



DZONE'S 2019 GUIDE TO

Automated Testing

FACILITATING CONTINUOUS DELIVERY



RESEARCH PARTNER SPOTLIGHT



Key Research Findings

BY JORDAN BAKER PUBLICATIONS ASSOCIATE, DEVADA

Demographics

For this year's Automated Testing Survey, we polled 498 IT professionals, with a survey completion rating of 73%. Below is some of the basic demographic information based on the responses of these survey takers.

- On average, respondents have 17 years of experience in the field.
- Respondents work for organizations based in two main geographic regions — the USA (41%) and Europe (32%) — while they live in three main geographic regions — Europe (34%), the USA (27%), and South Central Asia (14%).
- Respondents tend to work for enterprise-level organizations:
 - 23% work for orgs sized 1,000-9,999
 - 22% work for organizations sized 10,000+
 - 20% work for organization sized 100-499
 - 60% of respondents work on teams of 10 people or fewer.
- Respondents are engaged in developing three main types of software:
 - 82% develop web applications
 - 46% are working on enterprise business applications
 - 25% develop native mobile apps
- Survey takers tend to perform one of three main roles:
 - 34% are developers/engineers
 - 20% are developer team leads
 - 16% work as architects

- Three main programming language ecosystems are used by respondents' organizations:
 - 78% use Java
 - 69% use client-side JavaScript
 - 40% use Python
- The ratio of respondents who work at an org with a dedicated DevOps team is 50:50.

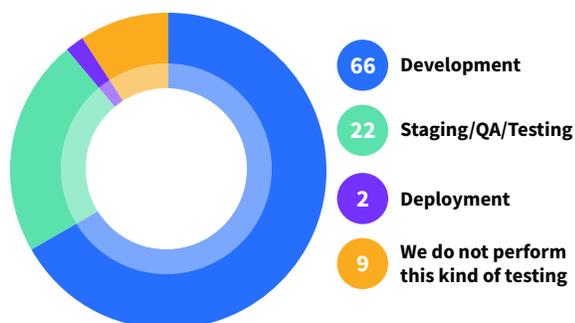
Automated and Manual Testing: A Continued Shift Left

Even though the shift-left mentality and methodology has been around for several years, it continues to gain momentum. In 2018, when we asked respondents at what stage in the SDLC do they typically begin automated testing, 54% reported development and 31% said staging/QA/testing. In this year's survey, 66% of respondents told us that they begin automated testing on their software during development, and 22% begin in staging/QA/testing. Interestingly, while automated testing continues its rather seismic shift to the left, manual testing methodologies saw little to no change over last year's survey. In 2018, 51% reported beginning manual testing in development, while 40% reported beginning in staging/QA/testing. This year, 49% reported performing manual testing in development and 44% claimed to do so in staging/QA/testing. But, again, neither of these year-over-year changes are statistically significant enough to signify a large shift in manual testing's place in the SDLC.

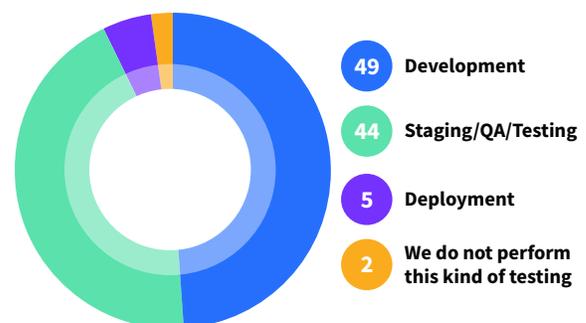
While the above highlights the usage patterns of those developers who have adopted either automated or manual testing, we also gathered data around why certain respondents' organizations opted against one or the other of these testing strategies. Among those survey takers whose organizations/teams do not perform automated testing, the most common reason for doing so, reported by 28%, is that it is not considered a critical priority. In 2018, 33% reported not considering automated testing a critical priority. This matches up nicely with the continued adoption and leftward shifting of automated testing discussed above. When it comes to those whose organizations do not perform manual

SURVEY RESPONSES

At which stage does your organization usually begin automatically testing software?



