

THE BUYER'S GUIDE

to Web & Mobile Test Automation Tools



Getting Started With (Or Replacing) Test Automation Tools

If you're in the market for a test automation solution, you're in the right place.

In DevOps today, manual testing doesn't cut it. It's slow, tedious, and creates bottlenecks. Test automation is the only way to keep up with the pace of DevOps. And it's a key enabler for continuous testing, Agile, and DevOps. Even though test automation is currently the preferred testing method, organizations still face barriers when trying to implement it.

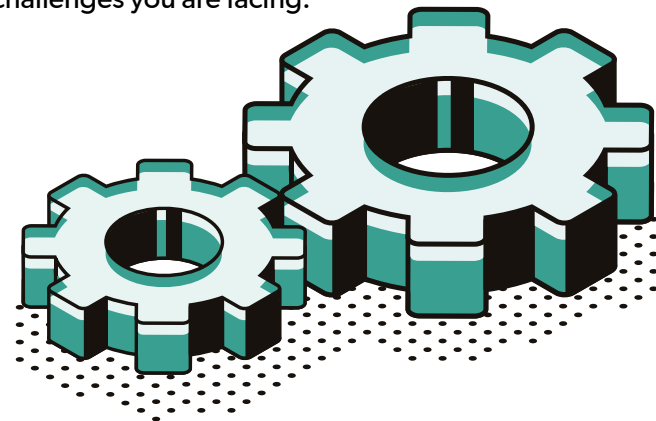
The reality of today's testing is that there are fewer silos between teams. Quality is now considered the responsibility of everyone — no matter their skillset or objectives. Successful test automation can only be achieved with the help of the right tools.

Because test automation executes tests faster and more often, teams must also have tools that manage the significant volume of data that is created. Most testing

is executed through continuous integration (CI), which is black box. That means that teams can only achieve visibility through test reports. And unfortunately many open source test automation frameworks fall short in analytics.

Testing tools and technology can help alleviate automation challenges you are facing. The testing tool market is varied and fragmented with a lot of players to consider. Some are just test creation vendors. Some only do reporting. Others are just a lab.

Keep reading to learn more about the test automation landscape and understand how to overcome specific challenges you are facing.



Contents

Key Trends in Test Automation	4
Benefits of Test Automation	11
Top Challenges in Test Automation	13
Things to Consider in an Automation Testing Tool	18
Meeting the Needs of Different Personas	24
Now What? Narrowing Down Your List	25
About Perfecto	27
Test Automation Tool Checklist	28

Key Trends in Test Automation

Take a look at what's on the horizon in software testing and DevOps. These trends are important to understand and consider when evaluating tools that need to be a long-term, effective solution.

DESIRE TO BE CI/CD READY

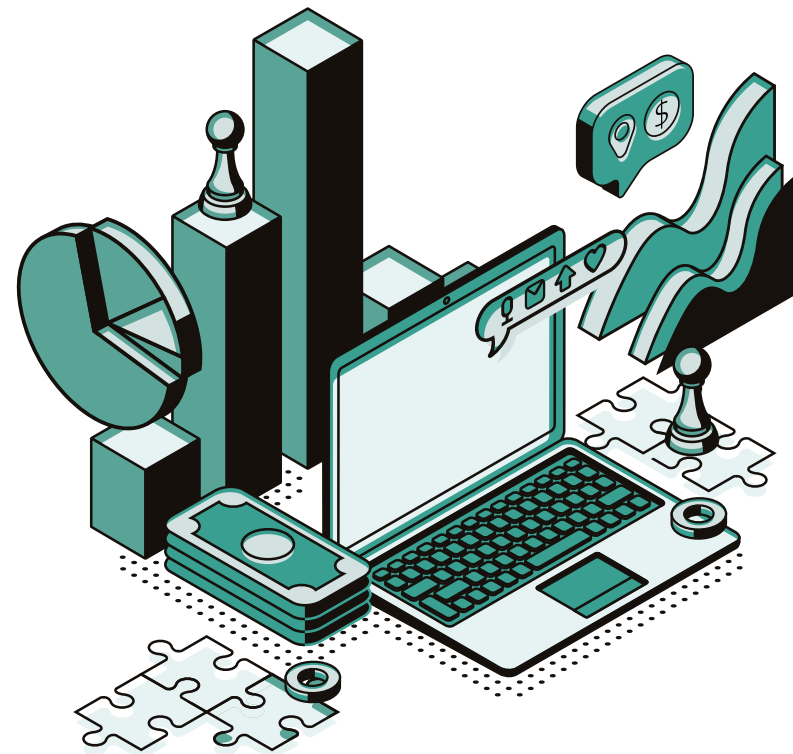
One key trend in test automation is the growth of teams seeking to establish a CI/CD pipeline.

CI provides DevOps teams with consistency. And it gives developers the chance to check in and validate their code with each change. CD ensures that these code changes get pushed through to their proper infrastructure environments. A CI/CD pipeline allows teams to execute and analyze in minutes, understanding where the problems are in the code.

What's more, the CI/CD pipeline is a best practice in DevOps, and it's something more and more teams are striving toward.

What to Look for in a Tool

A visual dashboard of the CI pipeline that can show build acceptance testing trends, health, and overall length. It should also include links directly to the test reports for further analysis and improvements.



MARKET BECOMING MORE FRAGMENTED AND COMPLEX

The mobile and web app market continues to grow more and more complex. Most notably shaking up the market in 2023 are the recent releases of iOS 16 and Android 13. Both releases have included UI and UX enhancements, productivity upgrades, and boosts to privacy and security.

With the new iOS 16 and iPadOS 16 releases, users can get richer accessibility support through live captions in FaceTime calls, conversations, audio, and videos. In addition, the new OS version will automatically detect doors and red signs and instruct the user on how to unlock them. The upcoming release will also come with other accessibility enhancements like voice control call hang-ups and more.

The new Android update — which will be supported on most of the Pixel devices like the new Pixel 6A Smartphone and other Android devices — will see Google include some nice UI changes like themed app icons for more end-user

customized design of the home pages. Google has also included changes like multilingual support and the accessibility-friendly Audio Description functionality to its platform.

The new iOS 16 and Android 13 releases are not only impacting apps and browser functionality but also redefining the mobile application market from a test coverage perspective. For this latest release, Apple terminated support for nine older and legacy devices. This continuous development process highlights the importance of assessing the quality changes that version updates have on apps.

These changes will heavily impact browser and device coverage, and will require changing the overall platform support considerations. Teams will need to have a focused iOS 16 and Android 13 device matrix along with some legacy platforms like iOS 15 (and possibly even older depending on their client's device usage and analytics), and of course a relevant lab the most recent iOS and Android versions.



