed, which leads to inconsistencies and inaccuracies in the testing process. It is essential to have a more streamlined and automated approach to test automation, with automated deployment, environment setup, and test data creation and updates, to improve the efficiency and accuracy of the automation testing process.

6 VALUE: Not considering all direct and indirect costs when calculating the ROI of software testing automation.

The costs of test automation are often not fully considered, such as the costs of tools, labor, environments, and expertise, as well as the value of early defect detection, including defect prevention and coverage increase. This can lead to inaccurate ROI calculations and underestimation of the true value of test automation. However, only the costs of manual labor and defect leakage are usually considered, which can further limit the accuracy of ROI calculations. It is critical to consider all relevant costs and benefits when calculating the ROI of test automation to obtain a more accurate assessment of its value.

Uncommon metrics you must consider before calculating software test automation ROI

To effectively manage software test automation ROI, it is essential to track and measure key (uncommon) metrics throughout the project lifecycle. These metrics can provide valuable insights into the performance and progress of the test automation and enable the whole team to identify areas for improvement and make informed decisions.

1 Automated Test Trust Factor

The Trust Factor or Trust Percentage metric is used to measure the amount of functional test coverage that is re-executed by the business team. This metric is an indicator of how much trust the business has in the automated tests and the coverage they provide. The Trust Factor is calculated by dividing the total number of automated functional tests that are not re-executed by the business by the total number of automated functional tests available/executed by the development team. For example, if there are 100 automated functional tests available and the business re-executes 80 of them, the Trust Factor would be 20%. A high Trust Factor indicates that the business has confidence in the automated tests and is relying on them to validate the functionality of the software. On the other hand, a low Trust Factor may indicate that the business does not trust the automated tests and is relying more on manual testing or redundant testing. This metric is applicable only if the business team is re-executing the same tests that the development team had already executed in earlier phases. Also, it is recommended to be calculated considering a specific time frame and compare it to an equivalent period, for instance, per quarter or per year basis.

$$\text{Automated Test Trust Factor} = \frac{\sum \text{Automated functional tests not executed by the business team}}{\sum \text{Automated functional tests available executed by the development team}}$$

2 Test Automation Scope Clarity Index

Test Automation Scope Clarity Index metric measures the clarity and effectiveness of the test automation strategy in a software development project. This metric assesses how well the test automation scope is defined, understood, and aligned with the overall business. The Test Automation Scope Clarity Index is calculated by comparing the actual test automation scope with the business scope, both based on a sum of user stories or functional requirements). This is done by assessing the alignment of the test automation objectives, requirements, and test cases with the overall and current business requirements, more specifically business workflows. A high Test Automation Scope Clarity Index indicates that the test automation scope is well-defined, understood, and aligned with the current business rules, workflows, and processes, while a low Test Automation Scope Clarity Index indicates that there is a lack of clarity in the test automation scope, which can result in ineffective and inefficient test automation. By monitoring and improving this metric, project teams can ensure that the test automation scope is clearly defined, communicated, and understood by all stakeholders.

$$\text{Test Automation Scope Clarity Index} = \frac{\text{Test Automation Scope}}{\text{Business Scope}}$$

3 Autonomy Percentage

The Autonomy Percentage metric measures the degree of automation in the software development process. This metric assesses the percentage of test runs that are executed without any human intervention. The Autonomy Percentage is calculated by dividing the total number of test runs executed without human intervention by the total number of test runs executed. A high Autonomy Percentage indicates that the test automation process is efficient and effective, with a high degree of automation in the testing process, while a low Autonomy Percentage indicates that the test automation process is less efficient and less effective, with a greater degree of human intervention in the testing process.

$$\text{Autonomy Percentage} = \frac{\sum \text{Test runs executed without human intervention}}{\sum \text{Test runs executed}}$$

4 Test automation maintenance ratio

The Test Automation Maintenance Ratio metric measures the effectiveness and efficiency of the test automation process in a software development project. This metric assesses the average amount of time spent on maintaining existing automated test scripts. The Test Automation Maintenance Ratio is calculated by dividing the average person-hours spent on maintaining existing automated test scripts by the average person-hours spent per 40-hour week. A high Test Automation Maintenance Ratio indicates that a significant amount of time is being spent on maintaining existing automated test scripts, which may be due to issues such as test script failures, unknown changes to the application under test, or other factors. A low Test Automation Maintenance Ratio indicates that the test automation process is efficient and effective, with minimal time spent on maintaining existing test scripts. By monitoring and improving this metric, project teams can identify and address issues with the test automation process, such as script stability or maintenance overhead.