At which stage does your organization usually begin manually testing software?



testing, one statistic jumps of the page. In 2018, 7% of respondents reported the main reason for not adopting manual testing was the lack of a QA/testing team. In 2019, 43% told us they had no dedicated QA/testing team. This seems a troubling stat. One possible answer to this large year-over-year change is that, as testing is continuously shifted left into the development phase of the SDLC, developers are becoming de facto testers on top of their roles of software engineers.

Now that we've covered where automated and manual tests occur in the SDLC at a high-level, let's drill down a bit to examine the types of tests that are automated and performed manually. Among those respondents who perform automated testing, four main tests were reported: integration, component, performance, and story-level. Comparing the data we received around these tests to our historical data from 2018, we find that the percentage of those using automated integration tests stayed stable (77% in 2018 vs. 78% in 2019), while automated story-level tests (36% in 2018 vs. 46% in 2019) and component tests (52% in 2018 vs. 59% in 2019) both grew in popularity. Automated performance tests, however, fell slightly year-over-year, dropping from a 51% adoption rate in 2018 to 46% in 2019. Among those performing manual tests, four types of tests proved predominant: user acceptance tests, usability tests, story-level tests, and post-deployment tests. Interestingly, despite the seemingly stagnant state of manual testing outlined above, we found that all four of these manual tests grew year-over-year in terms of adoption rates. Here's a quick breakdown of this historical data:

- User Acceptance Tests
 - 2018: 60%
 - 2019: 67%
- Usability tests
 - 2018: 50%
 - 2019: 58%
- Story-level tests
 - 2018: 41%
 - 2019: 47%

- Post-deployment tests
 - 2018: 36%
 - 2019: 40%

THE ROLE OF DEVOPS IN AUTOMATED TESTING

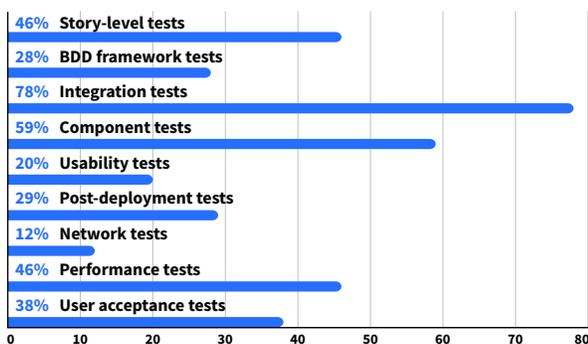
THE EVOLVING ROLE OF DEVOPS

As noted in the Demographics section at the beginning of this report, the ratio of respondents' organizations who have and don't have a dedicated DevOps team is an even 50-50 split, which didn't change significantly from last year (in 2018 48% of respondents reported having a DevOps team). But for those who do have dedicated DevOps teams, the role of these teams is gravitating more to a cultural role than a technological one. In our 2018 Automated Testing survey, 67% of survey takers reported that the main goal of their DevOps team was to help the organization adopt the best Continuous Delivery tools; in 2019, this rose to 74%. Additionally, last year 46% of respondents told us that their DevOps team worked to increase the collaboration and break down silos between Dev and Ops; this year, that figure rose to 53%. All the other, more technical, DevOps team functions either did not grow or shrank in importance. The most shocking, however, was the decreased impetus to develop and deliver software across the entire stack. In 2018, 52% reported this a main goal of their DevOps team; this year, 42% said this type of software delivery was a main goal for their DevOps team. Also, improving frequency of deployments (62% in 2018 vs. 63% in 2019) and introducing automation across the SDLC (56% in both 2018 and 2019) remained of equal importance year-over-year.