What to Look for in a Tool

An online lab that keeps up with market dynamics for both web and mobile. It also must support the latest platforms and OS versions. This enables teams to keep testing against the most current market devices and desktop browser permutations.

NEED TO AUTOMATE MORE TEST CASES

The features and capabilities available to today's mobile application developers are seemingly endless. From biometric authentication to scanning bar codes on your own, the application development experience is growing vaster to keep up with the needs of today's consumer.

As application development capabilities keep growing, the needs for more comprehensive application testing are growing as well.

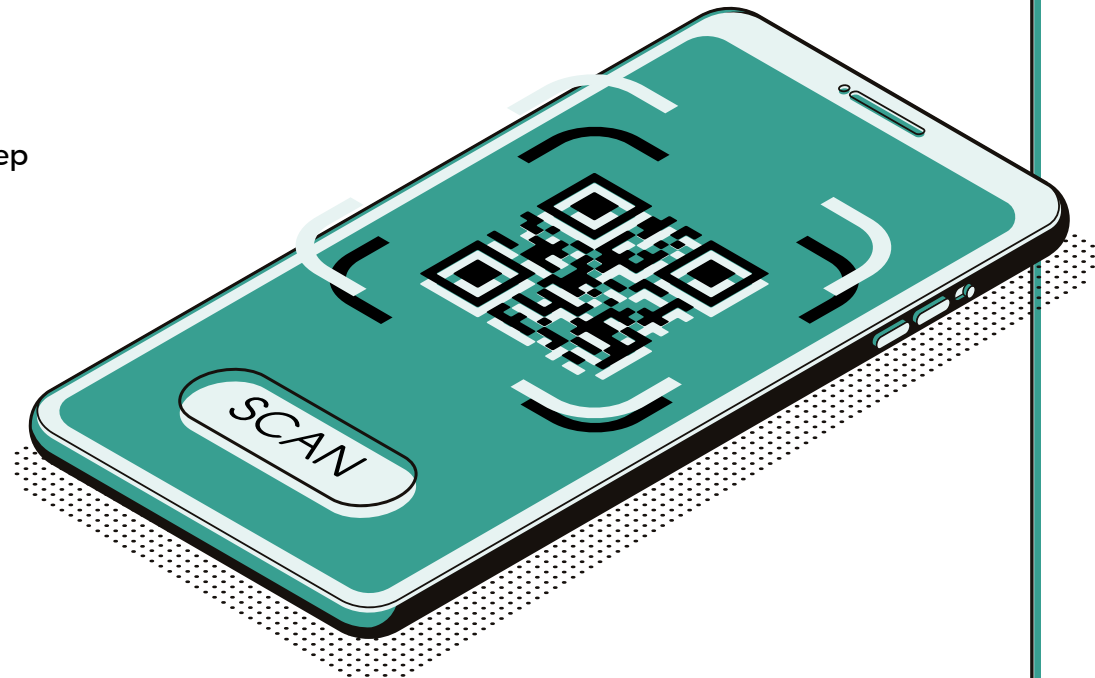
While testing features like voice recognition and multi-factor authentication manually is always an

option, teams cannot rely on manual testing as a long-term, scalable solution. Testers need to test these capabilities quickly, efficiently, and with the rest of their application.



What to Look for in a Tool

A thorough platform that supports advanced test automation capabilities, from biometric authentication and two-factor authentication to geolocation testing and image injection.



EMERGING AUTOMATION FRAMEWORKS

Automation frameworks like Flutter, Appium, and Playwright have either grown in popularity or released major version updates in the last year. The emergence of these frameworks has made a major impact on the app testing market.

While Flutter and Appium have been around for a while, their most recent releases are making a splash in the world of mobile app development. Flutter's latest 3.0 release offers users a wealth of new support capabilities (including foldable phones), enhanced app experiences, more control over app initialization, and upgrades to code review tools. Appium's 2.0 release allows test engineers to cover not only mobile applications, but also desktop apps, Flutter apps, and more.

Playwright is a cross-browser testing framework that has piqued the interest of testing teams in recent months, rivaling other popular frameworks like Selenium and Cypress. Playwright includes many features that make it a great choice for

developers and testers seeking a speedy and seamless testing framework aligned with the modern web, such as native input for mouse and keyboard, auto-wait, web-first assertions, and tracing.

It is essential for teams to familiarize themselves with the ins and outs of each framework to choose which one will be the right fit for them when building their apps. Every framework has its strengths and weaknesses, and different frameworks will ultimately appeal to different teams based on their business needs.



What to Look for in a Tool

Unified access to both mobile and web along with support for desktop web and mobile app test automation frameworks (Selenium, Appium).

LOW CODE/NO CODE FOR DEV AND TESTING

Another trend in the test automation space is the continued growth in popularity of codeless or low-code testing tools.

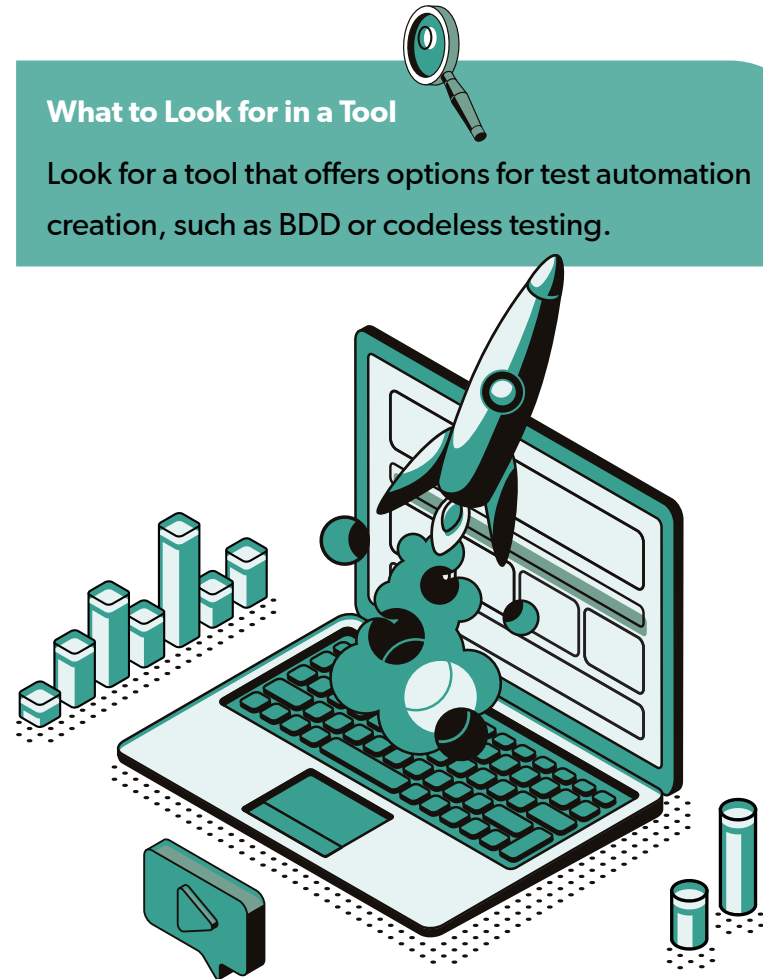
These tools are ideal for members of the DevOps team without hard, technical coding skills. A business tester without coding skills will benefit greatly from a codeless testing tool, which would allow them to partake in the test automation creation activities they previously have not participated in.

