State of Testing 2018

SMARTBEAR

Content

Preface

This survey was designed to establish benchmarks for the software testing industry regarding the methodologies, practices, and tools used by QA professionals in 2018 to build, validate, and deliver software. As our second annual survey, this report also captures emerging trends year-over-year and provides insights into what teams globally think the future will hold. The following report will cover:

- Development & Delivery Models
- Testing & Quality Assurance Practices
- API and UI Testing: Tools, Techniques, and Challenges
- Future Expectations of the Industry

Methodology

At SmartBear, we conducted a global online 45-question survey over the course of two weeks in July 2018 and collected a total of 2,676 responses. These were collected via online advertisements, social media, community forums, and in collaboration with industry influencers and SmartBear partners. You can find the collaborators at the end of the report.

An internal database of software professionals responsible for a wide variety of quality assurance and development functions were invited via email to participate in the survey. The findings presented are based upon the completed responses from over 1,700 Manual Testers, Automation Engineers, Developers, Consultants, QA Managers and Analysts from more than 16 different industries globally. A copy of this report and five $100 dollar amazon gift cards were offered as incentives for full participation.

Introduction

Software over the last two decades has revolutionized the way we live. In the 90s, it was every parent's nightmare for children to get in the car with strangers. Today, Uber Pool has defied and re-defined the wishes of parents everywhere. In fact, cars have even started to drive themselves. In the next 10 years, we could move to a fully autonomous ride hailing service.

From flip phones and floppy disks, to 3D-video games, pacemakers and the birth of artificial intelligence-based devices like Alexa, technology has advanced at an unimaginable rate. The rise in what software can accomplish has correlated with a boom in technological dependency. With software having seeped into almost every aspect of day-to-day life, the value and importance of software development and testing has skyrocketed. Gartner expects global IT spending to reach $3.7 trillion this year.

Teams world-wide are pressured to build better, test faster, and deploy quicker – all in the hopes of delivering flawless user experiences to meet market and business demands. With our flagship testing survey last year, we set out to evaluate the use of various tools, development methodologies, and testing techniques used by software professionals, and to set benchmarks to measure against for years to come.

With our second annual survey, we aimed to investigate the evolution of the industry and answer questions such as, “How fast is automated testing growing,” and, “Are release cycles actually getting shorter?” The spectrum of practices and tools available to QA professionals is almost just as broad as the products in the market and our goal was to better understand what today's trends are.

As with last year, we set out to gather feedback from the entire community. With 45 questions and over 2,600 responses from a myriad of job roles across every major industry, we were able to once again gather valuable data that successfully paints the picture of where we stand today and how software is transforming.

The year-over-year insights are fascinating. So go ahead and dig in. You'll discover more of what your peers are doing today in testing.

If you are one of the contributors that took the time to fill out the survey, thank you. You provided valuable feedback and insights and we greatly appreciate your time and efforts in completing the survey.

- The SmartBear Software Team

High-Level Takeaways

The Industry Reliance on API Testing is Growing

The Role of Testing Continues to Expand Beyond the Traditional Tester

Automation is Not Growing as Fast as We Might Expect

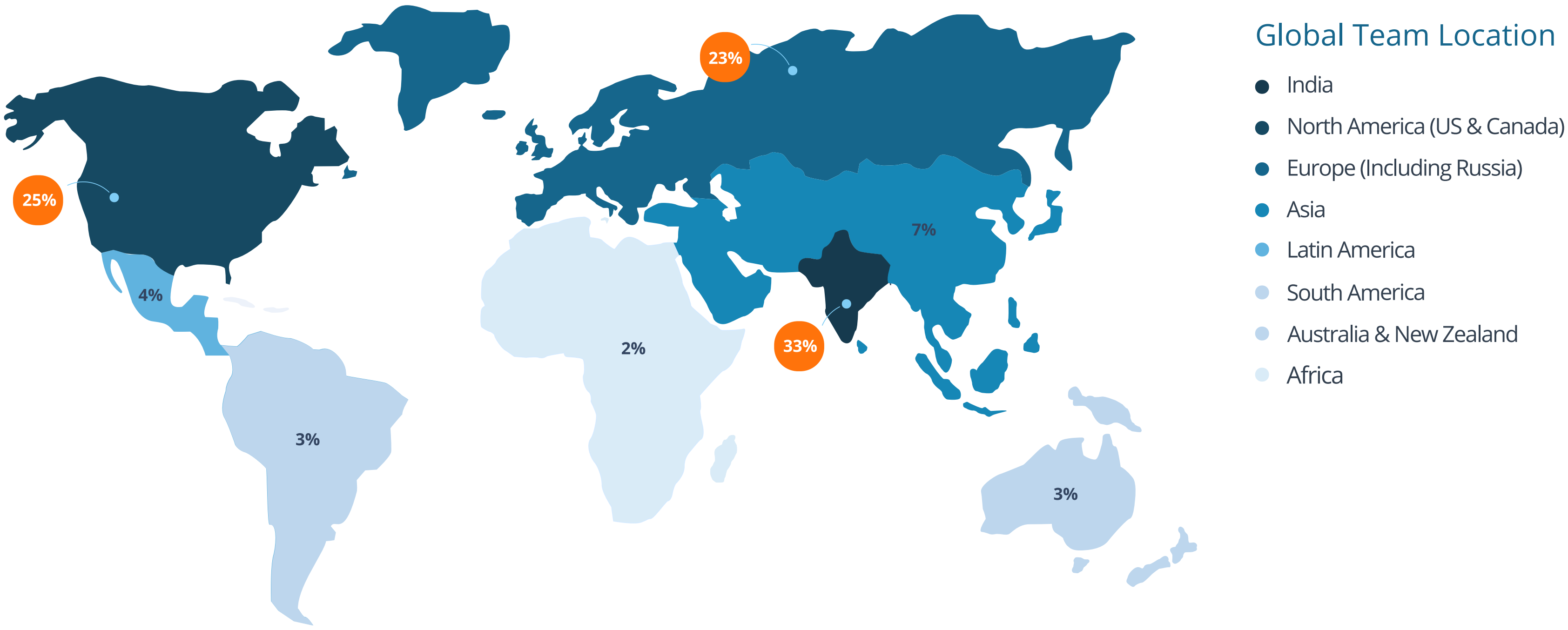
3rd-Party API Usage Has More Than Doubled

One-Quarter of Teams Are Releasing At least Once a Day

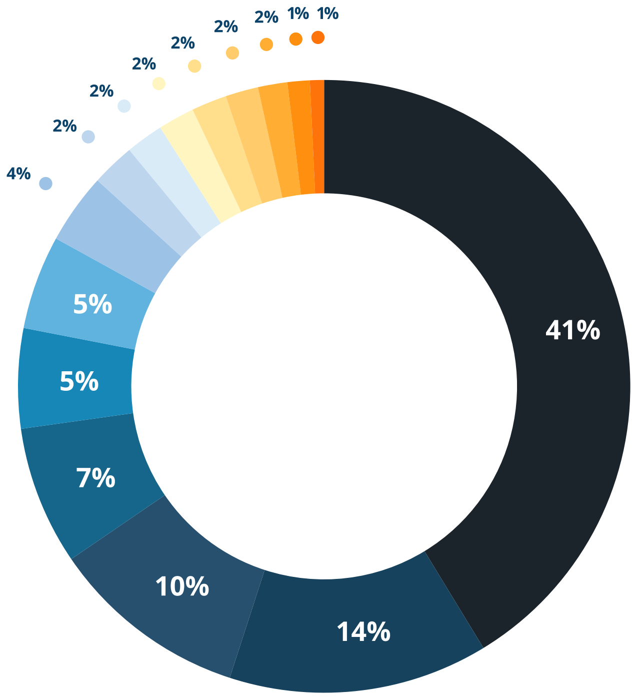
Demographic & Firmographic Data

The SmartBear 2018 State of Testing survey includes responses from over 2,600 individuals located all over the world, working across all facets of the software testing industry, from application development to quality assurance.

Thousands of individuals globally responded to the State of Testing survey.



Respondents work in over 16 different industries, from software to finance and gaming.



Industry

- ● IT/ Service (Consultants)
- ● Finance, Banking, Insurance
- ● Computer Hardware, Software, or Consumer Electronics
- ● Other
- ● Health Care Services
- ● Telecommunications, Utilities
- ● Automative, Transportation
- ● Government, Aerospace, Defense
- ● Manufacturing
- ● Media, Advertising, Communications
- ● Retail or Wholesale Trade
- ● Education
- ● Hospitality, Travel
- ● Pharmaceuticals and/ or medical devices
- ● Gaming

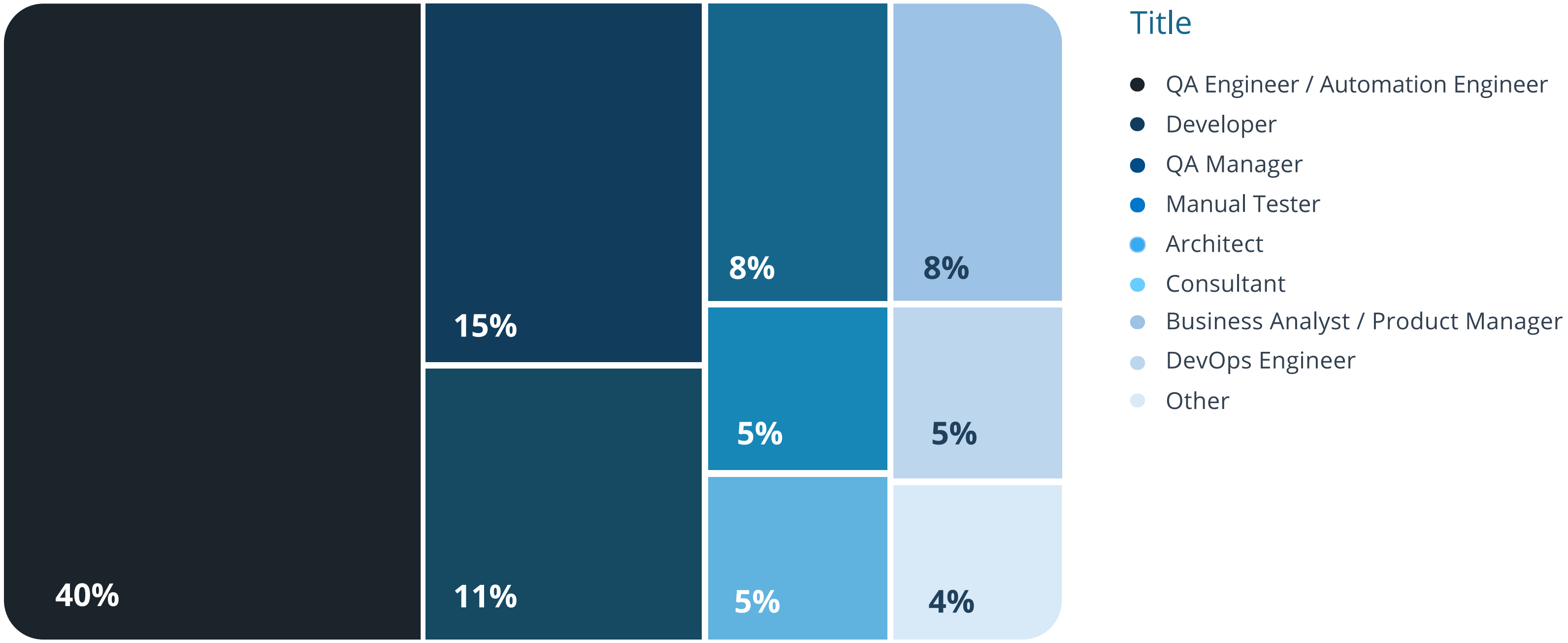
Company sizes ranged anywhere from under 25 to over 10,000 people.

