



THE DZONE GUIDE TO

# Automated Testing

Improving Application Speed & Quality

VOLUME I

RESEARCH PARTNER SPOTLIGHT



# Key Research Findings

BY G. RYAN SPAIN  
PRODUCTION COORDINATOR, DZONE

## DEMOGRAPHICS

434 software professionals completed DZone's 2017 Automated Testing survey. Respondent demographics are as follows:

- 41% of respondents identify as developers or engineers, and 27% identify as developer team leads.
- The average respondent has 14 years of experience as an IT professional. 56% of respondents have 10 years of experience or more; 21% have 20 years or more.
- 40% of respondents work at companies headquartered in Europe; 34% work in companies headquartered in North America.

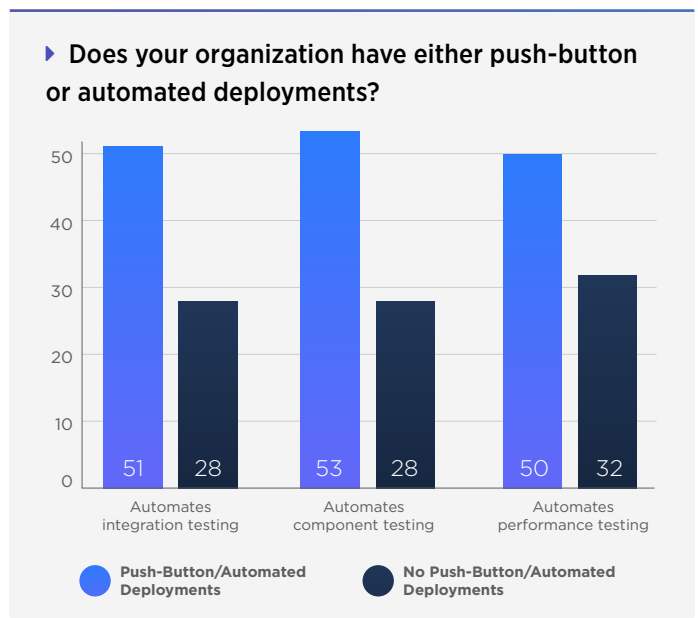
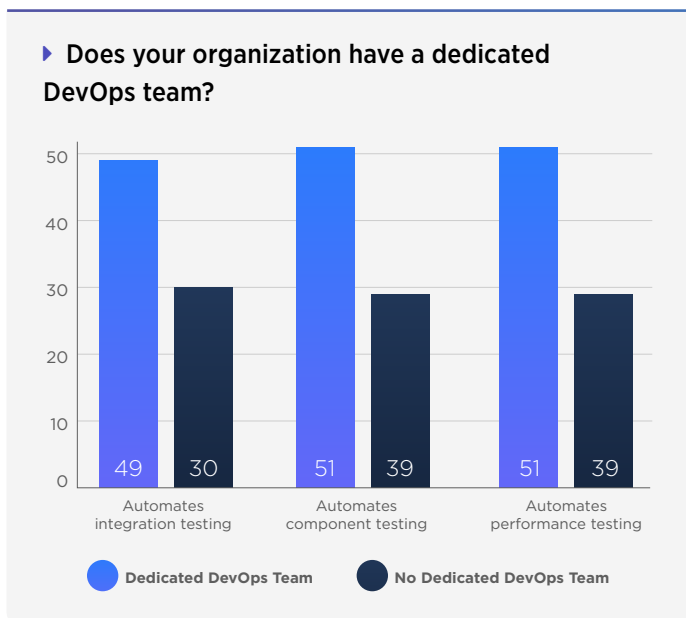
- 18% of respondents work at organizations with more than 10,000 employees; 20% work at organizations between 1,000 and 10,000 employees; and 28% work at organizations between 100 and 1,000 employees.
- 83% develop web applications or services; 49% develop enterprise business apps; and 36% develop native mobile applications.

## AUTOMATED TESTING

We asked survey respondents which tests in their organization's pipeline(s) are automated and which tests are performed manually. The most popular automated tests were integration (61%), component (58%), and performance (56%). While 22% of respondents automate none of these tests, 17% automate one of the three, 25% automate two, and 36% of respondents automate all three. For manual testing, the most common responses were user acceptance (78%), usability (70%), and story-level tests (63%). Across all manual and automated testing, manual testing had 36% more responses than automated testing. We also asked about a wide array of tools for automated testing. The most popular tools amongst our respondents were JUnit (61%), Selenium (46%), JMeter (45%), SoapUI (29%), and Cucumber (21%). 44% of respondents say their organization's Continuous Integration processes extend into an automated Continuous Delivery pipeline from code check-in to production deployment.

