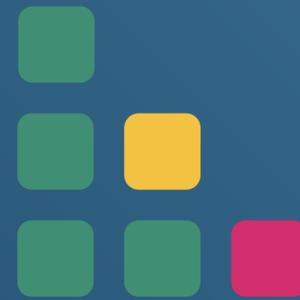


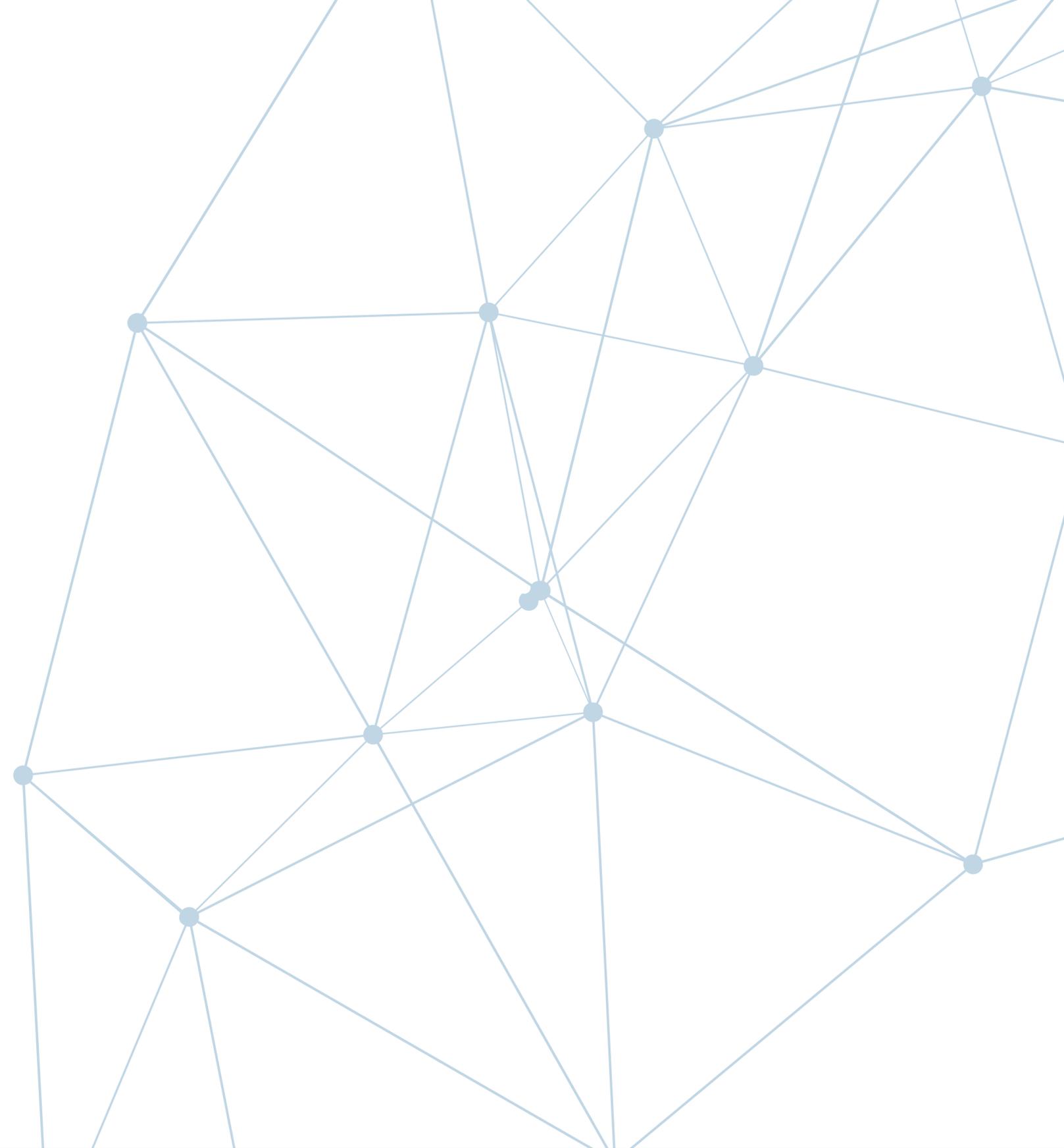
TestRail



2022 Software Testing & Quality Report

Annual Survey Report

- 3** Introduction
- 4-8** Current Trends in Development
- 9-22** Current Trends and Challenges in Testing
- 23-27** What Business Goals Does Testing Help Achieve?
- 28-29** Closing Thoughts





Introduction

Welcome to the 2022 Software Testing & Quality Report!

