

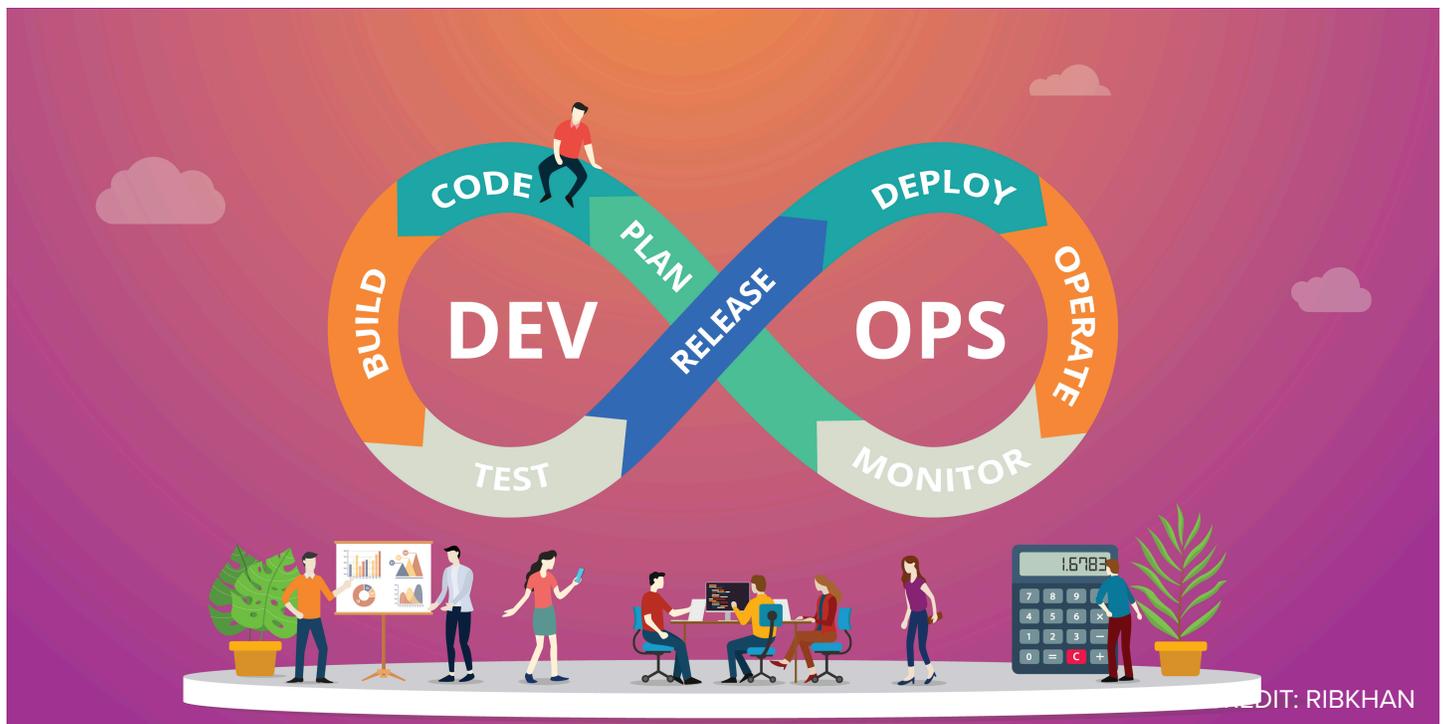
GIGAOM

RESEARCH BYTE

Scaling Testing Across the DevOps Pipeline *Delivering Software at Speed without Sacrificing Quality*

JON COLLINS

TOPIC: **DEVOPS**



Scaling Testing Across the DevOps Pipeline

Delivering Software at Speed without Sacrificing Quality

TABLE OF CONTENTS

- 1 Summary
- 2 Definition
- 3 Motivation
- 4 Process and Strategy Considerations
- 5 Leadership Perspectives
- 6 Vendors Offering DevOps Testing and Quality Solutions
- 7 Conclusion
- 8 About Jon Collins
- 9 About GigaOm
- 10 Copyright

1. Summary

All companies are software companies, goes the adage. While definitions of digital transformation may vary, enterprise boards agree that digital technologies unlock new opportunities for closer customer engagement and new business models. This being the case, time to ROI has become an essential metric; if you cannot develop something as fast as the competition, then you will lose out. The added complication is that nobody knows exactly what customers want; therefore, giving people the ability to try new things has become more important than delivering a gold standard.

