5 KEY CONSIDERATIONS FOR

Evaluating UI and Endto-End Automated Testing Solutions





Why invest in test automation?

Nearly every software-driven company is trying to innovate faster and release code more frequently. Yet, delivering poor software quality can lower customer satisfaction and hurt profitability. Software testing remains a bottleneck in many organizations due to slow authoring and instable tests. A robust test automation platform is designed to accelerate the software testing process and ensure high-quality releases. The benefits include:

	BUSINESS DRIVER	DEV INITIATIVES TO ACCOMPLISH DRIVERS	SPECIFIC GOALS TO COMPLETE INITIATIVE	HOW TESTIM SUPPORTS YOUR GOALS
Time	Increase Revenue	Deliver innovation faster	Shorten feedback loop, shift left	Coded or codeless tests, branching, parallel runs, grids
Quality	Customer Satisfaction	Increase test coverage	Build test automation	Fast authoring of AI- stabilized tests
Cost	Reduce Dev Costs	Improve productivity	Create a stable test environment that aligns to developer workflows	Dev tools integrations, code or codedless, advanced debugging

Test automation allows teams to author and execute tests faster, easily maintain tests, report bugs quickly and clearly show testing results.

Below is a 5-step process to help you identify the best test automation platform that not only solves your challenges today but gives you flexibility to change as your software development processes and tools evolve over time.

STEP	01	IDENTIFY NEEDS AND CHALLENGES
STEP	02	CONSIDER VENDOR ATTRIBUTES
STEP	03	EVALUATE CURRENT COSTS AND BUDGET
STEP	04	DETERMINE REQUIRED FEATURES
STEP	05	ANALYZE POST-PURCHASE NEEDS



01 - Identify needs and challenges

Your test automation challenges and needs will vary depending on where you are today. Most organizations will fall under one of three situations:

Situation 1: No test automation solution

Organizations in this category tend to be small or midsized with limited or no QA support staff. Their development processes and toolchains are simple yet standardized with limited variability. To this point, they have met their testing needs manually, though the cost and effort is starting to grow. Such systems typically require only basic integrations with bug tracking tools and CI solutions.

Situation 2: Underperforming test automation solution

Often these organizations have tried an open source solution and found it to be less automated than they had hoped, requiring special skills, long

test authoring times, or constant maintenance of tests. Others may have tried a vendor tool that promised to solve the problems of open source solutions, but didn't have the maturity or functionality to fully test their applications.

Situation 3: Already own a big legacy ALM suite

Larger, often established organizations with complex development processes fall under this group. These organizations have well-defined QA processes and globally distributed teams. Standardized tools may exist, but it's likely that there is a wide variety including legacy technologies that require custom integrations.

Define your business drivers

Once you determine your particular situation, the next step involves defining the exact business drivers that you are trying to resolve with a test automation platform. These include, but are not limited to:

- Accelerate time to market
- Improve software quality
- Integrate and align with your development tools
- Reduce software development/deployment costs
- Reduce time to identify and fix bugs
- Increase test coverage and improve traceability
- Foster better team communication and collaboration

Identify your challenges

Software applications are becoming more sophisticated, leveraging open source and third-party components that add functionality but also increase complexity. Further, organizations are quickening release cycles to stimulate innovation, learning, and growth. These trends place pressure on QA teams to automate and streamline testing processes. Consider these questions when evaluating test automation solutions:

- Is slow test authoring impeding your ability to effectively test new functionality in a timely manner?
- Do your QA teams spend too much time maintaining or updating tests?
 How does that impact your costs?
- Does your team spend a lot of time maintaining your testing environment due to different operating systems and configurations?
- Do you have distributed teams that require visibility into testing status for better collaboration?
- Does your test automation solution fit into your DevOps tools and processes to simplify branching, test runs, status, and bug reporting?



02 - Consider vendor attributes

Once you have identified your needs and challenges, you are ready to start vetting potential vendors. On the surface, many test automation solutions appear quite comparable, with vendors claiming similar features and benefits. A deeper look, however, may reveal differences in product maturity, fit with your tools and processes, customer support, and costeffectiveness.