## DEVOPS TRENDS

It's no surprise that automated testing and other DevOps practices go hand in hand. 49% of respondents working



at organizations with dedicated DevOps teams said one of that team's goals was introducing automation across the entire SDLC. Looking at the three most popularly automated tests, we found that respondents who said their organization automated these were much more likely to have one of these dedicated DevOps teams. 49% of respondents whose org automates integration tests said they have a DevOps team, as opposed to 30% who said their org does not automate integration tests. For component tests and performance tests, the difference was 51% with dedicated DevOps teams compared to 29% of respondents at orgs not automating these tests. Respondents answering that these tests are automated were also much more likely to say their organization performs push-button or automated deployments; for example, for integration tests, this difference was 51% vs. 28%. These respondents were also significantly more likely to believe their organization has fully achieved Continuous Delivery.

**TOOLS AND AUTOMATED TESTING**

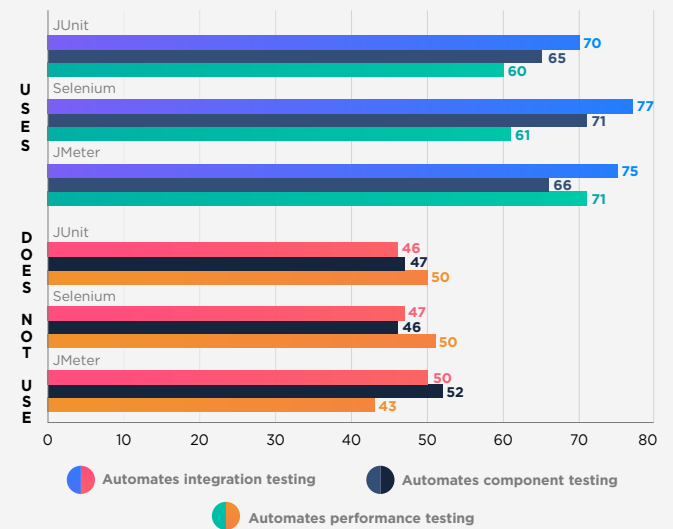
Responses regarding the most popular testing tools were also connected with these commonly automated tests. 70% of respondents whose organization uses JUnit said they automate integration tests, compared to 46% of non-JUnit users. For JMeter this difference was 75% vs. 50%, and for Selenium it was 77% vs. 47%. These trends apply to component and performance testing as well. Performance tests, while not as dramatically different as the others for users of JUnit and Selenium, were automated by 71% of JMeter-users, vs. 43% of non-users. Considering it was more likely for respondents to automate more than one of these popular tests, even starting test automation in one area seems to have an impact on other tests.

**TOOLS AND LANGUAGES**

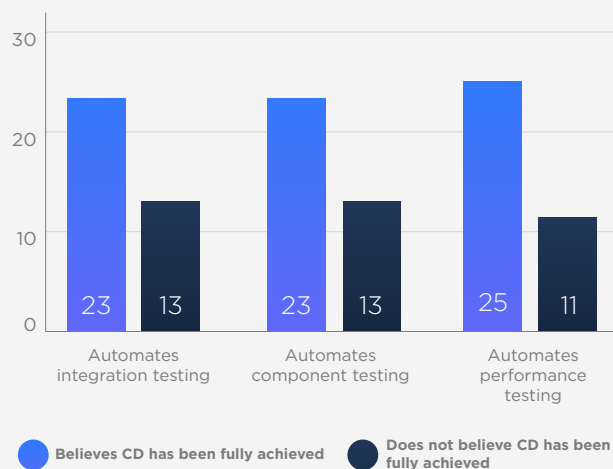
Given how language-specific testing tools are, the popularity of JUnit and JMeter amongst our respondents makes sense. 86% of respondents work at an organization that uses the Java ecosystem, and 62% of respondents work at organizations where Java is the primary programming language. 68% of respondents working at an organization that uses Java at all said their org uses JUnit, and 50% said they use JMeter. Of the respondents who work at primarily Java organizations, 77% said they use JUnit, and 53% said they are using JMeter. So these two open source testing tools are taking hold in the Java world.