$$\text{Test Automation Maintenance Ratio} = \frac{\text{Average person hours spent on maintaining existing automated test scripts}}{\text{Average person hours spent per 40 hour week}}$$

5 Organizational Readiness – Measurement of organizational maturity to support Test Automation

The Organizational Readiness metric assesses the maturity level of an organization in terms of its ability to support test automation, considering the organization's culture, resources, and infrastructure that are necessary to support test automation effectively. The Organizational Readiness metric involves a few key aspects such as **test automation strategy** (the extent to which the organization has developed and implemented a test automation strategy that aligns with its overall business goals and objectives), **skillset and training** (the level of expertise and training of the individuals responsible for developing and maintaining test automation scripts), environment and infrastructure (the quality and adequacy of the organization's infrastructure, including tools, hardware, and software), and stakeholder engagement (measuring the extent to which the organization's stakeholders are involved and engaged in the test automation process).

By measuring those key aspects, the Organizational Readiness metric provides a comprehensive assessment of the organization's ability to support test automation. A high score on this metric indicates that the organization has a high level of maturity to support test automation effectively, while a low score indicates that the organization may need to invest more in terms of culture, resources, and infrastructure to support test automation effectively. By monitoring this metric, the organization can ensure that it has the necessary support structure in place to achieve the desired level of test automation maturity. We recommend each key aspect be scored on a scale of 1 to 5, with 1 being the lowest and 5 being the highest level.

$$\text{Organizational Readiness} = \frac{\text{Test Automation Strategy Level} + \text{Skillset and Training Level} + \text{Environment and Infrastructure Level} + \text{Stakeholder Engagement Level}}{4}$$

6 True Test Automation Cost to Value Trend

The True Test Automation Cost to Value trend measures the ratio of dollars spent to dollars returned through automation. It should be measured per release, as it gets lower over time if done properly. The cost of automation should include the expenses incurred for the environment, labor, tools, and data. On the other hand, the value returned should consider labor deferred, early defect detection and defect prevention savings (which leads to less introduction of low-quality code by developers), and coverage improvement. The True Test Automation Cost to Value Trend is an important metric because it fully determines the ROI of automation efforts and whether they are worth the investment.

$$\text{True Test Automation Cost to Value Trend} = \frac{\sum \text{Cost of automation (environment, labor, tools, data)}}{\sum \text{Value Returned (labor deferred, early defect detection savings, defect prevention savings)}}$$

QAC's Software Testing Automation ROI Scorecard

Each metric can be scored on a scale of 1-5, with 1 being the lowest and 5 being the highest, or based on its percentage or unit value.

(Uncommon) Metric	Description	How it is calculated	Score
Automated Test Trust Factor or Trust Percentage	Measures the amount of trust the business has in the automated tests and the coverage they provide.	Dividing the total number of automated functional tests that are not re-executed by the business by the total number of automated functional tests available/executed	Percentage
Test Automation Scope Clarity Index	Assesses how well the test automation scope is defined, understood, and aligned with the overall business.	Comparing the actual test automation scope with the current business scope and determining how aligned or different they are	Percentage
Autonomy Percentage	Assesses the percentage of test runs that are executed without any human intervention	Dividing the total number of test runs executed without human intervention by the total number of test runs executed.	Percentage
Test Automation Maintenance Ratio	Measures the average amount of time spent on maintaining existing automated test scripts	Dividing the average person-hours spent on maintaining existing automated test scripts by the average person-hours spent per 40-hour week.	Percentage
Organizational Readiness	Assesses the maturity level of an organization in terms of its readiness to support test automation	Considering the readiness ratio of the organization's culture, resources, and infrastructure that are necessary to support test automation effectively	Scale of 1-5, with 1 being the lowest and 5 being the highest
True Test Automation Cost to Value Trend	Determines the ROI of automation efforts and whether they are worth the investment	Dividing the dollars spent (including costs with expenses incurred for the environment, labour, tools, and data) by the dollars earned through automation	Dollar amount

Conclusion

Having effective software test automation metrics is paramount for companies to measure the return on investment (ROI) properly. While conventional metrics such as the number of executed tests, test failure rates, and the duration of test execution are fundamental, they may not adequately stimulate the intended behaviour of the overall software development organization. Therefore, it is imperative to go beyond the standard metrics to enhance ROI for test automation programs. Doing so will enable stakeholders to make informed decisions on how to continue investing in test automation, optimizing the test automation process to ensure that their investment is cost-effective and brings value to the organization.

If you're interested to learn more

Contact us



Consultants
an ALLEN company

qaconsultants.com