Company Size

10%	1 - 25
15%	26 - 100
21%	101 - 500
12%	501-1000
19%	1001-10,000
23%	10,001+

The survey audience was comprised of a diverse set of job functions, including Automation Engineers, Manual Testers, and Developers.

The majority of respondents (40%) are QA Engineers / Automation Engineers. The 4% who responded with 'other' represent job titles including: Infrastructure Manager, Project Manager, SE Manager, Programmer, Continuous Delivery Team Lead, Devops Specialist, and those in Director level positions, or those who perform a combination of the job titles listed.



Application Details & Development Models

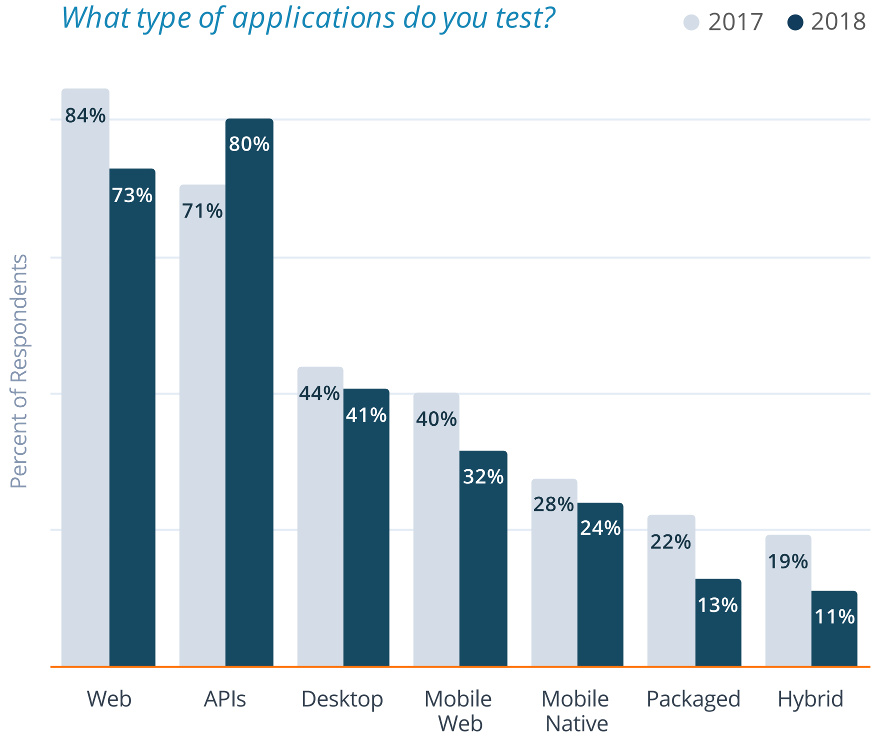
The API economy is booming. Those who reported testing APIs grew from 71% in 2017 to 80% in 2018. As the backbone of application development, APIs are the key to enabling the vast network of systems, platforms, and apps that make up the Internet of Things (IoT) to connect. Over the past few years, the number of APIs being consumed has skyrocketed. They are being integrated into more platforms than ever, so it is not surprising to see this type of growth and to expect it to continue.

While those testing other application types, including web and mobile, dropped year-over-year, the total number of respondents testing *more than one application type fell as well.* In 2017, 15% of the survey audience reported testing multiple application types, as compared to 21% this year, indicating QA professionals are narrowing their testing efforts to only one application type.

Six-percent (6%) of respondents reported testing two other application types, which were new answer choices included in this year's report.

- Mainframe applications - 6%
- Embedded applications - 6%

What type of applications do you test?

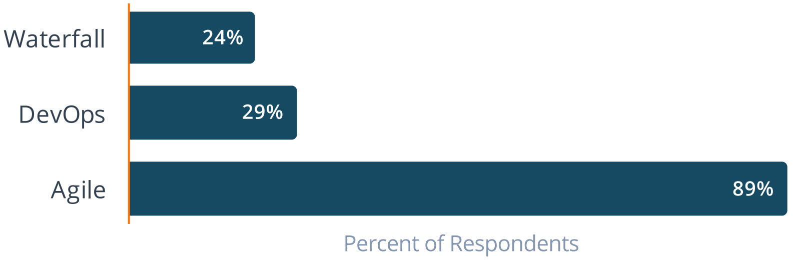


The Waterfall model is still used by almost a quarter of QA professionals.

While Agile is the dominant development model used by quality assurance teams (89%), just under a quarter are still using the Waterfall model (24%). It's important to note, that respondents were asked to select all models they follow. Of those who answered this question, 63% reported following only one model, while 32% selected two, the most popular combination being Agile and DevOps.

The concepts behind Agile and DevOps are similar. Both models are indicative of a fast, flexible, and iterative development lifecycle, but DevOps places a higher focus on team collaboration and Agile focuses on continuous implementation via models like Scrum and Kanban.

Which software development models does your team follow?

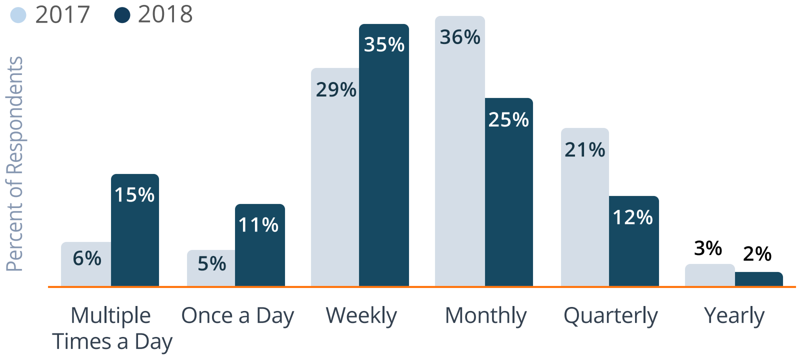


The number of teams releasing product updates on a daily basis more than doubled in 2018.

The software development industry has been under pressure for the last few years to adopt practices and methodologies that enable teams to test and release faster. According to the respondents of our 2018 survey, these initiatives are having a positive impact and release cycles are getting shorter.

Fifteen percent (15%) of respondents reported releasing multiple times a day, as compared to 6% in 2017. A higher percentage of individuals also reported releasing on daily and weekly basis in 2018 than in 2017, while those who reported releasing on a monthly or quarterly basis reported dropped by 11% and 9% respectively.

How often do you deploy / release your application?



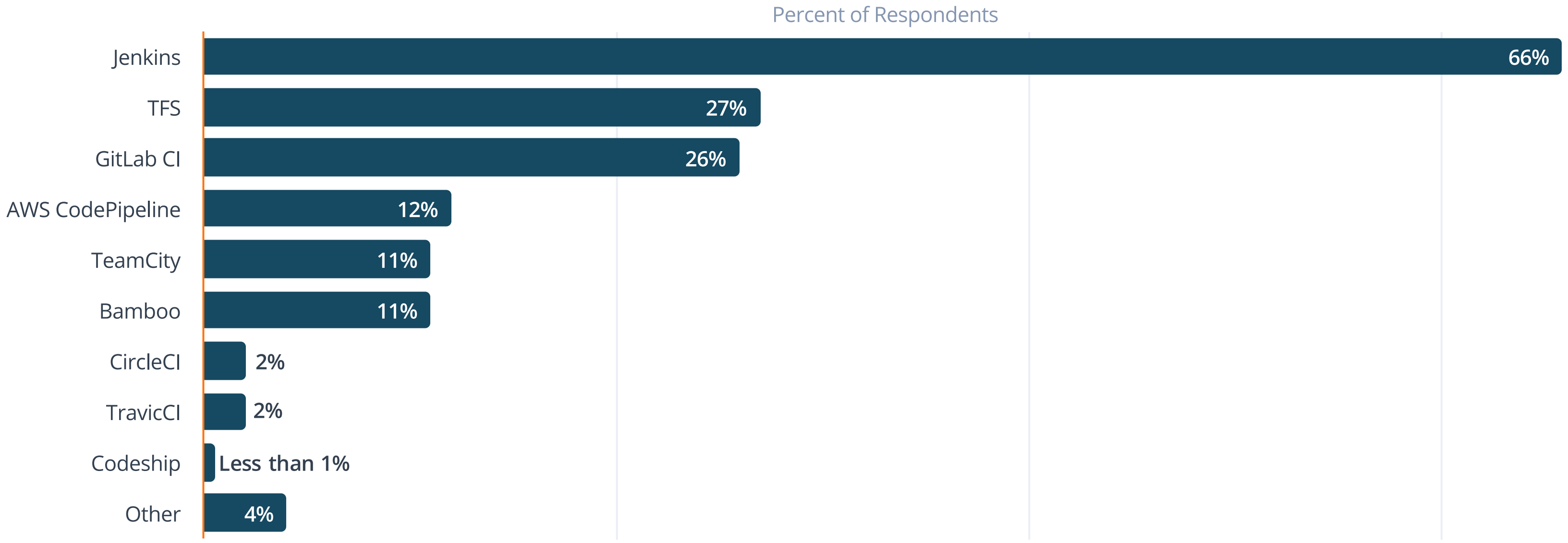
Jenkins is the most popular continuous integration tool used by software professionals.

Sixty-six percent (66%) of survey participants use Jenkins as their continuous integration (CI) server. Continuous integration is an essential practice for Agile and DevOps teams. Given the popularity of the Agile model that we saw earlier, it was no surprise to find that 100% of the survey audience is using a CI tool.

Answers for those who responded with 'other' include:

- Oracle Fusion
- Autorabbit
- Wildfly
- Go
- VSTS

Do you use any of the following CI servers?

