Test Automation Buyer's Guide

mabl

> *"While test automation is not new, past automation approaches have not helped app dev teams keep up with increased demands of modern delivery."*

The Forrester Wave™ for Continuous Functional Test Automation Suites, Q2 2020

In the past, automated testing was seen as a “nice to have” luxury — businesses relied largely on manual software testing to ensure applications were delivered error-free. But today’s companies realize that to be competitive, they must embrace automation. Automated software testing frees up valuable developer and QA resources, allowing those employees to focus on higher-value tasks. And while selecting and implementing an automated testing solution is not without its challenges, few investments can have a more significant impact on your customer happiness and your bottom line.

In this guide, we’ll talk about why test automation is so important and explore the true cost of poor software. We’ll also discuss how you can tell if your business should consider implementing automated testing, how to build a business case for test automation, and how to manage the transition to quality engineering. And finally, we’ll give instructions on how to effectively evaluate test automation solutions and identify key features to look for when buying an automated testing platform.

01

The True Cost of Poor Quality Software

> *“As organizations undertake major digital transformations, software-based innovation and development rapidly expands. The result is a balancing act trying to deliver value at high speed without sacrificing quality. Generally, however, we are not very good at balancing. Software quality lags behind other objectives in most organizations. That lack of primary attention to quality comes at a steep cost…While organizations can monetize the business value of speed, they rarely measure the offsetting cost of poor quality.”*

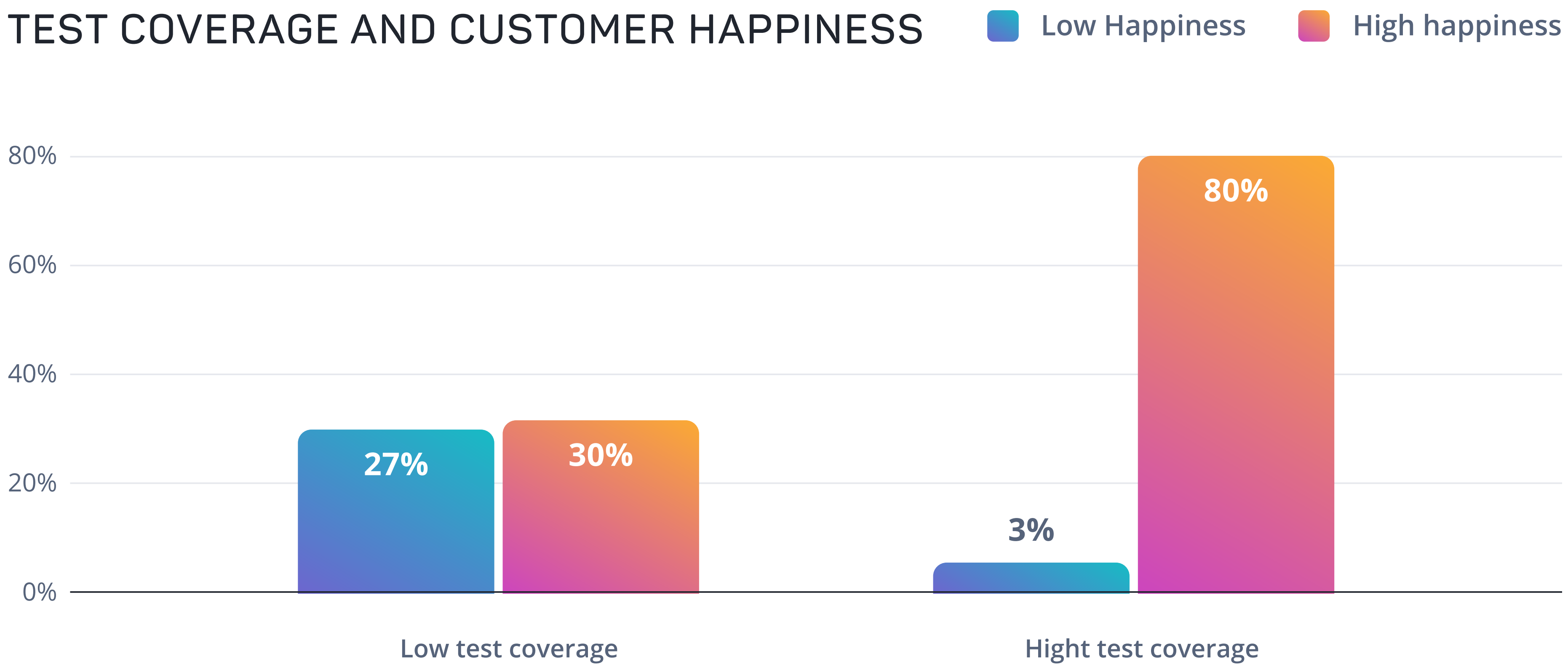
Consortium for Information and Software Quality,

