WHITE PAPER

AUTOMATED TESTING: THE GLUE THAT HOLDS DEVOPS TOGETHER

In order to reap the full benefits of DevOps, organizations must integrate software testing into their continuous delivery pipelines, rather than keep software tests inside a silo. The software testing solution they choose should be highly automated, scalable, secure, and able to support fast, on-demand testing.

SAUCELABS

TABLE OF CONTENTS

EXECUTIVE SUMMARY

On the surface, DevOps may seem to be a simple and straightforward idea. The very term DevOps implies that implementing a DevOps workflow is as easy as achieving collaboration between developers and IT operations teams.

Yet adopting DevOps effectively is more complex than simply uniting Dev and Ops. Alongside developers and IT Ops teams, quality assurance engineers who oversee software tests are an equally crucial component of a successful DevOps operation. Organizations that attempt to implement DevOps without building automated software testing into their continuous delivery pipelines will fall short of achieving the full benefits of DevOps.

Automated testing is the key to successful integration of quality assurance into DevOps workflows. It is the only way to ensure that quality assurance is as continuous, agile and reliable as the rest of the DevOps operation.

In this sense, automated testing serves as the glue that binds together all of the other processes that comprise the continuous delivery pipeline. Without automated testing, DevOps just doesn't work.

DEVOPS BENEFITS

To understand why automated testing is crucial for effective DevOps, it is necessary first to identify the benefits that organizations seek to achieve by following DevOps principles.

The primary benefits of a well-designed and maintained DevOps environment include:

- **Seamless communication across all parts of the organization.** DevOps does this by eliminating the silos that have traditionally separated different teams from one another.
- **Software changes that are delivered on a rapid, continuous and reliable basis.** This requires software updates to be broken into small parts that can be designed, written, tested and put into production continuously, in contrast to the "waterfall" rhythm of traditional software delivery.
- **Maximum agility.** When software delivery is agile, applications in your toolchain can scale easily in response to fluctuations in demand. In addition, software delivery teams have the ability to switch easily between development frameworks and tools according to changing needs or preferences.

- **The elimination of unforeseen delays in software production.** These delays typically result from having to fix problems with code after it is in production, at which point rollbacks are costly and time-consuming. DevOps can help to avoid this risk by ensuring that code is tested automatically, as part of the continuous delivery pipeline, before it goes into production.

WHY DEVOPS NEEDS SOFTWARE TESTING

As noted above, the term DevOps implies that software delivery can be optimized simply by facilitating collaboration between development teams and IT operations teams. That is not true. Software testers also need to be seamlessly integrated into the continuous delivery chain and work alongside development and IT Ops teams.

To be sure, integrating development with IT Ops is one important part of achieving DevOps. Software is delivered faster and more reliably when programmers are in constant communication with the system admins who deploy and manage software in production. And speed and reliability increase when both of these groups enjoy maximum visibility into the state of application development.

Yet this type of collaboration is not enough on its own to make DevOps work well. If software testers remain in a silo and do not participate in the continuous delivery chain, a number of problems arise. These problems include:

- **The inability to test software updates at the pace at which they are produced and deployed.** If automated software testing is not part of the continuous delivery pipeline, the code changes that developers write cannot be tested continuously. They will instead have to be tested irregularly, whenever the testing team is able to address them. Under these conditions, the risk of releasing problematic code into production greatly increases, as does the likelihood of delays that prevent updates from reaching production continuously.

- **Agility is undercut.** Even if the rest of the software delivery chain is agile, failure to integrate automated testing into the continuous delivery pipeline will undercut the organization's ability to derive value from that agility. Programmers will lack the ability to switch between development frameworks easily because they will not be able to assure that the testing team is ready to support the change. Software delivery will not scale

because software tests cannot scale when they are not integrated with the continuous delivery chain.

- **Quality suffers.** Part of the value of DevOps is its ability to standardize and streamline software delivery processes. When automated testing is not part of the continuous delivery pipeline, tests remain irregular and ad hoc. That undercuts the overall quality of the application being developed.
- **Rollbacks become common:** To make the most of DevOps, the software delivery chain should be fully automated and continuous. The introduction of bugs that force developers to roll back code once it has been written is a serious hamper to continuity. Without automated testing, the likelihood of rollbacks is high, because code is pushed down the pipeline before it is tested properly.

For all of these reasons, organizations that leave software testing in a silo rather than integrating it into the continuous delivery pipeline fail to achieve the value of DevOps—even if the rest of their DevOps operation is well designed.

AUTOMATED SOFTWARE TESTING: THE GLUE BETWEEN DEV AND OPS

Automated software testing is also crucial for DevOps because tests are the medium that ties development to IT Ops, and testing ensures that updates flow smoothly from the beginning to the end of the continuous delivery pipeline.