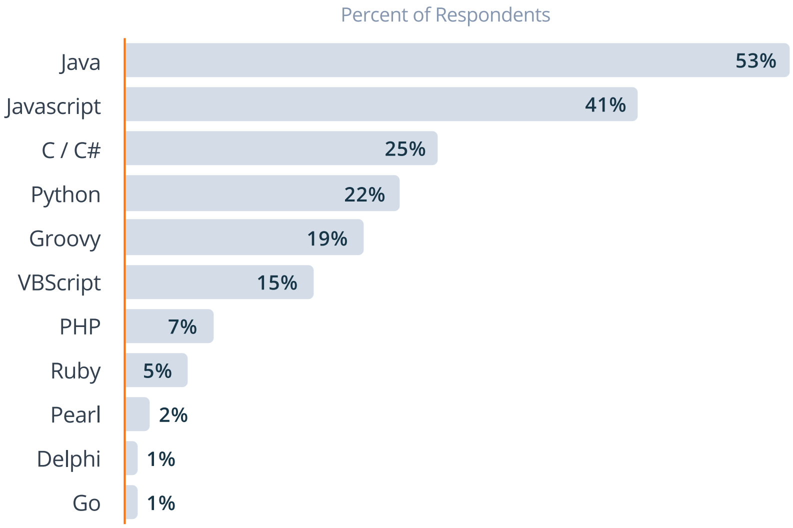


Java reigns as the most popular language.

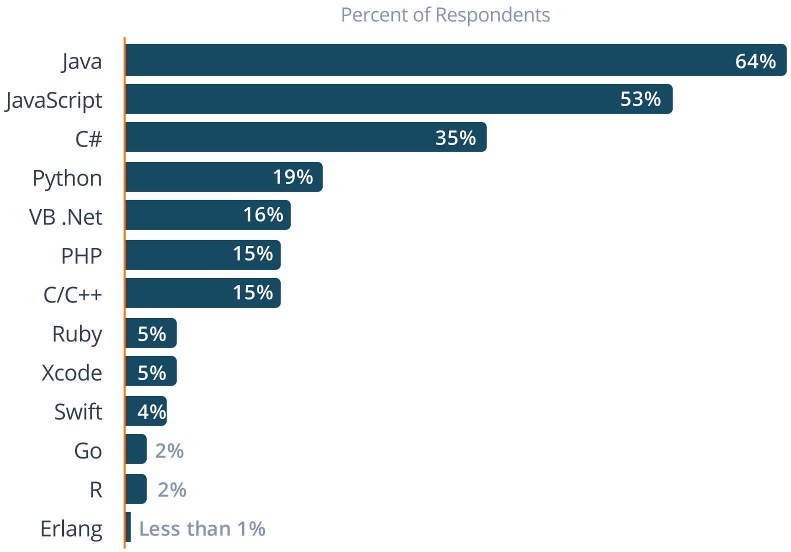
While a higher number of respondents reported using Java to write their application (64%) than to create test scripts (53%), the language is the most widely used for both use cases today. For those who reported writing applications, a slightly higher percentage (35%) reported using 2 or more languages than those who use more than one for writing test scripts (32%).

Self-reported scripting ability shifted minimally year over year, with 37% of the survey audience reporting their ability to write test scripts or code as intermediate in 2017, compared to 41% in 2018. Those who reported themselves as proficient in scripting remained flat at 9%.

What languages do you use to write your test scripts / code?



What languages do you use to write your application?



Testing Approaches & Tools

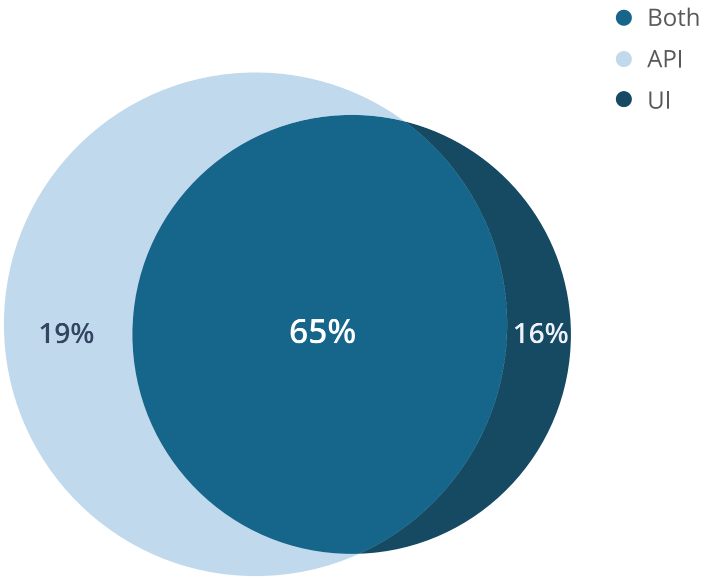
Software professionals responsible for both API and UI testing increased from 2017 to 2018.

Survey respondents who reported testing the front and back-end of their applications in 2018 increased to 65%, up from 63% in 2017. What we found to be surprising last year, was the extent to which individuals were themselves implementing both practices, each being stages of the development lifecycle typically believed to be conducted by completely separate job functions.

While it's expected to see that QA Managers and Consultants indicated being responsible for both, a higher percentage of other job functions also selected 'both' over their traditional responsibilities, indicating the individual software practitioner is implementing API and UI testing.

Fifty-one percent (51%) of developers reported implementing API and UI testing, as compared to 40% being responsible for just testing APIs / Web Services. What this shows, is that as the SDLC quickens, the silos between phases and job functions are blurring.

Which of the following are you responsible for testing?

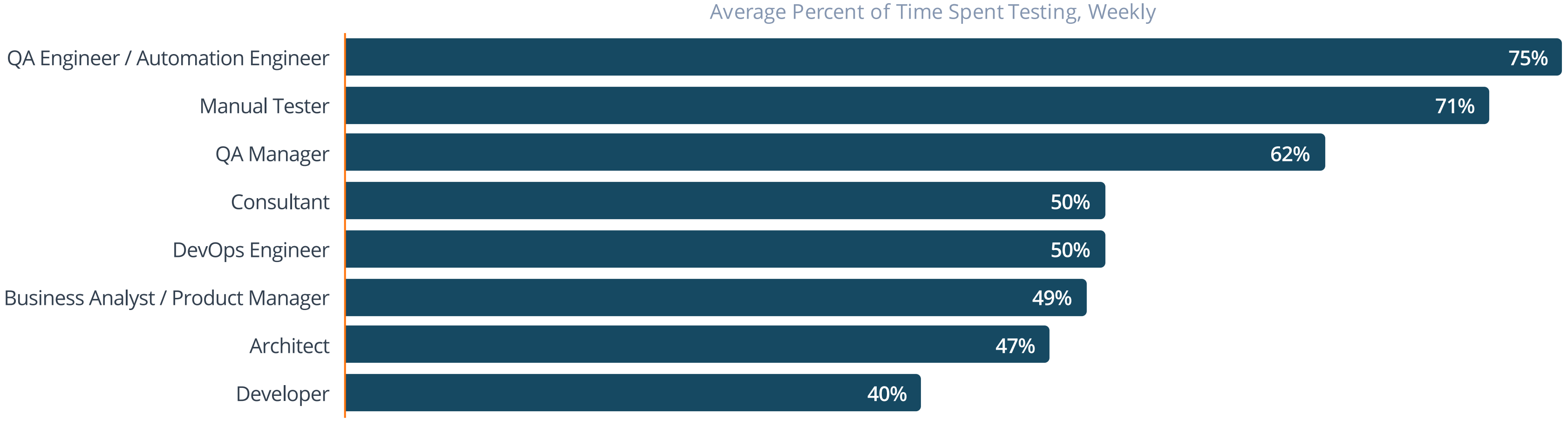


The job of testing is expanding well beyond the role of the traditional tester.

Years ago, it was not unheard of for teams to implement the bare minimum of testing required to validate a product prior to launching. QA had been considered secondary to meeting the development and deployment deadlines. Today, the importance of testing has grown exponentially and team members beyond that of the traditional tester are expected to lend a hand.

Across the board, the survey audience spent more than 40% of their week testing, with QA Engineers / Automation Engineers spending the most amount of time (75%). Developers spend 40% of their week testing, proving that teams are shifting left. Getting developers involved in testing means the process can be done sooner, and feedback provided faster to those responsible for making application updates. This could be a key driver behind the increase in teams delivering in shorter cycles.

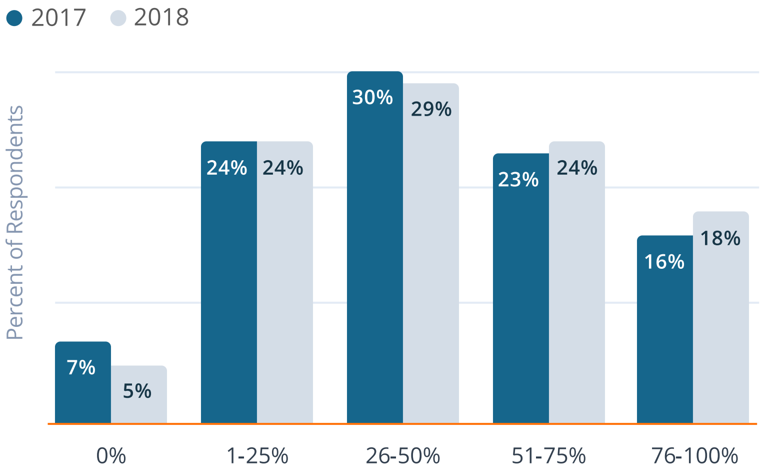
What percentage of your week do you spend testing?



Automation is growing, but not quickly.

It's been widely acknowledged across the industry that automated testing is essential for teams looking to adopt more iterative development cycles and to release faster. Only 5% of respondents this year reported not automating any of their tests, which dropped from 7% in 2017, indicating that purely manual testing cycles are fading. However, only 24% of teams are automating 50 to 75% of their tests, with even fewer automating 76 – 100% of their tests (18%). These numbers grew marginally from 2017 to 2018. While it feels that automation is picking up steam, it's not growing as quickly as the industry might expect.

What percentage of your tests are automated?



Automated API Testing is the fastest growing testing practice.

