IDC TECHNOLOGY SPOTLIGHT

# Using Predictive Analytics and Advanced Test Automation to Boost Business Assurance

*January 2018*

By Peter Marston

Brought to you by Infogain, Powered by IDC

*Expectations on delivering business value from application testing and assurance have increased. Application development and deployment teams face increasing pressures to deliver value faster amid evolving business needs, application enhancement requests, and business requirement changes. As a result, Agile and DevOps application delivery methodologies have emerged as methods to help organizations address such dynamics. However, while adoption of Agile and DevOps has increased, many organizations face challenges of aligning and optimizing their testing resources and struggle with shortening test cycle times to drive greater value for application delivery. Application development and deployment teams not only are tasked with delivering assurance faster but also need to be more intelligent about what they test and possess sound cognitive abilities that predict and prevent application issues from surfacing before applications are even tested. This paper examines the use of automation within assurance organizations and also explores the role Infogain's business assurance solutions can play in aiding organizations with enhancing and streamlining their application testing and quality assurance (QA) activities.*

## Introduction

### *Progressive Application Delivery Is Spreading into the Enterprise*

In the past, many organizations relied on testing and assurance personnel and software to help ensure application quality and drive business value. Such traditional testing approaches leveraged human resources to execute massive amounts of application testing tasks manually with the assistance of software technologies. However, while such conventional approaches for application testing helped with raising application assurance levels and reducing the likelihood of application defects being released into production, they tended to be time consuming, make it difficult to prove ROI, and be fractured in their ability to proactively, cognitively, and predictively address and eliminate application testing issues before they arose. IDC has found that organizations are facing new dynamics with application delivery, and those dynamics are forcing organizations to evolve their approaches to testing and assurance. In fact, organizations are driving more Agile and DevOps delivery into their application portfolios (see Figure 1), and those approaches to testing and assurance are being shaped by:

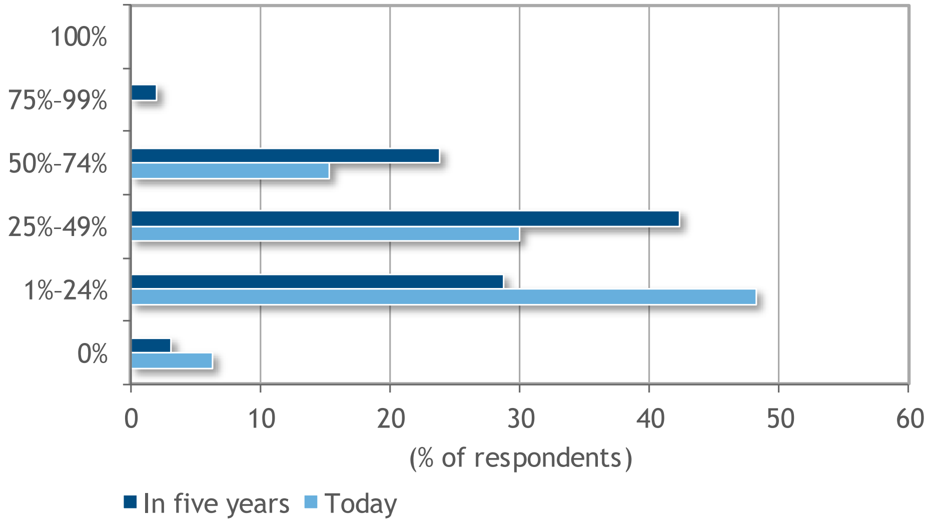
- **Increased speed of business change.** Newer communication channels such as the web and social media are speeding up information consumption and exchanges. The speed by which information becomes available, disseminated, and consumed means that organizations need to be able to respond to urgent matters quickly or risk damage to their reputation or loss of business. Organizations face increased pressure to respond to unplanned business needs, and according to IDC research, 53% of businesses need to respond to urgent events within a 48-hour time period. This means that organizations need to possess not only measures and mechanisms that enable rapid response but also methods to automatically and predictively detect application runtime dangers to ensure that customer-facing or internal business applications are always available and running to support users in times of unexpected need.