But codeless testing and low-code solutions are not just for team members without coding skills. They are also gaining traction with developers and test automation engineers. While these two personas have coding skills, no-code/low-code solutions help speed up the process.

Codeless and low-code solutions are faster and less error prone. That means these team members get to spend less time creating and maintaining tests and more time on other critical DevOps activities.

What to Look for in a Tool

Look for a tool that offers options for test automation creation, such as BDD or codeless testing.



SPECIFIC SOLUTIONS BASED ON AI/ML

More and more segments and domains in the digital markets are starting to invest and leverage artificial intelligence (AI) and machine learning (ML) capabilities.

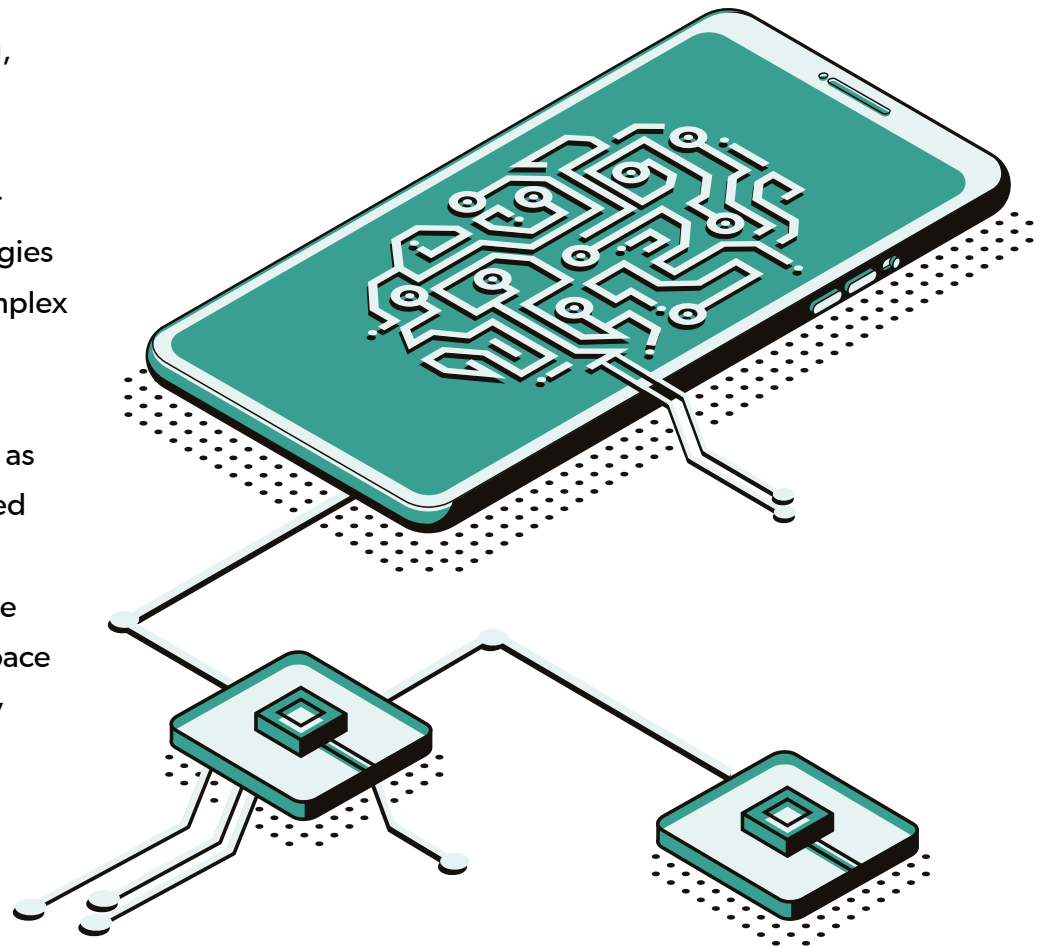
With the growing complexity of developing, testing, and navigating big data, combined with the need for faster response to market events, organizations are looking at smarter engines that use AI and ML. These technologies can help optimize some of the time and complex tasks that are slowing them down.

We've seen both big data analysis — as well as codeless test automation — being introduced recently using AI and ML capabilities. In the future, we will continue seeing more of these technologies evolving to help address the pace of innovation and the complexities that they bring to enterprises.

What to Look for in a Tool



Find a test automation tool that offers smart insights through AI or machine learning, especially in test creation and test reporting.



GROWTH OF CLOUD ADOPTION

Another growing trend in the test automation space is cloud adoption. Cloud adoption is becoming more of a common practice at every DevOps stage. However, cloud is still a commodity.

Cloud computing benefits are numerous. They're easy to scale. Their robust infrastructure and networking boosts speed. They're best positioned to support continuous testing and fast feedback for developers upon each code change. And they make it easier for geographically-dispersed teams to collaborate.

Plus, cloud computing boasts strong security that meets the strict requirements of global enterprises, especially when compared to an on-prem solution.



What to Look for in a Tool

Find a cloud-based solution with a unified set of capabilities all within one end-to-end tool.

DEVSECOPS/DEVQUALOPS

It's no secret that there are major security challenges in the DevOps process. On top of security concerns, organizations are still struggling to instill quality inside the cycle. One reason why these pose such a challenge is that security and quality are not fully automated within the DevOps activities. For that reason, we're seeing the rise of **DevSecOps** and **DevQualOps**.

DevSecOps is the mindset that everyone is responsible for security, which is achieved without impeding speed and scale. Rather than security being examined and tested at the end of the cycle, security is incorporated at each stage of DevOps.

And, now with DevQualOps, testing practices are evolving to ensure high quality software releases. DevQualOps ensures quality across the DevOps lifecycle, testing, environment delivery, and reporting. This is only expected to grow in the coming years.