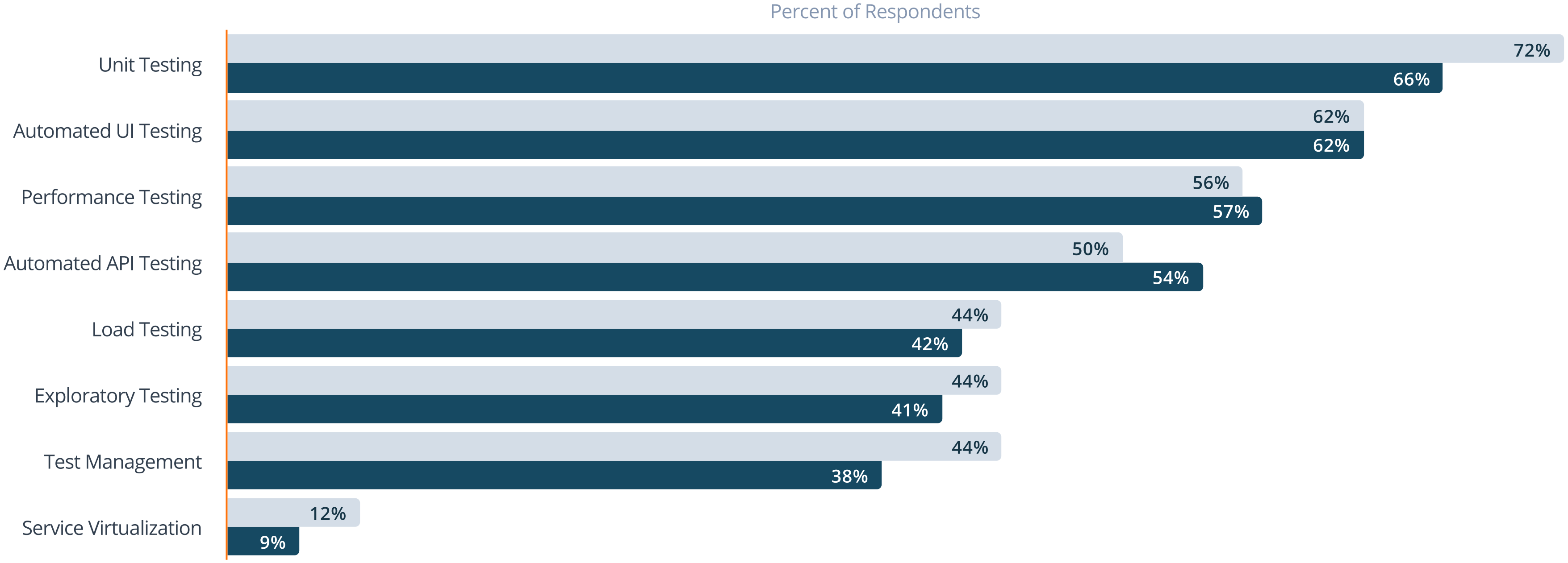
The use of Automated API testing grew from 49% in 2017 to 54% in 2018. In fact, the only other testing practice that grew in use year-over-year was performance testing. While this could indicate a difference in the mix of responses we received this year, it validates industry talks regarding the booming API economy and the increasing reliance technology has on APIs, which we will discuss later in this report.

Given the importance put on automated testing, it's no surprise to see that the manual testing practice, exploratory testing, has decreased year-over-year.

We added three testing types to the question this year to capture more information from our audience about their practices. These showed that regression testing is the most commonly used practice among all respondents, and less than a quarter are implementing Test Driven Development (TDD) or Behavior-Driven Development (BDD).

- Regression Testing = **74%**
- TDD = **23%**
- BDD = **19%**

Which testing practices does your team use? ● 2017 ● 2018



Almost one-third of unit tests are still being run manually.

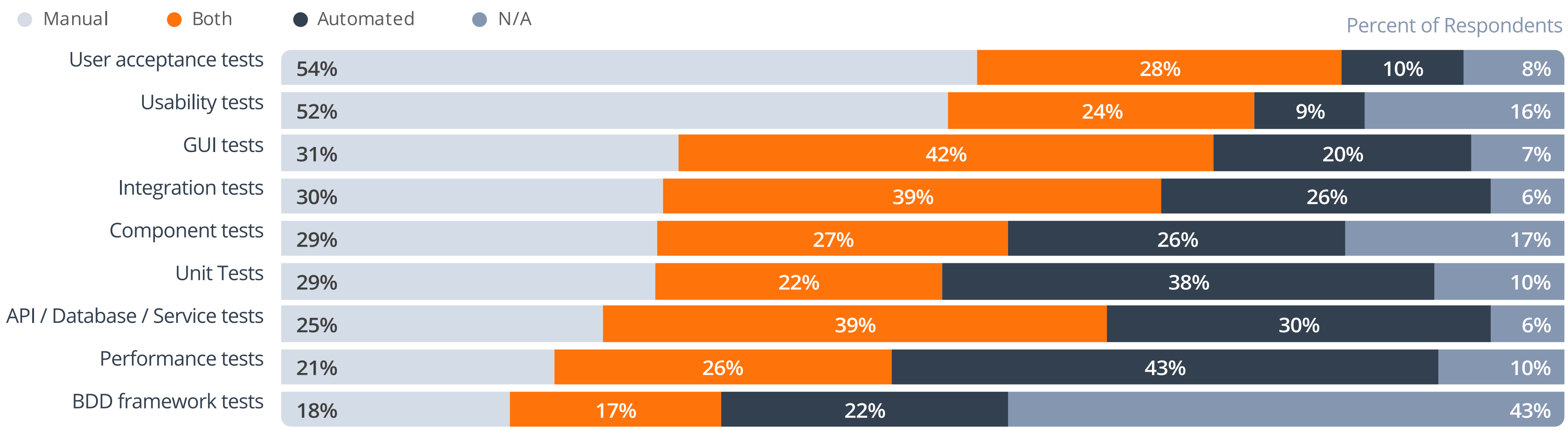
In our 2018 survey respondents answered performance, API, and unit tests are the top three most-automated methods of software testing. In this context, we are assuming that the 29% of individuals not automating their unit tests are simply not automating them in a pipeline.

Unit tests are the some of the most effective, resilient tests we can write and run against our software and should be automated upon each deployment, not only run on-demand by developers. By having unit tests in our pipelines, we should be able to reduce our mean time to resolution and speed our average build times.

While there is no magic number, software teams should have significantly more Unit tests than End-to-end tests. Ten-percent (10%) of user acceptance tests and 9% usability tests are being completely automated. Both practices are excellent use cases for manual testing since they require that a full system or workflow is validated from the end-user standpoint.

The human mind here is key, so what is intriguing is that there are a handful of QA professionals who have found scenarios for user acceptance and usability tests in which automation excels more than the programmer or tester's opinion.

Which tests does your team run manually, with automation, or via a combination of the two?



Load Testing

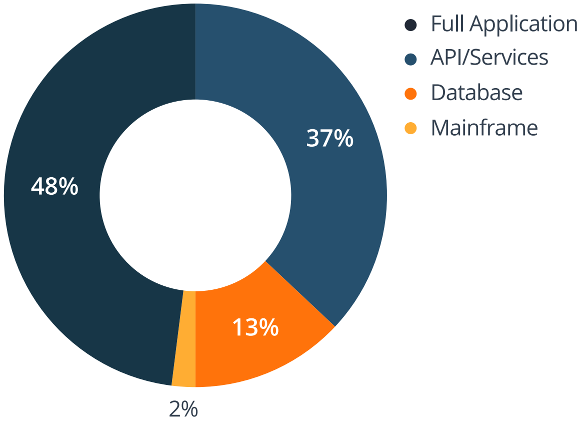
Load testing is a critical component of any development cycle as it ensures that parts of an application under test, whether it’s an API, a database, or even the full application itself, can handle real-life load conditions. For web-based products, this could mean the number of site visitors, and for APIs, the number of times a single API is hit.

Less than half of all QA professionals are load testing their APIs.

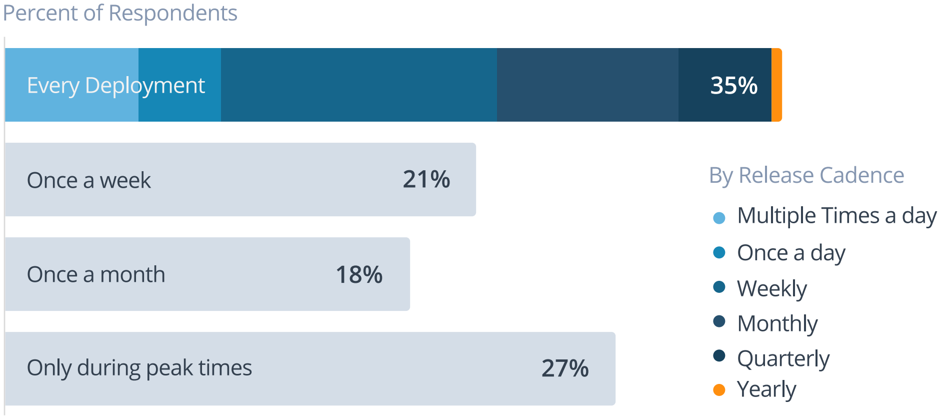
Despite the importance of and reliance on APIs across the industry, just over one-third of survey participants (37%) are load testing their APIs.

Across the board, less than half of respondents (48%) reported load testing their entire application, meaning there is plenty of room for teams to make improvements to their software's performance, and ultimately, consumer experience. Of those that do implement load or performance tests, 35% of respondents reported releasing every deployment, with the majority occurring on a weekly basis.

What do you load or performance test against?



How often do you load or performance test your applications?



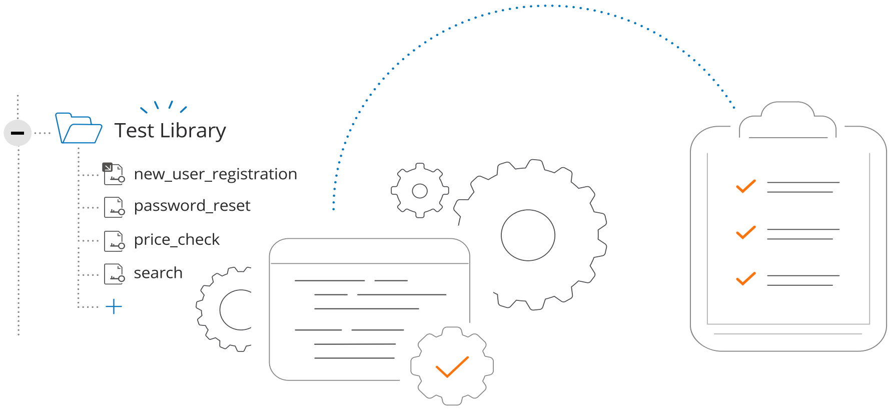
Test Management

Test Management is a process that can help teams track test cases, whether they are automated or manual, and link them to requirements, tasks, and defects. The definition of test management as a practice, can range from tracking test cases in an excel spreadsheet, to using a project management tool such as JIRA, or a standalone test management platform.

More than half of all teams have invested in one or more standalone test management tools.

Sixty-seven percent (67%) of the survey audience has invested in standalone tools. Despite the large mix of platforms available in the market, Jira is the dominant choice for teams today, with 59% of respondents indicating they use it for managing their test cases.

Less than half are using other major products including: QAComplete (8%), Zephyr (8%), TestRail (8%), qTest (4%), and Hiptest (2%). Answers for those that selected 'other' include: HP ALM, Rally, SpiraTest, and in-house made tools.



Thirty-three percent (33%) of teams are still using some form of spreadsheets for test management. For those teams who are struggling with spreadsheets and looking invest in a standalone tool, there are a few things to consider. Selection criteria for choosing a test management tool include:

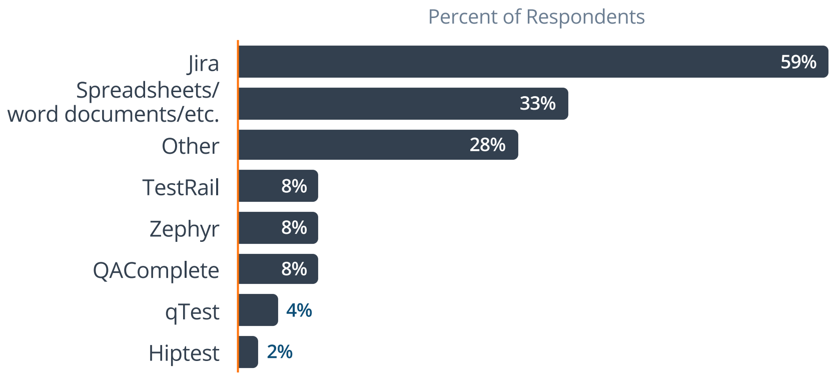
- Adapt to the way your team works whether that's an Agile team, Waterfall team, etc.
- Integrates with Jira and other testing tools that you're already using
- Allows for permissions controls and settings

Teams using a test management tool are more likely to automate.