The Cost of Poor Software Quality in the US, A 2020 Report

Poor quality software doesn't just frustrate users and create headaches for your customer support team; it can have a considerable impact on your company's bottom line. The Consortium for Information and Software Quality's 2020 report found that in the United States alone, poor quality costs organizations more than $2.08 trillion. And unsuccessful development projects — largely the victim of quality issues — account for $260 billion.

Mabl's Testing in DevOps survey of 600+ testers, developers, and leaders also speaks to the impact of software testing on costs and revenue. Among survey respondents, higher test coverage was directly correlated to higher customer product ratings. In fact, 80% of teams who described themselves as having "high test coverage" also reported high customer satisfaction. It's no surprise that only 30% of low test coverage teams reported high customer satisfaction.

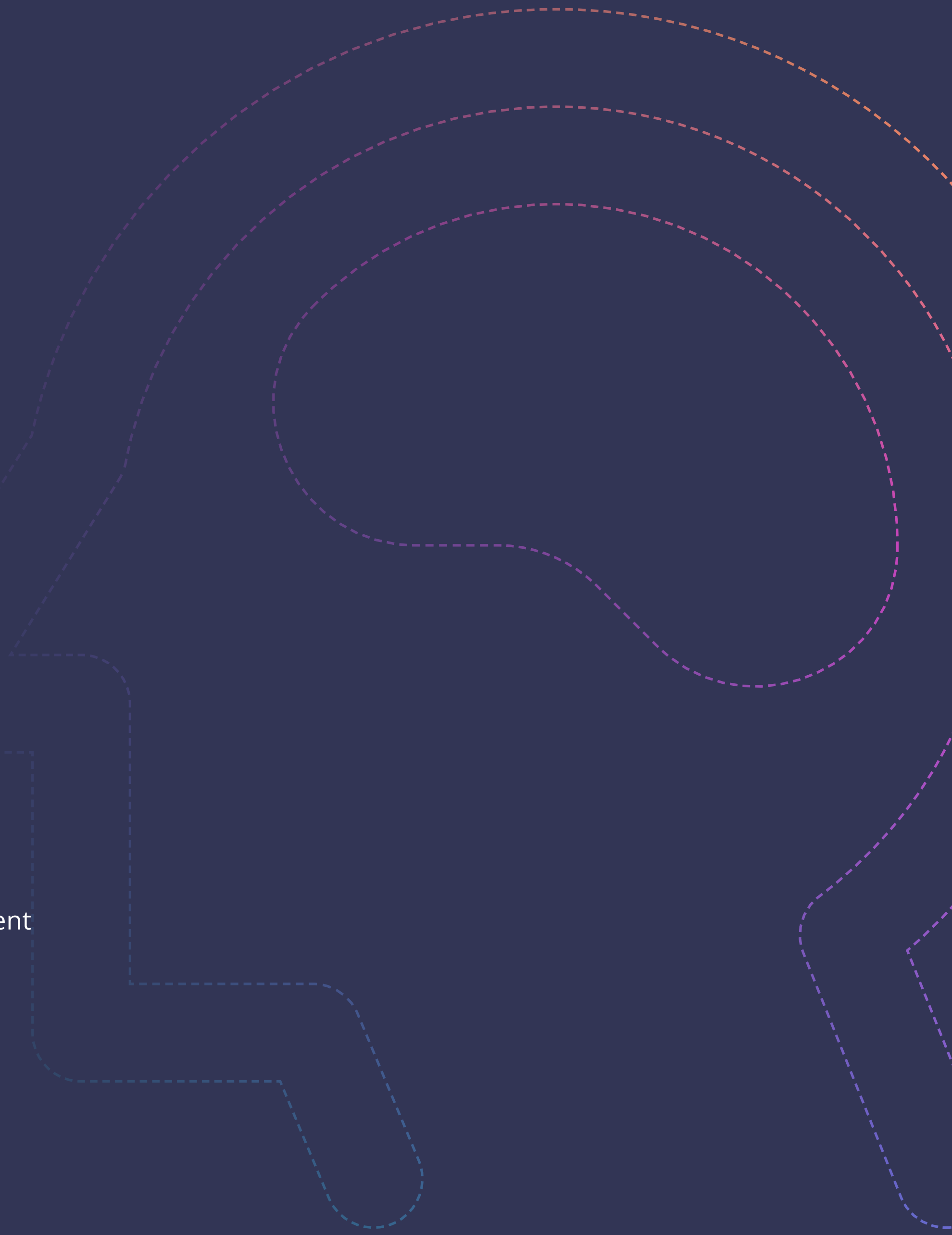
TEST COVERAGE AND CUSTOMER HAPPINESS



02

Understanding When It’s Time to Search for a New Solution

How do you know it’s time for your organization to research, select, and implement a test automation solution? Here are some signals to watch for.



Testing is a deployment bottleneck:

Is your team deploying software into production with minimal testing? Or are testing activities holding up deployments to production? If you constantly feel behind or like you can never "catch up" with all of your testing tasks, your current framework may be hindering product velocity.

You can't see trends beyond pass/fail:

Pass/fail information without context can send you scrambling for answers, slowing down development cycles. Did it fail in the app, the test, the infrastructure, or somewhere else? Does your current testing solution help you identify problematic areas? Does it provide unified logging and dashboards to surface insights and quality trends when you need them? Without easy access to test data and trends, you might be spending hours each week searching for failure causes and analyzing trends.

Not enough people can contribute to test automation:

Are your current testing processes easy for everyone - from developers to manual testers - to learn and use? Does a fraction of your team have the knowledge to script tests? When scripting isn't required for test creation and maintenance, more members of your team can contribute to increase test coverage and optimizing quality.

Test coverage is declining or inaccurate:

