

This IDC Technology Spotlight examines the use of automation within assurance organizations and explores the role that testing automation services can play in aiding organizations with enhancing and streamlining their quality assurance activities.

# Advancing Testing Automation to Bolster Innovation and Competitive Advantage

January 2019

**Written by:** Peter Marston, Research Director, Application Life-Cycle and Cloud Services

## Introduction

Application delivery teams face pressure to develop, test, and release applications faster amid constantly evolving business needs. To meet this challenge, many teams have turned to more progressive methodologies such as agile, DevOps, and hybrid application delivery. However, as more progressive application delivery has grown in popularity, new impediments have emerged that undermine testing scope and application quality.

The result? Application development and deployment teams struggle to ratchet up their application testing speeds without sacrificing testing scope or quality. This IDC Technology Spotlight examines how delivery teams can advance their testing automation capabilities.

### AT A GLANCE

#### KEY TAKEAWAY

IDC believes that advancing testing automation as part of application life-cycle management will grow in importance over the next several years as organizations seek competitive advantage.

## Progressive Application Delivery Spreads to the Enterprise

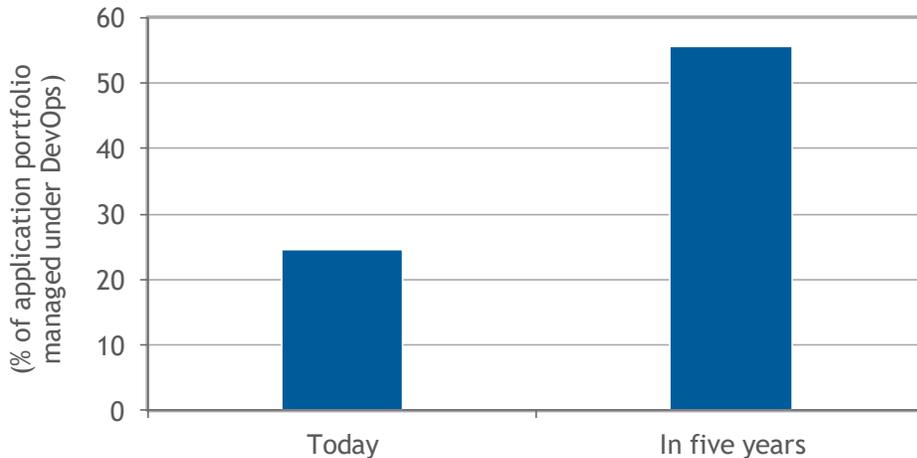
Application delivery teams have traditionally relied on utilizing labor to test applications and ensure quality. Today, teams are approaching application delivery differently. IDC has found that nearly 98% of organizations have adopted DevOps to develop, test, and manage their applications. On average, organizations use DevOps methodologies to manage nearly 25% of their application portfolio today, and they expect to apply DevOps to more than 55% of their application portfolio in the next five years. Testing and assurance approaches are evolving to match the need for:

- » **Increased speed of business change.** Communication channels such as the web and social media have sped up information consumption and exchanges. The speed with which information becomes available and is disseminated and consumed means that organizations need to respond to urgent matters quickly or risk loss of reputation or business. IDC research shows that 53% of businesses need to respond to urgent events within 48 hours. Organizations thus need to possess measures and mechanisms that enable rapid response and automatically (and predictively) detect application runtime dangers to ensure application availability in times of crisis and unexpected need.

- » **Increased time-to-value expectations for application development, testing, and deployment.** Application deployment cycles have shortened. IDC research shows that more than 50% of organizations expect their service provider to deploy a mobile application within 10 days or less. Using DevOps delivery for application development can shorten development cycles, but application delivery teams must have their testing cycles in tune with development and deployment to ensure testing tasks execute at speed without sacrificing quality.
- » **Increased expectations on application return on investment (ROI).** A significant portion of an application's value is rooted in the productivity gains the application generates for users. IDC's research shows that nearly 50% of organizations expect their mobile applications to generate a 10–25% ROI over a five-year period. Organizations expect that capital used for application testing and deployment will generate business value and not simply be a use of cash for the process. Application delivery teams must ensure testing activities generate value to uphold ROI expectations even when DevOps release frequency increases, testing scope expands, and testing cycle time shrinks.