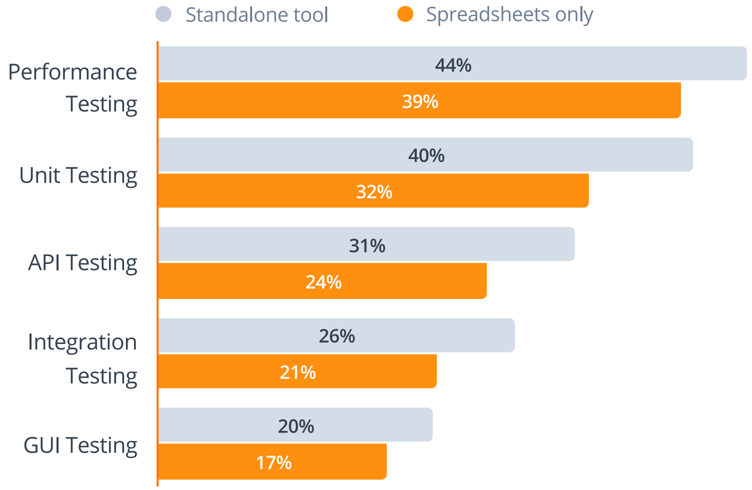
There are many benefits for teams who choose to invest in a test management tool including an improvement in collaboration, prioritization of tests, reduced duplication and errors, and increases in automation.

This year, we discovered that survey respondents using a standalone test management tool are more likely to automate their testing across a majority of testing types including: integration testing, unit testing, UI testing, API testing, and performance testing, as compared to those who use spreadsheets and word documents.

What tool do you use for test management?



Test automation and test management tool use.



API Testing

APIs are the building blocks of modern software and business innovation, enabling a connected-data revolution that is transforming the way we design, build, and test applications and services. From GPS mapping, to payment processors, APIs play a critical role in connecting the world around us. They have allowed companies like Twilio, Stripe, and Slack to thrive and grow by providing specialized functions that easily connect into existing ecosystems and applications.

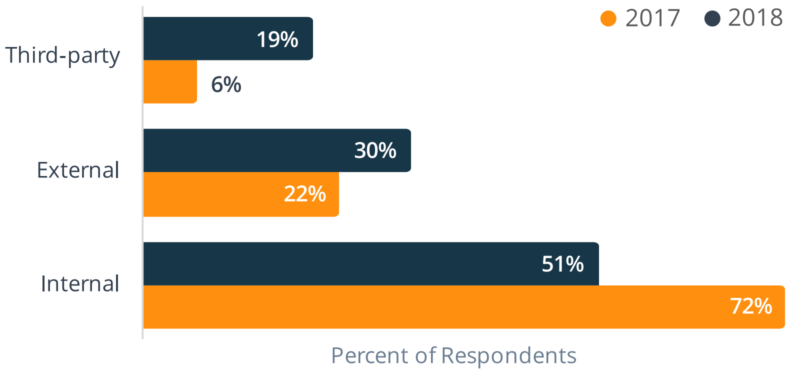
This next section will dive into the current trends, challenges, and the overall state of API testing for today's development and QA professionals.

External API testing is growing.

2017, 71% of respondents were testing some kind of API or web service and in 2018, that number grew to 80%.

The breakdown of types of APIs tested has also drastically shifted year-over-year. More developers and testers are testing third-party and external facing APIs, rather than just the internal APIs that make their application functional in a SOA or Microservice Architecture. Using third-party APIs for testing can be notoriously expensive and difficult, as API providers can charge teams each time they use the API.

Which type of API do you test against most frequently?



The increase in testing of third-party services could be attributed to two factors. First, the general increase in third-party software used in applications has led to an increase in making sure they are tested. Second, teams are understanding and better utilizing service virtualization, allowing for increased availability and reduced costs for external APIs.

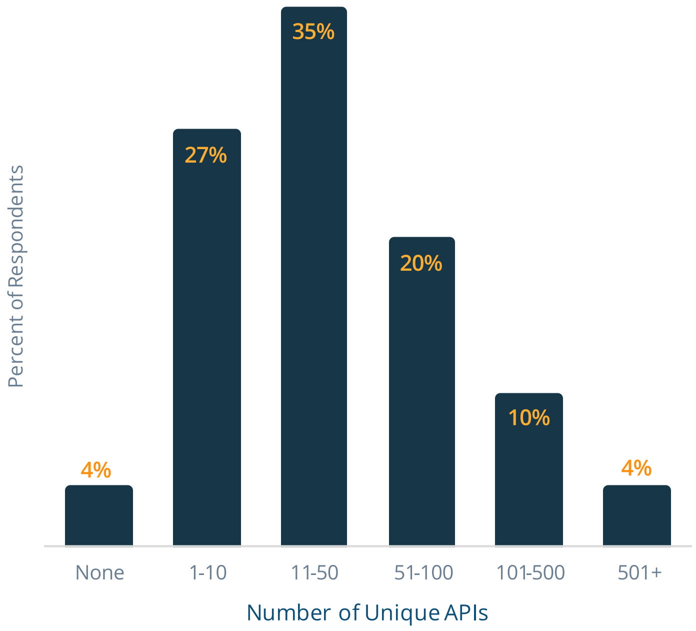
The number of respondents that do nothing to remedy external API dependencies also dropped from 16% to 12%, a step in the right direction. Teams continue to use a mix of service virtualization, mocking, and stubbing to get these external APIs tested sufficiently.

APIs are growing in numbers.

Think most teams only have a few endpoints in their application's architecture? Think again. Thirty-one percent (31%) of survey respondents are testing between 51 and 500 endpoints, with those testing 50 or more endpoints having more than doubled since 2017.

This is great news for software teams as it points to the growth of separate APIs fueling service-oriented architectures. This data point backs the perceived growth in Microservices and Cloud Infrastructure.

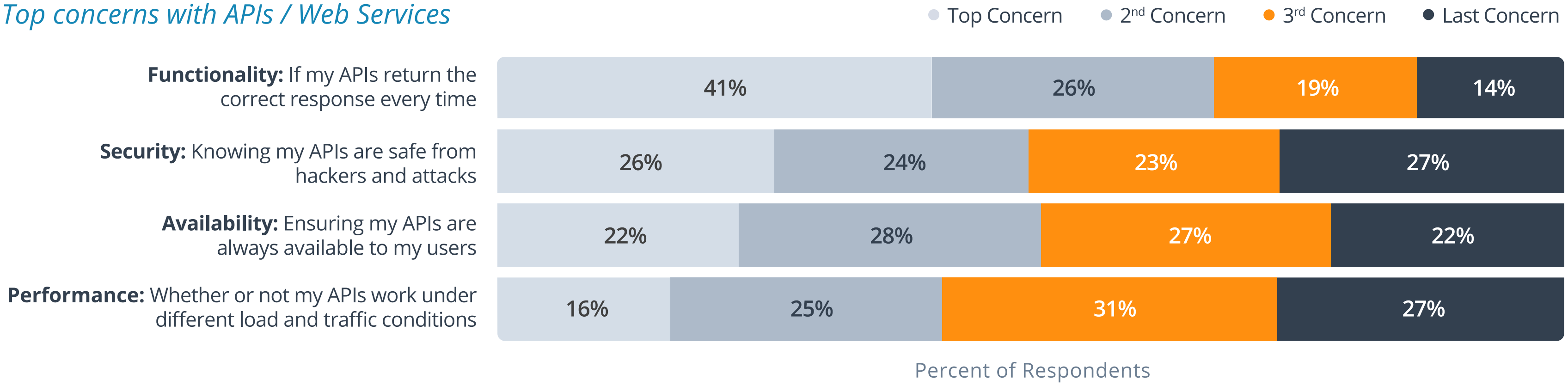
How many unique API endpoints do you utilize in your testing and development?



Functionality is still the name of the game.

When we asked users last year to rank their top concerns in order of importance surrounding APIs, functionality was the resounding winner. When polled this year, our users answered nearly identically. Functionality was once again the top concern, meaning API testers are most concerned as to whether of not their APIs will return the correct response. This year however, security and availability traded places.

Top concerns with APIs / Web Services

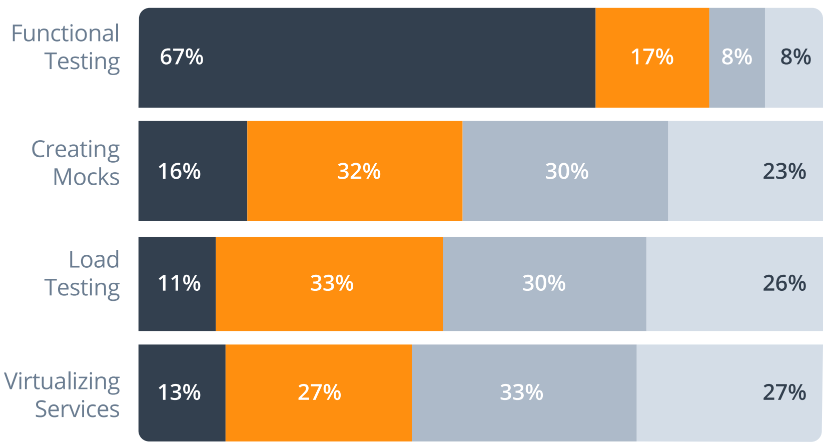


Teams spend more time on functional API testing than on any other practice.

If we look at how much time is spent on various API testing practices, respondents' answers seem to correlate to which API testing concerns are actually more difficult to tackle. More time is spent creating functional tests than any other challenge - including creating monitors or load tests for performance testing.

Virtualization and mocking still take quite a bit of time for teams though, which could be attributed to the centralization of infrastructure across organizations - unlike code, pipelines, and tooling that is left in the hands of the end users (devs and QA) - virtualized services are often the responsibility of central IT and Ops teams.

Time spent on API testing



UI Testing

GUI testing is a critical part of software development. Most end users focus on the usability and aesthetics of the software they use and can be blinded to the underlying scripts and APIs that make their applications functional. A poorly designed UI can lead to unsatisfied consumers, so it's vital to the success of any software to prevent bugs from reaching production and to provide the user as close to a flawless experience as possible.