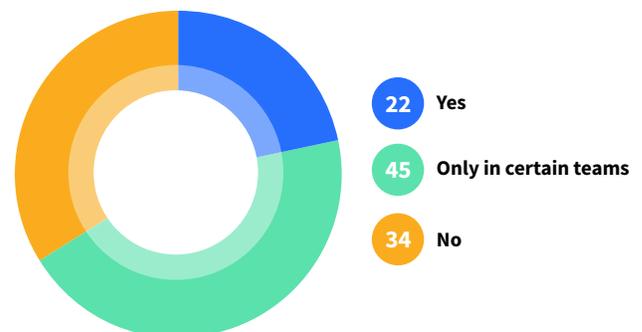
Despite the growing cultural role of DevOps in enterprise software organizations, it still has a large technological part to play. If we compare the data on those respondents whose organizations have dedicated DevOps teams to the stats we gathered on respondents' software build processes, DevOps's continued role in automation becomes clear. Of the 58% of respondents who told us they break their builds up into stages, 34% have a dedicated DevOps team, and of the 46% who include automatic checks to proceed, 21% have a DevOps team. Additionally, while a total of 34% of respondents have security issue detection built into their SDLC, 21% of those respondents have a DevOps team in their

SURVEY RESPONSES

Which tests in your organization's pipeline(s) are currently automated?



Does your organization practice test-driven-development (TDD)?



organizations create agility and automation throughout their build process and implement proper testing and security practices.

TESTING STRATEGIES

Much like the role of DevOps, the role of testing within the SDLC seems to be undergoing an evolution. In 2018, 28% of respondents reported to practice test-driven development (TDD) across all their development efforts. In 2019, this number fell to 22%, while the percentage of respondents who do not practice TDD at all remained stable (36% in 2018 vs. 34% in 2019). The number of respondents using TDD selectively, however, saw a year-over-over increase. In 2018, 36% of survey takers told us they use TDD "only in certain teams." In this year's survey, that number rose to 45%. If we compare these numbers to our data on organizations with DevOps teams, we find that those orgs who have dedicated DevOps teams are more likely to practice TDD across the board than those that don't. 15% of those respondents who use TDD on all their projects also have a dedicated DevOps team, versus 7% who have no DevOps team, but still adhere to TDD for all their development efforts. Interestingly, TDD adoption among "only certain teams" proved equally popular among organizations with and without DevOps teams.

Despite this year-over-year shift in the frequency of test-driven development efforts, the benefits that respondents reported to derive from TDD remained the same. Just as in our 2018 survey, the top four benefits of TDD were: improved quality of code (80%); less time spend debugging (59%); easily maintainable code (49%); the ability to use tests as documentation (45%). Intriguingly, after running these numbers against our stats on DevOps adoption, only one of these benefits seems to occur more frequently in organizations that had dedicated DevOps teams. Among those who reported seeing improved code quality due to TDD, 54% work for organizations with DevOps teams, while 46% do not.

Moving from TDD to the continuous testing paradigm, however, we see DevOps have a more considerable effect. When we asked survey takers which of the four continuous testing pillars they saw as the most challenging, we saw the following results (number one receiving the most votes for "most challenging," with number four receiving the least votes for "most challenging"):

1. Maintaining a stable and up-to-date test lab
2. Test automation creation
3. Test execution and orchestration
4. Test report analysis

Among those respondents who reported to work in an organization with a dedicated DevOps team, we see these results turned almost upside down. If we look at the four pillars of continuous testing through this DevOps lens (again, with number one receiving the most votes for "most challenging," with number four receiving the least votes for "most challenging"), we see the following:

1. Test execution and orchestration
2. Test report and analysis
3. Maintaining a stable and up-to-date test lab
4. Test automation creation

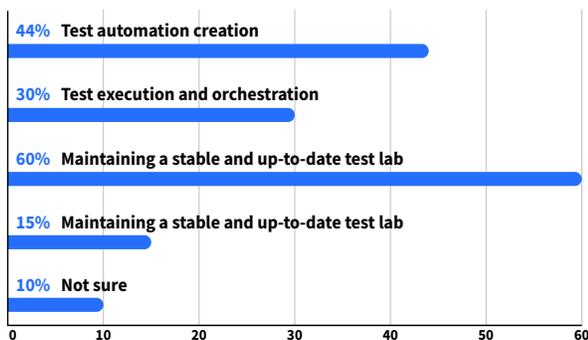
Tellingly, those survey takers who work for companies with a DevOps team found test automation the least challenging of the four pillars.