Benefits of Test Automation

If you aren't already convinced that you need a proper test automation tool or if you are looking for some help selling it internally, here are a few of the major benefits of adopting test automation tools.

1 REDUCED TESTING FATIGUE

Test automation limits the potential for human error — especially for those mind-numbing tests that have to be repeated over and over again.

2 FASTER BUG DISCOVERY

Test automation gives developers the chance to see problems while code is being written — not just at the end. This helps eliminate duplicate code. It also keeps code simpler, cleaner, and easier to scale.

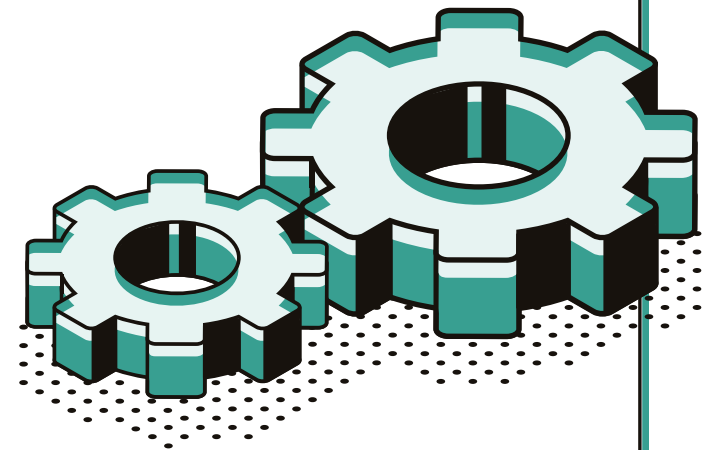
In addition, test automation incorporates testing earlier into the build cycle and development phases to uncover defects. Teams benefit from fast feedback and quicker MTTD and MTTR.

3 IMPROVED RISK MITIGATION

If an automated test fails, code can be fixed immediately — well before it makes its way to production. This not only keeps costs down, but it helps DevOps managers make better decisions before releasing software updates.

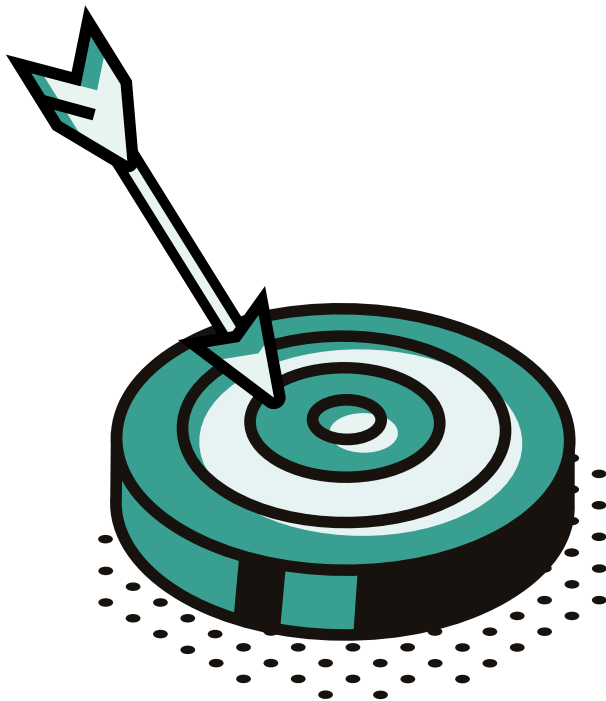
4 MORE EFFICIENT SOFTWARE AND APP TESTING

Test automation allows testing on multiple platforms in parallel. This is especially beneficial for batch testing at night or other types of time-intensive, high-volume testing.



5 BETTER MANUAL TESTING

Test automation gives real-live humans more time to test parts of software that machines can't. Highly-targeted manually testing brings critical, human intuition to the table. Manual testing is done only where it's needed — such as ad hoc/exploratory testing, visual and UI testing, and reviewing test results.



6 BETTER BUSINESS OUTCOMES

Test automation can help your team achieve better business outcomes. Test automation expedites the identification of business risks, helps provide data-driven decisions around the continuous delivery of value to customers, and more.

Top Challenges in Test Automation

Test automation has become a top priority for digital enterprises as part of their DevOps journey. However, there are challenges and roadblocks with achieving those goals.

As you examine and consider test automation tools that are right for your team, make sure that each tool or vendor has a good solution to help you avoid these common pitfalls.

THE LACK OF NECESSARY SKILLS

Test automation is an art. But, not everyone is an artist. Although test automation offers teams a number of benefits, not everyone has the right skills to do it successfully. Because of this, manual testing still occurs as teams compensate for any existing automation gaps. Or, they see manual testing as a stable alternative.

However, this results in delays and a disconnect with the R&D team, who is usually two or three steps ahead.

In order to match the R&D team's pace, testers need to complete their cycles in four to six hours.

Unfortunately, manual testing cannot meet the development pace, which forces it to be pushed to the end of the cycle. As a result, teams need to add a manual regression — also called “stabilization phase” — at the end of the sprint. This extends the release cadence, rather than reducing it.

Overcoming This Challenge

Creating test automation requires technical skills with fluency in development languages such as Java, Python, JavaScript, Objective-C, and more. In addition, it requires creativity to consider and develop sufficient test scenarios so that there are no critical escaped defects in production.

Additional skills that are required include setting up technical environments such as test labs, advanced reports, virtual machines, databases, and more.

2 TEAMS DON'T STABILIZE TESTS BEFORE AUTOMATING

For teams who are just getting started with automation, the process is similar to riding a bike. Before you take off, you need balance.

We often see teams building all their tests and running them together without stabilizing them first. The result is often failure, frustration, and trust issues with automation. Because of this, teams revert back to manual testing.



Overcoming This Challenge

This challenge is less of a technology fix and more about partnering with the right vendor — one that can help define the right test automation strategies, goals, and pace. And one that is committed to seeing you succeed.

Partner with a vendor to utilize best practices for building a test automation foundation, like this modular execution approach:

- Build your test locally over a real device/browser.
- Test daily to ensure a balanced execution and that there are no blockers.
- Tie your execution to CI and run your test constantly. Or, at least every night/
- Increase your digital platform coverage and execute more frequently.
- Maximize your automation coverage up to 90% to get into the DevOps-friendly zone.

3 LAB MANAGEMENT IS BURDENSOME

There are still teams that prefer to build and maintain their own labs. While this isn't necessarily a bad thing, in-house labs are harder to manage — and they're expensive.

Labs can quickly become out of date with new operating systems, devices, and browser versions consistently being released. As a result, teams can spend a lot of time maintaining and running their lab as opposed to testing.

