



# The State of Testing 2017

SMARTBEAR

## Content

Preface & Methodology.....	3
Introduction.....	4
Demographic & Firmographic Data.....	6
Software Testing.....	11
Development Models.....	13
Testing Approaches & Methodologies.....	15
Test Automation.....	20
API Testing.....	22
UI Testing.....	27
The Future of Testing.....	33
Survey Partners & Collaborators.....	39

## Preface

This survey was designed to establish benchmarks for the software testing industry regarding the ways in which teams are testing the graphical user interface (GUI) and application programming interfaces (APIs) of their applications in 2017. The following report will cover:

- | Software Development & Testing Methodologies
- | API Testing: Approaches, Practices, & Concerns
- | GUI Testing: Test Creation, Techniques, & Challenges
- | Future Expectations of the Industry

## Methodology

SmartBear Software conducted a global online survey over the course of four weeks in June 2017 and July 2017 and collected a total of 5,219 responses. Survey responses were collected via online advertisements, social media, and in collaboration with industry influencers and SmartBear partners. An internal database of QA professionals responsible for API and UI testing was emailed an invitation to participate in the survey. The findings presented are based upon the completed responses from over 3,400 QA Engineers, Software Developers, Manual and Automation Testers, Consultants, and Analysts from more than 15 different industries globally. A copy of this report and five \$100 dollar amazon gift cards were offered as incentives for participation.

## Introduction

Is the line between development and testing continuing to blur? What are the top challenges with UI test automation? How do teams react when third-party APIs aren't available? How much testing is done in the cloud?

These are just a few of the questions we had for QA professionals and development teams around the world. Billions of people use software every day and the demand for quality applications is only increasing.

From ride-hailing apps and video-streaming services, banking transactions to video games, to driving a car and crossing the street, we interact with technology from the second we get up in the morning to the last moment before we fall asleep. This boom in software technology has disrupted how we live our day-to-day lives and has increased pressure on development, testing, and operations teams to build and release applications faster than ever while also providing a great end-user experience.

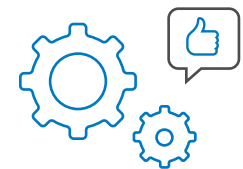
With our flagship testing survey, we set out to investigate the evolution of the various tools, development methodologies, and testing techniques that leading software teams use to ensure software quality and quick delivery cycles. Our goal was to better understand the current state of the software testing industry and discover what the future might hold.

What makes our survey unique is the breadth of information we set out to gather from the entire community. With 41 questions and a total of 5,219 responses, we were able to gather valuable data from developers and QA professionals in a variety of job functions, levels, company sizes, and industries worldwide, that successfully paints the picture of today's testing industry from end-to-end.

This report is fascinating. It reveals the thoughts of developers and testers like you (as well as a few managers, consultants, and analysts), and presents key insights into today's trends, opportunities, and challenges. We invite you to keep reading to discover what your peers are doing today in testing and what it means for you.

If you are one of the contributors that took the time to fill out the survey, thank you. You provided incredible feedback that we hope to build on in the years to come.

- SmartBear Software



## High-Level Takeaways

Over half of QA Engineers and Software Developers in Test are responsible for testing both the front-end and back-end of their applications.

Teams that are defined as agile are less likely to deploy their applications once or multiple times a day.

Internal APIs are being tested against more than external or third-party APIs.

Test stability, object identification, and test maintenance are the top three challenges respondents face when automating their UI tests.

Having APIs return the correct response is the top concern API testers have with testing APIs and web services.

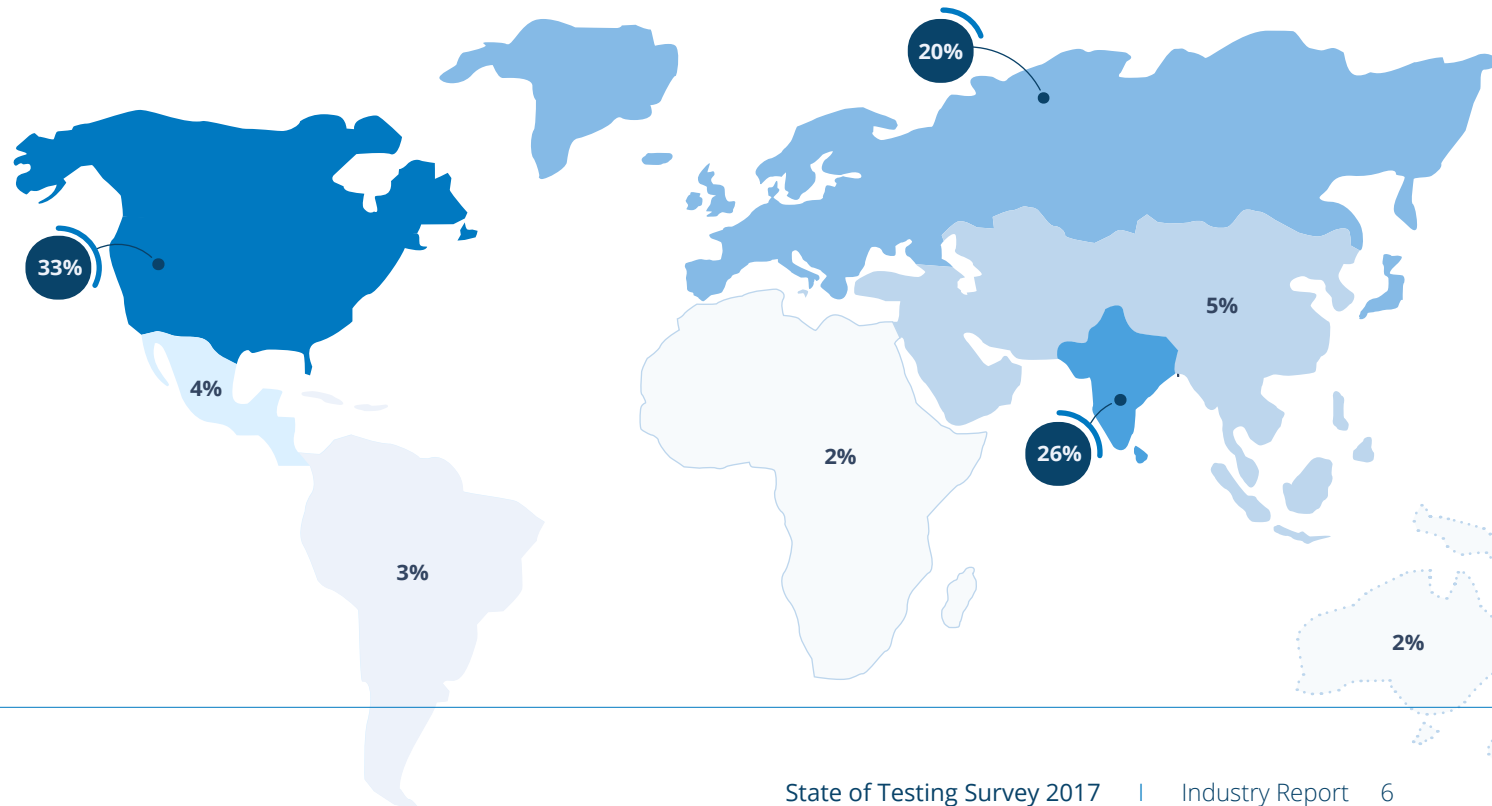
Testers expect test automation to outpace manual testing in the next two years.

# Demographic & Firmographic Data

SmartBear Software's 2017 State of Testing Survey includes responses from over 3,400 individuals working across all facets of the software testing industry. This section provides further insights into the participants of the survey – the countries they work in, the industries they represent, their company sizes, team dynamics, and their individual roles.

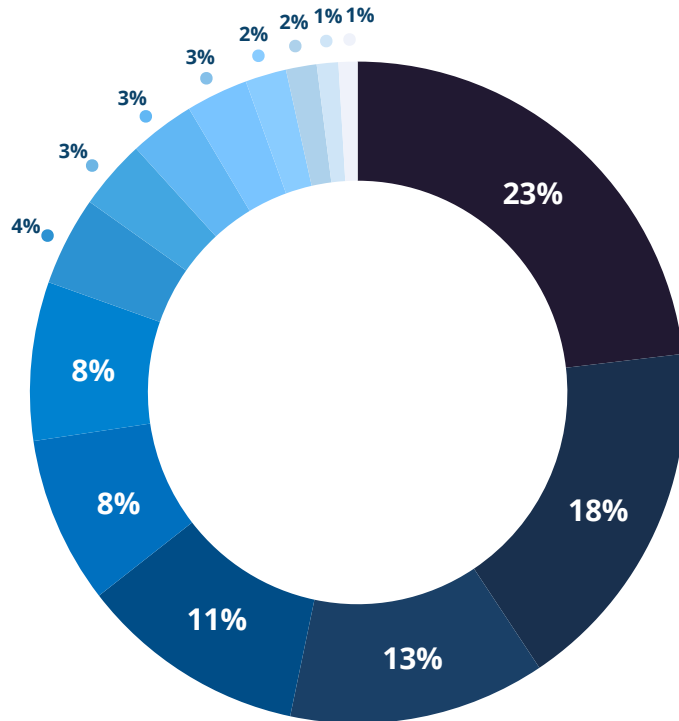
## Thousands of individuals worldwide participated in the State of Testing survey

- North America (US & Canada)
- India
- Europe (Including Russia)
- Asia
- Other
- Latin America
- South America
- Africa
- ⊕ Australia & New Zealand



n = 3457

Respondents work in a wide variety of businesses that span more than 15 different industries.