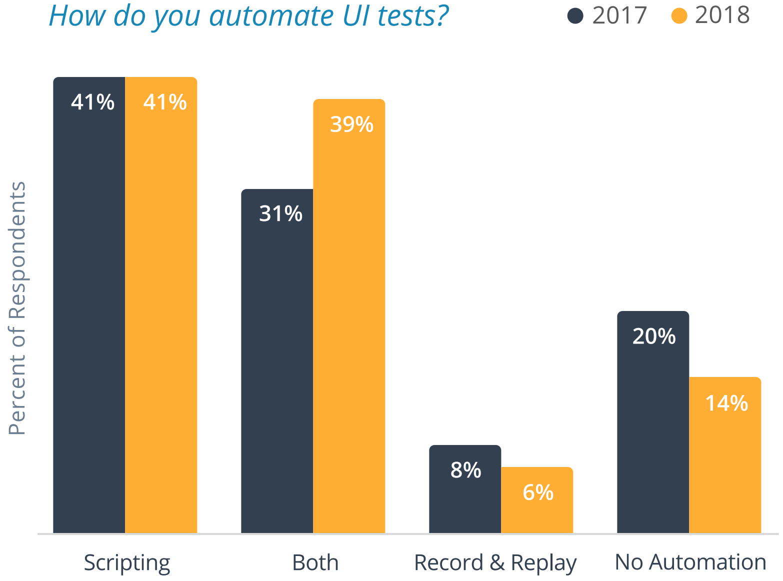
UI testing is challenging though. As an application's visual elements can change frequently, it becomes exponentially more difficult to test. With users adopting new technologies daily, the types of software that need to be tested are also transforming. This next section of the report digs into GUI testing practices, the challenges around automating UI tests, and the current state of web and mobile testing.

The majority of UI tests are still created using scripting.

Many automated testing tools emphasize their ability to develop robust test scripts without having to write code. The record and replay feature alleviates many of the hassles associated with manual testing and enables faster UI testing by tracking and recording hard-to-script actions, such as mouse movements and keystrokes.

While there continues to be an emphasis on record and replay tools in the market, UI testers are more likely today to write scripts to create UI tests. Survey respondents using scripting to build their front-end tests remained flat year-over-year at 41%, while those who indicated relying solely on record and replay dropped by 2%.

How do you automate UI tests?

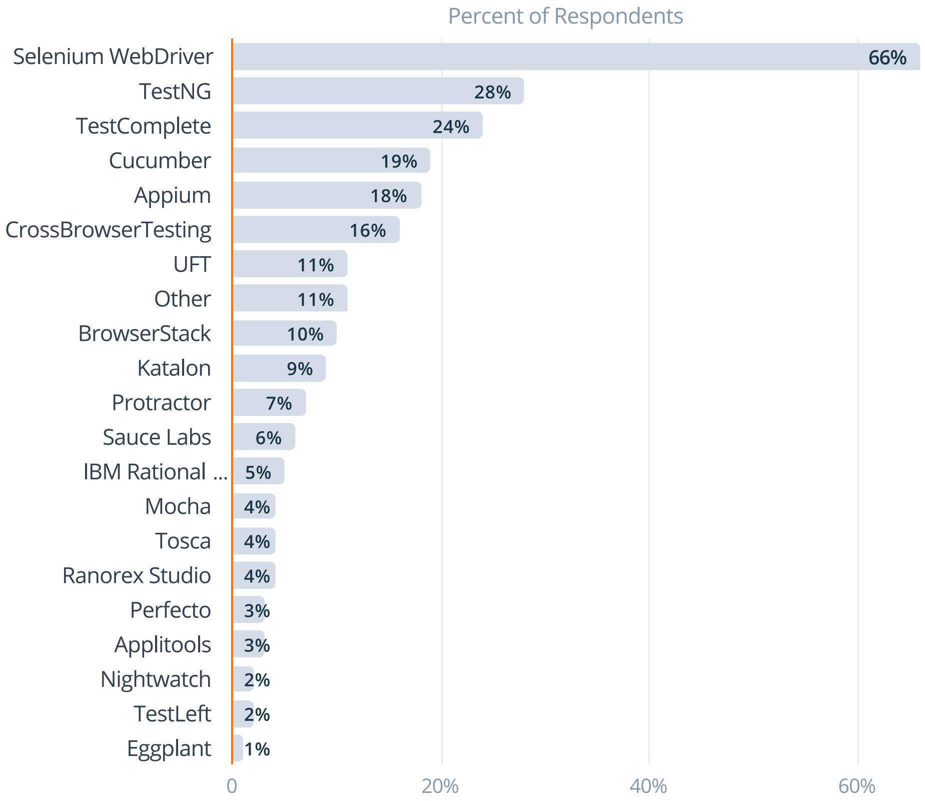


Selenium WebDriver dominates as the most popular UI testing tool.

When it comes to UI test automation, there are many front-end testing tools and frameworks available to teams today, but the most commonly used is Selenium WebDriver, with 66% of respondents indicating they use the tool. There are many other open-source frameworks and platforms out there to help assist in automated UI functional testing, but there are also plenty of standalone options including TestComplete, CorssBrowserTesting, UFT, and Ranorex.

For the 11% who selected 'other,' answer choices included: WinAutomation, TestCafe, testim.io, and custom developed tools.

What UI testing tools do you use?

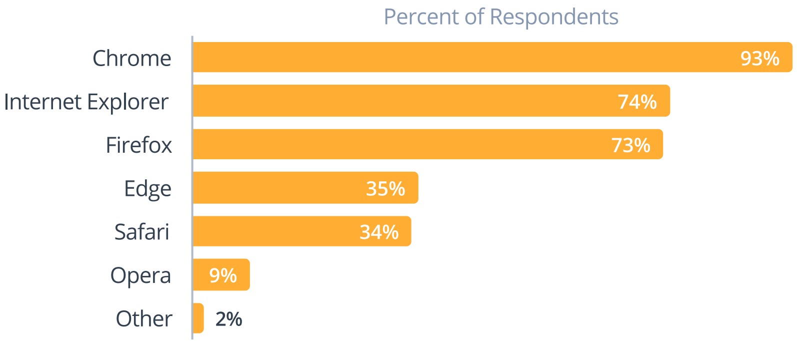


Chrome, Firefox, and Internet Explorer remain as the three most popular browsers to test applications on.

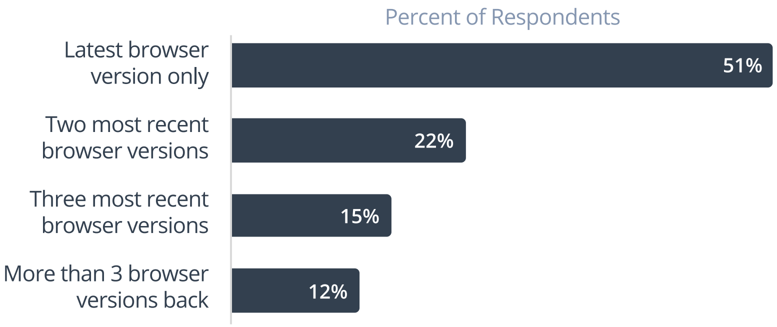
Chrome is the most popular browser to test applications on today (94%), followed closely by Firefox (74%) and Internet Explorer (73%). This is no surprise when comparing to last year's results where Chrome, Firefox, and Internet Explorer were also the three most popular browsers to test applications against. The other popular browsers that testers are testing on include Edge (35%), Safari (34%), and Opera (9%).

Many browsers, like Chrome, include automatic updates to browser versions unless specified, so it's understandable that half of QA teams focus exclusively on testing on the latest browser version (51%). When we look at the latest two browsers, we find that this number increases to 73%. When we include the latest 3 versions we, find the percentage of testers increases to 88%.

Which browsers do you test your applications on?



How many browser versions do you test on?



Although testing across multiple browsers and versions can be tedious due to the number of combinations available, even with automated testing, we found that 87% of survey respondents are testing on 2 or more browsers, an increase from 65% in 2017. This increase in the number of browsers being tested on leads us to believe the need for an automation and cross-browser testing strategy will continue to be vital to the success of any UI test strategy. There will always be room for teams to expand test coverage by adding older browser configurations or new browsers to their arsenal.

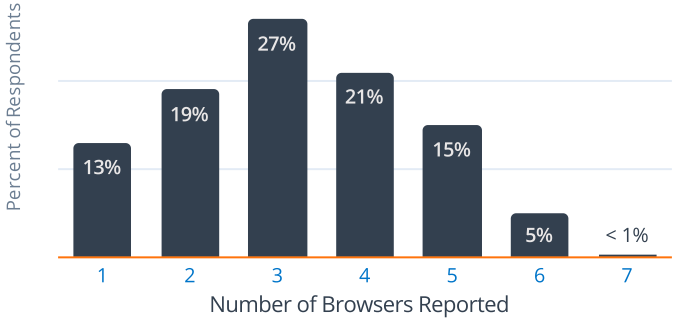
Over half of survey participants test on mobile devices.

The mass adoption of smartphones has intensified the need for mobile UI testing, so it was no surprise to find that over half of the survey respondents reported testing on mobile devices. Of those who responded to testing on mobile devices, 81% indicated they test on iPhones and 75% on Samsung devices. However, cell phones aren't the only types of mobile devices out there that require testing. Tablets and e-readers are becoming more advanced, widely-used, and require front-end validation as well.

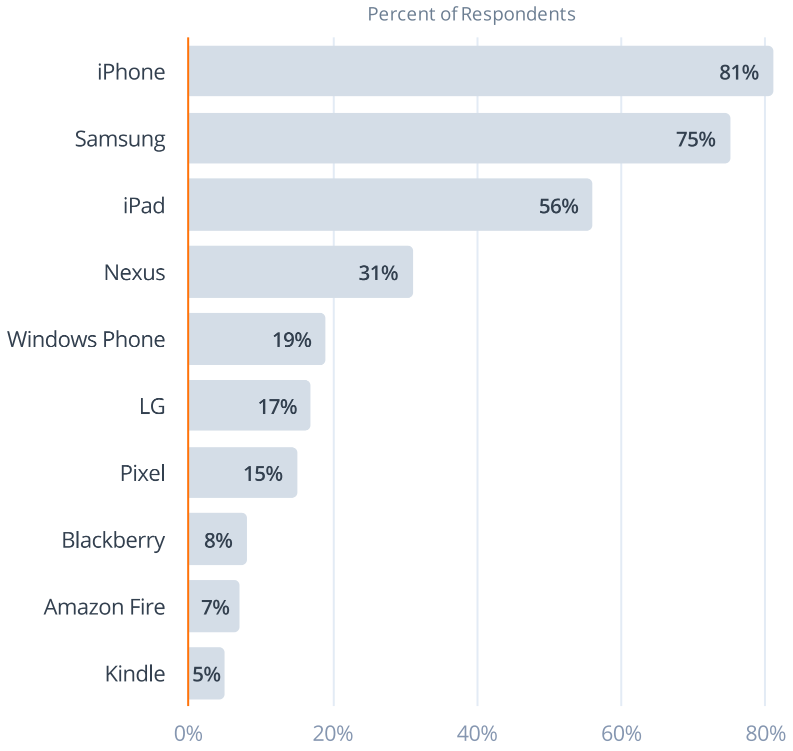
In addition to browsers and browser versions, testers now need to ensure their applications function properly on a myriad of devices, operating systems, and versions. In contrast to web testing, 54% of mobile UI testers do not exclusively test on the latest configuration of mobile devices. We found that people are more likely to test on iPhones, Androids, and iPads, over devices like Kindles, Blackberry, and Windows machines.