A QA leader at a large US-based bank recently made this point about the issue:

“We are in the business of creating and maintaining the bank’s testing activity. We are not in the business of managing IT resources. This tradeoff should always guide us, and we should keep the balance in favor of testing at all time.”



Overcoming This Issue

Having an always-on, always-up-to-date, cloud-based lab is critical for continuous testing. Many vendors only offer test automation or test creation without a lab solution. However, this won't be enough for your testing needs. Selecting a vendor that offers lab management for both browsers and devices is key.



4 TEST AUTOMATION BECOMES DYNAMIC

Different from the Waterfall methodology, the DevOps process requires unique testing capacity at each and every step of the software development lifecycle (SDLC). The execution layer should be geared towards automation at scale, which allows teams to scale up or down at different points of time.

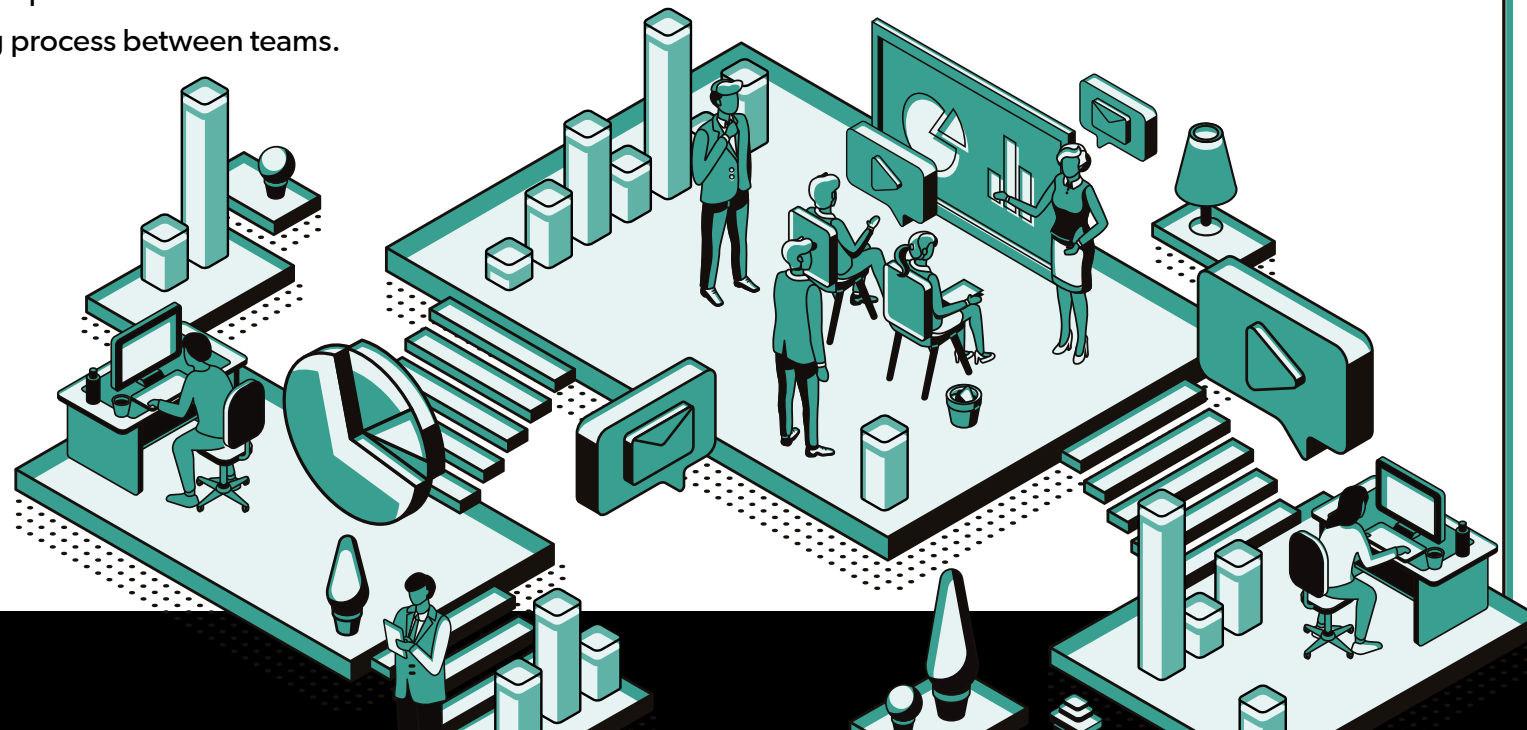
In order to achieve coverage metrics, teams should be able to run their tests in a parallel mode and define priorities for executions. They should also be able to queue and better orchestrate their overall testing process between teams.

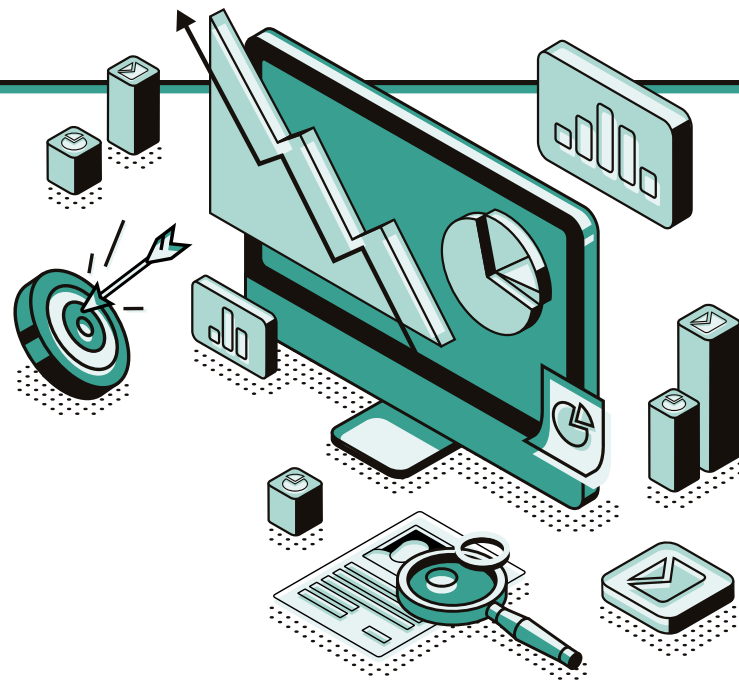


Overcoming This Issue

Automating at scale requires the right combination of people, skills, and tools.

The best and most successful approach is to start small and then scale. However, many teams do not have the right culture or process. Teams need to build a robust foundation that can scale, be re-used, and will be reliable over time.