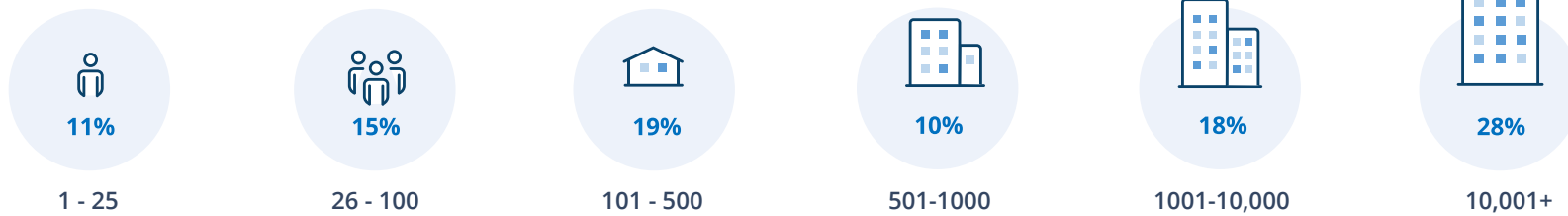


#### Industry

- Computer Hardware, Software, or Consumer Electronics
- Finance, Banking, Insurance
- Other
- Internet or Web Services
- Health Care Services
- Telecommunications
- Automotive, Transportation
- Retail or Wholesale Trade
- Government, Aerospace, Defense
- Media, Advertising, Communications
- Education
- Pharmaceuticals and/or medical devices
- Gaming
- Non-profit, non-governmental organizations

n = 3302

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



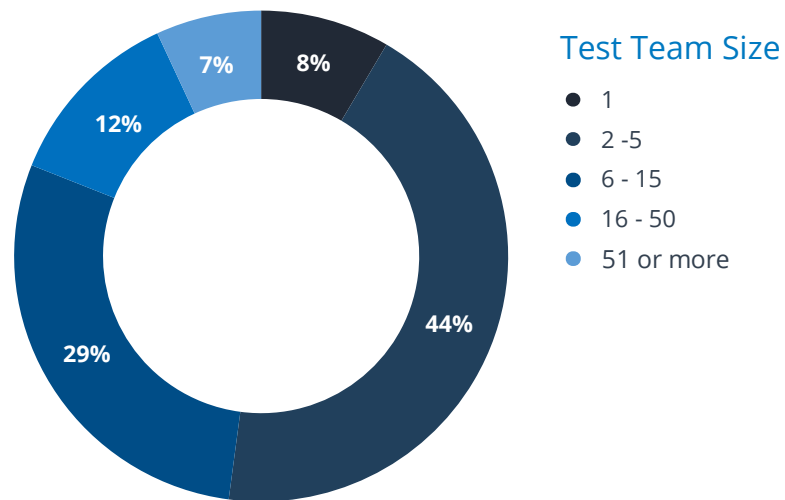
n = 3446

## The majority of development teams have 2 to 15 people that also test software.

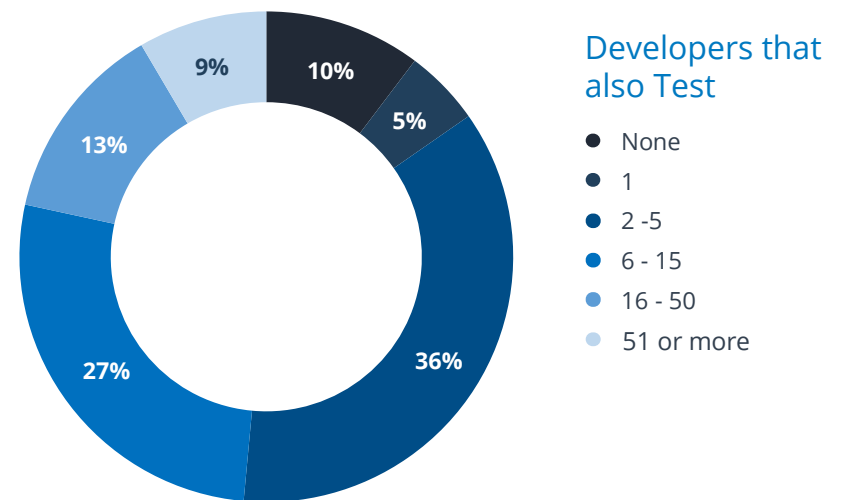
Forty-four percent (44%) of **testing teams** have 2 to 5 people and more than half of **development teams** have 2 to 15 people that are also responsible for testing (63%). With the growth of methodologies like Agile and DevOps, testing teams have been shrinking and the line between roles increasingly blending.

An interesting indicator of this trend is that of the companies that employ 10,001+ employees, more than half of the teams' sizes remain under 15 people.

- | 33% of teams have 2 to 5 people.
- | 33% of teams have 6 to 15 people.



n = 3432

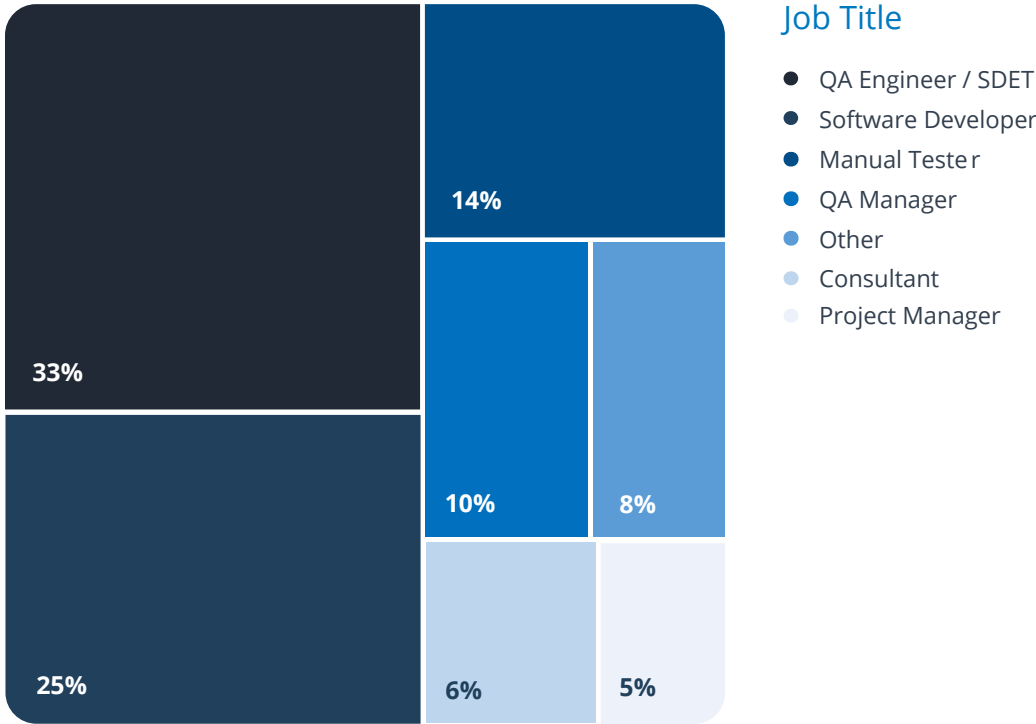


n = 3447



# The survey audience works in a diverse set of job functions and levels, including Software Engineers, Developers, and QA Managers.

Thirty-three percent (33%) of respondents are QA Engineers or Software Development Engineers in Test (SDET). 'Other' job titles represented include UI Developers, Business Analysts, and Automation Engineers.



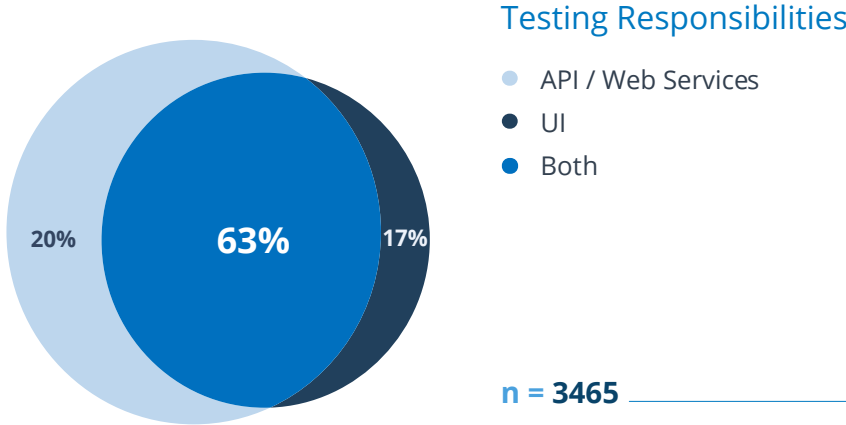
n = 3452

# The majority of testers are responsible for both API and UI Testing.

One of the most notable discoveries from the survey was that the majority of respondents said they are responsible for both API and UI testing (63%).

Based on job title, there is a small shift in the focus of testing responsibilities. Developers, Project Managers, and Consultants are more likely to test only APIs, while Manual Testers and QA Managers are more likely to test only the UI of an application. Of the QA Engineers that participated in the survey, 68% said they were responsible for both back-end API and UI testing.

This forces us to put the growth of methodologies like Agile or DevOps, and trends like 'shifting left' into perspective. Moving forward through the rest of the survey, readers should note that much of the data presented refers to the responses from the individuals whose job it is to conduct both API and UI testing. In some key instances, this differs from the responses of those who focus strictly on one or the other.



## Testing Responsibilities by Job Title

	QA Engineer/ SDET	Software Developer	Manual Tester	QA Manager	Other	Consultant	Project Manager
API / Web Services	14%	37%	11%	7%	20%	31%	21%
Both	68%	53%	62%	77%	65%	56%	66%
UI	19%	10%	28%	17%	14%	12%	13%
<b>Responses</b>	<b>1,130</b>	<b>866</b>	<b>335</b>	<b>335</b>	<b>270</b>	<b>216</b>	<b>158</b>

**n = 3451**

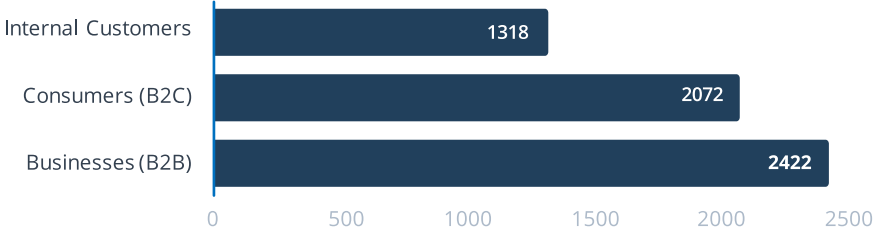
# Software Testing

## B2B software is tested more than B2C software.

Seventy percent (70%) of developers and testers are working on software intended for businesses, whereas only 60% are working on applications intended for consumers.

n = 3456

### Intended End Users



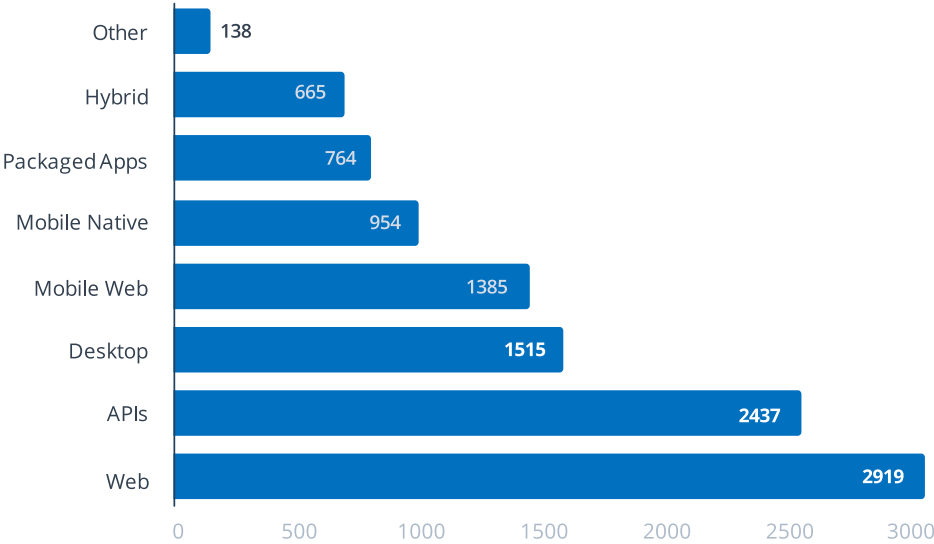
# Web applications are the most common type of application being tested.