*\*Margin of error calculated with 95% confidence interval*

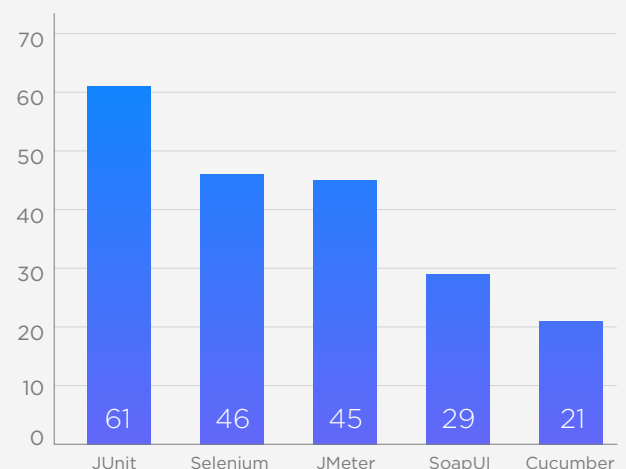
► Does your organization automate integration, component, or performance testing?



► Do you believe continuous delivery has been achieved in your organization?



► Which of the following testing automation tools does your organization use?



# THE CROSSROADS OF TESTING

Automated testing is seen as one of the key components of achieving Continuous Delivery in an organization. Thought leaders often suggest to automate everything. However, as of right now, not everything can, or should, be automated, particularly tests that rely on real-user input. So, developers stand at a crossroads: which tests can be automated now, and why? We asked almost 400 developers about which tests they performed manually and which were performed automatically.

## MANUAL TESTING

### USER ACCEPTANCE TESTS

**TESTS** ensure that a user is satisfied with a product's function. These can be used to test user stories and ensure that they have been implemented correctly. Since this is based on the subjective opinions of end users, this is not conducive to automation.

**78%** of respondents perform manual user acceptance tests.

### USABILITY TESTS

These are tests that are performed directly with users to determine how easy it is to use a piece of software. Because results rely on the opinions of real users, this cannot easily be automated.

**70%** of users perform manual usability tests.

### POST-DEPLOYMENT TESTS

**TESTS** can vary between smoke checks to ensure the application is running and testing any major bugs that become apparent once the application is live. Because of the troubleshooting and unique circumstances involved, this is not an easy task to automate.

**63%** of DZone's audience perform manual post-deployment tests.

### PERFORMANCE TESTS

determine the speed or effectiveness of an application or network. Performance tests may be manually performed to determine the source of a performance bottleneck or diagnose an issue, but automated performance testing is useful to get a consistent, up-to-date picture of an application's performance. task to automate.

**55%** of users automate performance tests, while **45%** perform them manually.

## AUTOMATIC TESTING

Also called component tests, **UNIT TESTS** take individual pieces of an application's source code, called units, and ensures they are operating properly. Since these units don't rely on external dependencies, automating them is a relatively straightforward process.

**58%** of respondents perform automated unit tests.

### INTEGRATION TESTS

are performed when two pieces of software are combined and tested as a single unit. It should be an easy task to ensure that two pieces of software can communicate between each other, and can be easily automated.

**61%** of readers perform automated integration tests.

# Nine Critical Considerations for Testing Responsive Websites Using Selenium

BY CARLO CADET - DIRECTOR PRODUCT MARKETING, PERFECTO MOBILE

Responsive website design is becoming a method of choice for many organizations. Among the primary motivations for embracing responsive design are:

- Consistent user experience across all platforms
- Improve marketing results by being mobile friendly
- Lower maintenance cost

One code base across so many platforms and form factors raises the bar on quality and therefore the testing strategy.

In the below checklist, you can find the most critical testing consideration for a RWD that will ensure good UX, and sufficient test automation coverage. All of the below considerations can be automated using Selenium framework or cloud-based tools.

## 1. VISUAL VALIDATIONS

Does your website look right across all platforms like desktop browsers, smartphones, tablets, and IoT-supported devices?

Does the site look okay in various orientations, like portrait and landscape, as well as in various languages?

## 2. ENVIRONMENT CONDITIONS

Validate performance across all expected conditions. Factor in variables such as incoming events, background apps, location services, and changing network conditions (Data, Wi-Fi, Airplane mode, etc.)

## 3. NAVIGATION

Validate CSS breakpoints across different form factors and orientations. When the site is launched across these displays, the navigation and the content of what is being displayed to the user changes (above the fold and beyond the fold content, hamburger menus, etc.).

## 4. PLATFORM COVERAGE

Analyze web traffic to determine coverage strategy, identifying the mandatory platforms and OS versions to be tested throughout the SDLC.

## 5. ACCESSIBILITY COMPLIANCE

Assess compliance with accessibility requirements across the market you serve. Using tools like WAVE that can be integrated into your Selenium scripts or be a stand-alone tool for your test engineers, is a good choice (out of few others) to adopt.

## 6. PERFORMANCE

Performance optimization is key, especially considering Google's recent prioritization of mobile-friendly sites. Performance testing for source and data loading, caching controls, and functional scenarios are proving to be effective tools for achieving critical performance gains.

