Codeless Testing Automation

Codeless Testing Automation

GUIDE TO CHOOSING THE RIGHT TEST AUTOMATION TOOLS

INDEX

INTRODUCTION

There are many misgivings about Software Test Automation – some of which stem from being overwhelmed by the sheer volume of information out there; others from ill-informed decisions; yet others from the wrong choice of Test Automation Tools; and still others from haphazard implementation. This treatise therefore sieves through the information overdose, to put the spotlight on presenting an informed view on Test Automation, as well as on when and how to approach it.

It must be remembered that Test Automation is not a magic wand that puts everything in place with just one wave of the hand. It requires time and effort to study the organization's needs, so that well-informed decisions are made, which will bring good returns on investment in Test Automation tools, rather than be a drain on the company's resources. Test Automation must therefore be viewed from a multi-dimensional approach.

In order to get the most out of test automation, it is important for organizations to review the need for it and only then hop on to the bandwagon. The starting point is the basic question: Is Test Automation required for the organization? If the well-analyzed answer is yes, then an important consideration is: Which tool **best suits** the organization?

A noteworthy point is that the best Test Automation tools in the market are not necessarily the tools that best suit your organization. The choice has to be for the **right tool**, as tools that may excellently suit one entity may not necessarily suit the needs of another. There are many considerations that will help narrow down the choice to the **Right Test Automation Tool**. This Whitepaper will explore these in detail, in order to help organizations reap the maximum benefits, by investing in the most appropriate Test Automation Tool.

ABSTRACT

In the world of Test Automation, tool selection plays a vital role. Wrong tool selection can spell failure for the Test Automation program. This failure is also accompanied by wasted time, money and efforts. On the other hand, time and efforts invested in selecting the right tool, will make the future easier for all concerned. This Whitepaper therefore seeks to throw light on this important topic and has been presented in two sections.

The first section titled '**The What and When of Test Automation**' is sub-divided into two parts:

- The first part presents an overview of Automated Testing and contrasts it withManual Testing for a better understanding.
- The second part explores 'when' it is time to automate, and also presents a checklist each, for situations that favor Manual Testing, and those that favor Automated Testing.

The second section titled '**Choosing the Right Test Automation Tools**' is the crux of this Whitepaper and provides guidelines for investing wisely in the tool that best suits the organization.

These guidelines are summarized below:

- Analysis of the Project's Needs
- Features Considered Necessary in the Tool
- Team's Programming Knowledge and Skills
- Budgetary Constraints
- Ease of Test Case Creation and Test Case Maintenance
- Script Reusability
- Support for Keyword and Data-driven Testing
- Integration Capabilities
- Compatibility with Modern Methodologies like Agile and Devops
- Functional Reporting
- Facilities for Collaboration between Stakeholders
- Training, Technical Support, and Updates.

THE WHAT AND WHEN OF TEST AUTOMATION

This section presents an overview of Automated Testing and Manual Testing, in order to contrast the two, and also understand the place of each of them in contemporary software testing. This understanding provides the foundation on which the decision of what and when to automate is based.

01 The 'What of Test Automation'

Test Automation is the practice of running tests automatically by using test automation tools to manage and maintain test data, execute tests, and analyze the test results, to improve the quality of software. The process consists of writing test scripts or programs that are replicated by automation tools, to automatically execute the various steps that would have been physically done in Manual Testing. The test scripts execute the given instructions and present the test results verifying whether they are as expected. Test Automation tools help to speedily generate values or events in response to the test script/program. There are different types of Automated Tests like Unit Testing, Integration Testing, Smoke Testing and Regression Testing.

In order to make the right decision on what is more suitable for the organization, it is important to have an unbiased understanding of Automated Testing and Manual Testing, and hence this is presented below:

Automated Testing seeks to provide quick, error-free verification of the app which in turn helps meet the dire need for speed, accuracy and security which are so vital to software testing today. The focus is on standardizing testing to ensure consistency and reliability, thus building confidence in the app via superior verification of the software. To reap full benefits of test automation, it is important that the codes cover all relevant aspects and take cognizance of every complexity, as the app consists of hordes of integrated and inter-connected codes. Meticulous planning, choice of right automation tool/s, and careful comprehensive implementation are vital for perfecting test automation.

Manual Testing QA professionals verify as well as validate the software, keeping in mind smooth and seamless usage of the app in users' hands. The tester gets into the shoes of the user, and the emphasis is on providing superior user experience, which is a very important consideration for app success. The human touch that Manual Testing provides, especially in case of Exploratory Testing, and where the touch and feel of the app is important – makes it a winner when human perception is an important criterion for app success. In short, Manual Testing provides human validation that the app will meet users' expectations.