Web technologies have gained a significant amount of traction among businesses and consumers over the years, and there is no sign of this trend slowing down. To meet the growing demand for an online presence, an impressive number of organizations and test teams are building applications for the web today.

Eighty-five percent (85%) of survey respondents are developing and testing web applications.

n = 3455

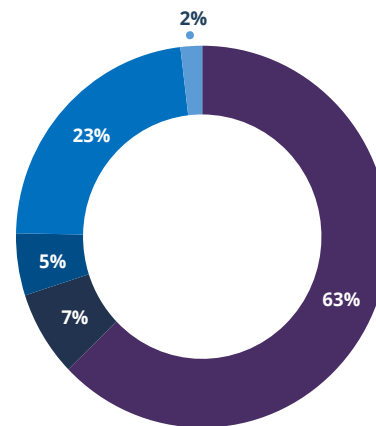
### Types of Applications Being Tested



# Development Models

## Agile is the dominant model for software development.

The majority of teams are embracing an agile approach to software development (63%). While Waterfall is still being used by 7% of testing teams, it is outpacing teams following a DevOps model (5%). Most other teams will vary their development process based on the project at hand (23%).



What software development model does your team follow?

- Agile
- Waterfall
- DevOps
- Depends on the Project
- Other

n = 3406

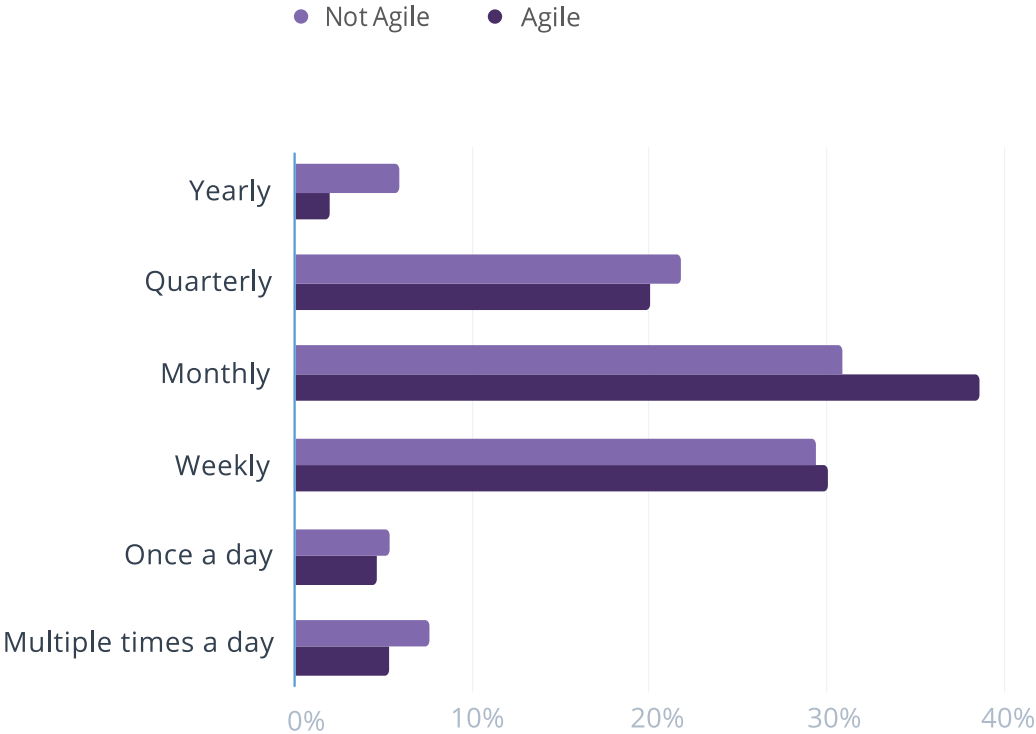
# Teams are not as agile as they think they are.

While most respondents say their teams are following an Agile model, many don't implement some key agile best practices. Team sizes are small and over half use a continuous integration (CI) server - the most popular being Jenkins (57%). However, a key indicator of true agility is the ability to deploy on a frequent and regular schedule.

The definitions of Agile and DevOps have typically been considered synonymous with rapid delivery cycles. While teams that follow a DevOps model are more likely to release multiple times a day, more than half of the teams that labeled themselves as 'Agile' are releasing applications on a monthly, quarterly, or even yearly basis. Interestingly, there was no significant difference between release cycles for teams that follow an Agile model versus teams that follow a Waterfall model. In fact, teams that follow other models (including 'It Depends on the Project,' 'Waterfall,' and 'Other') are more likely to release daily or multiple times a day than agile-labeled teams.

This is interesting as it reveals what teams think it means to be agile and how this deviates from the current definition. Teams that vary their development processes by project for example, are seemingly more 'Agile.' Their approach is inherently flexible and it drives towards the release cycle the agile ideology teams strive for.

Agile Release Cycle

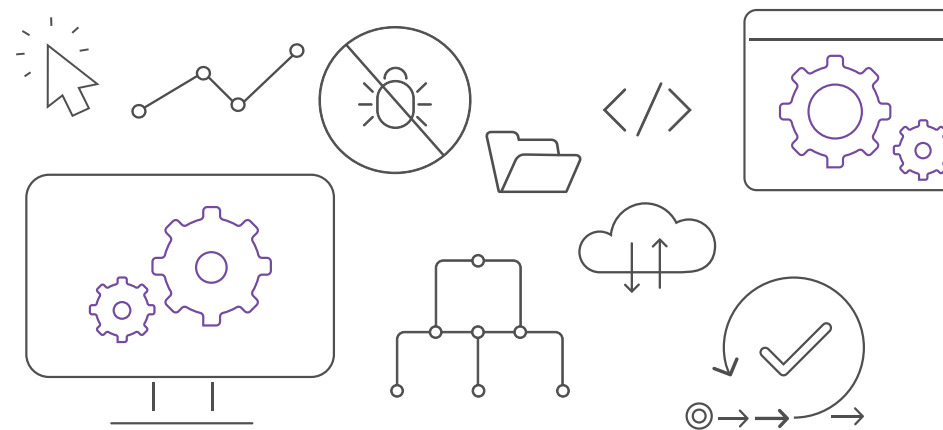


n = 1085 n = 2136

# Testing Approaches & Methodologies

**Unit testing, automated UI testing, and performance testing are the most commonly used testing techniques.**

There are a number of testing techniques at the disposal of QA teams today to help them ensure they deliver high-quality applications quickly. With so many available, it can be hard to choose which ones to utilize. Testing as a whole drives towards the goal of delivering a positive user experience, but each testing technique focuses on specific pieces of an application that will require different amounts of time, skill sets, and tools to implement.



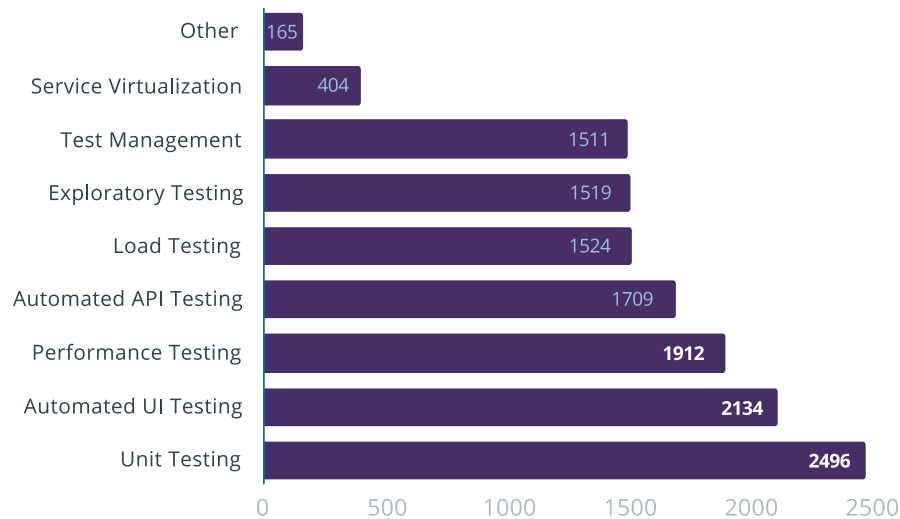
Exploratory testing is a common manual testing practice used to check systems on the fly without previously designed test cases. According to the survey, mid-sized companies employing 101 to 500 people are more likely to conduct exploratory testing than small companies with 1 to 25 employees, or large companies with over 10,001 employees.



Teams are using a wide distribution of testing practices to ensure their applications behave and look as expected, but the top three most used are unit testing (72%), automated UI testing (62%), and performance testing (56%).