- **Heightened time-to-value expectations for application deployment.** Organizations face increasing pressures for the speed by which full application deployment cycles can be completed. Using mobile applications as an example, a recent IDC application services survey revealed that more than 50% of organizations would expect their service provider to deploy a mobile application within 10 days or less. This means that business expectations are growing more intolerant of application deployment delay. Organizations must have rapid testing cycles to ensure applications are available, and up and running quickly, or risk detrimental impacts on business continuity.
- **Increased expectations on application ROI.** Organizations view application investments with increased expectations. Much of an application's value is rooted in the productivity gains the application generates for its users, and when an application is being serviced — whether for patches, enhancements, testing, or upgrades — application downtime due to testing and maintenance directly affects user productivity levels and resulting cash flows drawn from productivity. Using mobile applications as an example, IDC's research has shown that nearly 50% of organizations would expect their mobile applications to generate a 10–25% ROI over a five-year period. This means that organizations expect capital used for application testing and deployment to generate business value and not simply be a use of cash. Cash flow benefits (e.g., productivity gains, cost savings, revenue generation) must be delivered.

## FIGURE 1

### Percentage of Application Portfolio Managed Under DevOps (Today and in Five Years)

*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 = 403

Base = all respondents

Notes:

The survey was managed by IDC's Quantitative Research Group.

Data is weighted by industry by employee size.

Use caution when interpreting small sample sizes.

Source: *Application Services Survey,* 2017

### *Progressive Shifts in Application Delivery Spawn Increased Challenges*

As organizations are advancing their application deployment methods to match evolving business climates and needs, QA organizations face increased pressures. The increased adoption of DevOps and Agile forces QA to shorten testing cycles. Shortened cycles and increased expectations on faster application deployment mean that QA must devise ways to protect and improve test case scope while speeding up testing cycles or risk:

- **Rising development and application maintenance costs.** Shortened time to execute testing tasks can result in QA organizations shrinking the number of tests they do to respond to needs to release applications more quickly into production. Reducing the number of test cases that are run not only can expose applications to vulnerabilities but also can cause post-deployment rework that can be more costly to fix post-deployment versus pre-deployment.
- **Insufficient test coverage.** While the pace of application delivery demands has accelerated, QA organizations still need to ensure proper test case coverage across areas such as security testing, functional testing, usability testing, integration testing, regression testing, and performance testing. Failure to ensure all necessary testing areas are covered not only can lead to unrealized application value and drive overall application costs higher but also, in cases of data security, can damage organizations' brand reputation and bring about legal ramifications.
- **Bad decisions based on disparate tools and fragmented information.** Adding more DevOps tools to the application portfolio may assist organizations with supporting certain elements of continuous development and integration, but a lack of cohesiveness across the tools and full DevOps processes can lead to error-prone decisions. The inability to integrate tools along the DevOps process value chain can hinder an organization's ability to see holistic operational performance across all facets of DevOps processes and mask root causes of defects found in the testing cycle. As a result, organizations may make misguided decisions for defect remedies that don't end up resolving core issues.

### *Intelligence and Automation Are Increasingly Critical to Circumvent Challenges*

Although more progressive methods for application delivery have created new challenges for application testing, QA organizations can overcome those hurdles by evolving their approaches to testing, gaining more insights into what they test, and implementing automation in key parts of the testing process. More specifically, organizations can rise above testing challenges through:

- **Smarter testing and test case prioritization.** QA organizations need to employ more progressive strategies to be prepared for DevOps delivery. Being smarter about testing helps ensure 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 testing and deployment can help organizations surface risks as well as understand testing tasks that may be unnecessary. By approaching quality assurance in smarter ways, organizations can better assess and triage critical areas that must be tested manually and those that can leverage heavy automation.
- **Driving more predictability with testing.** Traditional approaches to application testing have centered on QA being the last step in a process to detect defects before code is released into production. While such approaches often helped with reducing the number of defects released into production, the approaches were flawed with helping organizations target and optimize the overall application life cycle. Implementing tools and processes that assist with spotlighting key linkages of testing to previous development phases (e.g., requirements definition, creative design, and development) can aid organizations with driving predictability in testing and help enable a mentality shift from QA being used as a last mechanism for defect detection to being a source of defect prevention.

- **Leveraging automation.** Applying automation to testing that goes beyond test script execution and test case development can further aid organizations with driving more value from their quality assurance program. Leveraging machine learning tools for areas such as regression testing and performance testing can help organizations not only execute testing tasks faster but also identify systemic issues with code development that may have gone unnoticed before. Automation can also help organizations become more predictive in areas where certain tests drive more value and enable them to be more adaptable to fluid business conditions.