A typical DevOps delivery chain starts with developers, who design application changes and then write the code to implement them. The delivery chain ends with IT Ops, which is responsible for pushing the updates into production and maintaining them.

Software tests are what come in the middle. Integration tests ensure that changes or new features written by developers can be added to the application without breaking it. Usability testing identifies flaws that developers might not have foreseen when designing code, and prevents those problems from reaching end users. Device compatibility tests ensure that code written and built in development environments will work as intended in real-world settings, which are much more complex and involve many more hardware and software variables than development environments.

In each of these respects, software tests are the glue that binds the code written by developers to the production-level application deployed by IT Ops.

A continuous delivery pipeline that lacks automated, continuous testing will not enable developers and IT Ops teams to interact effectively with one another.

HOW TO ENABLE CONTINUOUS TESTING

In order to integrate software testing effectively into a continuous delivery pipeline, DevOps teams should implement testing solutions that enable and reinforce DevOps goals. When choosing a testing platform, look for the following essential features:

- **Support for a variety of frameworks.** The integration servers or code repositories that your DevOps team uses today are likely to change in the future. So are other aspects of your development process. For example, the web app that you develop today may be transformed in the future into a hybrid app that takes advantage of HTML-based and native features at the same time. Since your development needs will change in ways that you cannot fully predict, it is important to seek a testing solution that can support a broad array of frameworks and adapt to suit your changing needs. Otherwise, you will undercut the agility of your continuous delivery pipeline because you will be wed to particular tools for developing and managing your code.

- **The ability to scale.** Your testing platform should be able to perform tests as quickly as needed, and it should support as many parallel tests at one time as you require. On-premises testing solutions are unlikely to offer the necessary scalability because they will be constrained by limited hardware resources. In contrast, a cloud-based testing platform can scale as seamlessly as the rest of your continuous delivery pipeline.

- **The ability to test quickly.** In order to avoid delays to your continuous delivery chain, you need to be able to perform tests quickly. Performing parallel tests on a large scale is one way to achieve this. Another is running compatibility tests on simulated devices first (since these tests take less time), and performing more time-consuming tests on real devices later in the pipeline, just before code enters production.

- **High automation.** DevOps teams achieve their speed and agility in part by automating as much of the software delivery process as possible. Your testing solution should be as automated as the rest of your DevOps toolset. You should be able to trigger tests, analyze results and share testing information across the organization (using features like ChatOps

integration) in a completely automated fashion. While manual tests may always be required occasionally, the rest of your testing should be as automated as possible.

- **On-demand testing.** To avoid kinks in the continuous delivery pipeline, you need to be able to perform tests whenever they are necessary. There are two ways to do this. One is to maintain a massive on-premises testing environment with enough resources to perform tests whenever you need them. The other solution is a cloud-based testing solution that can complete tests quickly upon request. The second solution is much more cost-efficient because it obviates the need to implement and manage an expensive on-premises test grid that is underutilized most of the time. Cloud-based testing also allows you to avoid false failures generated by on-premise test grids, as well as the inefficiencies that can arise from having to find and fix bugs in on-premise test infrastructure.

- **Security.** In a DevOps environment, all members of the team—including software testers—have an important role to play in keeping applications secure. Testing platforms, therefore, need to include security features, such as the encryption of test data when it is exchanged over the network, and robust access control policies.

A software testing solution that includes these qualities will allow your developers and IT Ops teams to work together as efficiently as possible. It will ensure that the code you put into production is reliable and stable across diverse environments. And it will empower your organization to derive full value from its migration to a DevOps-based workflow by maximizing the agility, visibility, scalability and continuity of your software delivery pipeline.

ABOUT SAUCE LABS

Sauce Labs ensures the world's leading apps and websites work flawlessly on every browser, OS and device. Its award-winning Continuous Testing Cloud provides development and quality teams with instant access to the test coverage, scalability, and analytics they need to deliver a flawless digital experience. Sauce Labs is a privately held company funded by Toba Capital, Salesforce Ventures, Centerview Capital Technology, IVP and Adams Street Partners. For more information, please visit saucelabs.com.

SAUCELABS

SAUCE LABS INC. - HQ
116 NEW MONTGOMERY STREET, 3RD FL
SAN FRANCISCO, CA 94105 USA

SAUCE LABS EUROPE GMBH
NEUENDORFSTR. 18B
16761 HENNIGSDORF GERMANY

SAUCE LABS INC. - CANADA
134 ABBOTT ST #501
VANCOUVER, BC V6B 2K4 CANADA