Building on the notion of continuous integration and delivery, DevOps responds to this need for agility and speed. For many organizations, however, software testing practices contradict these goals. Software quality is as important today as it ever was. However, the way in which software is tested needs to evolve to fit with innovation practice.

In this report, we look at how quality and testing can become an integral element of DevOps, starting and ending with customer needs and covering every stage along the way. We review how software quality and testing practice is evolving to meet the needs of innovation as characterized by DevOps: an evolution we are calling DevQualOps.

Key findings of this report are:

- Business pressure to deliver new software functionality is forcing legacy testing practices onto the back burner, increasing cost and risk.
- Modern cloud-native applications, business speed, agility drivers, and DevOps approaches to delivery and operations influence how testing needs to take place.
- The rationale for software testing remains the same whether an application is developed quickly or slowly, by a line of business or IT, customer-facing or not.
- DevQualOps encompasses activities that assure an appropriate level of quality across the DevOps lifecycle, across software testing, environment delivery, collaborative planning, and reporting.
- Automation, enablement, visibility, collaboration, and control are key to delivering DevQualOps across the application estate.
- An organization's ability to integrate testing successfully into the development and operations lifecycle can be seen as a symptom of DevOps maturity.
- A "start, measure, manage" approach enables the organization to break from "anything-goes" approaches and start the journey towards DevQualOps maturity.
- The ability to measure quality is the ultimate test for gauging how well quality investments are benefiting the business and its customers.

Overall, we review how to turn application testing into a catalyst for innovation and value creation rather than being perceived as a constraint.

2. Definition

To be clear, the principles of software testing do not change whether an application is developed quickly or slowly, or whether it is customer-facing or not. The purpose of software testing is to identify defects (bugs) in the software, which developers can then resolve: a generally accepted principle is that the sooner a problem is identified, the cheaper it is to fix. So, for example, if a software requirement is identified as flawed prior to development, it can be changed without even having to rewrite code, which has to be the cheapest fix of all. And at the other end of the pipeline, a problem found when an application is already in use can be very expensive in both impact and cost of resolution.

Traditional application stacks are easier to scope and bound, and therefore, to test (ie., involving executable code running on a desktop computer, accessing a server back-end). However, modern applications differ in ways that influence what needs to be tested and how.

For example, modern applications are:

- Often web-based or mobile
- Based on cloud platforms or open-source libraries
- Strongly customer-focused as a portal onto the business
- Offering integrations via RESTful APIs or plugins
- Event-driven, with little control over how they will be used.

Each area of testing needs to be considered in light of these factors, as well as taking into account DevOps agility and speed. Our conversations with CIOs suggest that innovation speed can sometimes overrule test coverage; in layperson's terms, let's get the software out there and see if it's what is needed, ensuring fitness for purpose before making sure it is watertight. DevQualOps balances these needs and assures an appropriate level of quality across the development and operations lifecycle, by:

- Minimizing the impact and potential cost of software defects to the business;
- Supporting testing needs across unit, integration, acceptance, and regression testing;
- Offering appropriate test data and environments to enable software testing;
- Enabling stakeholders (ie. customers) to participate in the testing process;
- Providing mechanisms to ensure the software is only deployed when it conforms to set criteria;
- Incorporating feedback and reporting to enable developers to resolve bugs; and
- Assessing test results and data so managers can set application strategy and planning.

For modern applications created using a DevOps approach, testing can never be finished; it is, therefore, a question of ensuring that an application is of sufficient quality for its need. In the next sections, we shall review the challenges to making this happen, and what decision-makers can do in response.

3. Motivation

The notion of a constant or perpetual beta product first arrived in the late 2000s, during what was termed the Web 2.0 age. As websites became more interactive and spawned online applications, it was seen as acceptable to offer workable versions of new products on the understanding that (a) features would be released on an almost daily basis; (b) they might not always work; and (c) any faults would be quickly resolved.