Your organization sets up test coverage goals to monitor your testing quality, but how well are you meeting them? Can you assess if you're testing the most important user journeys? If you find that your team is aiming for an arbitrary coverage number, or is having difficulty improving test coverage, you may be letting too many bugs fall into production, affecting your users' experience.

Test maintenance has become your primary task:

In many high-velocity organizations, consistent updates to the application can create brittle tests. As a result, teams spend hours each week updating scripts to prevent false negatives. If your team wasn't required to maintain tests each week, they could be spending that time on activities to increase test coverage instead.

Defect resolution delays releases:

Does your team spend more time fixing bugs than building new features? Or worse, how many users experienced a bug before your team was able to implement a fix? If defects are difficult to identify in your application - and take even longer to resolve - user happiness can be significantly affected.

You are experiencing higher customer churn:

One-in-three customers will switch brands after a single bad experience. If your organization is seeing an uptick in customer churn, identifying any defects that are hindering the user experience is critical to rebuilding their loyalty. If your customers are citing poor quality or performance as their reason for finding another vendor, your business would likely benefit from automated testing.

Testing activities are siloed from the development team:

A core pillar of DevOps is collaboration, and to successfully adopt DevOps, teams need to integrate testing activities into the whole team's workflow. Are developers using a different solution to test than QA? Are tests required to pass before merging code? Your test solution should integrate with the tools your team is already using to trigger tests in development, and notify the right teams when there is a failure.

Unplanned work is taking over sprints:

While some unplanned work is typical for sprints, there should be a balance between new feature development and issue resolution. When time spent on bug resolution takes over feature development, teams should consider integrating testing further into development. Integrating test automation into your development pipelines can simultaneously catch bugs earlier in development and increase product velocity.

03

How to Build A Business Case for Investing in Test Automation

Everyone wants software to get to market faster. After all, the faster you can respond to market trends and customer needs, the bigger your competitive advantage. But helping management understand the critical role that test automation plays in the development process can be difficult.