FIGURE 1: *Application Delivery Evolves*

**Q** *What percentage of your company's total business application portfolio would you estimate is being managed under a DevOps methodology today, and what would you estimate that percentage to be in the next five years?*



n = 501

Base = all respondents

Note: Data is weighted by employment size.

Source: IDC's Application Services Survey, 4Q18

## Progressive Shifts in Application Life-Cycle Management Spawn Challenges

Use of DevOps has forced application delivery teams to shorten cycle times for application deployment. Condensed cycles coupled with elevated expectations for faster application deployment mean that testing teams must maintain (or expand) testing scope while speeding up testing execution or risk:

- » **Increased development and testing costs.** Less time to execute testing tasks can result in teams shrinking the number of tests they perform to deliver applications more quickly into production. However, running fewer test cases can expose applications to vulnerabilities and cause post-deployment rework that can be more costly than fixing the problem prior to deployment.
- » **Insufficient test coverage and increased business and reputational risks.** DevOps can drive the frequency of application releases higher. A tactic to address increased release frequency is to reduce test case coverage to speed up testing cycles. However, reducing testing scope and failing to ensure coverage across all necessary testing areas can lead to a greater likelihood of application defects and increased risk. As a result, overall application costs can be driven higher because of application rework and reputational harm for production defects.
- » **Organizational fragmentation and ineffectiveness.** DevOps is an organizational change management and cultural shift initiative. While fundamental processes of design, build, and run still hold true for any type of application life-cycle management, DevOps delivery requires IT organizations to approach and execute delivery differently. Organizational structures must have tighter coordination between delivery phases and greater collaboration and transparency across IT and line-of-business personnel. Switching to DevOps methodologies without the right organizational structure, business processes, leadership, and technology can yield higher levels of ineffectiveness in application delivery.

## Advanced Automation Is Critical to Circumvent Challenges

Although more progressive methods for application delivery have increased challenges for application testing, application delivery teams can overcome those hurdles by evolving their testing approaches. Delivery teams can rise above testing challenges through:

- » **Developing smarter testing strategies.** Application delivery teams need to employ more progressive testing strategies to optimize application delivery. Testing smarter ensures that coverage of major impact test cases isn't affected. Additionally, understanding where new development work has direct (and indirect) linkages to other facets of deployment can help application teams surface risks, understand which testing tasks may be superfluous, assess and triage critical areas that must be tested manually, and determine which tests can leverage automation to drive value.
- » **Driving more predictability with testing.** Traditional testing approaches can prevent defects from being released into production, but they can be limited in effectiveness and value. Using automation instead of manual testing enables organizations to focus on value-added operational work such as improving the testing process to understand the impact of code changes and developing predictability in how application code changes affect production applications before tests are run. Employing higher levels of automation around test execution and other routines in application testing exercises helps application delivery teams focus more on value-added work such as test strategy development, defect predictability and prevention, and impact analysis.

- » **Leveraging machine learning (ML) to enhance current automation assets.** IDC research shows that organizations estimate 37% of their testing activities are automated, with nearly 50% of testing activities to be automated in three years. Introducing ML tools for code reviews, regression testing, and performance testing lets organizations execute testing tasks faster and identify systemic issues with code development that could have gone unnoticed. ML used on top of testing automation also helps organizations become more predictive in areas where certain tests drive more value, such as security testing. It also streamlines test script management and maintenance.

## Benefits of Testing Automation

Advancing testing automation levels yields benefits that include:

- » **Faster testing cycle times.** Incorporating higher levels of automation and intelligence within application testing enables application delivery teams to proactively prevent defects versus finding and fixing them. High automation levels combined with analytics that spot code issues before tests are run and identify high-priority testing tasks can also help improve upstream development activities, enhance resource utilization, and avoid unnecessary costs throughout application delivery.
- » **Increased innovation.** Enhancing existing automation capabilities can refocus quality assurance and testing resources on more value-added work and innovation and away from manual test case execution, monitoring, and defect detection. Shifting the role of testers from test execution and monitoring to test strategy development and business risk mitigation helps application delivery teams drive more value for their organizations and fosters a culture of innovation.
- » **Cost savings and improved ROI.** Through intelligence and automation, organizations can better target and speed up the cycle time for executing the full range of application testing tasks. This has a twofold effect. First, leveraging automation for testing saves on labor costs by redirecting tasks traditionally done by labor to automated execution. Second, labor can be assigned to other testing and application life-cycle management areas that require deeper levels of business logic and industry expertise to drive higher resource productivity. By applying testing automation to a larger portion of testing areas and more deeply in the application portfolio, organizations can generate significant cost savings over time to achieve higher cash flows and boost ROI.