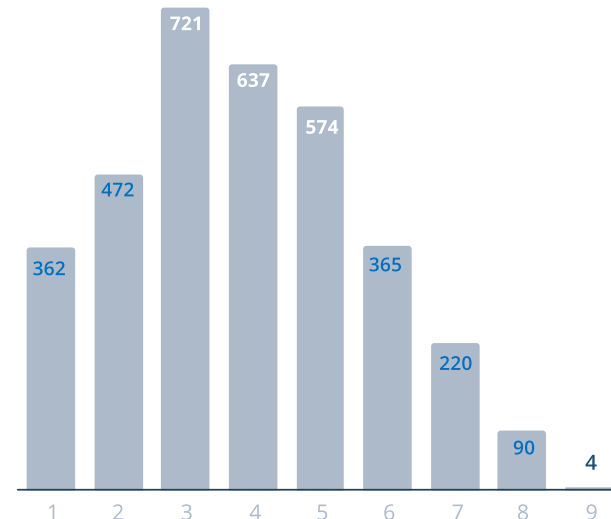
In fact, the majority of teams are using 3 to 5 techniques. By focusing their efforts on a few key practices, teams can maximize the value they derive from their testing process. The most common 'Other' types of testing reported being used by respondents included security testing, regression testing, and manual testing.

### Test Practices



n = 3445

### Number of Test Practices Used



n = 3445



## Less than half of teams practice test management.

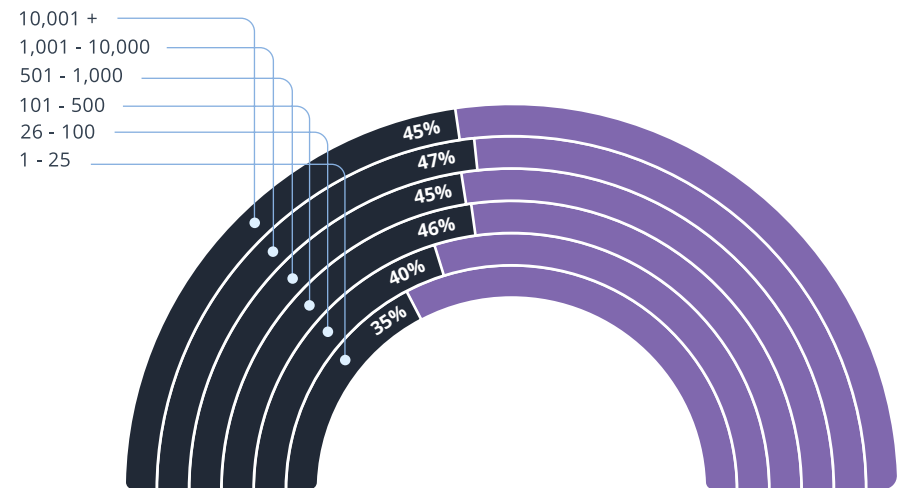
Only 44% of teams practice test management. Test Management is a process that can help teams track test cases, whether they are automated or manual, and link them to requirements, tasks, and defects. While the definition of test management can range from tracking test cases in an excel spreadsheet, to using a project management tool such as JIRA or a standalone test management platform, the fact that less than half of respondents practice it suggests that there is potential for teams to increase efficiencies and test coverage.

Another interesting finding is that there was no significant difference in adoption of test management by team size, as the team size distribution of respondents who practice it is similar to the survey audience as a whole. This counters any notion that test management is used strictly by larger organizations.

As testing becomes increasingly automated, and practiced by members of the extended development team beyond pure software testers, it will be interesting to see if test management becomes more widely adopted in future years.

## Use of Test Management

● Test Management ● No Test Management



n = 3445

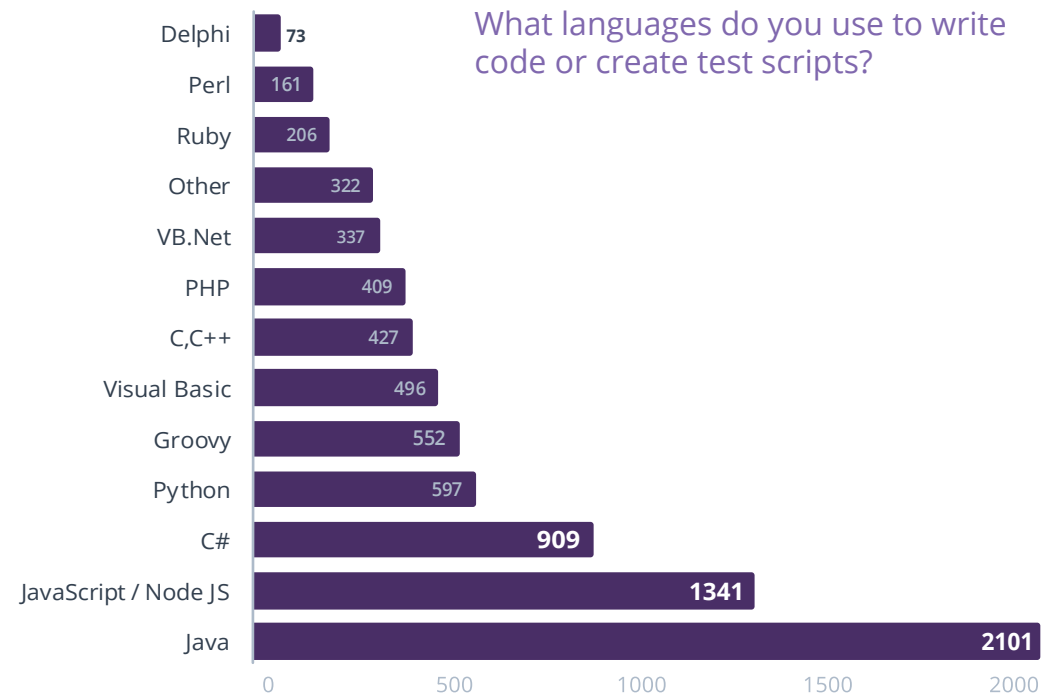
## Java reigns as the most-used scripting language.

Java is the most popular scripting language among participants, with 62% of respondents using it to write code or create test scripts. Java, PHP, Python, and Groovy are the languages most likely to be used among API testers. Survey participants responsible for both API and UI testing are more likely to use Python and JavaScript.

More than half of the survey's testers also reported using only one or two scripting languages (65%). For those who use multiple languages, the most popular pairings included:

- | Java / Javascript
- | Java / Groovy
- | C# / Javascript

What languages do you use to write code or create test scripts?

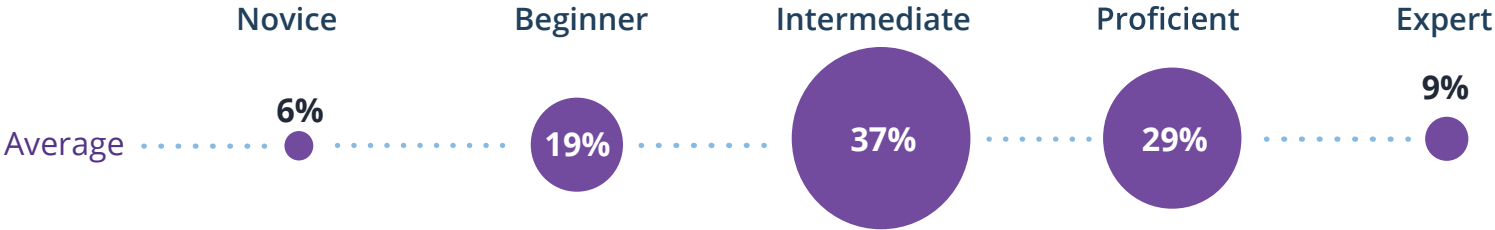


n = 3400

# Testers reporting a higher proficiency in scripting work at larger companies.

Close to one-third of the survey's testers rated their scripting ability as intermediate and less than half described their scripting skills as proficient or expert-level. Interestingly, the larger the company, the higher the testers rated their skill in scripting.

## Script Ability by Company Size



Company Size	Novice	Beginner	Intermediate	Proficient	Expert
1 - 25	8%	27%	34%	24%	8%
26 - 50	8%	20%	38%	27%	7%
101 - 500	7%	22%	39%	25%	7%
501 - 1000	6%	18%	37%	29%	10%
1,001 - 10,000	5%	18%	37%	31%	10%
10,001 or more	4%	15%	38%	32%	10%

n = 3410

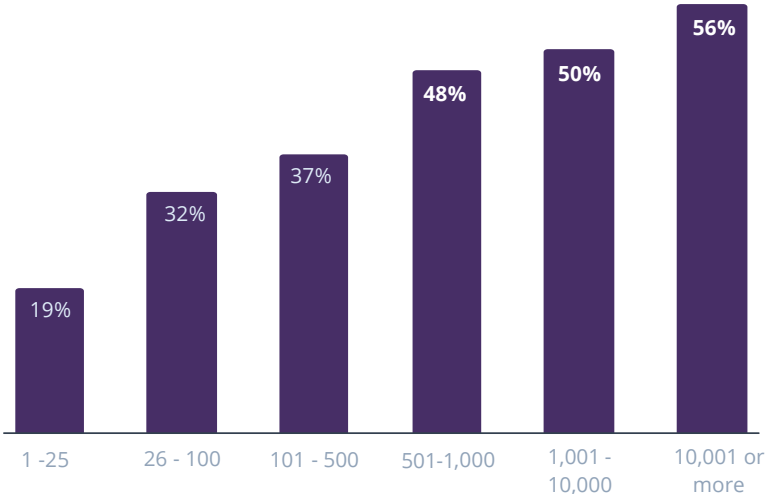
# Test Automation

## Less than half of all tests are automated.

Over the past few years, automated testing has been steadily adopted by companies around the world to improve test efficiency and coverage, but there is still room for teams to expand their use of the practice. Survey respondents noted that on average, only 44% of their tests are automated.

Large companies with 10,001 or more employees automate more of their testing on average (56%) than small companies with 1 to 25 employees (19%).

Percentage of Tests Automated (Average) by Company Size



n = 2946

# Handling frequent changes to an application is the top challenge with test automation.

The number one challenge testers face with automating their tests is the frequency with which the functionality of their applications change. Constant modifications to an application under test will force teams to alter their test scripts and cases, run additional tests, and can cost an exorbitant amount of time and money.

It's no surprise then, that teams responsible for testing the APIs or the GUI of their applications spend a significant amount of their time either executing tests or writing test cases. Although the majority of testers are responsible for both API and UI testing, those who test exclusively at the services layer tend to automate more tests than those who focus on the UI layer only.