As web-based applications have matured, the beta idea has fallen by the wayside but the principle of continuous delivery still holds, for a simple reason: each new feature offers new potential for business value. Consider travel booking sites. Each customer interaction with a travel site could lead to a potential sale of a flight, hotel, or service. If site A releases an option to link, for example, flights and airport parking, this may prove to be useful enough for travelers to default to site A, which means other sites have a competitive disadvantage until they also offer the feature.

The pressure to release new features is therefore very high. From a business case perspective as well, the moment a feature is released, it will start to generate value, which accumulates over time. Even the simplest version of a service can be of worth; for example, consider the insurance company that created a very basic inquiry form for a new product. It started to generate telesales even as the more complex application form was being built. If every company is a software company, feature delivery is the key to business value.

While true, this can result in software quality best practice falling by the wayside. The most obvious consequence is that poorly tested features are released with impunity, leading to the inevitable set of problems: poor customer experience since the application does not work; potential data loss, privacy or security breaches; possible loss of business; and of course, the time to resolution which takes effort away from new feature creation, undermining the very point of continuous delivery.

A lack of software testing strategy and policy can cause other consequences. Applications may have very different pipelines with different testing approaches (from testing everything to minimalist approaches), making it harder to understand where to apply effort. A historically lax approach to testing can create a significant backlog of features that have never been fully tested.

At the same time, it isn't as simple as saying "do more testing." Today's applications need to run on multiple target platforms, work across a wide variety of mobile devices and web browsers, and integrate in complex and unpredictable ways with other applications. Applications are by their nature complex and fragmented; an application may be made up of hundreds (or more) of microservices, making it almost impossible to test every single way in which one microservice might communicate with another.

4. Process and Strategy Considerations

In response, we could fall back to the idea of the “devil’s triangle” between functionality, speed of delivery, and quality: the principle is that you can have any two, but not all three. So, if the speed of delivery and quality are to lead, the cost needs to be borne by functionality, which will be a difficult pill to swallow for any software product manager. Ultimately, the required approach is to consider how to balance all three, based on specific ways to raise quality to an acceptable level without undermining either functionality or speed. To achieve this, DevQualOps should be implemented as follows:

1. **Automate:** Automate as many testing activities as possible. While open-source frameworks exist for automated test execution (such as Selenium and Appium), you can also automate how test cases are defined, environments are deployed, results analyzed, and coverage assured.
2. **Control rollout:** Use feature flags to switch parts of applications on or off, or present alternatives, depending on runtime configuration criteria. Test a new feature on a limited group prior to broader release.
3. **Enable developers:** While it might be true that coders do not want to spend their lives as testers, tools exist to enable developers to code, while still being able to test and review. Such tools can align with peer review to motivate teams around raising the bar on quality.
4. **Be consistent:** DevOps pipelines for different applications can benefit from similar structures, policies, and approaches, such as the ‘minimum necessary’ sign-off of a feature prior to deployment. Establish these in collaboration between business, IT, and quality teams with tools that enable you to adjust policies over time; for example, raising acceptability thresholds for new feature releases, or initially reporting defects rather than blocking a release.
5. **Create visibility:** Perhaps most importantly, get a view of quality across your application estate and set priorities, identifying where attention needs to be spent. Follow a start, measure, and manage approach, where the first step is to gain a picture of what is currently happening. Then create metrics that matter to your business. Finally, optimize based on these metrics.
6. **Employ Pareto:** Given that you will never be able to test every possible scenario, nor every facet of your application estate, set priorities based on both the most common/at-risk applications, and the most significant features. Once you are on top of this 20% subset, you can expand.
7. **Involve stakeholders.** Quality decisions need to be made by those most impacted—in this case, business units, customers, and partners. In this way, testing can become a source of dialog between stakeholder groups.

The goal of DevQualOps is to increase testing consistency and uniformity while still allowing flexibility. Done correctly, it reduces overall costs, freeing up resources for further testing and improvements.