## Trends

Key trends emerging in application life-cycle management are creating opportunities to test in a more holistic and streamlined manner. IDC has observed that:

- » **Application portfolios are expanding, and testing speed is critical to success.** Application portfolios are growing and forcing application delivery teams to change. According to IDC's *Application Services Survey*, conducted in 4Q18, 16% of organizations estimate having 250 or more applications in their portfolios. In five years, 26% of organizations expect their portfolios to have 250 or more applications. IDC research also finds an increasing intent to remove organizational silos through increased application integration across varying application types. Organizations need better and faster ways to execute testing tasks to support the increased testing scope as well as higher levels of integration across more applications.

- » **Artificial intelligence (AI) systems are being deployed to derive more insights from data and enhance testing speed.** Digital technologies such as the Internet of Things (IoT) create new opportunities and challenges around aggregating and managing the data they generate. This data provides insight into how to improve business operations and customer experiences. Organizations are leaning on AI tools with ML capabilities to unearth insights. IDC research shows that nearly 45% of organizations have deployed or plan to deploy cognitive systems within their organizations to drive stronger operations, improve top-line growth, and enhance customer experiences.
- » **Advanced automation is no longer just an option; it's a competitive advantage.** Use of containers and microservices further unlocks DevOps capabilities within organizations. Containers and container orchestration streamline and automate code delivery and help achieve faster time to market for new application functionality through code use and reuse. IDC research finds that 78% of early DevOps adopters have invested or plan to invest in containers and container orchestration to streamline and drive higher automation levels in their application delivery. While organizations are excited about the speed that automation can bring to code delivery, they must be aware of the implications that faster code builds and an increasing volume of code builds will have on their testing and quality assurance teams and activities.

## Considering Tech Mahindra

Tech Mahindra's AutoMaTics is a platform that helps application delivery teams build upon existing automation assets and drive higher value out of their testing and quality assurance capabilities. The platform is a scriptless solution that supports automation for user interface (UI) testing, application programming interface (API) testing, microservices testing, mobile device testing, and connected device testing. AutoMaTics can convert existing manual testers into automation testers with a short learning curve to transform the test automation practice. Tech Mahindra claims that in some large accounts, up to 85% of test cases have been automated by manual testers with a shorter learning curve. AutoMaTics integration with model-based testing tools such as Conformiq makes it a unique solution in the testing automation space and brings higher ROI.

AutoMaTics' scriptless functions allow users to work with a thin-client graphical user interface (GUI) to compose flexible testing scripts. No code writing is required to perform automation. The solution is based on open source technologies. AutoMaTics core is Selenium based, so it can deploy in a wide range of applications and technology environments supported by Java and Selenium. AutoMaTics contains a library of scripts and operations that enables users to quickly set up and extend existing test automation in their application testing activities.

The primary benefits of AutoMaTics center on its more than 5,000 embedded test cases and the time savings it provides on script maintenance. Testing teams can leverage the platform's automated capabilities to refactor test scripts from release to release instead of using testers to perform that maintenance. Modification of test scripts is executed with the user interface instead of coding scripts. UI manipulation for test case enhancement eases the maintenance work required, saving test engineers significant time in updating code for test scripts. Dashboards provide test engineers and managers alike with analytics on test execution status, testing trends, and test case performance.

AutoMaTics' centralized automation enables testing teams to increase reusability of test cases and scripts and reduces the labor required to maintain test scripts and bug fixes. The platform's "flexi system" enables automation for both regression and progression systems, which speeds up regression cycles and reduces costs to maintain regression scripts when code changes occur for application enhancements. AutoMaTics integrates seamlessly with an organization's existing testing tools and assets. Toolsets such as Rally, Cucumber, JUnit, Apache Maven, and Jenkins are easily integrated to enable application delivery teams to strengthen existing automation in their DevSecOps, continuous integration/continuous delivery (CI/CD) pipelines.