Percentage of Tests Automated (Average) by Testing Responsibilities



n = 2958

## Top Challenges of Test Automation

- The application function changes frequently
- Not enough time to test
- Employees lack the right skill sets
- Managing test cases
- Don't have the right test environments available
- Delivery process doesn't support automation
- Don't have the right tools

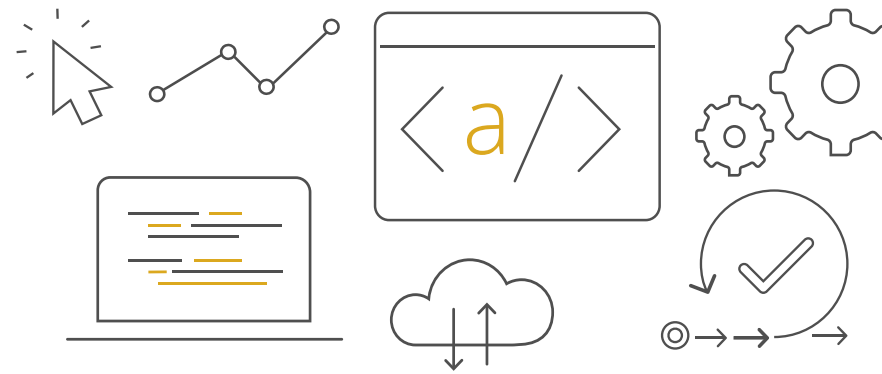
# API Testing

Over the past few years, you've been hearing the phrase "software is eating the world." From smart watches to cars, millions of products use APIs, and with the explosion of SaaS, mobile device, and web applications, there has been a growing demand for interconnectivity. APIs play a critical role in providing an easily maintainable way of connecting software. They are the legos that link together and connect the vast universe of applications that make up the Internet of Things (IoT). This next section dives into the current state of API testing and uncovers the biggest challenges developers and QA professionals face.

## Nearly three-quarters of teams are testing against internal APIs.

Over 80% of survey participants reported they were responsible for testing APIs. Eighty-two percent (82%) of those that test APIs use under 50 unique endpoints in their daily testing and development.

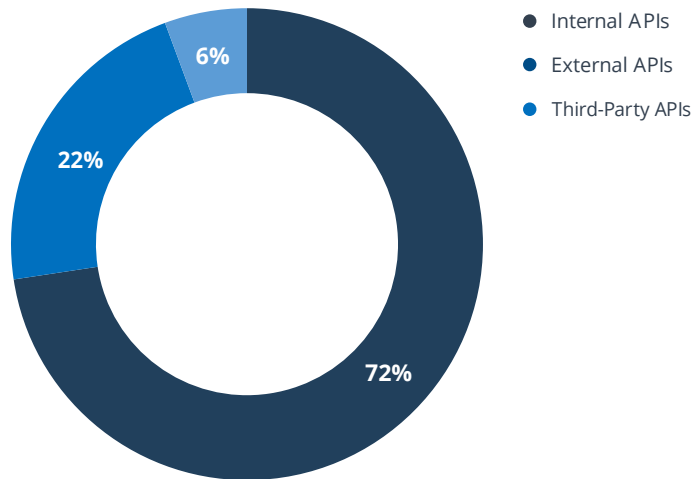
As expected, the number of endpoints used in daily testing and development increases as company size or test team size increases.



The majority of respondents test against internal APIs, with only 27% of respondents reporting they primarily test against external APIs (those developed by their team for use by someone else), or third-party APIs (those developed by other companies for use by their team). In fact, only 6% of the survey's respondents said they primarily test against third-party APIs.

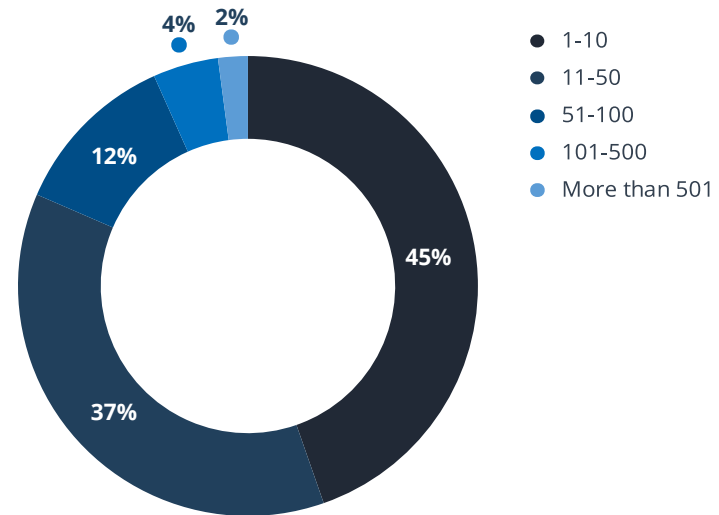
Using third-party APIs for testing can be notoriously expensive and difficult, as companies can charge teams each time they use the API. Companies also may not provide the sandboxes teams need to test against their APIs, meaning those APIs may not always be available. As the use of APIs continues to grow, it's important to note that the majority of APIs being tested are not intended for public consumption.

### Which type of API do you test against most frequently?



n = 2815

### API / Web Service Endpoints Used in Daily Testing and Development.



n = 2779

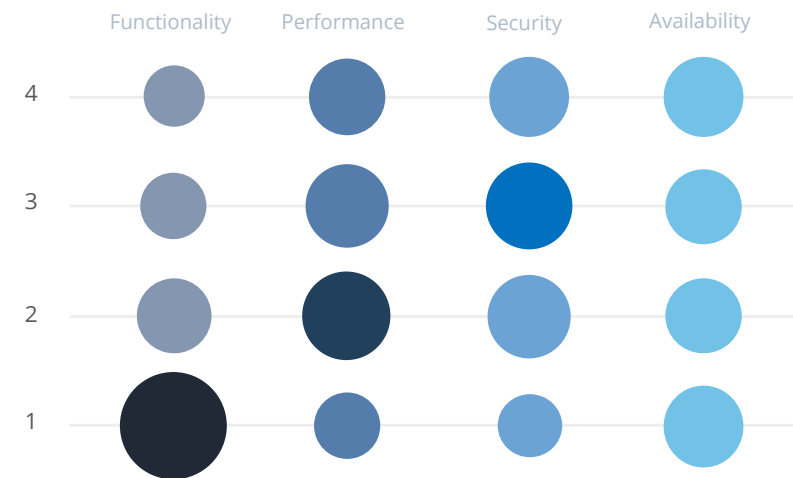
## Ensuring APIs return the correct response is the top concern respondents have with testing APIs and Web Services.

Regardless of testing internal or external APIs, the primary concern API testers have is whether or not their APIs will return the correct response every time. As a result, it wasn't too surprising to discover that API testers spend more time conducting functional API testing than creating mocks, load testing, or virtualizing services.

- **Functionality:** If my APIs return the correct response every time.
- **Performance:** Whether or not my APIs work under different load and traffic conditions.
- **Security:** Knowing my APIs are safe from hackers and attacks.
- **Availability:** Ensuring my APIs are always available to my users.

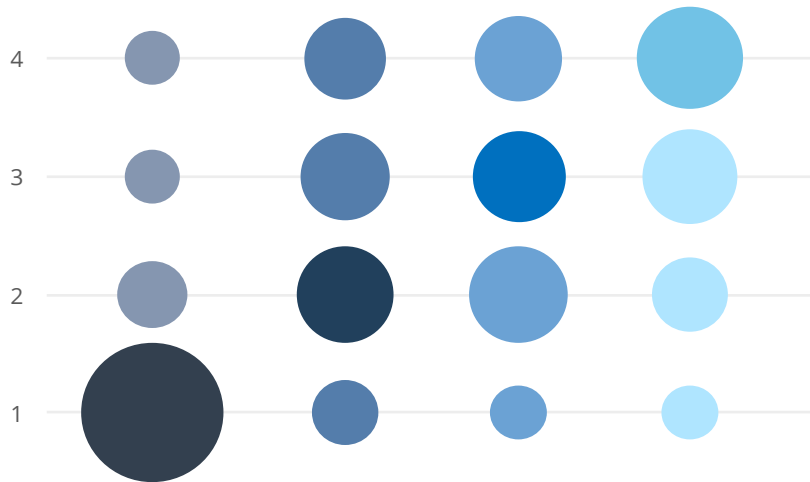
## Top Concerns in API Testing

1 = Most Concerning    4 = Least Concerning



n = 2337





## Time Spent on API Testing Practices

1 = Most Time Spent 4 = Least Time Spent

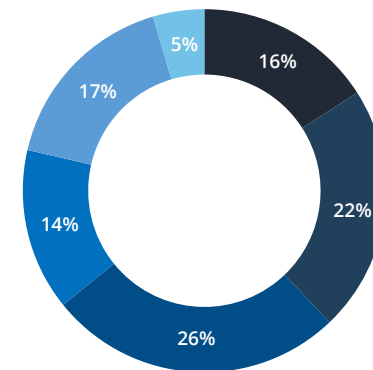
- Functional Testing
- Creating Mocks
- Load Testing
- Virtualizing Services

n = 2520

## Unavailability is the biggest challenge testers face when testing against third-party APIs.

The biggest obstacle testers face when testing against third-party APIs is that they aren't always available. In the instances where an API isn't available, 26% of testers will create mocks using a script or code-based tool in order to mimic those APIs, while 16% of respondents will do nothing. Seventeen percent (17%) of API testers also noted they had no idea they could do something about unavailable APIs.

## Response to Unavailable Third-Party APIs



- Do nothing
- Create mocks on a UI-based tool to mimic APIs
- Create mocks on a script/code-based tool to mimic APIs
- Use a service virtualization solution to emulate APIs
- I had no idea we could do something about unavailable APIs
- Other

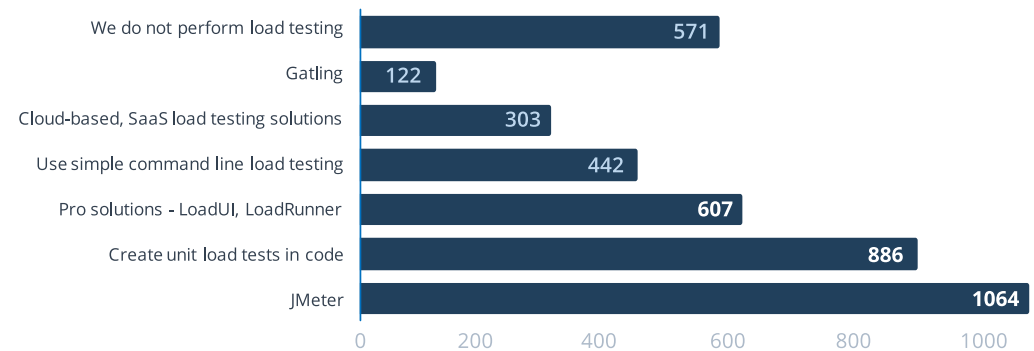
n = 2781

## Using JMeter is the preferred method for API load testing.

The most commonly used methods for API load testing are using JMeter or creating unit tests in code, while 17% don't load test their APIs at all. It's important to note, that 40% of API testers who conduct load testing will load test every release cycle.