To help make the business case for investing in test automation, you must do three things:

- Tie the testing automation change to the value it provides
- Connect quality success with business success
- Approach testing as a team sport

Leverage data to show automated testing's value

To build a business case around how quality impacts the business, leverage your data. To start, understand your current state:

- How long does it take to create a new automated test?
- How long does it take for tests to run?
- How many hours does the team spend on test maintenance?
- How many hours does the team spend on manual testing?
- How many bugs are reaching production per month?
- How long does it take to resolve each bug?
- What is your current test coverage?

Depending on your product and team goals, it may be useful to break these numbers down by time spent per week or by sprint. There are also nuances in test cases, so you may look to quantify each metric by the complexity of a test case.

Run a proof of concept (POC) or proof of value (POV) with a new testing solution to understand the impact of automated testing on team efficiency, test coverage, and product velocity. Leverage this page to document your current state, and understand the return on investment a test automation solution can provide.

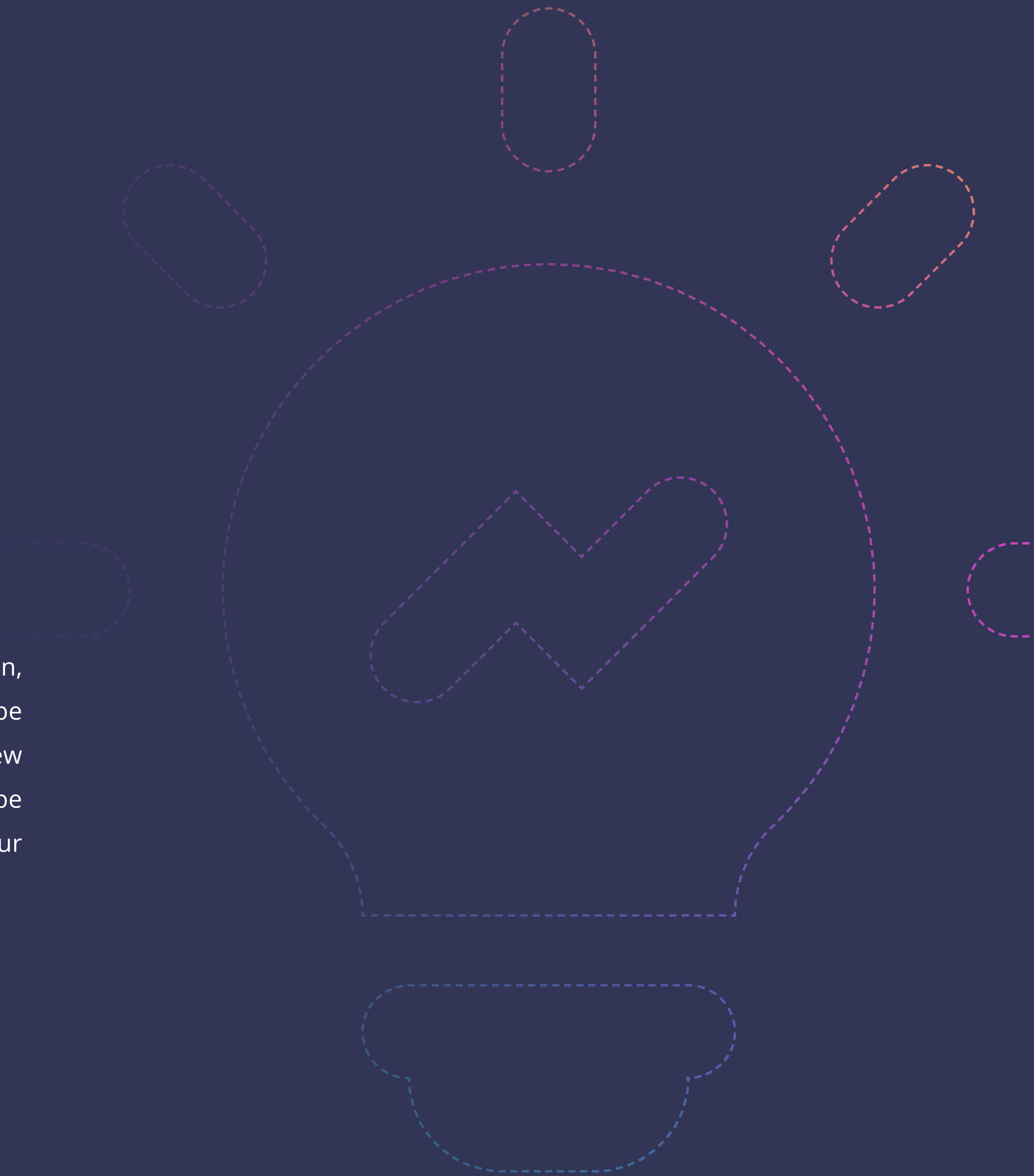
Connect quality success to business success