5. Leadership Perspectives

An organization's ability to integrate testing successfully into the development and operations lifecycle implies DevOps maturity. Realistically, while many enterprises are doing DevOps at some level, in practice they might be less advanced than they think. The ability to deliver high-quality applications via DevOps validates the effectiveness of the pipeline and approach in use. With this in mind, we consider the following perspectives:

- **CFO or COO.** DevQualOps reduces the cost of software delivery, albeit requiring an up-front investment of time, effort, and tooling. Software quality becomes something to agree on up-front, at a board level, to be enacted by stakeholder groups.
- **Business Executive.** DevQualOps provides better engagement with customers and employees, together with business-level measurements to ensure applications are delivering what is needed.
- **CIO or Senior IT.** DevQualOps gives back control of all applications and website estate, reaching a shared agreement with lines of business of policy and thresholds. It provides a purview over the state of software quality and risk to feed strategy, planning, and prioritization.
- **Quality Manager.** DevQualOps strategy works across new applications, sites, and services enabling maturity to advance.

6. Vendors Offering DevOps Testing and Quality Solutions

Below we present a non-exhaustive list of vendors that play in the DevQualOps space, describing what each does in terms of features.

Applitools

Applitools sets its stall around machine-learning-based functional and visual testing of apps, websites, and desktop user interfaces. Applitools reviews interfaces in the same way a person might, looking for errors, inconsistencies, and changes. Functional and visual testing is required across the DevOps lifecycle. However, given that larger applications and sites can have many thousands of screens across multiple targets, undertaking visual checks manually is a daunting (if not impossible) task.

The Applitools core product works as a component of the DevOps pipeline, integrating with most common test environments and continuous integration platforms via SDKs. It reduces the testing workload through automation by decreasing test script lines of code (by up to a factor of five). It also increases the scenarios in which testing can be done; for example, enabling real-time testing of operational sites and applications.

Applitools is adding features to align with the development pipeline. For example, it can link a specific interface error to the problem code, plus pull a request to fix it. Applitools also offers an analytics dashboard (recognizing that customers are just as likely to use the Applitools API to pull data into their dashboard of choice). Recently, the company is embracing the fact that interface-based tests enable application functions to be tested automatically, rather than at the code level.

Unsurprisingly, the company is focused on enterprise organizations, which are more likely to face the pain of testing complex interfaces. For similar reasons, Applitools has traditionally worked with more customer-facing, compliance-oriented sectors, such as financial/insurance, retail, and healthcare. More recently its focus has moved to encompass organizations facing testing bottlenecks, as they look to scale their digital transformation activities.

Experitest

Experitest offers a test management platform (SeeTest) and a hosted laboratory environment (Experitest Grid) for user interface testing of mobile and web-based applications. The hosted environment enables tests to run in parallel, speeding up testing, and increasing test coverage.

Test automation is based on de facto test frameworks: Selenium for cross-browser and Appium for mobile, as well as XCUITest and Espresso. Experitest offers Appium Studio for mobile test definition, which includes an open interface to integrate testing of specific features, e.g. fingerprint recognition.

A portal offers reporting and analytics of test results. Experitest can be API-driven, enabling its

integration with continuous integration tools, such as TeamCity, Jenkins, and Bamboo, as well as with a number of testing frameworks.

Functionize

Functionize was set up to solve the testing bottleneck to agile development in general, and DevOps in particular, using machine learning. It does this in a number of ways across software testing activities, enabling the autonomous execution of test activities, reducing the time taken to write cross-browser test cases and addressing their continued effectiveness as applications change. First and foremost, it uses adaptive natural language processing to enable writers to create test cases in plain English. This can save a great deal of time, particularly when multiple versions of a single test case are required.

Functionize is cloud-based. During execution, it applies AI and machine learning to learn from past test case data across their (anonymized) customer base and increases the robustness of test cases in response to subtle application changes, such as a revised URL. A root cause analysis engine can look at test failures and diagnose ways to repair the test in question, and new tests can be generated based on how users interact with the application concerned.

Functionize competes somewhat with de facto testing frameworks, such as Selenium, and notes its “plain English” approach as a differentiator. The solution focuses on UI testing but can also test microservices, web services, and APIs.

Typical Functionize customers are mid-market and enterprises, who are facing testing complexity challenges. Functionize can integrate with existing test management solutions, as well as run as a stand-alone. The company has a focus on partnerships, using services and consulting partners for its go-to-market activities.

Gremlin

Gremlin offers a hosted fault detection and protection platform, using Chaos Engineering principles, i.e., simulating failure scenarios in a controlled way to reveal weaknesses in systems and infrastructure. The capability responds to the fact that modern, microservices-based applications are massively complex and constantly changing.