## The Benefits of Smarter Testing and Automation

Use of predictive analytics and automation tied to application testing yields benefits that include:

- **Increased agility, flexibility, and productivity.** Incorporating intelligence and automation capabilities within application testing can aid with building agility and flexibility by freeing up IT resources to focus on higher value-added initiatives. By repurposing resources, the IT organization is afforded higher levels of agility and flexibility to deploy resources more effectively and efficiently within the organization where, in turn, higher levels of productivity can be generated by those resources.
- **Cost savings and improved ROI.** Incorporating intelligence and automation capabilities within application testing can help reduce costs and improve application testing ROI. Through intelligence and automation, organizations can better target and speed up the cycle time it takes to execute application testing tasks. This has a twofold effect. First, leveraging automation for testing can help organizations save on labor costs by redirecting tasks that were traditionally done by human resources and moving them to automated execution. Second, organizations can redirect labor to other testing and application life-cycle management areas to drive higher levels of resource productivity. Applying testing automation to a large scale of applications and testing areas, organizations can be positioned to generate significant cost savings over time, which in turn can generate higher cash flows and boost ROI.
- **Intelligent business insights and predictive issue avoidance.** Incorporating analytics capabilities within application testing can aid organizations with proactively identifying and preventing application defects versus finding and fixing them. Possessing the right analytics that spot code issues before tests are run or identify high-priority testing tasks that circumvent application risks can help organizations improve upstream development activities, enhance resource utilization, and avert unnecessary costs throughout the overall process of application delivery.

## Key Application Portfolio Trends

Over the years, IDC has witnessed several trends within the application services arena that are causing organizations to evolve how they approach application testing as well as their overall application life-cycle management. Most recently IDC has seen that:

- **App portfolios are growing and forcing quality assurance organizations to evolve.** According to IDC's *Application Services Survey,* conducted in 2Q17, 19% of organizations estimate that they have 250 or more applications in their portfolios today, while 43% of organizations expect their portfolios to have 250 or more applications in five years. In addition to the growth in application portfolios, IDC research has shown an increasing intent to remove organizational silos through increased application integration across varying application types. This means that organizations need to find better and faster ways to execute testing tasks to support the massive amounts of integration that are happening now and will continue to happen in the future.

- **Use of open source is growing, and testing needs have become more expansive.** IDC has found that organizations are leveraging open source more within their application portfolios. IDC's 2017 *Application Services Survey* found that, on average, a fifth of the application portfolio within organizations is custom developed on open source today. In five years, organizations expect that percentage to move up to one-third of their application portfolio. This means that further progression in using open source will force organizations to approach application testing more dynamically and devise new methods and support for regression testing, back-end code packaging, and unit/functional testing.
- **Automation is taking a greater foothold.** As organizations add more applications and extend the use of open source within their application portfolios, they seek to drive more automation and cognitive computing as part of various aspects of their business and application life-cycle management. IDC's 2016 *Application Services Survey* found that nearly 20% of organizations rated implementing custom cognitive applications within the organization's application portfolio as a high or very high priority today, and more than 50% rated it as a high or very high priority in the next five years. This means that as organizations increase the size of their application portfolios, they will continue to seek areas to drive higher levels of automation within business and IT processes to fend off rising costs and develop competitive advantages.

## Considering Infogain Business Assurance Services (UAP and PAQ)

Infogain's business assurance services help organizations drive higher value out of their testing and quality assurance capabilities. The business unit is part of a larger IT services firm based in Los Gatos, California. Of Infogain's 4,000 total employees, roughly 500 workers are dedicated to its business assurance services unit, which services more than 100 clients.

A key point of Infogain's business assurance services differentiation is its focus on value creation. Unlike competitors' offerings on application testing services that center services on late cycle regressive testing and "gut feel"–based test coverage, Infogain's business assurance services center on early cycle progressive testing and intelligence-based test coverage. To achieve this shift in approach, Infogain has developed two key solutions: Unified Automation Platform (UAP) and Predictive Analytics for Quality (PAQ).

Infogain's UAP consists of services and intellectual property that provide an efficient, flexible automated quality solution that helps enable testing early in the software development life cycle (SDLC). It delivers ROI through its usage of open source tools, high test asset reuse, and an ability to unify clients' existing automation tools and frameworks in a single, end-to-end platform. UAP is aimed at organizations that utilize Agile development for application delivery and supports continuous integration and continuous delivery (CI/CD) and DevOps frameworks. The solution is meant to aid client organizations with testing as rapidly and efficiently as possible, helping clients target additional productivity gains of 30% through automation and more efficient processes.