We developed this report in the hopes that it sheds light on the current state of the QA and software testing community. We aimed to learn more about your current goals, challenges, and key focus areas like new approaches to development, trending automation techniques, frameworks, and more.

So, what's changed in the world of QA and software testing in the last year? (Beyond the need to set up a home office, that is.)

The last year has served as a catalyst for cultural transformation worldwide, and software testing is no exception. But how did it affect software testing and QA specifically? What trends in testing and software development emerged? What were some of your primary objectives and business goals? What key challenges did you face? These questions are what we set out to answer.

To answer these questions, we invited TestRail users and the broader testing community to answer an in-depth survey about their perspectives on key QA and testing trends. This is the fourth annual survey we have run and we have now gathered enough data and insights to launch this inaugural Software Testing & Quality Report. By looking at the trends, patterns, and sentiments revealed by your responses, we want to paint a picture of where QA and the state of testing is headed and how you fit into the picture.

Over the last four years, we have received more than 15,000 responses to our annual surveys. To give you a sense of who is participating in these surveys, here's a breakdown of some of the key demographic highlights from this year's questionnaire:

- Most respondents were testers or QA engineers, followed by test leads and test automation engineers/SDETs
- Respondents came from 30+ industries, with the largest segment of respondents currently employed by companies that produce computer software
- Respondents represented almost 100 countries around the world

As you read this report, be on the lookout for trends around agile, DevOps, and other development methodologies, shifts in the focus on different testing methodologies, and surprising data about the adoption of test automation. You can also keep an eye out for questions about the business impact and ROI of testing, top challenges faced by QA teams these days, and people's primary objectives around quality right now.

Finally, on behalf of everyone here at TestRail, our genuine thanks go out to those who took the time to answer the survey and made this report possible. As we march into 2022 and beyond, we hope some of the findings from this report help you and your team reflect on your current software development and QA processes and support your continued efforts to improve the quality practice within your organization!

Current Trends in Development

Current Trends in Development

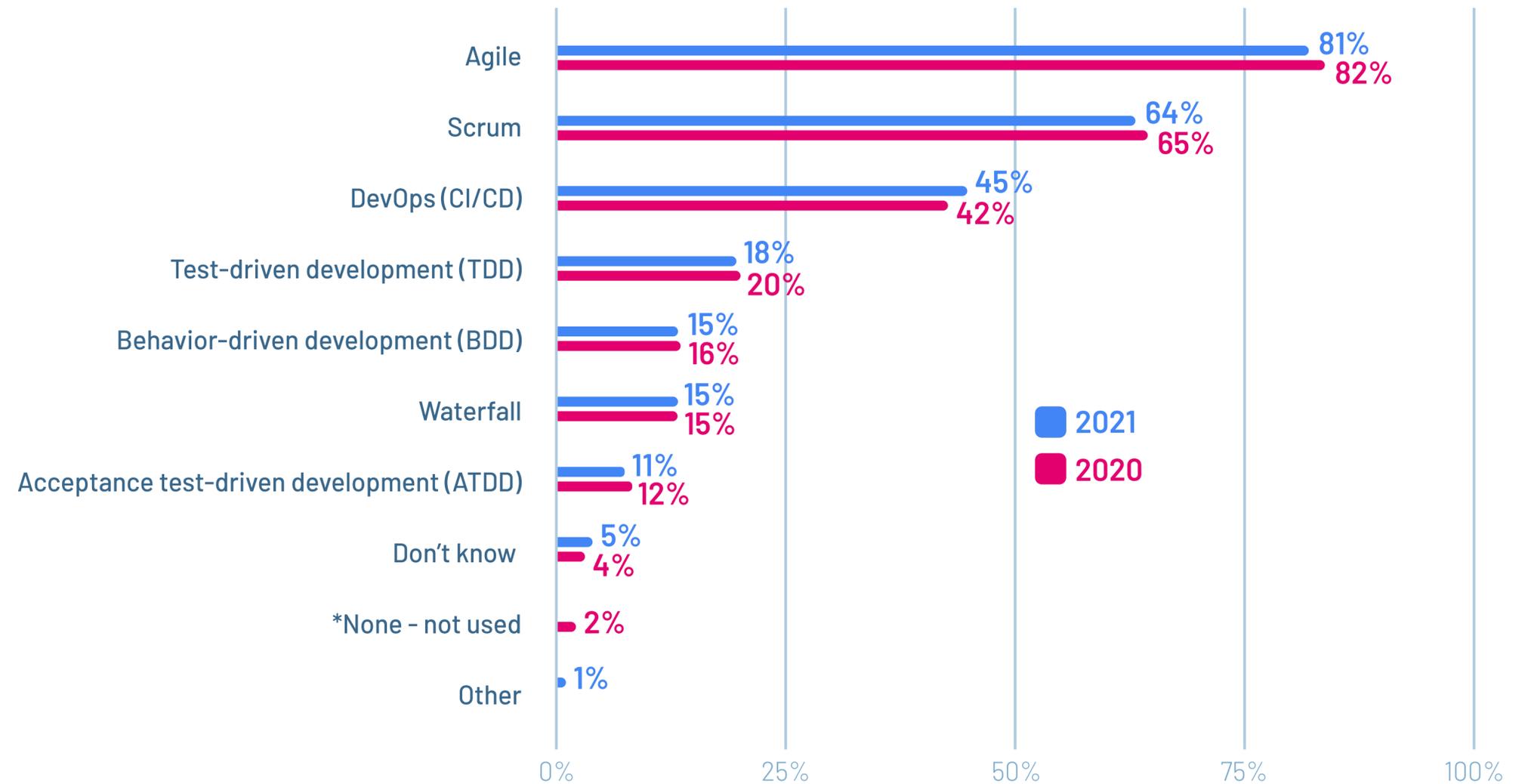
Development methodologies and techniques