The capability allows teams to test for infrastructure failures, such as latency and performance, in a proactive, safe, programmatic, and reproducible way. This delivers both higher levels of resilience and reduces the time required to diagnose and resolve failures. It currently integrates with the Spinnaker open-source build and deploy tool, and is planning to integrate with more CI/CD tools over time.

Gremlin can also simulate application-level failures for more granular attacks through their offering ALFI (Application Layer Fault Injection). These also work in serverless environments. It incorporates the ability to control the impact of a simulated fault, or indeed, switch off the simulation at any point.

Gremlin is targeted at both internal business and customer-facing systems and sites. Common use cases are where an organization is moving to the cloud or looking to adopt a microservices approach and wants to test the migration; or where an operations department is looking to test its monitoring and alerting capabilities, or train teams in fault diagnosis and resolution.

IBM

Unsurprisingly, given its large development software portfolio, IBM has a broad testing offering that covers technologies from mainframe to Kubernetes and which ties into the wider IBM DevOps pipeline platform. IBM DevOps and its UrbanCode portfolio are discussed further in [GigaOm's Value Stream Management report](#). Of specific interest are IBM Rational Testing products:

- **IBM Rational Test Workbench**, offering API, functional UI, performance, and regression testing.
- **IBM Rational Test Virtualization Server**, enabling parts of an application to be simulated during functional tests.

IBM typically works with larger enterprises. A central goal is to shift testing left in the DevOps cycle, increase coverage, and minimize the impact of testing activities through automated continuous testing. The company looks to help with the cultural and tooling changes that are a particular challenge for such organizations.

Note: we continue to monitor the IBM partnership with development partner HCL Technologies, which covers both the IBM UrbanCode and IBM Rational portfolios. IBM owns these portfolios but has previously divested other software products to HCL following successful partnership engagements.

OverOps

OverOps sets its stall around the concept of pre- and post-production software reliability, notably the fact that it is better to identify and resolve a software issue through automated analysis of executable code, rather than (the more standard practice of) analyzing logs to diagnose problems after they have happened.

The OverOps solution offers runtime application analysis and anomaly detection, alongside analysis of historical data to identify whether an error has been seen before. The result is both specific identification of issues, plus broader statistics and insights — for example, this includes the idea of 'slow-downs' to identify performance degradation over time.

Importantly, the OverOps solution is designed to feed information back to developers as well as quality assurance engineers, via both custom portals and with links to service management tools such as ServiceNow or PagerDuty. The software incorporates the notion of quality gates, such that deployment

can be halted if code quality falls beneath a defined threshold.

The company tends to work with larger companies that have experienced the challenges and consequences (e.g. revenue loss) of run-time software reliability, and which recognize the need to change their approach.

Percy

Percy has set its stall around what it calls continuous visual testing focusing on changes to user interfaces. Percy identifies differences from previous versions and enables the changes to be reviewed by software engineers in an automated fashion, rather than being manually intensive (or, as the company points out, having bugs discovered by customers following the release).

The rendering process is not simple. For example, Percy offers a “snapshot stabilization” feature to ensure minor discrepancies (e.g. due to font rendering) are not seen as a visual difference. It can also deliver results based on tens, or hundreds of thousands of screen snapshots at a time. The result is to promote the notion of visual coverage in much the same way that code coverage is seen as a metric for functional testing.

The product works with source code repositories such as GitHub and Bitbucket and integrates with the CI/CD toolchain to provide visual reviews as part of day-to-day development workflows.

Percy’s product is offered as a service, with a free plan (based on the number of screen snapshots) to enable organizations to get started. While the product can work with all web-based interfaces, the company aligns with engineering teams that already have agile adoption and CI practices, and need to scale the way they assure visual tests.

Perforce/Perfecto

Perfecto (recently acquired by Perforce) offers a continuous testing platform for mobile and web applications. It enables DevOps teams to create the functional, non-functional, unit, and integration tests, execute these at scale and then analyze results with quality dashboards. These can integrate with CI tools as well as ChatOps collaboration platforms and tasking tools, such as Slack and Jira. A key objective around using Perfecto is to reduce business risks associated with code quality, as soon as possible, within DevOps processes.