5 SORTING THROUGH ALL THE DATA

It's not enough to execute your tests quickly with automation. You also need visibility into test results. And you need to analyze the results just as quickly. When DevOps automates at scale, one of the results is a huge influx of data, which needs to be reviewed and analyzed.

Teams often find they are swimming in a sea of data made up of log files, artifacts, and test results. A challenge for teams is to identify the beneficial information buried in the data. Effective triaging and sorting through the noise is key for achieving fast feedback.

Overcoming This Issue



Making sense of all the data is a big undertaking for many teams, especially those that are not equipped with the right tools.

Find a test automation tool that goes beyond basic reporting. Teams need machine learning capabilities, heatmaps, detailed test reports, root cause analysis, dashboards, cross-platform comparison for visibility, HAR files for network traffic analysis, and more.

Things to Consider in an Automated Testing Tool

EMULATORS/SIMULATORS VS. REAL DEVICES

When reviewing test automation tools, you'll want to take note of the types of devices they have — or if they test on emulators and simulators.

All three types of testing have their own benefits and drawbacks. Emulators and simulators make testing apps on multiple mobile devices easier. But, keep in mind that emulators and simulators do not offer the test coverage of real devices. For that reason, real device testing is the most accurate option.

A combination of both virtual and real devices can provide teams with the optimal testing coverage. In the early sprint phases, when the features are still being shaped, it makes a lot of sense to run smoke tests, unit tests, regression tests, and other fast validations against emulators and simulators from the developer environment.

Later in the build process, real devices can be used for things such as for UI, UX, and battery functions.

When the coverage requirements and the quality insights are greater, launching the full testing scope in parallel against real devices is the most effective process.

VENDOR EXPERTISE

When you're examining potential test automation tools, you should also examine the vendors themselves. When you invest in a solution, it is important that the vendor invests in your success as well.

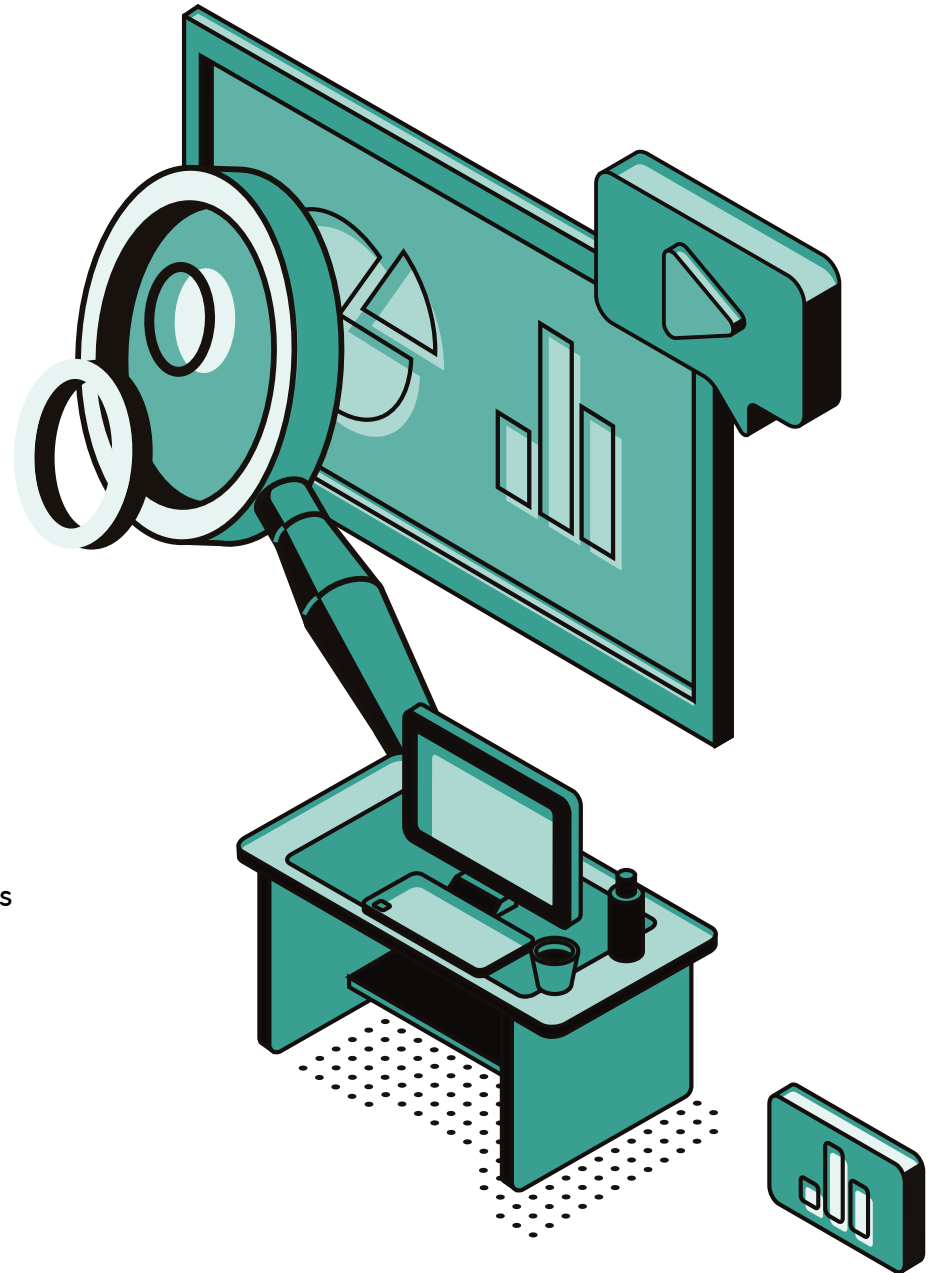
You'll want to ask about how they can get you up and running quickly. And, you should ask about their implementation plan. Consider their expertise in the field — how much guidance and best practices can they offer for your testing strategy? Investing in a partner that is committed to making you successful rather than just selling you software is critical.

TASK MANAGEMENT

Automation testing tools should do more than just run tests automatically — they should make tasks easier to manage as well. Ask yourself what detailed feedback you really want in a tool. What reporting and insights are crucial for you and your team?

For example, if you want faster feedback in your tool, look for one that supports test-driven development processes, or TDD. It can provide faster feedback which allows teams to find the bugs immediately as they develop features.

In addition, your test automation tool should give you a centralized place for analyzing test results and individual reports. It should also track progress across all types of test cases — including manual, exploratory, and automated tests. Tasks and results from test cases should also be easy to share among team members and categorizable by feature, component, and sprint.