## 7. LOCALIZATION

Validate location services scenarios. Design scenarios for both location specific data and the "traveling user".

## 8. SECURITY

Personalization strategies are driving an increasing quantity of private user data process by sites. Add data privacy scenarios to test suites. This includes authentication rules and types, cleaning of private data upon session termination.

## 9. DON'T FORGET QUALITY ANALYSIS & VISIBILITY

Testing a RWD site across multiple platforms, means, dealing with large amount of test data. Having a quality dashboard after each test automation execution that is tag-driven for easy filtering enables data-driven and risk-based decisions.

# Drive Testing at the Speed of Agile

It's hard to believe that after 30 years, 70% of testing is still performed manually. A major bottleneck in the SDLC, legacy testing remains a barrier to speed and quality – unable to keep up with today's agile, continuous testing model.

But as more organizations adopt test-driven, agile development methods, they gravitate towards test automation – enabling test teams to automatically generate reusable test assets like test cases, test data, and test automation scripts right from requirements.

Model-based testing (MBT) helps you avoid costly defects that stem from poor requirements. It enables you to automate testing activities, shortening testing time dramatically. And so, your high-quality apps are delivered faster at lower costs.

The question then becomes, in what ways does MBT drive test automation effectiveness?

Start with modeling requirements as an active flowchart, versus writing them in inefficient text-based methods. With 64% of defects coming from poorly-defined requirements, modeling using a flowchart eliminates unclear requirements – while boosting collaboration and communication.

Next, generate optimized sets of test cases automatically. This means creating test cases, test data, and test scripts automatically, right from the flowchart as user stories are created, and testing the functionality at maximum coverage with the smallest set of tests.

Finally, automate the 'change in requirements process.' This cuts the time wasted on manually finding and fixing tests when requirements change, because as changes occur they automatically initiate impact analyses and create or repair tests to maintain test coverage – while building up a library of reusable test assets that can be run or rerun as test automation artifacts.

CA offers comprehensive solutions that automates the most difficult testing activities – from requirements engineering through test design automation and optimization. These capabilities help you test at the speed of agile, enabling you to build better apps, faster.



**WRITTEN BY GEDEON HOMBREBUENO**

PRINCIPAL PRODUCT MARKETING MANAGER, CA TECHNOLOGIES

## PARTNER SPOTLIGHT

# CA Agile Requirements Designer



Improve software quality, reduce testing costs, and speed up application delivery.

### CATEGORY

Continuous Testing

### NEW RELEASES

Continuous

### OPEN SOURCE

No

### STRENGTHS

- **Requirements Engineering.** Map requirements to a visual, active flowchart, and reduce requirement ambiguity by 90 percent and software defects by 56 percent.
- **Test Design Automation.** Automatically generate the smallest number of test cases needed for 100 percent functional coverage and test automation scripts, linked to the right data and expected results.
- **Enable Agile Testing.** Automatically generate test cases, test automation scripts, and test data for all functionalities being delivered in every sprint.
- **Test Case Optimization.** Import test cases, remove any duplicates, and shorten test cycles by 30 percent.
- **Manage changing requirements.** Automatically identify the impact of a change and update test cases in minutes.

### CASE STUDY

Williams is a Fortune 500 energy infrastructure company, providing natural gas processing and transportation. Based in Tulsa, Oklahoma, it employs 5,600 people through operations across the US.

To keep pace with business demands, Williams needed to deliver higher quality applications and updates more quickly and with fewer defects. Manual testing processes were hampering its ability to achieve this.

Williams implemented a suite of CA Technologies solutions to automate and improve software testing. CA Services also provided customization, education, and implementation support to provide end-to-end release management.

Williams has improved the speed and quality of software delivery, which frees up resources for new business projects, and will help the company achieve its business goals of providing the best service for less cost.

See their story [here](#).

### NOTABLE CUSTOMERS

- Williams
- GM Financial
- Sprint
- Citigroup
- Level 3 Communications

**WEBSITE** [ca.com/us/products/ca-agile-requirements-designer.html](https://ca.com/us/products/ca-agile-requirements-designer.html)

**TWITTER** @CAinc

**BLOG** [ca.com/en/blog-highlight](https://ca.com/en/blog-highlight)

# Continuous Testing

Enable continuous testing across your software delivery lifecycle.

Adopt next-generation testing practices to test early, often, automatically, and continuously.



Only CA offers a continuous testing strategy that's automated and built upon end-to-end integrations and open source.

Enable your DevOps and continuous delivery practices today.

Explore [ca.com/continuous-testing](https://ca.com/continuous-testing)