Mobile device iterations are released and adopted much less frequently than those of browsers, and some companies are more consistent in sunsetting versions than others. There are only three or four iPhone models in the market at any given time for consumers to use, but Android users have thousands of options to pick from. Not only are different device versions available, but Android users can choose from different brands as well (Samsung, Nokia, Motorola), each with their own screen sizes and resolutions. This fragmentation is great for consumers who like to have options, but can be a massive challenge for developers and testers.

Number of browsers applications are tested on.



Which mobile devices do you test your applications on?

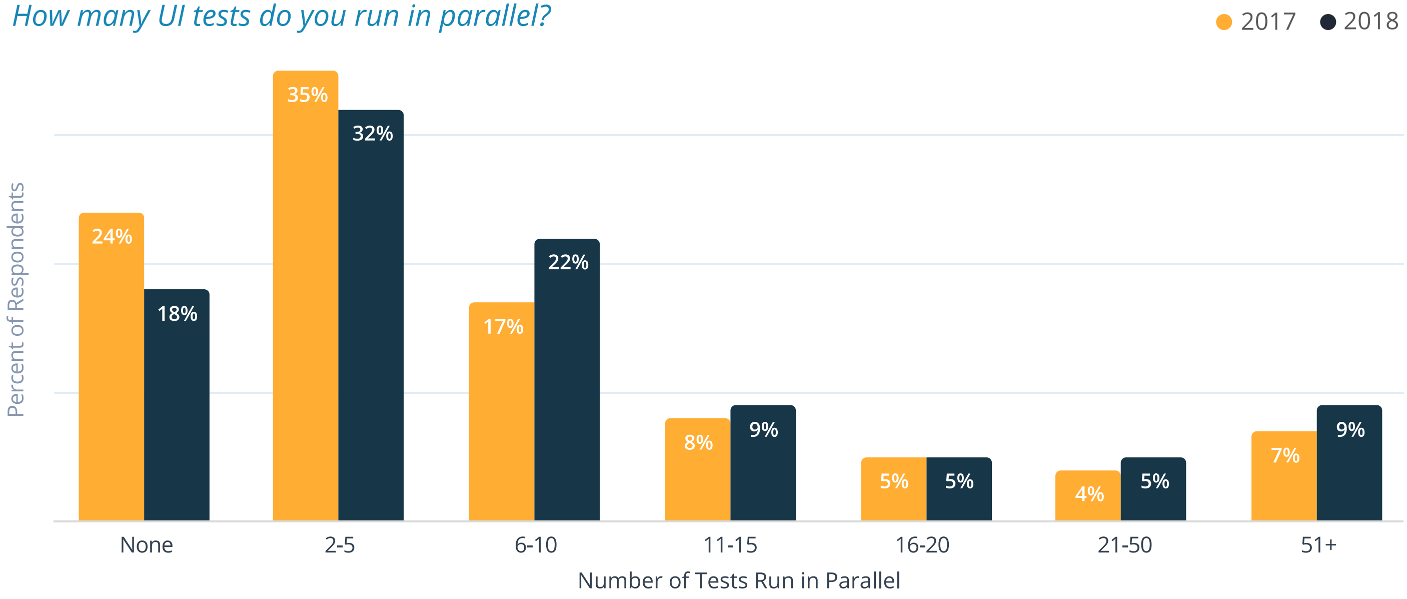


The number of testers executing tests in parallel has increased.

Running tests concurrently on multiple devices expands test coverage while reducing testing time. This alleviates many challenges UI testers face when running tests across multiple environments. Today, the majority of testers are still only running a few tests in parallel, or none at all. However, the benefits of parallel testing seem be spreading as teams look to optimize their time and testing cycles. The number of survey respondents not running any tests in parallel decreased from 24% in 2017 to 18% in 2018. Year over year, we also saw an increase in the number of sets of tests being run in parallel, with those running 6 to 10 tests concurrently increased from 17% to 22%.

This indicates that there is an overall increase in testers who are not only looking to run tests concurrently, but increase the number of tests they want to run in parallel as well. Given the explosion in the number of devices available to consumers today, each with their specifications and operating systems, it's understandable as to why we are seeing an increase in teams adopting parallel testing.

How many UI tests do you run in parallel?



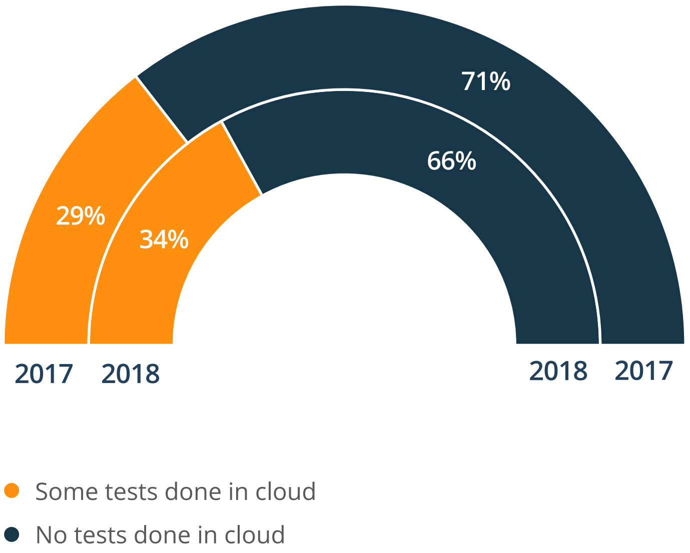
One-third of software professionals are now leveraging cloud-based testing.

Cloud testing has recently emerged as an asset in the software development industry. With cloud-based test tools, test environments are available on-demand and will often be updated regularly, giving testers continuous access to the most recent browsers, device versions, operating systems, and resolutions.

Cloud-based platforms are also inherently collaborative, allowing testers and managers to view the progress of their tests and development processes from one location.

Given the benefits of cloud-based testing, it was no surprise to find more survey respondents reported testing in the cloud this year (34%), as compared to last year (29%). More than half of all teams are still not leveraging this recent advancement however, indicating there are yet even more methods teams can take advantage of to reduce testing cycles and get to the market faster.

What percentage of your tests are performed in the cloud?



The Future of Testing

With the expansive universe of technological devices available to consumers today and the new tools and practices cropping up to build and test them, we were curious to learn how respondents felt about the future of the industry.

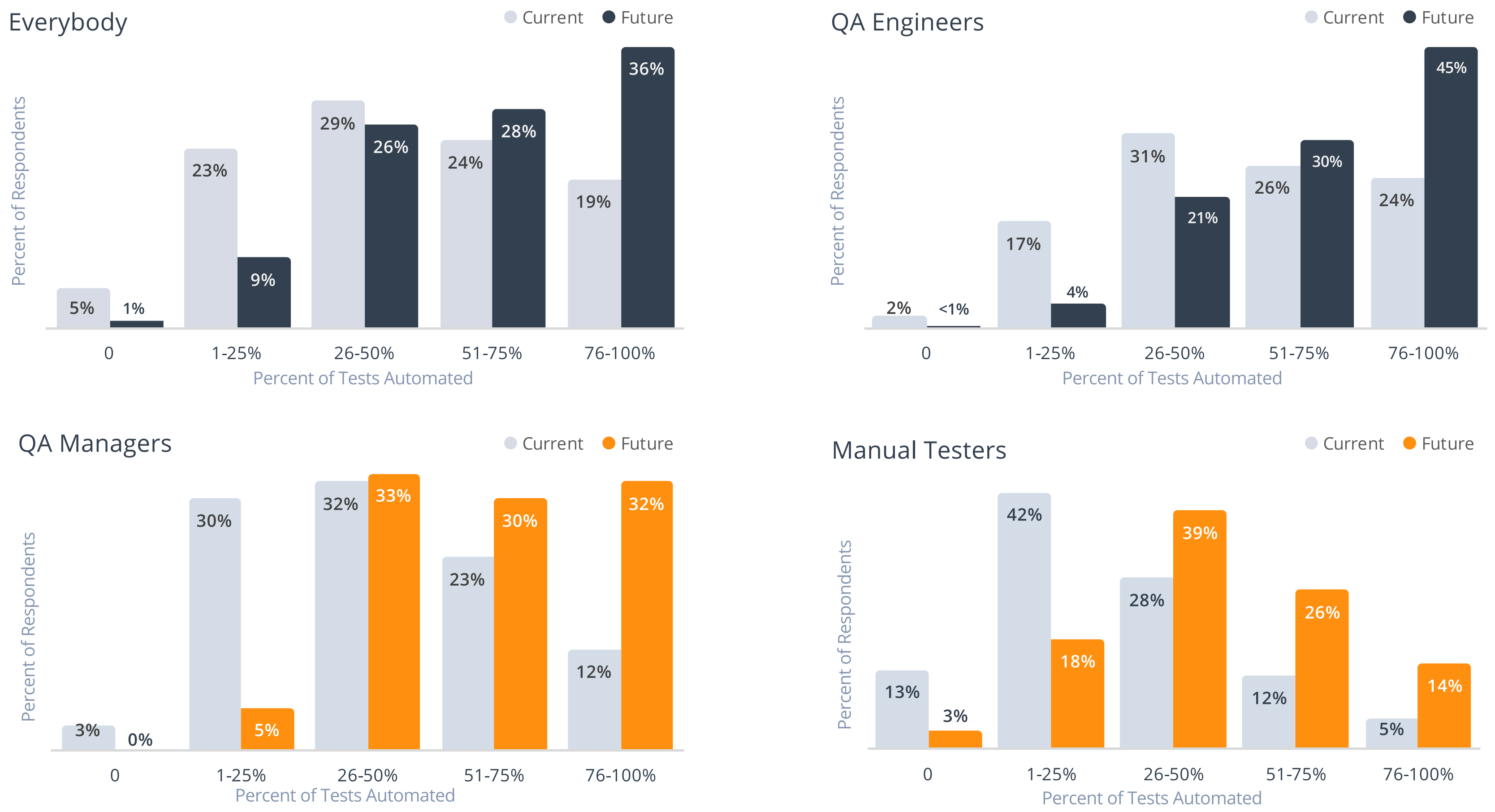
Artificial Intelligence and Machine Learning are no longer far-fetched ideas, but pratices that have weaved their way into modern development practices and are being leveraged for better testing quality. Mixed reality is the forefront of what's next and wearable mobile devices are integrating with the human body. It's both a terrifying and exciting era to be in software. What will the next two years hold and what technologies can we expect to see more of?

15% of teams deploy more than once a day.

All job roles, from Analysts to QA Engineers and Manual Testers, expect automation to continue to grow.

Unsurprisingly, Manual Testers have a more tapered expectation of the growth of test automation as compared to QA Engineers. Only 14% of Manual Testers would expect to automate 76-100% of their tests in the next two years as compared to 45% of QA Engineers. Manual testing is a vital practice for teams to implement as it can help identify tricky bugs that can't be detected with automated tests. While Manual Testers understand that the use of automation will continue to expand (there was 10% drop in those expecting to run all of their tests manually year-over-year), most still expect to automate less than half their tests.