HEADLESS BROWSERS VS. REAL BROWSERS

Another feature to consider in an automated testing tool is the type of browsers available to you. Does the tool offer real browsers for testing, or does it offer headless browsers? Or does it offer both?

As with emulators and real devices, a mix of headless browsers and real browsers is recommended for optimal test coverage. Headless browsers are gaining traction in the automated web app testing space because they provide fast feedback and are easy to set up. What's more, headless browsers play an important role in unit testing, providing fast feedback without the need to set up a lab or a grid, while real browsers can provide the best UI testing and test coverage. Overall, headless browsers complement the overall testing objectives for teams.

INTEGRATION AVAILABILITY

Choosing a testing tool with the flexibility to integrate with your existing workflows will make the transition to automation much easier.

If you've already got a solid task management or CI/CD platform, determine whether it can be integrated into an automation testing tool you're interested in. Developing, organizing, and reviewing tests from within your existing CI server could really speed up your regression and Agile cycles.

The same applies to any other fan-favorite collaboration or case management platform that you want to continue using. Make sure you thoroughly check the list of integrations available so you can continue using tools you've already come to love and trust.

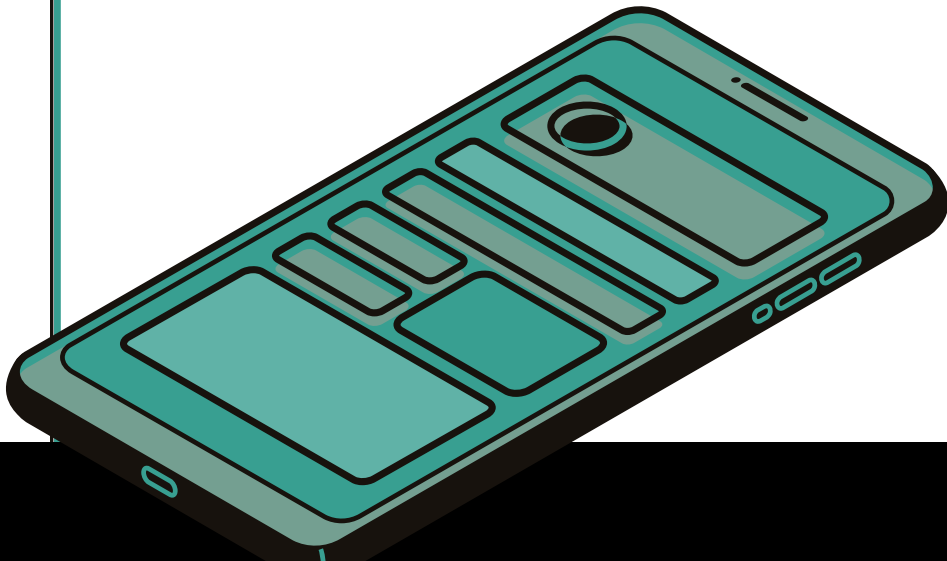
TESTING ENVIRONMENTS

Be specific about the number and types of technology combinations you'll need — including the specific hardware/OS/browser combinations that are vital for your testing.

Check whether the automation testing tool offers support for both desktop and **mobile application testing**. Then ask yourself these questions:

- Do you have any need for testing local, onsite devices?
- Do test environments also support manual testing, and how easy is it to configure them according to different projects, releases, & sprints?

Answering these questions will help you determine what you need in your test automation tool.



CLOUD VS. IN-HOUSE

Another feature to consider in a test automation tool is whether it's cloud based or in-house.

On-premise technology can be a headache for both users of the tool and the IT department. The teams must ensure that the solutions are properly installed and up to date on each type of machine. On-premise tools are also more difficult and costly to scale as you grow. And, if your team is spread across the globe, your speed will more than likely be compromised.

On the other hand, cloud platforms allow users and enterprises to meet a number of robust requirements across scalability, speed, collaboration, quality, and security — freeing up team time to focus on product innovation.

Cloud-based tools are faster for geographically-dispersed teams and foster more collaboration. They're much easier to scale, with little to no set up. And their level of security is much more advanced than on-premise solutions.

COMMERCIAL VS. OPEN SOURCE

Another consideration to make pertains to whether the tool is open source or commercial.

Open source automated testing tools are free, but they often lack the support that growing enterprises need. For teams facing the increasingly fragmented testing space, open source testing tools don't cut it for test coverage. What's more, the time and effort put into maintaining test environments and fixing issues cuts into any potential savings of a free tool.

There is no single framework — commercial or open source — that matches the testing skills and objectives of an entire feature team. Teams must mix and match tools and frameworks through due diligence and periodic assessment in case there are newer and better solutions available.

REPORTING CAPABILITIES

If the tool you select doesn't offer robust reporting and analytics capabilities, you'll have to do it yourself. This will cause you to spend valuable time

and resources analyzing test results. Doing this yourself is time consuming, and it means you'll most likely be working with outdated data. Remember, time is money.

You'll want to find a tool that offers robust, built-in reporting capabilities — ones that go beyond basic pass/fail reporting. Look for features such as CI dashboards, detailed test reports, cross-platform visual validation, heatmaps, root cause analysis, and machine-learning powered smart insights. These robust reporting capabilities will allow you to identify bugs earlier in the process and fix them faster.

BDD VS. TRADITIONAL AGILE TESTING FRAMEWORKS

Another consideration pertains to BDD, or behavior driven development. This option enables you to create test scripts in easy-to-use English phrased steps in a simple-to-use environment.

If members of your team — such as business testers — lack coding skills, BDD can allow them to participate in the test creation process, as opposed to a traditional Agile testing framework.

BUDGET

Price is an important factor in choosing the right testing platform. It is important to understand the capabilities and features between the vendors you are evaluating. Some components are easy to compare. Others contribute differently to the total cost of ownership. When comparing different tools, don't lose sight of the advanced features that drive considerable value and cost savings as you scale your test automation.

For example, lab costs should be comparable across most vendors, but make sure you understand what you are paying for. Pricing will vary depending on the following:

- Mix of real devices or emulators and simulators
- Private, dedicated devices vs. shared, public, devices
- Data center coverage and location.
- Concurrency/parallel testing.
- Reporting and analytics capabilities.
- Skillset-matched test automation creation options.
- Implementation and onboarding services.

All of these will affect the price of the tool. Determining what you want ahead of time will help you get a more accurate quote. Most companies are more than happy to help you determine the right testing package for your team.

SUPPORT AND RESOURCES

Take a look at your current testing resources, and then determine how an automation tool can assist you and your team. Think about what type of support you want when you run into problems.