When asked about which development methodologies or techniques you currently use, the vast majority of respondents use agile (81%). Meanwhile, scrum remains strong at 64%, while test-driven development (TDD) and behavior-driven development (BDD) saw slight decreases from 2020.

The agile methodology has transformed the process of developing software and appears to be the primary development methodology used in the industry market. More organizations are undertaking digital transformations, including incorporating software and technology as core elements of their unique selling proposition, as they look to enhance the way they manage data and deliver digital services to external customers as well as internal users. The agile methodology is also closely tied to DevOps, which respondents identified with three percentage points more than in our 2019 Annual Survey. DevOps is slowly becoming more of an influence in the software testing market and, especially when combined with agile, promises to promote the overall goals of agile culture in delivering high-quality software applications.

Does your team use any of the following development methodologies or techniques today?

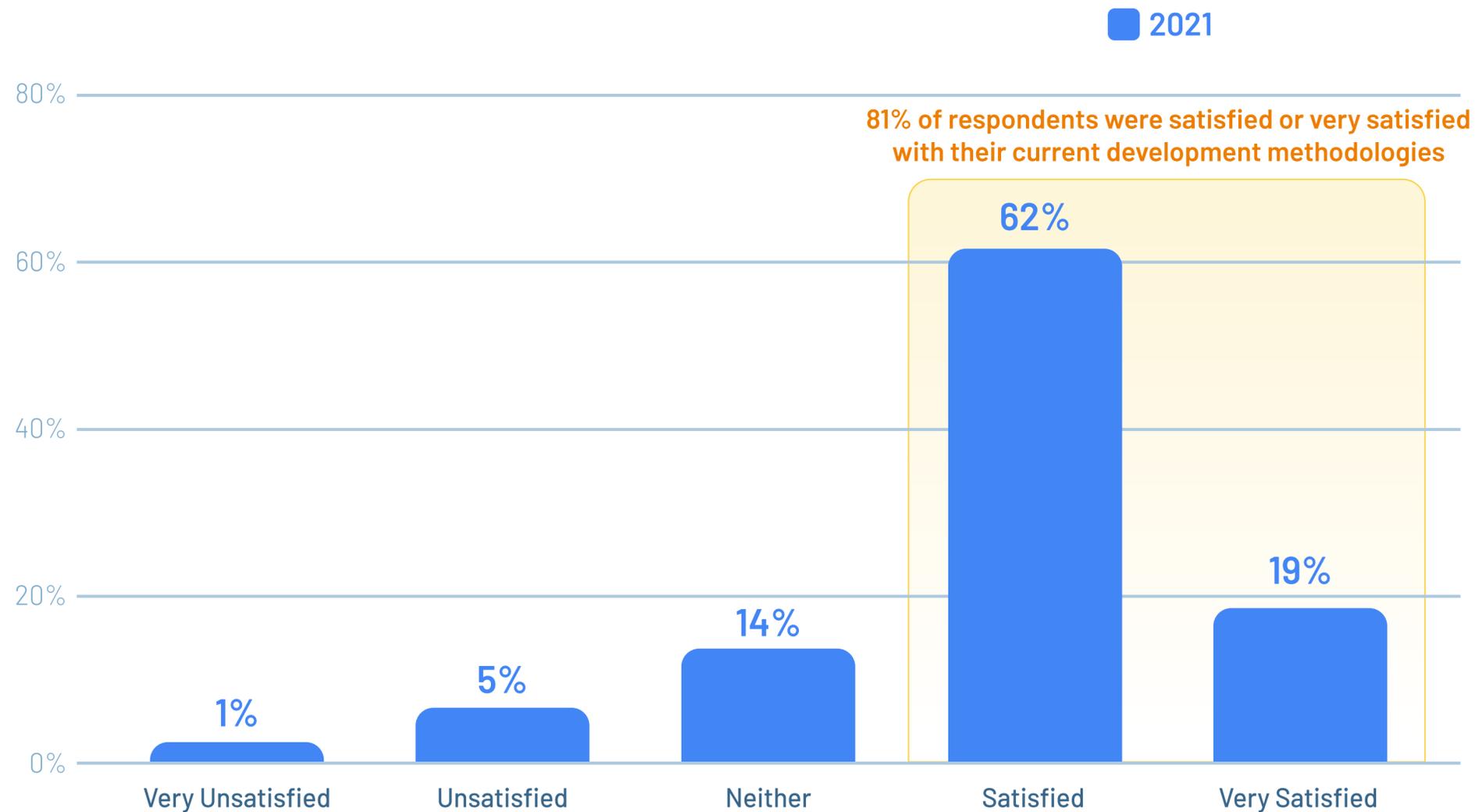
Percentage of respondents who selected the answer(s) below



*2021 data not available

How satisfied are you with your current development methodologies and techniques?

Percentage of respondents who selected the answer(s) below



Current Trends in Development

Development methodologies and techniques

Over 80% of respondents felt they were satisfied or very satisfied with the development methodologies they already have in place and only 6% felt they were not satisfied or very not satisfied. Satisfaction with current development methodologies depends on the applied development method(s) used. With over 80% of respondents using agile or some form of agile methodology, it seems likely that processes that involve quick iterations and more collaboration are closely related to satisfaction.

For those who were not satisfied with their development methodologies, common themes revolved around poor communication and lack of automation.



“Each team is doing their own thing independently, there is poor communication, and a lack of coordination between teams.”

“Testing needs to be closely knit with development as TDD helps you stay on top in terms of quality of the product. Right now, we feel that there is a big gap between developers and testers.”

“We are not fully developed and are too reliant on manual testing.”