TECHNOLOGY ADOPTION

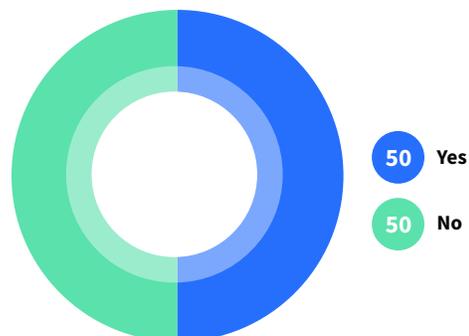
In order to continue to make strides in the areas delineated above, organizations have begun a shift toward the use of microservices and containers. And while these technologies do not necessitate the use of a DevOps paradigm, DevOps teams have found them a boon to their efforts. In terms of microservices adoption, among the general survey population 35% reported adopting a microservice architecture for some projects. But, of even more consequence, we found that 20% of survey takers have both had a dedicated DevOps team and had adopted microservices for some projects. Thus, well over half of respondents who have adopted microservices on certain projects also work in a DevOps development environment. When we look to containers, we see much the same pattern play out. Among the general population, 54% reported to have adopted container technology, while 32% of all respondents both had adopted containers and had a dedicated DevOps team. Again, over half of the users of this technology work in a DevOps-enabled organization. Additionally, when we asked respondents in what environments they use containers, 49% reported to use containers in a DevOps environment.

SURVEY RESPONSES

Which of the four containers testing pillars is the most challenging for you?



Do you have an officially designated DevOps team in your organization?



Diving Deeper Into Automated Testing

Twitter



[@RealGeneKim](#)



[@bridgetkromhout](#)



[@martinfowler](#)



[@bitandbang](#)



[@testertested](#)



[@techgirl1908](#)



[@jezhumble](#)



[@editingemily](#)



[@MaritvanDijk77](#)



[@kriscorbus](#)

Books

Agile Testing: A Practical Guide for Testers and Agile Teams

Learn about Agile testing, the tester's role in a software testing team, seven key traits of successful Agile testing, and more.

Just Enough Software Test Automation

Learn how to design, implement, and manage software test automation.

Implementing Automated Software Testing

Review the components, capabilities, and limitations of automated software testing (AST), and work through the entire AST implementation process.

Zones

DevOps [dzone.com/devops](#)

DevOps is a cultural movement, supported by exciting new tools, that is aimed at encouraging close cooperation within cross-disciplinary teams of developers and IT operations. The DevOps Zone is your hot spot for news and resources about Continuous Delivery, Puppet, Chef, Jenkins, and much more.

Agile [dzone.com/agile](#)

In the software development world, Agile methodology has overthrown older styles of workflow in almost every sector. The Agile Zone is your essential hub for Scrum, XP, Kanban, Lean Startup, and more.

Web Dev [dzone.com/webdev](#)

The Artificial Intelligence (AI) Zone features all aspects of AI pertaining to Machine Learning, Natural Language Processing, and Cognitive Computing. The AI Zone goes beyond the buzz and provides practical applications of chatbots, deep learning, knowledge engineering, and neural networks.

Refcardz

Continuous Testing

In this Refcard, you'll learn about what continuous testing really involves, why continuous testing is crucial for effective Agile and DevOps teams, and what the five-level path to continuous testing looks like.

Getting Started With Appium

In this Refcard you will learn everything you need to know about getting started with this open-source tool, from installing the Appium server to running your first tests. Download this Refcard now to see why Appium is "Mobile App Automation Made Awesome."

Test Design Automation

Download this new Refcard to get started with test design automation, explore the many benefits, and find real-world use cases.

Podcasts

Test Talks

Dive into all things from the world of software test automation, from recent news and book reviews to inspirational test automation thought leadership.