Is the help desk open 24/7? Determine whether tickets are created via phone call, mobile apps, live chat, or email. Having a real-life human available to help can boost morale and keep testing running smoothly. And it can accelerate your release delivery.

Check the company website and see if there are helpful, easy-to-navigate resources for handling issues independently. If so, it's a good sign they take support seriously.

Meeting the Needs of Your Entire Team

As you review and examine test automation tools, consider their impact on all members of your team. An ideal test automation tool should support all of the personas with an DevOps team so that everyone can contribute, collaborate, and supplement each other's skills. Learn more about the needs of DevOps team members:

1 DEVELOPERS

Developers need fast feedback upon each code change. When they check in the code, run a build, and make sure they didn't break anything, they need as much feedback as possible. Developers need tools with the ability to implement code quickly, integrate into their IDE, and integrate into the platforms they work on, including browsers, platforms, and tablets. More and more developers are also seeking low-code and no-code options. Writing code is time consuming, and these alternatives can save developers time.

2 AUTOMATION ENGINEER

Test automation engineers, or SDETs, need frameworks and tools that support automated testing of end-to-end scenarios. Like developers, test automation engineers also operate within a single CI/CD pipeline. Test automation engineers need frameworks that can provide test automation scenario coverage and examine the end user experience, or user journey. They need to be able to test features that are accessible to the end user — including Siri, image injection, Bluetooth, and GPS are also seeking low-code and no-code options. Writing code is time consuming, and these alternatives can save developers time.

3 BUSINESS TESTER

Business testers perform manual and exploratory testing. Because this can be time consuming and slow moving, they often provide feedback in later stages. Business testers do not have the coding skills that developers and test automation engineers have. Because of this, business testers benefit from tools that cater to all skill levels. These include BDD testing with tools like Cucumber, or codeless testing for easy and reliable test creation that doesn't rely on coding.

Now What? Narrowing Down Your List

Once you have gathered a list of test automation tools, you'll need to narrow down your choices and get internal buy in.

HOW TO NARROW DOWN YOUR CHOICES

After finishing your research of tools on the market, you'll likely have a number that you're interested in. In order to narrow down your choices, compare the key features that matter most to your team — whether that's quick access to testing platforms or the ability to plug and play.

You'll also want to see how the tool can fit in with your current toolchain. Remember, the tool should be an enabler for success. Changing your current processes to fit the tool will only create more problems. The tools need to fit your current process and solve specific challenges your team faces.

Often when you integrate a tool into your existing process, you have a few elements that are impacted.

In order to narrow down your selections, see how each tool works with the following:

- **Networking** — How does the tool integrate with your backend infrastructure? If you need to implement something, find a better option.
- **Security** — How well can you secure the test environment you have in house, the make up behind it, the platforms themselves, and the VMs? Can they be cleaned up? Who has access to it?
- **Scalability** — How can this tool help you scale up? How will it impact test coverage? How can you include support for legacy platforms?

To truly get a sense of how a tool will impact each of these components, sign up for a free trial of your top picks. Sometimes free trials offer limited functionality or a short timeframe to evaluate the tool. Alternatively, you may want to work with each vendor to set up a proof of concept (POC). This will give you a chance to try your hand at testing with the tool and see how it works with the rest of your toolchain.

HOW TO GET INTERNAL BUY-IN

After you have narrowed down your choice, your final step is to gain internal buy-in for your chosen test automation tool.

In order to do that, you'll need to make a case for the tool. Paint a picture of your current state, detailing the team's pain points and challenges, as well as their negative business implications.

Then, provide clear objectives on goals you want to achieve. Such KPIs to measure success may include:

- How many builds you are able to run.
- How many builds are successful.
- The level of automation you can achieve.

Mapping out just what you want to achieve with your test automation tool of choice will help leadership see your vision and support the adoption the tool.

Current State vs. Desired State	
Current State	Desired State
Negative Consequences	Positive Business Outcomes

About Perfecto

Perfecto powers exceptional digital experiences by combining the power of flexible test authoring, cross-platform execution, and AI-driven analytics into one secure, cloud-based web and mobile testing platform.

Visit www.perfecto.io for a free trial, demo, or to learn more.

RELATED RESOURCES

- [The 2023 State of Test Automation](#)
- [What Is Good Testing in 2022?](#)
- [ROI of Quality: Making a Business Case for Modern Testing](#)



Test Automation Tool Checklist

Use this checklist to help compare vendors in your search for a test automation tool. Score each 1-5.

VENDOR ASSESSMENT	VENDOR 1	VENDOR 2	VENDOR 3
Demonstrated customer success with active deployments.			
Ease of support with global reach.			
DevOps and test automation product portfolio.			
Stable financial position.			
Open source framework contribution.			
Enterprise grade SLA availability.			
Provide support & professional services based on teams experience.			
Organizational quality certifications and process control.			
TEST CREATION CAPABILITIES	VENDOR 1	VENDOR 2	VENDOR 3
Open source framework support.			
Leading scripting language support.			
BDD test automation.			
Codeless testing.			
Automation coverage.			
Object spy capabilities and inspection support.			

TEST EXECUTION CAPABILITIES	VENDOR 1	VENDOR 2	VENDOR 3
Burst mode capabilities/elastic scaling.			
Execution speed.			
Global location coverage.			
Real mobile device coverage.			
Simulator/emulator coverage.			
Browser support.			
Manual test executions.			
Automated test executions.			
Advanced automation (screen capture, image injection, barcode scanning, etc.).			
Parallel test execution.			

LAB	VENDOR 1	VENDOR 2	VENDOR 3
Advanced environment control.			
Unified lab for mobile & web testing.			
Datacenter infrastructure certifications & global infrastructure support.			
Public cloud support.			
Private/hybrid cloud support.			
Latest device availability, support, & deployment.			
Web browser support for Mac OS & Windows covering multiple browser types.			
Smart lab with advanced AI & self-healing capabilities, ensuring lab stability.			
Device access for advanced capabilities like (SSL, Proxy, and more).			
Secure network connection to lab infrastructure.			

TEST REPORTING AND ANALYTICS	VENDOR 1	VENDOR 2	VENDOR 3
Root cause analysis capabilities.			
Continuous integration support & reporting dashboard.			
Heatmap reporting capabilities for root cause analysis and inspection.			
Rich test artifacts			
Machine learning.			
REST API support for reporting engine.			
Device and web session streaming capabilities.			
INTEGRATIONS	VENDOR 1	VENDOR 2	VENDOR 3
Documented REST API.			
Jira integration.			
ALM tool Integration.			
CI/CD pipeline integration coverage.			
IDE integrations (Intellij, Eclipse, Visual Studio, Katalon, and others).			