Current Trends in Development

Adopting methodologies and techniques

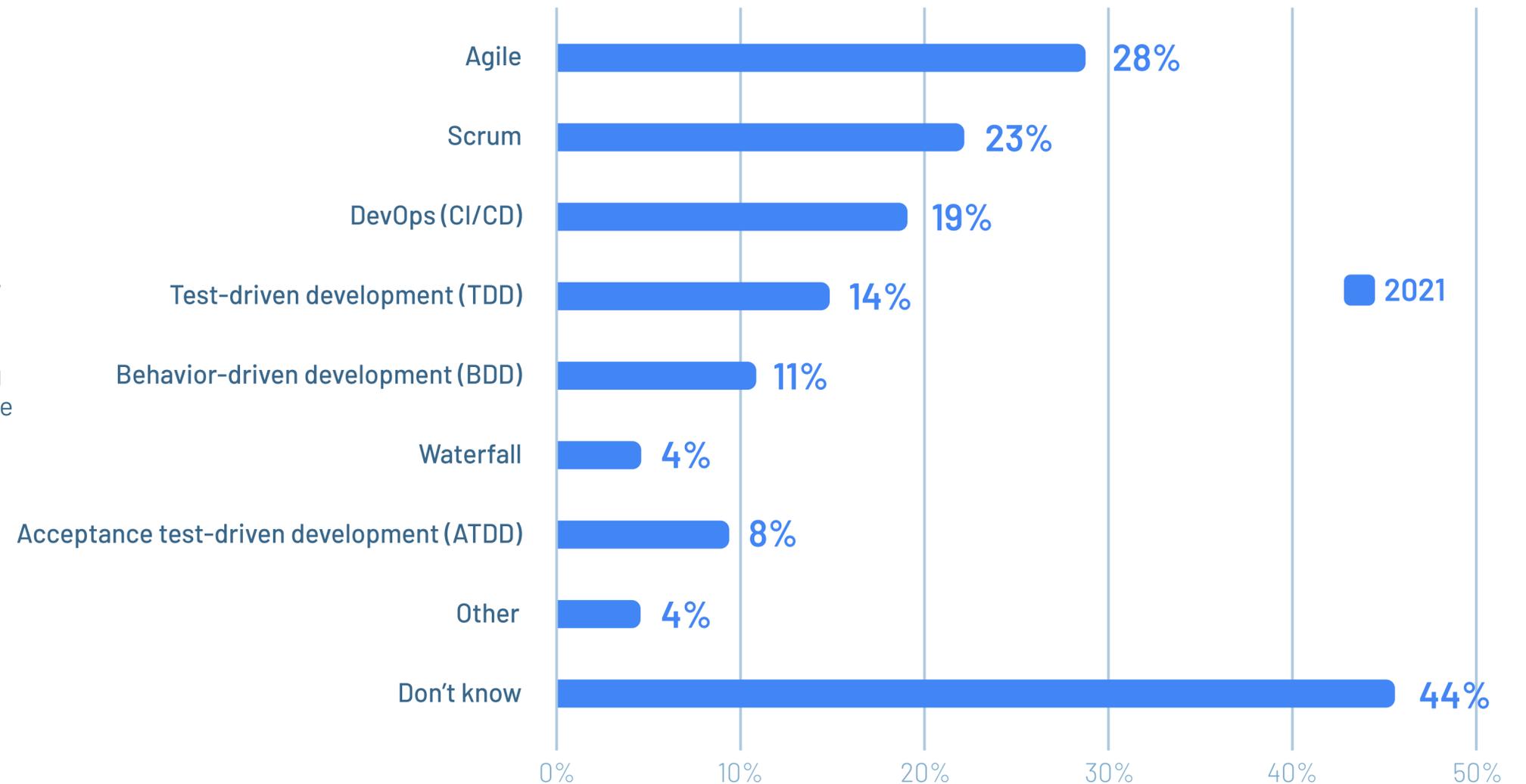
When asked if your team is considering adopting any development methodologies or techniques in the next year, almost half of the respondents didn't know if their teams were considering changes in development methodologies. Those who were considering changes in development methodologies were mainly considering adopting agile, scrum, and DevOps—in that order. TDD and BDD are in the conversation too at 14% and 11%, respectively.

Today, the changes in modern development require testing to adapt. Global trends like privacy regulations, cybersecurity attacks, increased leadership attention to digital infrastructure, and even Covid-19 also impact testing methods. As modern development practices quickly evolve, testing is required to keep up. By adapting testing practices, tools, and methods, teams can continue to ensure quality and on-time deliveries.

For those considering adopting new development methodologies or techniques in the next year, it's important to step back and ask questions to understand why you are considering a change and what you are trying to fix. Are there problems with the methods you're using today? What are they? Who is affected by them? Or, is there the expectation that—even if no large problem exists—an agile method would allow you to improve your current methods? Can you define what improvements you're anticipating or what you're hoping to achieve with a different approach?