The platform is already in use for various clients and is ideal for businesses that need stronger testing automation coverage for web and mobile testing as well as API and microservices testing. Tech Mahindra asserts that clients that utilize AutoMaTics have been able to drive 30% cost savings in their test design and test execution because of the solution's seamless integration, pre-integrated test workflow, defect tracking, and target device execution and orchestration capabilities. The company says that AutoMaTics clients have reduced script maintenance costs by 50% by leveraging the platform to execute tests and enabling skilled testing personnel to focus on test design and strategy. An added benefit is the enhanced quality that AutoMaTics generates through reinforced test case reusability and clean and consistent code generation for test scripts.

### Challenges

Rapid changes in business and technology environments are imposing greater pressures on service providers to perform exceptional service delivery, and client expectation levels on application services performance have risen. IDC research finds that application environments for development, testing, and production are becoming more complex. Highly federated and convoluted infrastructure environments — extending from on-premise to host-based and hybrid clouds and edge computing — create new challenges. Service providers not only must ensure application functionality and performance amid varied hosting and infrastructure environments but also must be equipped to support security testing and protect against security threats and weaknesses that more complex infrastructure and hosting environments may pose. Providers that continually invest and reinvest in their solution offerings to span a wide range of testing disciplines and that possess deep, integrated expertise through the application stack will gain competitive advantages against their rivals.

### Conclusion

Advancing testing automation as part of application life-cycle management helps organizations expand business capabilities and unlock value. By bolstering existing testing automation, application delivery teams are better positioned to reduce application and related business risks, and they can ensure application ROI by getting applications into the hands of users more quickly. IDC believes that advancing testing automation as part of application life-cycle management will grow in importance over the next several years as organizations seek competitive advantage. We believe organizations should:

Advancing testing automation as part of application life-cycle management helps organizations expand business capabilities and unlock value.

- » **Define clear and measurable goals and objectives.** Organizations must outline specifically what testing automation will and will not bring to their application delivery activities. They shouldn't limit benefits to testing; instead, they should link benefits to other aspects of application life-cycle management. Organizations should use automation goals and objectives as the foundation for how they will succeed with implementing comprehensive quality assurance across application life-cycle management.
- » **Assess the existing state of automation within application testing.** Many organizations have "pockets" of automation within their application testing but fail to link automation to other facets of application life-cycle management. It's important to get a macrolevel understanding of where automation can be applied within key testing functions and adjacent application life-cycle functions to perceive maturity levels as well as to spot opportunities where automation and continuous application testing can drive more value in other areas of application life-cycle management.

- » **Develop a governance and overarching performance monitoring model.** Even with increased levels of testing automation, organizations still need to develop a governance and oversight model to monitor performance, explore further automation, and uncover deeper adjacencies to other application life-cycle management areas. While the goal of testing automation is to enhance efficiencies, accelerate application testing cycles, and eliminate overhead and excess costs, organizations still must develop escalation paths and define measures of success to ensure testing activities and tools provide value. They should establish a set of resources that will guide, direct, and manage the program with business involvement and representation to better facilitate big wins and operational success.

**About the analyst:*****Peter Marston, Research Director, Application Life-Cycle and Cloud Services***

Pete Marston is Research Director for IDC, responsible for the Application Life-Cycle and Cloud Services practice. He develops research focused on the life cycle of application services markets, which include custom application development (CAD), testing, application management (AM), also referred to as ADM (application development and maintenance), and hosted application management (HAM).

 **IDC Custom Solutions**

**IDC Corporate USA**  
5 Speen Street  
Framingham, MA 01701, USA  
T 508.872.8200  
F 508.935.4015  
Twitter @IDC  
idc-insights-community.com  
www.idc.com

**This publication was produced by IDC Custom Solutions.** The opinion, analysis, and research results presented herein are drawn from more detailed research and analysis independently conducted and published by IDC, unless specific vendor sponsorship is noted. IDC Custom Solutions makes IDC content available in a wide range of formats for distribution by various companies. A license to distribute IDC content does not imply endorsement of or opinion about the licensee.

External Publication of IDC Information and Data — Any IDC information that is to be used in advertising, press releases, or promotional materials requires prior written approval from the appropriate IDC Vice President or Country Manager. A draft of the proposed document should accompany any such request. IDC reserves the right to deny approval of external usage for any reason.

Copyright 2019 IDC. Reproduction without written permission is completely forbidden.