Quality success and business success go hand-in-hand. Better testing equals happier customers and a more robust bottom line. Leveraging this ROI calculator, help your management team understand that by automating testing, employees can focus on innovation and high-value activities that will ultimately keep existing customers satisfied while also attracting new ones.

04

Tips for Managing Change

After proving the value test automation can bring to your organization, implementing new technology can be tedious. But what can be even trickier is convincing change-wary employees to embrace new processes and solutions. For your automated testing initiative to be successful, you must build change management activities into your evaluation process.



Get the timing right

Align your evaluation and readiness assessment with other compelling events in your company. For example, is your development team in the process of implementing other new processes and solutions? Is it almost time for your budget reset? Are you planning to adopt DevOps development processes? By aligning your test automation search with other changes in the organization, you'll increase your chances of gaining internal buy-in.

Take a crawl, walk, run approach

When you're making a change — especially when it comes to implementing a new solution or adopting a new process — you're making an investment. Reassure your team and others that you'll start small, build value, and grow from there. Gather metrics before, during, and after — you'll need them to continuously show others the value.

Consider all stakeholders

Certainly, your development and QA teams — along with management — are the primary stakeholders in the evaluation process. But if you want to empower other users in your organization to play a role in the testing process, you need to understand their needs, preferences, pain points, and goals. No one likes change forced down their throat; encouraging involvement from a cross section of stakeholders will ensure everyone feels their voice is heard in the process and the decision.

05

Steps to Evaluating New Solutions

After you've built your case for investing in an automated testing solution and have a plan to manage the cultural and process change, here are the 10 steps you need to take to effectively evaluate your options.

1

Consider what testing capabilities you need:

What should the new tool do, and what kind of capabilities should it have? How can test automation help your team navigate the changing world of software development? Involve your team in the conversation — be open to learning new things along the way and shifting your definition of success. Ideally, you should document the capabilities you are looking for. Skip ahead for a checklist template!

2

Understand your stakeholders and their pain points.

Certainly, your developers and QA team may have strong ideas about what you need in an automated testing solution. But they aren't the only ones whose opinions count. Your procurement, legal, and vendor management teams may also have a stake in the solution you choose, so work with your entire organization to identify their requirements. Ask your team early in the evaluation about security requirements, contract signature processes, or the turnaround time on contract reviews. This helps ensure you don't run into any momentum-sinking hiccups once you've made your decision.

3

Evaluate multiple solutions:

If you're going from a manual testing environment to automated testing, any kind of automation solution is going to provide some level of relief. But don't settle for the first test automation solution you see. Instead, keep your process focused and efficient so you can try multiple platforms.

4

Focus initial tests on high-value, critical paths:

When evaluating solutions, you may be tempted to resolve your toughest, most complex problem first — but this isn't your best bet. You're learning something new, so don't set yourself up for a frustrating experience that might introduce bias.

5

Create a scorecard:

Remember those capabilities you documented? Turn those into a scorecard. Maintain consistent standards, so you create a level playing field for each solution. Give each tool a rating based on how well it meets your criteria for desired features and capabilities — your ratings will help you make an objective decision. And hold on to your scorecards — products are always adding functionality, so you may want to refer back to them in the future.

6

Perform a proof of value:

After you have identified a shortlist of solutions to try, begin the proof of value process. This is where you'll use the tool in a hands-on way. As mentioned earlier, approach the proof-of-value process with a clear idea of the results you are looking to achieve.

7

Compare and contrast:

Evaluate solutions head-to-head. How long does it take to achieve the same goal in each product? How do creating tests, running tests, and test output differ? How does it feel to use each product? Is one tool more enjoyable to use than another? What are the unique features that matter? Let the answers to these questions help inform your decision.