Having reviewed what Test Automation is about and contrasted it with Manual Testing, it is important to understand when it is time to adopt or transition to Test Automation.

02 The 'When of Test Automation'

In deciding on when to automate testing, the first question that needs an impartial answer is: Does the project require Test Automation? True, Test Automation has immense benefits, yet it can't be denied that Manual Testing pays focused attention to the vital viewpoint of 'superior user experience'. The project's needs therefore become the starting point in deciding whether to automate and when to transition from Manual to Automated Testing.

For facilitating better decision making, a set of pointers follow, which spell out the situations that support Manual Testing, and situations that favor Automated Testing.

Manual Testing is preferable in the following cases:

- User Interfaces (UI) like visual aspects are to be tested where human insight is important.
- There are frequent changes in the UI of the system that is being tested.
- The project is a short-term one where Manual Testing proves to be faster than script writing.
- Exploratory or Ad hoc testing is required.
- Test cases cannot be automated.
- The project is in its initial phase. (However there is a growing trend to automate from the beginning, to enable early error detection and avoid costly rework later.)

In all other cases, Automated Testing is the way forward.

Automated Testing is the practical choice especially when:

- Test cases are consistently repetitive.
- There are frequent software updates.
- Speed, accuracy and security are non-negotiable.
- Test coverage is sought to be increased.
- Risks related to human element are high.
- The same set of test cases need to be run simultaneously, more so on multiple machines.
- A single functionality has to be tested with multiple data sets.
- Frequency of Regression Testing is high.
- The team has to simulate a large number of users for Performance Testing.
- Critical functionalities cannot solely rely on Manual Testing.

It can be concluded that there are still areas where Manual Testing is preferable, even though in general Automated Testing is seen as vital for the fast paced digital world. Ultimately it is the project's needs that will define which type of testing is more suitable. A careful study of the project is therefore important to make the right decision. A balance of both Manual as well as Automated Testing may be required to best suit the project's needs.

However, once the organization conclusively decides that its need for speed and accuracy is non-negotiable, then it must seriously explore the path of Automated Testing.

The next section of this treatise will deal with a very important aspect of Test Automation viz. selecting the right Test Automation tools.

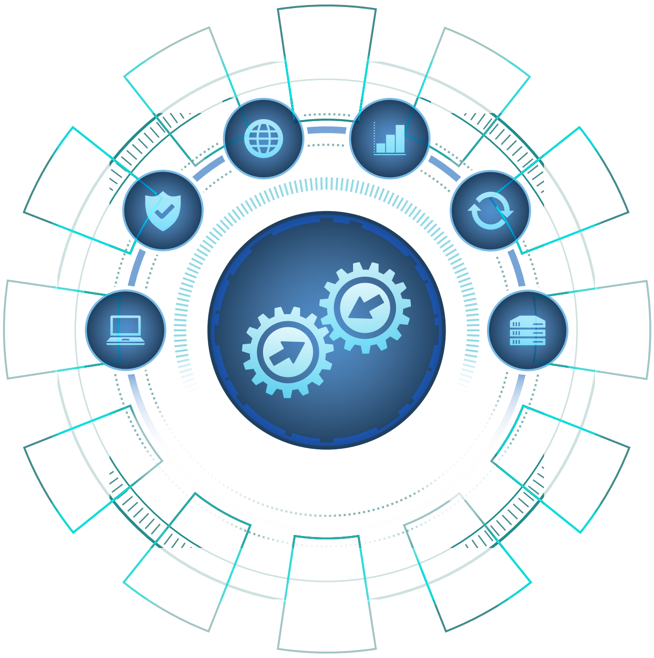
CHOOSING THE RIGHT TEST AUTOMATION TOOLS

Choosing the Right Test Automation Tool does not mean choosing the best tool available. There are Mobile Apps, Web Apps, and multiple Industry-specific Apps geared to meet industry specific needs and regulations. These differ in complexity and nature, and hence it cannot be presumed that an automation tool that works well for many projects will work well for all.