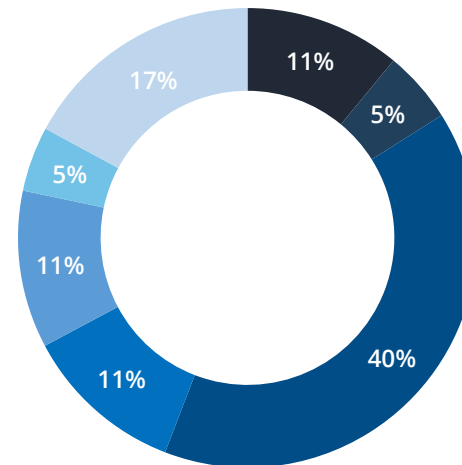
APIs are often the cornerstone of service-oriented architecture (SOA), so comprehensive API performance testing will ensure that APIs can handle the load that real users will encounter in production. With only 11% of respondents conducting API load testing every time code is checked in, there is plenty of room for teams to improve API performance.

## Methods for Load Testing APIs n = 2776



## Load Testing Frequency

n = 2797



- Every time code is checked in
- Every day
- Every release cycle
- Every other release cycle
- Every quarter
- Every year
- Never

# 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 are often blind 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 ever reaching the UI.

UI testing is challenging though, as an application's visual elements can change frequently. With users adopting new technologies daily, the types of software that need to be tested are also transforming. This next section 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 created using scripts.

Many automated testing tools today 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.

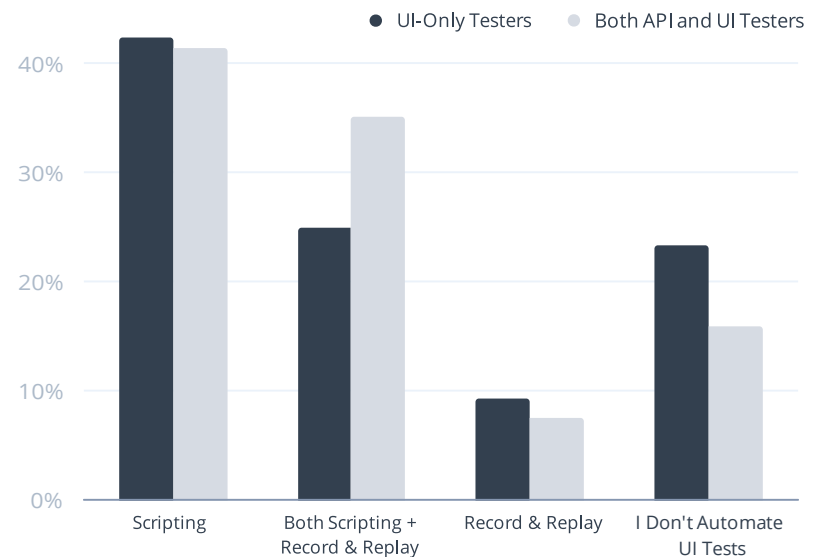


Despite the increased emphasis on record and replay tools, UI testers are more likely today to write scripts to create UI tests. In the next few years, it will be interesting to watch if this dynamic changes and what impact it will have on both record and replay tools as well as more script-heavy tools and open-source frameworks such as Selenium.

By comparison, survey respondents responsible for both front and back-end testing are more likely to use a combination of both scripting and record and replay to create tests. Sixteen percent (16%) of respondents don't automate their UI tests at all, meaning there is still plenty of room for teams to improve efficiency by taking advantage of the benefits of automated testing tools.

- | **8%** of testers use record and replay as their sole method for creating UI tests.
- | **42%** of testers use only scripts to build UI tests.
- | **35%** of testers use a combination of scripting and record and replay to develop UI tests.

## Method of UI Test Automation by Testing Responsibilities

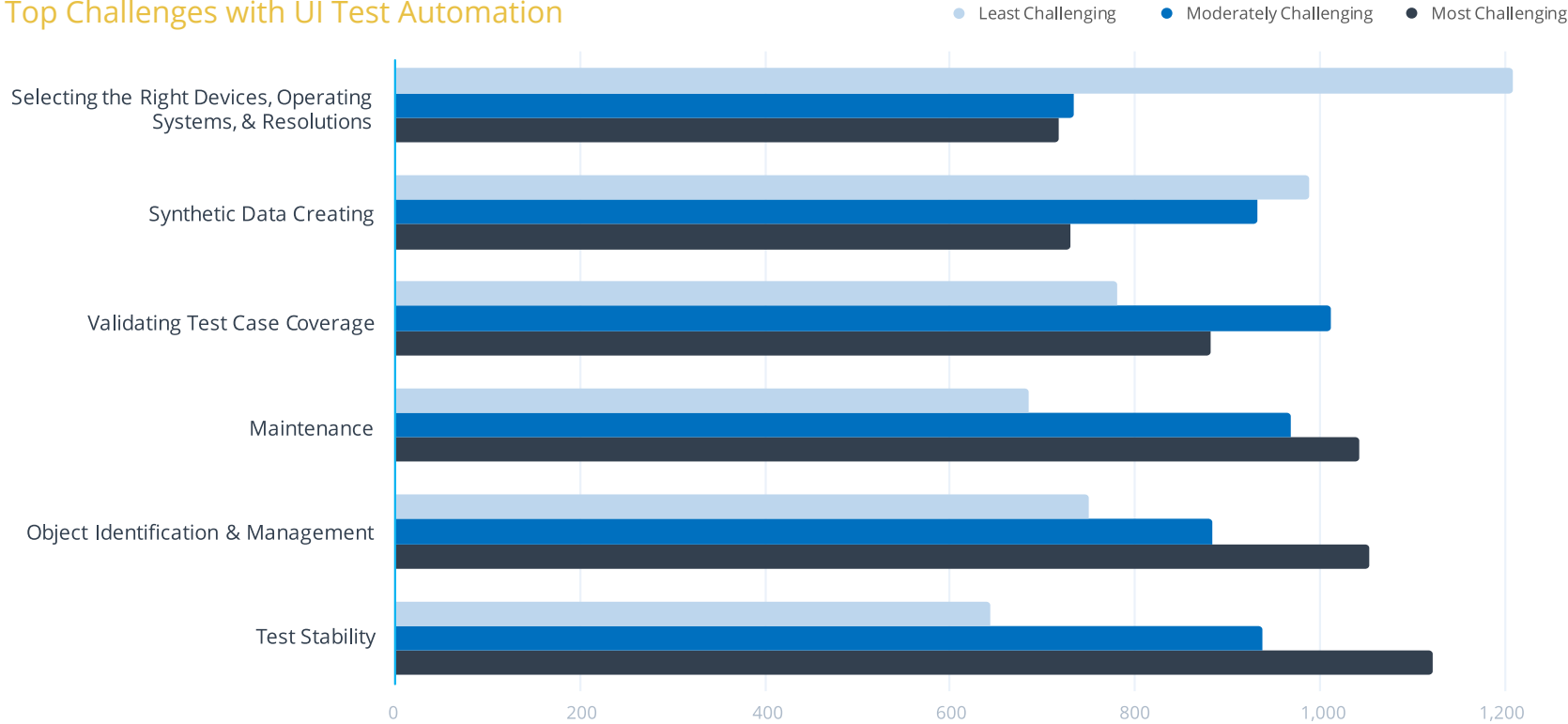


n = 2725

# Test stability, object identification, and test maintenance are the top automated UI testing challenges.

UI tests are flaky. The slightest change to an application's function or user interface can cause UI tests to break, resulting in additional maintenance overhead. So it was no surprise to discover that the top challenges testers faced with automating their UI tests were test stability, followed closely by object identification, and test maintenance.

## Top Challenges with UI Test Automation



n = 2704

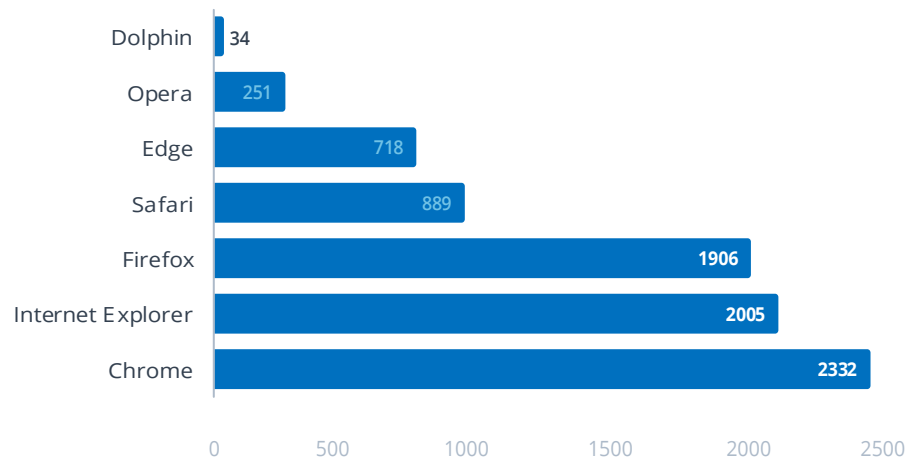
UI testers and those responsible for end-to-end testing agree that their top challenge is test stability. With the explosion of IoT devices, this problem won't get any easier. Already, tools have been looking to enhance UI test stability by providing stronger name mapping or built-in object recognition capabilities. In the coming years, it will be essential for software testing tools to transform and address these challenges.