8

Develop an implementation plan:

Just as important as evaluating vendors and test automation tools is developing an implementation plan. Your implementation plan will have as much bearing on your success as the tool you choose. Implementation can be tricky. Don’t rip and replace; plan incrementally to allow for learning and experimentation. Start with small, meaningful tests that give you quicker value.

9

Collaborate with your team to share insights and plan strategy:

As you’re evaluating new solutions, remember to collaborate with your team and share learnings with each other. By sharing tips and tricks that you’ve learned while using the platform, you’ll be able to gain tremendous efficiency and ensure everyone is successful.

10

Talk to others:

The testing community can be a valuable resource when evaluating solutions. Take advantage of opportunities like company forums or Slack channels to communicate with other users about their experiences with a specific tool.

06 • Key Features to Look for in a New Solution

- □ **Ease of use:** Look for a UI that makes it easy for everyone, regardless of coding expertise, to contribute to automation. Easy shouldn't mean simple, though. Developers and Quality Engineers should be able to refine complex tests with JavaScript. Balancing simplicity and adaptability will ensure your entire team can effectively use the solution.
- □ **Built-in intelligence:** Machine learning can help auto-heal tests alongside changes in your UI, help your testing team build test plans for new features, or adjust test execution wait times so you can reduce time spent on maintenance.
- □ **Cloud-based SaaS:** There won't be any infrastructure to manage or frameworks to connect. Plus, cloud test executions have infinite scalability and provide more test execution insights.
- □ **Unified solution:** One solution that can perform cross-browser, functional, end-to-end, accessibility, and performance testing gives you a clear picture of all testing activities for easier collaboration and reporting.
- □ **Insights and reporting:** Test data is a valuable resource for continuous improvement across the software development lifecycle. Look for platforms designed to help you quickly understand the state of your application's quality.
- □ **Simulate real user interactions and data:** Software testing provides limited value unless it reflects how your customers actually use your product. Test automation platforms that integrate with customer data platforms like Segment will ensure that your testing activities are aligned with high-value user journeys.
- □ **Integrates with current tech stack:** Your test automation solution should integrate with your existing tech stack, which might include source control, continuous integration, issue tracking, and collaboration solutions. These integrations can help dramatically increase product velocity, mean time to resolution, and help the team collaborate effectively on quality.
- □ **CLI (or developer-focused capabilities):** Command line tools can make it significantly easier to integrate automated testing into your CI/CD pipelines, especially for developers. Identifying testing solutions that support integrated workflows is crucial for making test automation scalable and sustainable.
- □ **Ability to test business-critical apps:** Third-party applications such as Salesforce or Workday are playing a growing role in every organization's tech stack. But many test automation solutions struggle to maintain relevant tests as these applications are updated without notice. Look for solutions that can test business-critical applications while maintaining testing velocity.

Start Your Test Automation Evaluation

Are you ready to evaluate test automation solutions? Mabl is here to help. Our team of Solutions Engineers help teams quantify the impact of test automation and evaluate if mabl can support your quality goals.

Learn how mabl can help your team easily create end-to-end tests that improve your software's quality without slowing you down. Companies like JetBlue, Charles Schwab, and Stack Overflow rely on mabl to increase test coverage, accelerate product velocity, and optimize application quality. Contact us today to **request a personalized demo** or **start a free trial**.

BONUS • Test Automation ROI Calculator

This ROI calculator helps quality teams quantify their hard work in terms of dollars. The goal is to show your management team time and dollars saved by switching test automation solutions. This data is never meant to reduce headcount or salaries; it is intended to highlight the time and money that could now be spent on performing higher-value tasks and contributing to test coverage. Leverage the editable fields below, or adapt this calculator for the metrics most important to your team.

Average testing stats per month:	# tests created		# hours to create each test		# hours spent on test maintenance		Total hours spent		Hourly pay per QE		Total cost per month
Current Solution	5	×	3	+	6	=	45	×	$35	=	$1575
Solution #1	0	×	0	+	0	=	0	×	$0	=	$0
Solution #2	0	×	0	+	0	=	0	×	$0	=	$0
Solution #3	0	×	0	+	0	=	0	×	$0	=	$0