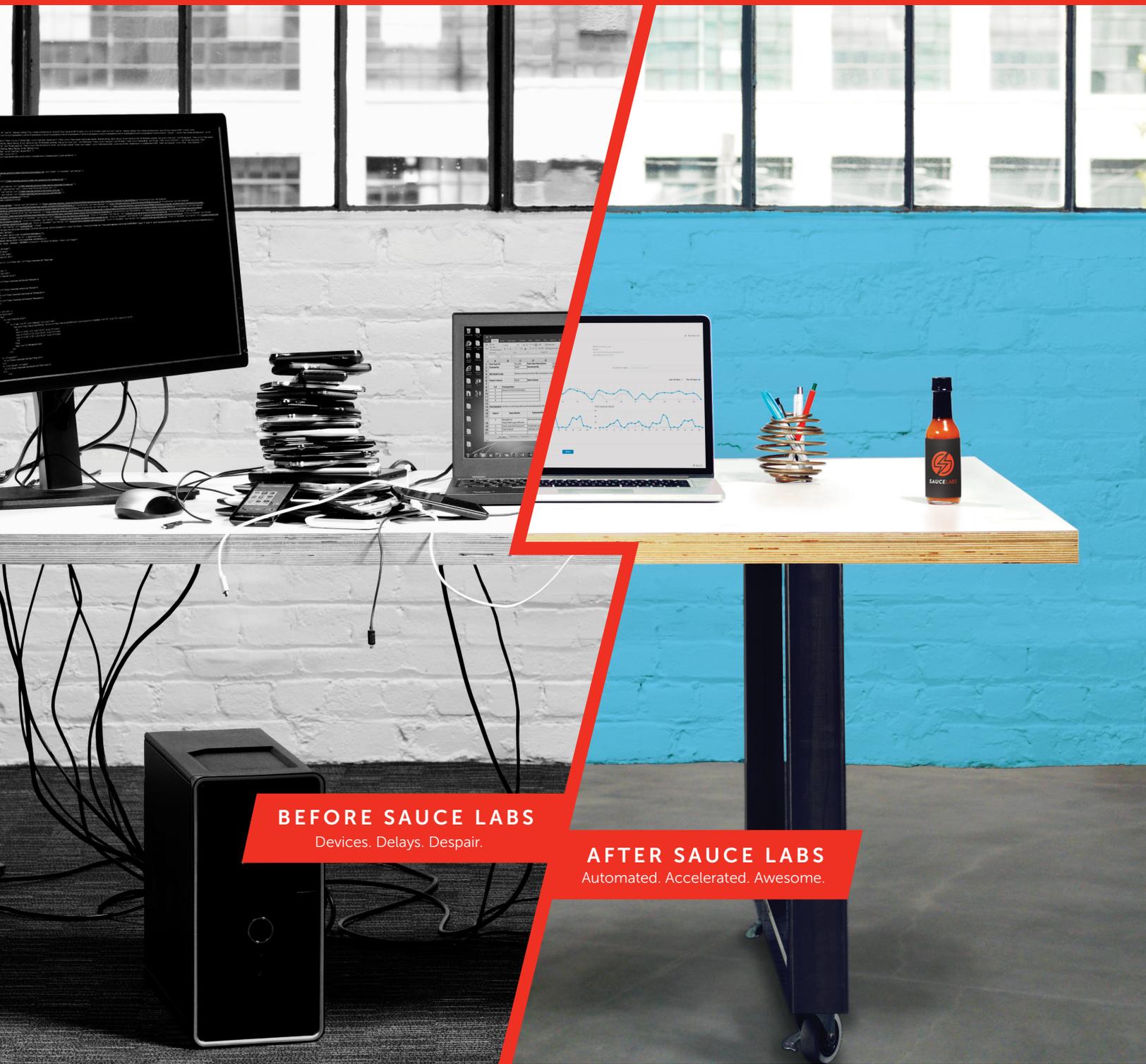
The Testing Show

Learn about myriad aspects of software testing, like advanced Agile with DevOps, automation struggles, continuous delivery, and more.

AB Testing

Explore the modern testing world from the perspective of Agile, data, leadership, and more.