## Four out of five teams test on multiple browsers, with half testing exclusively on the latest version.

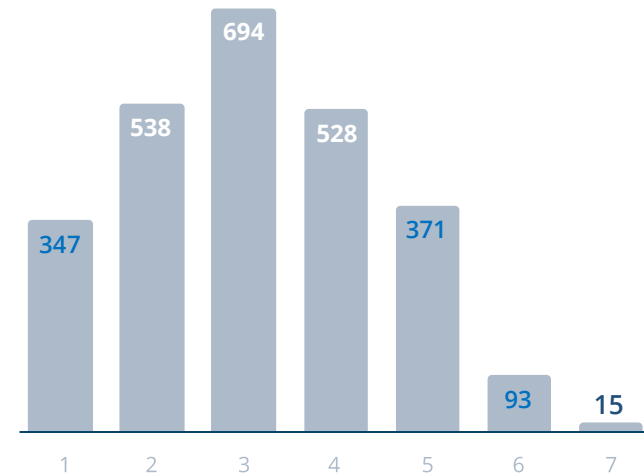
Chrome is the most common browser to test applications on today, followed closely by Internet Explorer and Firefox. Many providers, like Chrome, will include automatic updates to browser versions unless specified, so it's understandable that more than half of QA teams will focus exclusively on testing the most recent version (55%).

Although testing across multiple browsers and versions can be tedious due to the number of combinations available, even with automated testing, we found that 65% of survey respondents are testing on 2 more more browsers. Even with respondents testing across multiple browser types, there is still plenty of room for teams to expand test coverage by adding older browser configurations to their arsenal.

### Browsers Being Tested



### Number of Browsers Being Tested



n = 2586

## Two-thirds of participants test on mobile devices.

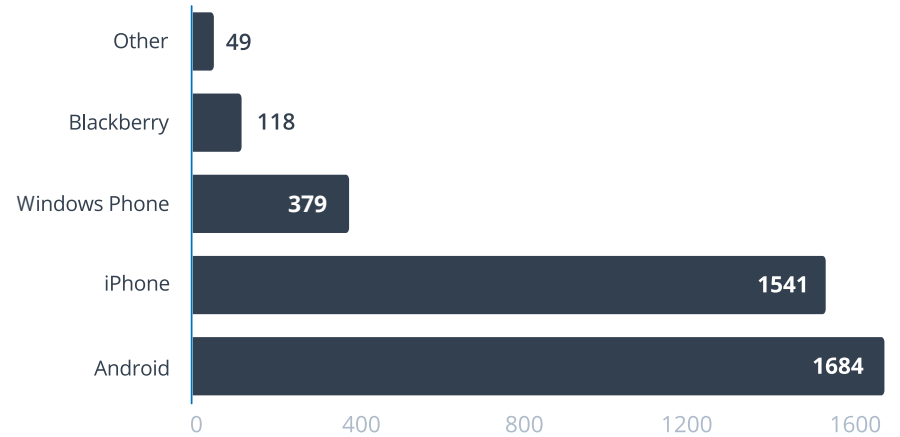
The mass adoption of smartphones has intensified the need for mobile UI testing, so it was no surprise to find that two-thirds of survey respondents reported testing on mobile devices. Eighty percent (80%) of survey participants reported testing on more than one mobile device, with the most commonly tested devices being iPhones and Androids.

In addition to browsers and browser versions, testers now need to ensure their applications function properly on a myriad of devices, operating systems, and device versions. In contrast to web testing, 64% of mobile UI testers **do not** exclusively test on the latest configuration of mobile devices.

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 (e.g., Samsung, Motorola, Nokia), 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.

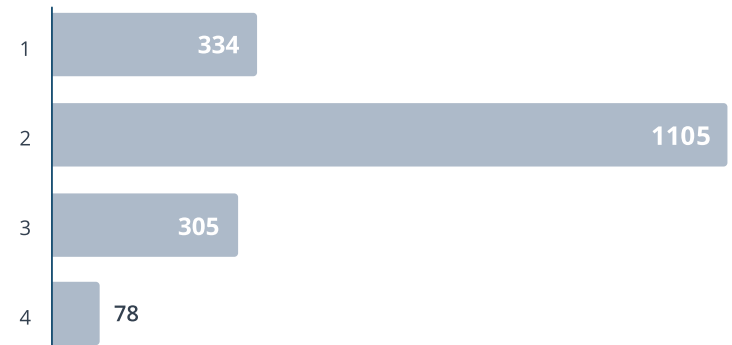
The volume of tests required to test against multiple devices and device versions, alongside the fact that 55% of mobile testers reported using manual testing, means a significant amount of resources are dedicated to mobile testing. Android fragmentation only exacerbates this issue, and with 36% of testers still focusing on the most recent device iteration, this indicates that teams are missing a giant sector of their user base as well as significant opportunities to improve efficiency.

## Mobile Devices Being Tested



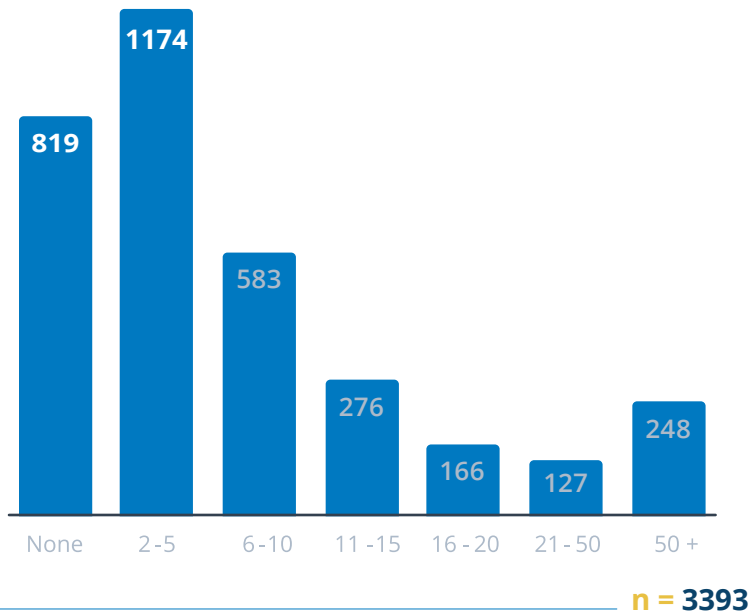
n = 1822

## Number of Mobile Devices Being Tested



n = 1822

## Number of Tests Run in Parallel



## Parallel and Cloud Testing Use

	No cloud testing	Some cloud testing
No parallel testing	13%	0%
Some parallel testing	21%	56%

n = 2486

## Web and mobile testers are more likely to conduct parallel testing.

Running tests concurrently on multiple devices expands test coverage and decreases testing times, which alleviates the challenges with testing across multiple environments. Today, the majority of testers are running only a few tests in parallel, or none at all. Unsurprisingly, participants who do web testing, either desktop or mobile, are more likely to conduct parallel testing (77%) than those who do not (66%).

- 35% of respondents are running 2 to 5 UI tests in parallel.
- 23% of respondents are not running any UI tests in parallel.

Digging deeper, participants responsible for end-to-end testing are 10% more likely to do parallel testing than those who focus on just UI or API testing. It will be interesting to watch how the continued expansion of devices, browsers, operating systems, and even resolutions will affect the software testing industry. A growth in the use of parallel testing could push paid test tools to enhance concurrent testing capabilities, or it could lead to the growth of open-source or cloud-based test tools that instantly provide the necessary test environments.

## Less than one-third of tests are run in the cloud.

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.

- On average, 30% of tests are run in the cloud.
- 34% percent of testers reported not running any tests in the cloud.
- Testers who conduct parallel testing are also more likely to run tests in the cloud.



# The Future of Testing

## API and manual testers expect test automation to grow faster than UI testers and managers.

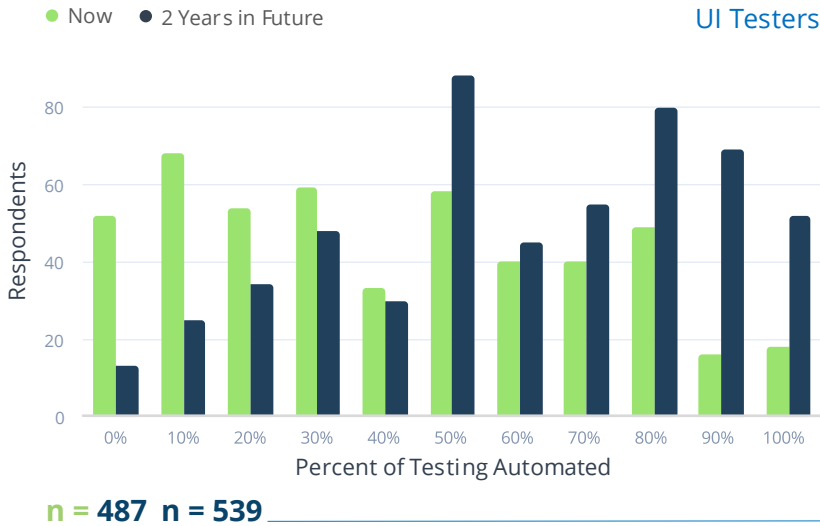
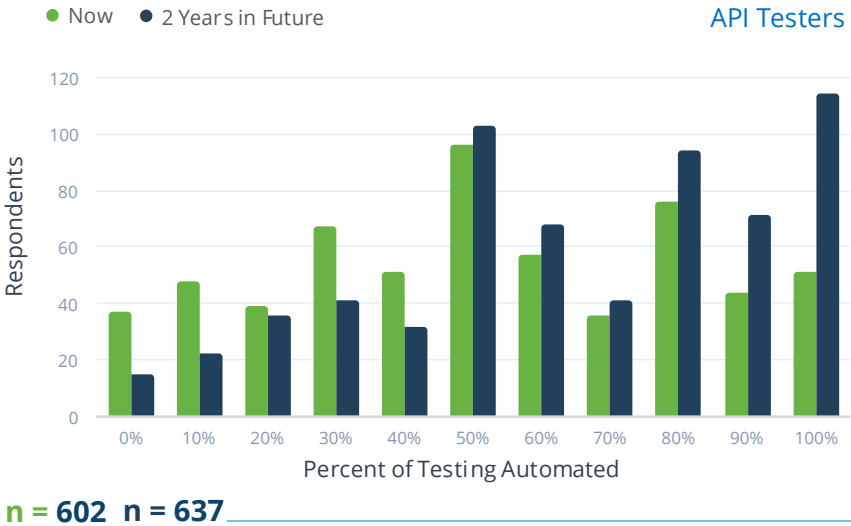
Manual testing is still a common practice used today by testing professionals to gauge the quality of software applications. While there has been much discussion over its purpose, especially for practices like exploratory testing, there seems to be little doubt that over the next two years, automated testing will become more prevalent.