There are a variety of Test Automation Tools some of which are code-based and others codeless. Jest is a JavaScript Testing Framework designed to ensure correctness of any JavaScript codebase. Selenium is an open-source Automated Testing Framework used to validate web applications across different browsers and platforms; and requires knowledge of programming language to customize it. Appium is an open-source automation tool for running scripts and testing native applications, mobile-web applications and hybrid applications on Android or iOS using a web-driver. Basic knowledge of programming language is helpful here. There are also 100% codeless, comprehensive platforms like BOTm, which is an accelerator BOT for automated and manual testing of mobile applications – developed for both Android and iOS devices.

The fact is that the market offers a host of automation tools which proclaim their claim to fame, but in reality, automation tools vary in features, complexity and purpose, and are most certainly not equal in performance, capabilities and offerings. Some tools have excellent test reporting features; others have great flexibility which makes integration with third-party tools easy for extending and customizing functionalities; some have user interface dashboards that promote smart capture of insights and analysis; others have robust test script creation tools; some are empowered with in-built AI & ML facilities to reapply test cases to recurrent situations and scenarios... and the list of differentiators goes on.

Clearly a lot of thought must go into selecting the right Test Automation Tool in order to ensure success of Test Automation. Time invested in zeroing in on the right tool will be substantially rewarding far into the future. Investing in the most suitable tool can bring great benefits and conversely the wrong tool will be a drain on time, efforts, and financial resources.



Guide to Selecting the Right Test Automation Tools

Analysis of the Project's Needs

A study of the project will reveal important aspects required in the tool, and this in turn will speedily narrow down the list to the most appropriate ones. Pertinent questions are:

- Is it a Web, Mobile, API or a Desktop App?
- Which are the Platforms that need to be tested?

  - Which Mobile Operating Systems for Mobile App?

  - Which are the supported Browsers for Web Apps?

  - Which Operating Systems for Desktop Apps?
- Which Programming Language was used for developing the app?
  This is relevant in case of code-based automation.
- Does the app need Cross-browser or Cross-device Testing?
  This gains relevance for Mobile Apps and Web Apps.
- Are there any other desirable or important requirements?

Features Considered Necessary in the Tool

A long-term view of what features are desired in the tool will be helpful to avoid future regrets. Hence it is important for the core team to come to a consensus about what features are a must have, what features are desirable, and what features are non-value adding for the project. This will help the organization avoid ending up paying for fancy features which are not value for money.

Some important features are referenced below:

- Check on supported platforms
- Applied application under tests
- Programming Languages
- CI/CD integration capabilities
- In-built AI & ML features
- Self-testing and self-healing ability
- Reporting functionality.

Team's Programming Knowledge and Skills

Test Automation Tools, as mentioned earlier, can be code-based or codeless. For code-based tools, programming knowledge is required for efficiently writing and executing test scripts. This is the case with most open-source tools. Consequently it is important to understand the skill sets of the team before finalizing the Test Automation Tool. If the current team is to be productively employed, the tool should be in the programming language that the team excels in. If the team does not have a strong coding and IT background, opting for a codeless tool would be the ideal way forward. If a new team has to be hired or put in place, then it is important to check the availability of the required skill sets in the market, and the costs that will be involved vis-à-vis the tool's human skill set requirement.

Budgetary Constraints

Analysis of the Project's Needs

The cost of test automation goes beyond the obvious and hence to decide on the right budget for the tool, or to meet the existing budgetary constraints, a holistic costing approach is
required which considers the following:

- Cost of the Test Automation Tool.
- Cost of machines required to run the tool.
- Cost of other infrastructure like Cloud hosting.
- Cost of manpower with the required skill sets for operating the tool.
- Man-hours required for training the team to use the tool.
- Time and resources required for creating and maintaining test cases.

An open-source solution that saves on the cost of the tool seems like the ideal option, but then the other considerations mentioned above come into play and could prove more expensive in terms of cost of manpower, machines, infrastructure, time value of money, learning curve etc. A codeless tool on the other hand will save on costly skilled manpower, learning curve, time and resources spent on coding, maintenance etc.

Ease of Test Case Creation and Test Case Maintenance

It's a no brainer that the tool selected must suit the project's needs, but what's also important is to check the cost of test case creation and maintenance which can be quite significant. An AI-powered self-testing, self-healing tool will greatly reduce time and resources spent in test case creation and maintenance. Since tools vary in their ease for test creation and maintenance, it is advisable to check the trial version of the shortlisted tools to compare how easy or complex it is to create and maintain test cases mandated by the project. It helps to note that tools equipped with the flexibility of pausing, can help enhance debugging.