For test creation, Perfecto provides an automation layer reflecting standard features of mobile and web-based apps, such as voice automation, image injection, and Bluetooth. It also offers a codeless test recording and playback facility. Perfecto incorporates machine learning in test creation, execution, and analysis. For example, rebooting stuck test devices and identifying the root cause of a test failure.

Perfecto also offers a cloud-based test laboratory to execute tests at a high scale, reducing the execution time in what it calls “burst mode” so it can maximize the number of tests that are run (say)

overnight. Perfecto can also work with open-source platforms, such as Selenium and Appium, for locally hosted testing.

From a collaboration perspective, Perfecto dashboards target different personas, such as developers, site reliability engineers, and (potentially non-technical) test teams. Dashboards can integrate results from other forms of testing, such as API or penetration testing. Note that the broader Perfocore portfolio incorporates static code analysis and quality reporting tools, such as Helix QAC and Klocwork, which also integrate with development pipeline tooling.

Perfecto is aimed at large enterprises across telecom, financial, insurance, retail, healthcare, and automotive, each of which has different use cases. For example, automotive uses more Bluetooth, whereas finance may have more need to test identification mechanisms.

Plutora

As described in the [GigaOm Research Byte on Value Stream Management](#), Plutora is looking to help organizations deliver software faster through better information aggregation, automation, and management. Addressing DevOps quality and testing, Plutora's solution set incorporates test environment management and orchestration tools, enabling complex test environments to be coordinated, controlled, and delivered.

Doing so for enterprise customers requires a view across agile/DevOps and traditional (waterfall) practices, as well as legacy tools, applications, and infrastructure. The key is to enable test management to plan and define test environments through the use of metadata, which can be managed centrally, so complex test environments can keep pace with changing requirements. Reports provide an overview of existing test environments and their status.

In addition to plugging into its end-to-end pipeline offering, Plutora integrates with operational management tools, such as ServiceNow, to close the loop back into development.

Sauce Labs

Sauce Labs' vision is to make testing fast, reliable, and easy throughout the software development process. They want to do this by balancing the needs of innovation, speed, and software quality while delivering continuous testing at scale. Sauce Labs has a strategic focus on automation and employs an accomplished team of solution architects to provide expertise for better testing practices.

Sauce Labs offers a hosted testing laboratory called the Sauce Labs Continuous Testing Cloud, which leverages the Selenium and Appium test automation tools, among others. Historically, the offering has focused on functional testing but the platform is broadening to address performance, visual testing, accessibility, and security. The company recently launched a lightweight testing solution, Sauce Headless, enabling development teams to check and identify issues earlier in the life cycle. The geographically distributed environment augments compliance with local privacy laws, such as GDPR,

and supports common identity management providers to enable provisioning of test resources.

The platform offers a comprehensive set of analytics based on tests executed from real-time dashboards, to indicate test status, as well as broader insights, such as trending information and areas of code that may be more problematic. This information can be accessed via a portal or integrated into a third-party dashboard using the Sauce Labs REST API. Sauce Labs integrates directly with common continuous integration tools, as well as collaboration tools such as Slack.

While Sauce Labs is used by many organizations, it focuses on enterprises facing the challenge of aligning their testing goals with the need for innovation speed. The company, therefore, pays attention to customer on-boarding, so they can grow in maturity and scale their ability to test across the lifecycle and the broader application base. For example, the company offers “Sauce Days” to help customers, as well as training, workshops, mentoring, and other professional services delivered on-site.

SmartBear

SmartBear offers tools that support and enable testing across the DevOps pipeline, from planning through to operational performance monitoring, many of which have been acquired in recent years or are based on SmartBear-supported open-source projects. Alongside tools to enable cross-browser and mobile testing (based on the Selenium platform), these include:

- Zephyr for test case management;
- HipTest for acceptance test design and execution;
- TestLeft for developer-centric functional testing; and
- SoapUI Pro and LoadUI Pro for API testing.

Despite the company offering more of a portfolio of products rather than a tightly integrated solution, SmartBear’s approach is to ramp organizations up as quickly as possible. Alongside the testing portfolio, it has an interest and capability in API management. This is relevant as it plays into the ability to integrate testing capabilities into the pipeline, as well as delivering on API and RESTful interface testing, which the company sees as a differentiator.