**62%** of all survey respondents think that the percent of automated testing used will increase.



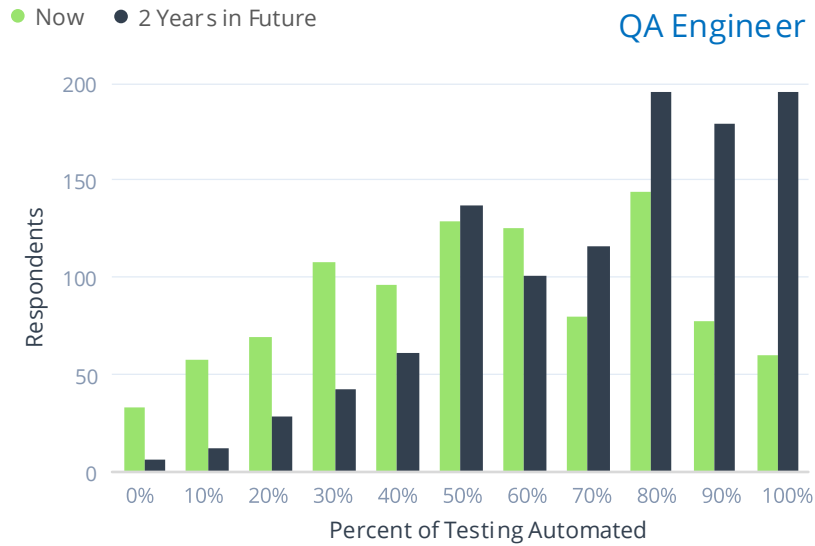
An interesting insight to look at is the side-by-side comparison of the percent of tests that are automated today versus what percent testers think will be automated in the future. Between API and UI testers, API testers currently automate more of their tests and they expect to be automating more than UI testers in the future. The expected shift to a predominantly automated process will be slower for UI testers.

### Future of Test Automation by Testing Responsibilities

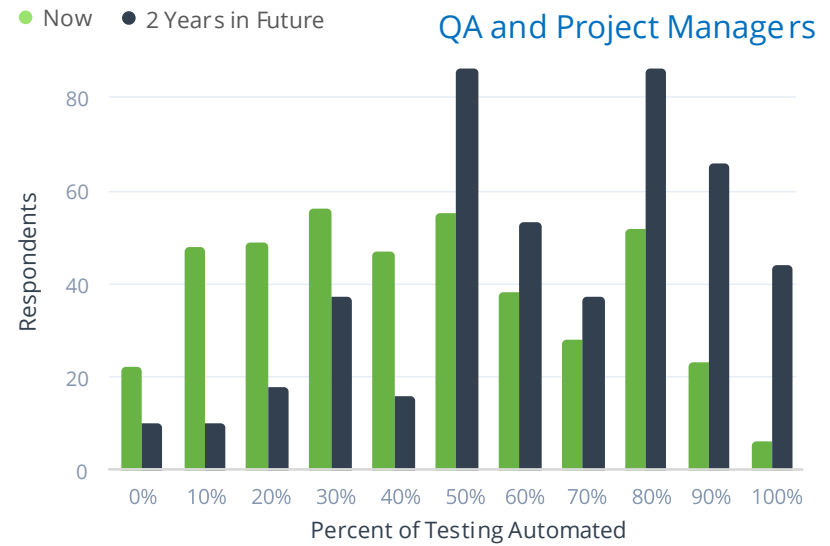


Compared to QA Engineers, QA Managers have a more tapered expectation of the growth of test automation. In two years, 50% of QA Engineers expect 80 to 100% of their tests to be automated, while 43% of QA Managers expect 80 to 100% of their tests to be automated.

### Future of Test Automation by Job Title



n = 979 n = 1072



n = 424 n = 463

## Delivery cycles will get even shorter.

More than half of respondents want to have shorter delivery cycles in the next two years. Sixty-eight percent (68%) want to deploy weekly, daily, or multiple times a day, as compared to the 41% who do so already. The demand to speed up testing and delivery will continue to drive the growth of Agile practices and tools, and could be the key driver behind the expected shift to a predominantly automated testing process.

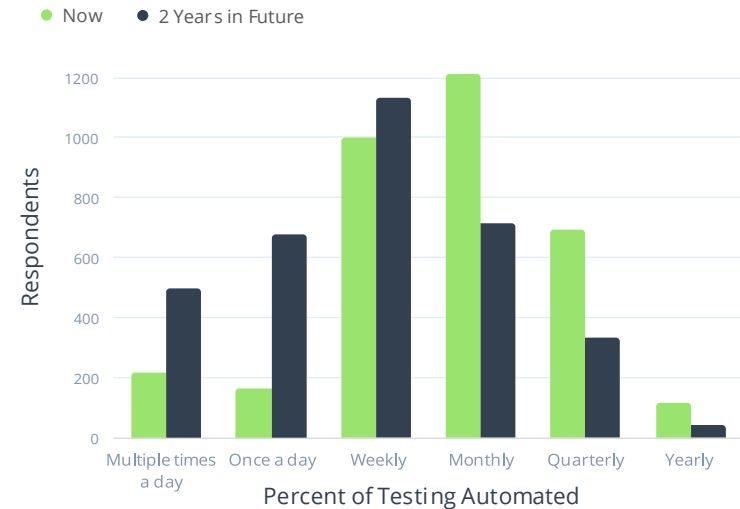
## Mobile web, native, and hybrid application testing will become a high priority.

Respondents expect to see the largest growth in mobile web and native application testing, as well as a spike in hybrid application testing. Testers also expect to test desktop applications less in two years (37%) than they do now (46%).

It will be interesting to see how the expected growth of web and native application testing will impact the testing and development tools in the industry. This growth could be the driving force behind high adoption rates for open-source frameworks that already successfully test mobile web apps such as Selenium and Appium.

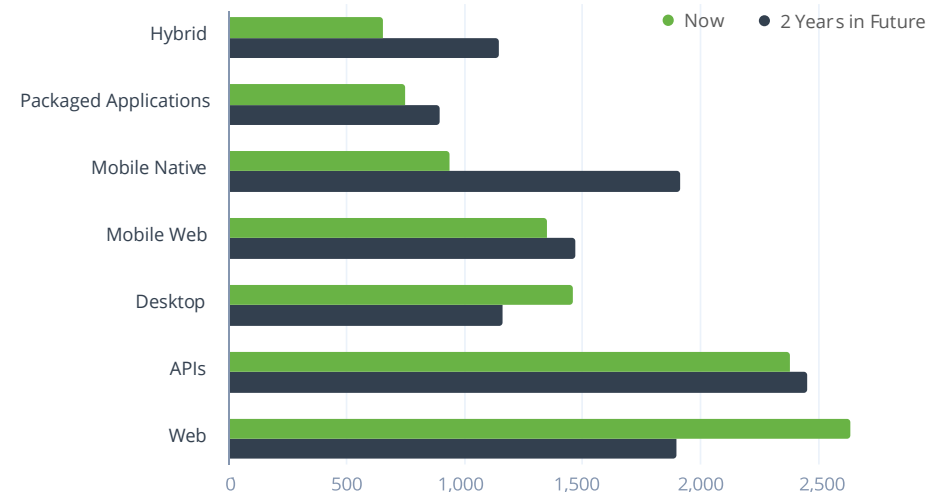
Additionally, this growth could drive companies to focus on building and testing progressive web applications. Progressive web apps combine some of the best aspects of web and native applications in a manner that enhances usability. They work for users regardless of browser choice, are easy to install as they aren't stored physically, and since they're accessed through browsers, will work across numerous devices, including desktops, mobile devices, and even tablets. A key benefit for companies is that progressive web applications also don't require app store approval as they behave like a website.

## Future of Release Cycles



n = 3396

## Future of Application Types Tested



n = 3141



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

The one open-ended question we asked our respondents was “How do you think the software testing field will look in two years?” We wanted to see what testers and developers at every level of their organizations think about the future of the industry and how they might fit in. This next section highlights some of the most interesting thoughts and perspectives of our respondents. Their responses reveal the vastly different mindsets of individuals throughout the community.

**“It would look more flexible and adaptable to changing software demands. Better techniques will be used to test a system. AI will be an important part in software testing tools and many common test strategies can be detected by tools themselves. Automated test script creation will become creative and less time consuming. Nevertheless, manually testing a software will not become obsolete.”**

-QA Engineer / SDET

**“Wizards and Robots will take place.”**

-QA Engineer / SDET

**“100% automation.”**

-QA Engineer / SDET

**“Automated testing will dominate.”**

-Software Developer

**“Manual testing will make a resurgence following a reality check that not all tests can or should be automated.”**

-Manual Tester

**“I think testing will push back from the trend of “automate everything” due to diminishing ROI and because stakeholders do not see automation solving IT’s problems. Stakeholders will want skilled humans back in the pipeline.”**

-QA Manager



**“More API testing, less UI.”**

-Application Architect

**“UI testing becomes more important. API testing should be mature.”**

-QA Engineer / SDET

**“More cloud, exclusively web applications, scalability will be the biggest challenge for every application out there.”**

-QA Engineer / SDET

**“Cloudy.”**

-Business Analyst

**“I hope security testing will become obligatory worldwide.”**

-QA Engineer / SDET

**“It will be more challenging as technologies are changing to meet the client demand and it will also be exciting for those who are passionate about learning new technologies. More focus would be on automation and exploring new areas.”**

-QA Manager

**“It will be more challenging given that new technologies are coming in everyday.”**

-Software Developer

**“Software testing will remain as long as software requires quality.”**

-QA Engineer / SDET

## 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.





## TestComplete

Functional Test Automation  
for Desktop, Mobile and Web

[Get Started](#)



## QAComplete

Test Management Platform  
with Best-in-Class Integrations

[Get Started](#)



## CrossBrowserTesting

Web-based, Mobile  
and Desktop Browser Testing

[Get Started](#)



## ReadyAPI

Write, Run, Integrate and  
Automate Advanced API Tests

[Get Started](#)



# SMARTBEAR

The Leader in Software Quality Tools for Teams

**6.5M+**

Users

**22K+**

Companies

**194**

Countries

We create the software tools that development, testing, and operations teams use to deliver the highest quality and best performing software possible, shipped at seemingly impossible velocities.

With products for API design and documentation, API and UI level testing, code review, and monitoring across APIs, web, mobile, and desktop applications, we equip every member of your team with tools to ensure quality at every stage of the software cycle.

[Get Started](#)



**SMARTBEAR**