A brief history of web and mobile app testing.



BEFORE SAUCE LABS

Devices. Delays. Despair.

AFTER SAUCE LABS

Automated. Accelerated. Awesome.

Find out how Sauce Labs
can accelerate your testing
to the speed of awesome.

For a demo, please visit saucelabs.com/demo
Email sales@saucelabs.com or call (855) 677-0011 to learn more.

 **SAUCE LABS**

Testing at the speed of awesome.

Automated Testing is Essential to DevOps and Continuous Delivery

Much has been written and said about DevOps, less so about the role of automated testing in a DevOps environment. In order to reap the full benefits of a healthy DevOps practice, organizations must integrate automated software testing into their Continuous Delivery pipelines. It is the only way to ensure that releases occur at both a high frequency, and with a high level of quality.

In order to integrate continuous testing effectively into a DevOps toolchain, look for the following essential features when evaluating an automated testing platform:

- Support for a variety of languages, tools, and frameworks. The programming languages and development tools that your DevOps teams use today are likely to change in the future. Look for a testing solution that can support a broad array of languages, tools, and frameworks.
- Cloud testing. On-demand cloud-based testing is the most cost-efficient option because it obviates the need to setup and maintain an

on-premises test grid that is underutilized most of the time. It also reduces the resource drain associated with identifying and resolving false positives, or failures due to problems in the test infrastructure.

- The ability to scale rapidly. Your testing platform should be able to perform tests as quickly as needed, and be able to do so across all required platforms, browsers, and devices. It should also be highly scalable to support as many parallel tests at one time as you require.
- Highly automated. DevOps teams achieve their speed and agility in part by automating as much of the software delivery process as possible. Your testing solution should work seamlessly with other components of your toolchain, most notably your CI and collaboration tools.
- Security. In a DevOps environment, all members of the team have an important role to play in keeping applications secure. Testing platforms, therefore, need enterprise-grade security features.

A software testing platform that includes these qualities will empower your organization to derive full value from its migration to a DevOps-based workflow by maximizing the agility, scalability and continuity of your software delivery pipeline



WRITTEN BY LUBOS PAROBEK
VP OF PRODUCTS, SAUCE LABS

PARTNER SPOTLIGHT

Automated Testing Platform



Sauce Labs ensures your favorite mobile apps and websites work flawlessly on every browser, operating system, and device.

Category Automated Testing Infrastructure

Case Study

With leading brands such as AOL, HuffPost, Yahoo Finance, Yahoo Sports, TechCrunch, and others, Verizon Media is home to media, tech, and communication products that more than a billion people love and trust. Its engineers develop and manage web, mobile, and server apps spanning tens of thousands of projects. With most of those teams relying almost solely on manual testing, Verizon Media needed an automated testing solution that would scale to their volume and run seamlessly alongside their CI/CD platform. They chose Sauce Labs, and the results have been transformational. By enabling engineers to programmatically access all the latest browser/OS combinations and run tests in parallel, at scale, Verizon Media now runs more than 2.7 million tests a month. Tasks that used to take 45 minutes to complete manually are now completed in five minutes or less. With Sauce Labs running in tandem with their proprietary CI/CD solution, Verizon Media saves a total of 4,500 hours per year in staffing and maintenance, making their investment in automated testing and continuous integration a worthwhile one.

New Release Daily

Open Source? Yes

Strengths

- Enterprise-grade cloud-based test infrastructure provides instant access to more than 900+ desktop browser/OS combinations, ~200 mobile emulators and simulators, and 2,000+ real devices
- Highly scalable, on-demand platform reduces testing time from hours to minutes when tests are run in parallel
- Optimized for CI/CD workflows, testing frameworks tools, and services
- Improve developer productivity with Saucer Headless & Sauce Performance

Notable Customers

- Salesforce
- Walmart Labs
- VISA
- Slack
- Starbucks
- Yahoo!
- American Express
- Verizon Media

Website

saucelabs.com

Twitter

@saucelabs

Blog

saucelabs.com/blog