Here are some key criteria to consider:

- Maturity is their solution missing key features? Does the vendor provide frequent feature updates? Is it old technology that makes it less usable or inflexible?
- Customer base does the vendor support customers like you? Do they have good online reviews? Can you live with reviewer's criticisms about their solution?
- Training services does the vendor provide any kind of training to help customers learn how to navigate their way around the platform?
- Support what kinds of technical support do they provide? Do they offer dedicated customer success representatives to help you be successful?
- Fit does the solution integrate with the 3rd party tools your team uses? Does the technology fit your team's skills so they can work effectively?
- Security does the solution meet your security needs? Do they have any



03 - Evaluate current costs and budget

It's not uncommon for organizations to focus on the license fee when evaluating automated testing tools and then fail to fully account for other hidden costs. The total cost of ownership (TCO) of any test automation platform includes the following costs:

- Fit to development methodology (separate QA team, QA inside the dev team, dev authoring tests)
- License, maintenance, and support fees
- Platform customization and integration to tool set
- Ramp-up time to get fully productive
- Data conversion and migration

The organization should determine their expected return on investment (ROI) after setting realistic expectations of benefits. Many test automation solutions require time before they can begin to demonstrate their true value. It's also equally important for the business to develop specific metrics (both qualitative and quantitative) for measuring the effectiveness of the test solution. These could be aspects such as improvements in software quality or increased test coverage.



04 - Determine required features

At this stage you have already established your goals, challenges, costs and vetted different vendors. It's now time to determine the exact features that you require in the test solution. You can use the chart below to compare broad-based features between different vendors:

- How to use this table:
- 1. Read through all of the features/capabilities and then prioritize them in order of importance.
- 2. Weight the importance of each feature to your organization using percentages from 0% to 100%.

3. Score the vendors from 1 (worst) to 5 (best) for each feature/attribute.

FEATURE	CAPABILITY	PRIORITY	WEIGHT	VENDOR 1	VENDOR 2
Test authoring	How easy is it to write automated tests? Your developers, testers, BAs and anyone who knows how to use your product should be able to do UAT. Can you record or code stable tests? Can all members of your team participate in QA?				
Maintenance	How much time is spent maintaining flaky tests? The solution should make it easy to determine exactly why the test failed without having to worry about the stability of the test itself.				
Toolchain integrations	How well does the tool integrate with your pipeline and release process? Does it integrate with Dev tools such as Git, collaboration tools like Slack, bug tracking tools like JIRA and CI tools like Jenkins and test management solutions?				
Stability & confidence	Do the tests break? How well do the tests handle code changes? Are there any Issues with timeouts? The test automation solution should help you troubleshoot quickly with error aggregation, screenshots, DOM of failed steps, network logs, and console logs.				
Multi- platform support for web & mobile	What's the process for running the tests continuously and on different browsers? Does the solution support different OS, browsers, and application for software you would like to interface with for combined issue reporting? Can you run tests on third- party test grids?				
Product direction	Does their product roadmap align with your future anticipated needs? Do they release new features on a regular basis?				
Reporting & traceability	Does the tool offer visibility into the possible meaning of the results as well as enriched information (e.g. logs, DOM, screenshots) to help trace why a test failed?				
Data migration	Does the tool allow you to reuse tests from previous solutions? If not, is it easy to recreate old tests on the new platform?				
Deployment	How easy is it to install and configure the test environment? Can the tool be used in the cloud via any web browser?				
Customer support	Does the vendor offer adequate customer support and training? What are their SLAs? How do they measure your success?				
Enterprise grade	Does the solution meet your requirements for security, data portability, user management, etc.? Enterprise-grade solutions should provide robust security features, easy data portability, audit trails, test case versioning, disaster recovery and more.				
Ramp up time	How long does it take for users to become operational and start realizing the benefits of why you purchased the solution? Ask to speak with customers to get a real idea of their on-boarding experience.				
Skill set fit & dev cycle fit	Does the tool require skills like coding or scripting? If so, are those skills accessible to your QA team?				
Risk management	Can you take your tests with you if you leave? Can you run them on another platform?				

Now calculate the weighted score (sum of weight X score for each line item). The vendor with the highest weighted score has the best set of capabilities that most closely match your priorities.



05 - Analyze post-purchase needs

The last step in the evaluation is understanding the onboarding process and the steps to achieve operational effectiveness. Will the vendor provide white-glove service to ensure your success?

The purchase of software is a start of a partnership. You need to carefully consider the following factors to make sure you're up and running quickly to realize the benefits of your investment:

- Implementation process Does the vendor have a well documented 30/60/90 day onboarding process which can be tailed specifically to your team's needs and timelines?
- Migration does the vendor handle data migration and if so, how much do they charge for the service?
- Seamless integrations will the vendor help you integrate with your dev stack and assist with testing to ensure everything is working correctly?
- Deployment will the product be deployed exclusively in the cloud or does the vendor provide an on-premises solution?
- Training/support will the vendor provide training and ongoing support to ensure product adoption? What is the vendor's support SLAs?
- Backup/disaster recovery what tools and mechanisms does the solution employ for backup and recovery tasks?
- Scalability how easily can the solution be scaled to handle both your near- and long-term workload requirements?
- Vendor lock-in if you decide to leave the vendor, can you take your tests with you? Can they be reused on another platform?



For more information contact us at info@testim.io or call +1-415-535-6690