Tricentis

The Tricentis “No-code” approach was founded on model-based software testing, in which the models need to be business readable. This abstraction is a deliberate departure from script-based testing, enabling (in principle) higher levels of automation with less technical expertise.

The Tricentis Tosca model-based engine can be used for web and mobile apps, as well as bespoke and packaged applications across UI and API testing. Alongside automation, Tricentis Tosca

incorporates features to support prioritization, test case design, test data management, and service virtualization to enable incomplete applications to be tested.

It incorporates machine learning within testing, test automation, management, and identifying areas of risk or incomplete test case coverage. Meanwhile, Tricentis qTest (previously QASymphony) incorporates a workflow engine to enable integration and orchestration of DevOps pipeline tools, such as GitHub, Jira, and Jenkins. Tricentis Flood offers a distributed load testing capability.

The company has an enterprise business focus, catalyzed by its partnerships with packaged software vendors, such as SAP and Oracle, and supported by its broader Robotic Process Automation (RPA) portfolio.

7. Conclusion

DevQualOps success aligns with general DevOps maturity. Whereas less advanced organizations may be exposed to greater levels of software quality risk, an integrated approach reduces quality costs and lessens business exposure. Looking forward, the most mature organizations will ensure application quality will be leveraged as a way of unlocking business opportunities.

In this innovation age, testing is not just about defects (which should be identified and resolved as early as possible) but is also about trialing new services and improving customer engagement. By moving application testing to the front, organizations can experiment proactively, learning what works and driving their businesses forward as a result. Case in point, a simple version of a new feature can be rolled out to a beta group using feature flags, testing whether it is useful prior to delivering a complete, fully tested version; multiple groups could be given different versions of a feature to find out which works best; and so on.

Software quality will always be a compromise, prioritizing different kinds of testing according to the task at hand. In some instances, it will be better to get a functionally incomplete application out to a specific group, to determine its suitability, whereas in others, functional completeness and robustness may be a higher priority. Both quality and DevOps managers require such metrics to confirm that the effort distributed across quality-related activities is delivering the expected results.

The ability to measure quality itself should be seen as the ultimate test. Not only to feed decision making (e.g. whether or not to deploy), and in terms of business outcomes (e.g. whether a new feature improves the user experience), but also to gauge how well quality investments are increasing quality as a whole, to the benefit of the business and its customers.

8. About Jon Collins



Jon Collins has advised the world's largest technology companies in product and go to market strategy, acted as an agile software consultant to a variety of Enterprise organizations, advised government departments on IT security and network management, led the development of a mobile healthcare app and successfully managed a rapidly expanding Enterprise IT environment. Jon is frequently called upon to offer direct and practical advice to support IT and digital transformation strategy has served on the editorial board for the BearingPoint Institute thought leadership program and is currently a columnist for IDG Connect.

Jon wrote the British Computer Society's handbook for security architects and co-authored *The Technology Garden*, a book offering CIOs clear advice on the six principles of sustainable IT delivery. He has written innumerable papers and guides about getting the most out of technology and is an accomplished speaker, facilitator, and presenter.

9. About GigaOm

GigaOm provides technical, operational, and business advice for IT's strategic digital enterprise and business initiatives. Enterprise business leaders, CIOs, and technology organizations partner with GigaOm for practical, actionable, strategic, and visionary advice for modernizing and transforming their business. GigaOm's advice empowers enterprises to successfully compete in an increasingly complicated business atmosphere that requires a solid understanding of constantly changing customer demands.

GigaOm works directly with enterprises both inside and outside of the IT organization to apply proven research and methodologies designed to avoid pitfalls and roadblocks while balancing risk and innovation. Research methodologies include but are not limited to adoption and benchmarking surveys, use cases, interviews, ROI/TCO, market landscapes, strategic trends, and technical benchmarks. Our analysts possess 20+ years of experience advising a spectrum of clients from early adopters to mainstream enterprises.

GigaOm's perspective is that of the unbiased enterprise practitioner. Through this perspective, GigaOm connects with engaged and loyal subscribers on a deep and meaningful level.

10. Copyright

© [Knowingly, Inc.](#) 2019. "Scaling Testing Across the DevOps Pipeline" is a trademark of [Knowingly, Inc.](#). For permission to reproduce this report, please contact sales@gigaom.com.