Infogain's PAQ solution is designed to bring analytical intelligence into the testing planning process. By leveraging data generated in the software development process, PAQ's suite of open source data management and statistical analysis tools apply machine learning-driven predictive intelligence to the test planning process by highlighting potential and likely points of failure in the system under test and provide a risk feedback loop throughout the SDLC. The solution is meant to aid client organizations with identifying the right tests to be as effective as possible.

Infogain's UAP and PAQ combine to form a Continuous Quality Engine (CQE) that enables organizations to quickly and predictively prioritize tests for automation and optimize test scheduling. At its foundation, the CQE employs a predictive analytics and machine learning engine that detects patterns in software development data, such as user stories, test data, and defect data, and post-application release data, such as error logs, defect logs, and multichannel customer feedback information. The assembly of information feeds into the automated test tools, framework, and repository for the UAP, which, in turn, churns out an optimized, automated test plan. Once tests are run, data is fed back into the analytics engine and machine learning is applied to drive more intelligent test plans and execution for the next sprint cycles in the Agile development process.

Business assurance services and the UAP were able to generate more than a 30% productivity gain in test automation scripts, reducing test execution time for sanity and regression suites by 60%. These benefits were the direct result of Infogain's enhanced automation solution. By leveraging existing client assets, such as test automation scripts, page objects, and web and desktop function libraries, Infogain integrated its UAP to create a factory design pattern interface. Utilizing the client's function library assets, Infogain then added its own mobile, web services, and database function libraries to create a UAP reporting and logging engine. These additions, combined with Infogain's UAP test manager, execution engine, mobile device lab, and performance engine, enabled the client to establish key performance indicators that didn't exist before, such as a test stability index, test automation review effectiveness, and estimation methodology.

### *Challenges*

While Infogain provides business assurance services that can address buyer needs, the service offering faces a headwind that has arisen within worldwide application services markets. IDC research has found that services buyers are showing increased interest in consolidating their vendor management for managed services they procure. Buyers have indicated that while they do not want to consolidate their services vendors down to just one provider, they do want to condense the number of vendors they utilize to streamline vendor management activities (see *U.S. Deal Making Study of Buyer Procurement Strategies for Outsourced and Managed Services,* IDC #US41967417, August 2017) and drive down vendor management costs.

## Conclusion

Automating application testing tasks and improving intelligence around application testing helps organizations expand and unlock business capabilities. By utilizing automation and advanced testing analytics, IT organizations can be better positioned to create cash flow benefits, which in turn, can fund and support other IT initiatives. IDC believes automation within application life-cycle management will continue to grow in importance over the next several years as organizations seek to drive higher levels of speed and cost savings in application delivery. IDC believes organizations should:

- **Define clear and measurable goals and objectives.** Organizations must outline specifically what automation will and will not bring to their application testing and DevOps activities. They must use these goals and objectives as the anchoring foundation for how they intend to (and will) be successful in implementing comprehensive automation across the various facets of 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 and predictive intelligence from testing to other areas of application life-cycle management. It's important to get a macro-level understanding of automation levels within the QA and adjacent application life-cycle functions to understand maturity levels, as well as help spot opportunities where automation of application testing can further benefit other areas of application life-cycle management.

- **Explore, size, and prioritize areas of application testing to automate.** As organizations explore and inventory their business for automation opportunities in application testing, they must measure how the activities link to or may affect various lines of business. Organizations must generate estimated costs, metrics, and risks to the line-of-business users for routine application testing tasks that can be automated (such as regression testing) and more complex testing tasks (such as usability testing and security testing) that may require more human input and oversight. They need to prioritize automation areas by level of business risk and impact as well as estimated use of operational cash.

- **Develop a governance and overarching performance monitoring model.** Even with increased autonomous computing in application testing, organizations still need to develop a governance and oversight model to monitor performance and explore areas for further automation and deeper adjacencies to other application life-cycle management cycles. While the goal of testing automation is to enhance efficiencies, speed up application testing cycles, and eliminate overhead and excess costs, organizations still must develop escalation paths and define measures of success to ensure any automated solution is providing value. They must establish a set of resources that will help guide, direct, and manage the program, such as a steering or management committee, and ensure that the program has line-of-business representation and buy-in.

## ABOUT THIS PUBLICATION

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.

## COPYRIGHT AND RESTRICTIONS

For more information on IDC, visit www.idc.com. For more information on IDC Custom Solutions, visit http://www.idc.com/prodserv/custom_solutions/index.jsp.

Global Headquarters: 5 Speen Street Framingham, MA 01701 USA P.508.872.8200 F.508.935.4015 www.idc.com