Future of Test Automation by Job Title



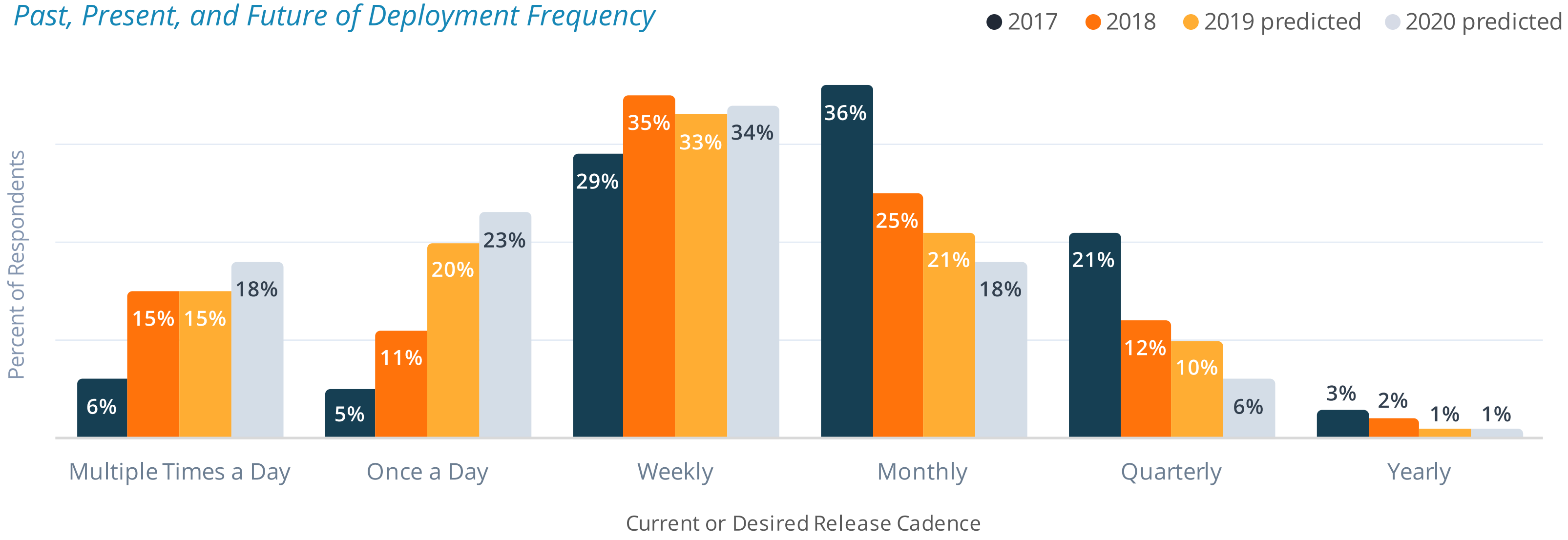
Global goals of shortening release cadences are being realized sooner than expected.

QA professionals globally have been looking to shorten delivery cycles. Last year, 68% of respondents were hoping to deploy weekly, daily, or even multiple times a day by 2019.

Some of those expectations were met a year early, and some have already been surpassed. In 2017, 33% of respondents predicted that they would be deploying on a weekly basis. Thirty-five percent (35%) of respondents this year reported doing so. Astoundingly, in 2017, 15% predicted they would be deploying multiple times a day. Survey respondents met that expectation this year.

Despite the slow growth of automation, the software development life cycle is getting shorter, indicating other practices that facilitate fluid and frequent development and testing, such as parallel or cloud-based testing, or industry trends like 'shifting-left,' are making an impact.

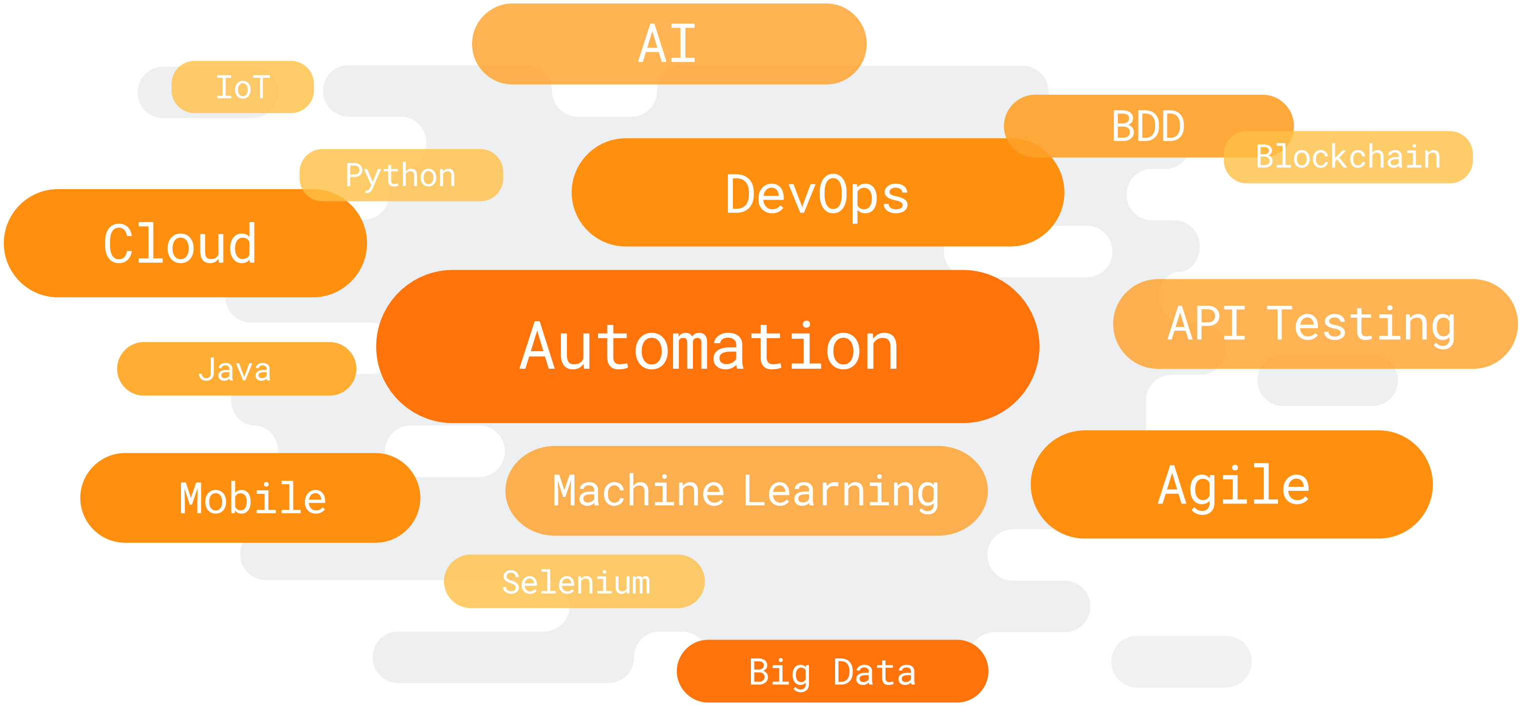
Past, Present, and Future of Deployment Frequency



QA professionals expect to also see more Artificial Intelligence, Machine Learning, and cloud-based testing in the next 2 years.

This year, we asked our respondents two open-ended questions in an attempt to better understand specifically what practices and technologies software professionals expect to see more of over the next two years, as well as their thoughts about what the future of the industry will hold. The word cloud below represents the top practices our survey audience expects to see more of over the next few years.

Top 3 practices and technologies users expect to see more of in the next two years.



What will the state of software testing look like two years from now?

The quotes below highlight the most interesting perspectives about the future of the industry from individuals across a variety of job functions. Once again, we were surprised by the stark differences in the mindsets of testers, developers, and others throughout the community. While some believe massive and evolutionary changes are imminent, others view the industry as mammoth-like, slow to and hard to change rapidly.

At the rate that AI is growing, as soon as you've built something you should be able to just tick a box to enable test automation

- Software Development Manager

I believe we need lots of AI driven testing tools so that as a developer i spend more time in adding features to my product then spending time on testing apps

- DevOps Engineer

I would hope that there is still a need for manual testers. There's been such a push for automation. We want that to help us work smarter and make our jobs easier. But, automation can't replace the human element, especially when you're dealing with complex applications. Also you still need people maintain to automation and test what the automation can't.

- QA Engineer and Manual Tester

Human overlords & robot workers

- QA Manager

I don't believe there will be any testers, only Developers and some will be assigned to test

- Manual Tester

Not hugely different, it doesn't change that fast. The majority is still doing what they did 30 years ago.

- Test Consultant

Software testing is a field that is going to evolve a lot in the next 2 years, considering the rise of Progressive Web Applications, IoT applications in everyday life introducing a number of APIs, Machine Learning and Analytics based applications, Blockchain and cryptocurrency applications

- Architect

Automation using AI will be unavoidable. Combination of API and UI automated tests will be highly prevalent rather than maintaining separate tests for each

- QA Engineer / Automation Engineer

It may disappear

- QA Manager

I can see the software testing field becoming purely a DevOps shop. The line between testers and developers will become increasing blurry.

- QA Manager

Similar. New tech, as always, but I am a firm believer in automating the easy stuff to allow humans to use their big brains to hunt out the dragons in the unknown, difficult to automate areas of applications (button mashing, weird combinations of user inputs, system edge scenarios like varying bandwidth in the wild and low battery conditions on devices).

- QA Engineer / Automation Engineer

Testing in the traditional sense is going to fade away and will take the direction of having developers who write tests as opposed to testers who test

- QA Engineer / Automation Engineer

The idea that it is possible to automate all testing will finally die, or everyone will realize that if testers could automate everything developers will be obsolete and Alan Turing proven wrong on the halting problem.

- QA Engineer / Automation Engineer

Not hugely different, it doesn't change that fast. The majority is still doing what they did 30 years ago.

- Test Consultant

Survey Partners & Collaborators

We would like to thank the following SmartBear partners and software testing groups for helping us collect the vast amount of data we gathered. You were influential in getting us the responses we needed to make this report a success.

MINISTRY OF TESTING

Cigniti Software Quality | Assured

PRIANTO

EMERASOFT

LIFEBOAT DISTRIBUTION

WiseClouds

SMARTBEAR
TestComplete

Functional Test Automation
for Desktop, Mobile and Web

Get Started

SMARTBEAR
Zephyr

Native Jira & Real-time
Test Management

Get Started

SMARTBEAR
CrossBrowserTesting

Web-based, Mobile
and Desktop Browser Testing

Get Started

SMARTBEAR
ReadyAPI

Write, Run, Integrate and
Automate Advanced API Tests

Get Started

SMARTBEAR

Think Bigger. Build Smarter.
Create Great Software.

6.5M+	22K+	194
Users	Companies	Countries

Get Started

SMARTBEAR