Script Reusability

Script writing can take up a lot of time and this is something that is ideally avoided in the speed-driven software world. The same code is required ever-so-often for multiple test cases. Having a tool which promotes code reusability would definitely save on time and efforts by avoiding the need for duplication.

Support for Keyword and Data-driven Testing

Some apps are heavily data driven which necessitates testing of an assortment of data at various interfaces. For such apps, the tool chosen should be one that supports and promotes keyword and data-driven testing.

Integration Capabilities

For uninterrupted testing, the tool should be able to integrate into CI/CD pipelines and external platforms. If there are existing/proposed process improvement or CI/CD tools, then the selected tool must support integration with these. Enhanced integration with Alexa and Whatsapp can greatly add to the cognitive features. Comprehensive integration goes a long way in improving test management, as well as enhancing team collaboration.

Compatibility with Modern Methodologies like Agile and Devops

The tool must support modern methodologies like Agile and Devops as these greatly contribute to avoiding bottlenecks in the Software Development Life Cycle (SDLC). The Agile model facilitates continuous iteration of development and testing, thereby speeding up the SDLC; and Devops which is a combination of practices and tools, is designed to speed up the organization's ability to deliver apps and services.

Functional Reporting

The Test Automation tool’s reports should be useful, analytical and comprehensive. It’s not the quantity but the quality and relevance of reports that matter. The tool should therefore generate reports which provide the information that the team needs; and will help speed up testing and decision making. Examples of useful reporting features are listed below:

- Screenshots for failed steps.
- Clear-cut indicators of failed test cases/steps which reveal where the error lies.
- Test execution videos which permit going back on the execution track to review failure points for analysis.
- Stack trace which generates when the app crashes because of an error or an exception.
- Time taken for execution of test steps and test cases.
- UI dashboard that provides smart capture of insights and trend analysis of test execution.
- Advanced features like WhatsApp Chabot through which resulting log reports can be sent directly to the user's physical device.

Facilities for Collaboration between Stakeholders

Test Automation Tools that promote and support collaboration between the Team, the Client, the Management, and other non-technical stakeholders, can help concurrent reviews by all stakeholders, so that corrective action and suggestions can be incorporated early.

Training, Technical Support, and Updates

Last but not the least is the training, technical support, and updates offered with the tool. Training offered by the vendor can save a lot of man-hours, which would otherwise be wasted on trying to understand how to use the tool. Trial and error definitely has no place in the field of test automation where speed and accuracy are non-negotiable.

24x7 technical support is important to ensure that there is no downtime in the testing process.

With the speed of digital innovation, care also needs to be taken to select a tool that incorporates current technology; and keeps abreast of innovation, releasing updates regularly to customers.

The terms and conditions for training, technical support and updates need to be carefully scrutinized to safeguard a smooth future run.

It is hoped that these guidelines will help organizations invest wisely in the best fit automated testing tool.

Most vendors provide a free trial to enable testers to get a feel of the tool and its capabilities. It’s advisable to take advantage of this to fine tune the final decision, and invest in the tool that best suits the project/organization.

CONCLUSION

Test Automation Tools are a must for projects where speed, accuracy and security are non-negotiable. However for test automation to succeed, it has to be a carefully thought out process. Selecting the right tool is as important as meticulous implementation of test automation. It is paramount that the tool selected must meet the project's requirements, since tools vary in their purpose, performance and offerings. Choosing the most suitable tool requires time and effort in the beginning, but will reap great dividends far into the future.

Selection of the right tool/s must be done keeping in mind the long term needs of the project. Points that need consideration include: Ease in developing and maintaining test scripts; Ease of test execution for non-technical users; Cross-browser testing ability; Support to keyword and data driven testing; Test reusability; Integration capabilities; Level of technical support and assistance; Language support for writing customized scripts; Support for continuous testing; Frequency of tool updates; and of course Pricing.

For Mobile App Testing, BOTm is a technically sound, value for money automated testing option which caters to the entire spectrum of mobile app testing. Its in-built state-of-the-art app testing technologies ensure speed, accuracy, and security; and additionally offer features like audio interaction with Alexa; CICT using Jenkins; Death of a Device Cloud; and Appium Converter. Visit **www.botmtesting.com** and sign up for a **Free Trial**, to experience the benefits of this world-class testing platform, which ensures affordable stress-free, error-free testing.

GET IN TOUCH

☎ 022 4050 8200

✉ sales@botmtesting.com | www.botmtesting.com

BOTm is the accelerator BOT for Automated and Manual Testing of mobile applications - developed for both Android and iOS devices.