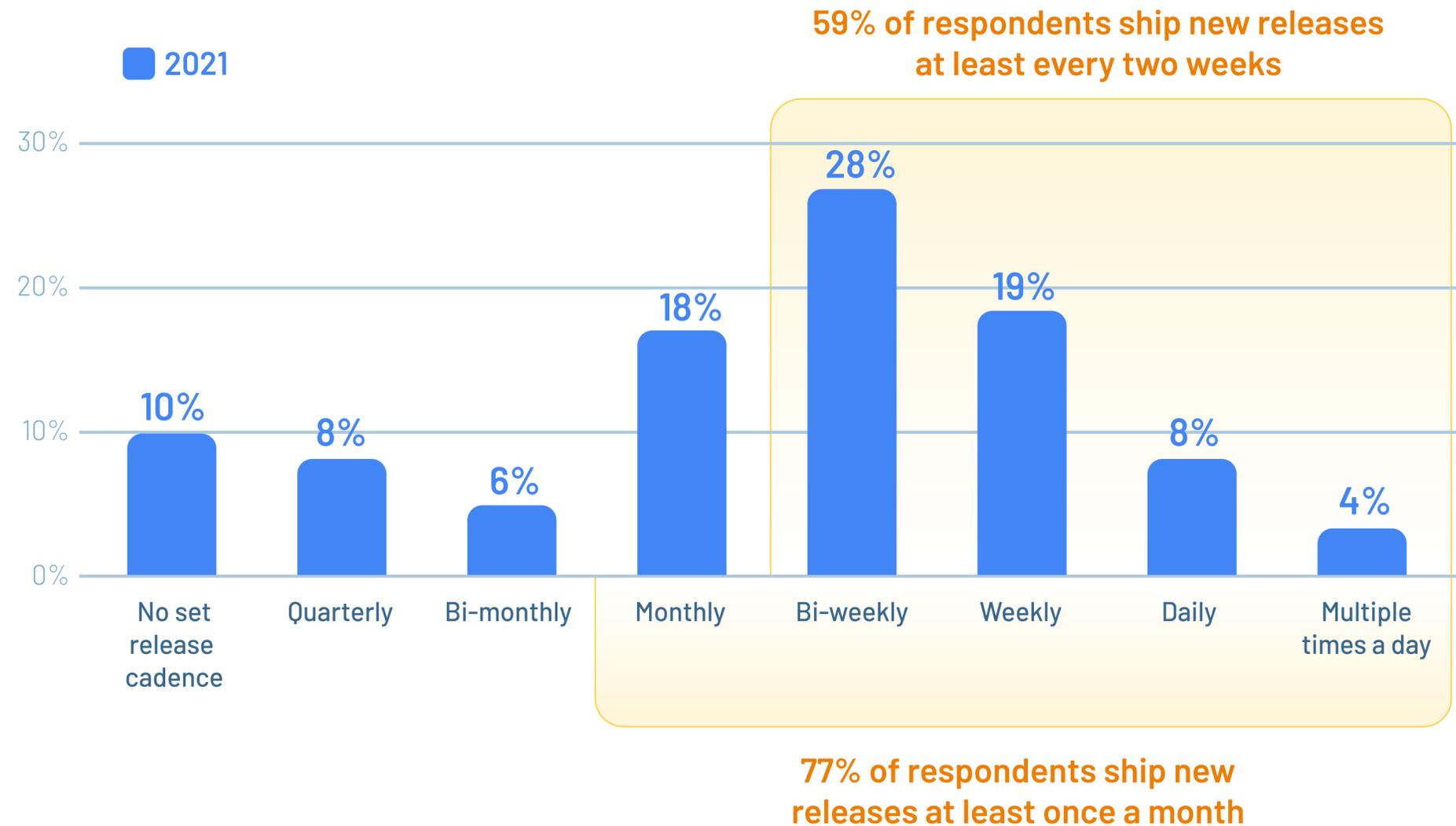
Is your team considering adopting any of the following development methodologies or techniques in the next 12 months?

Percentage of respondents who selected the answer(s) below



How frequent are your organization's release cycles?

Percentage of respondents who selected the answer(s) below



Current Trends in Development

Frequency of release cycles

"I feel the need - the need for speed." Pete "Maverick" Mitchell.

These words from Top Gun (1986) rang true for fighter pilots and race car drivers back then but they also stand true for the dynamic software development world of today, where it is becoming necessary to deliver code faster, safer, and with better quality.

When asked how frequent your organization's release cycles are, about 60% of respondents said they are releasing new code as often as every two weeks. Over 75% said they are releasing at least once a month. If quality has emerged as a differentiator for businesses to expand their customer base, enabling faster release cycles arguably comes next.



Current Trends and Challenges in Testing

Current Trends in Testing

Testing types, techniques, and methodologies

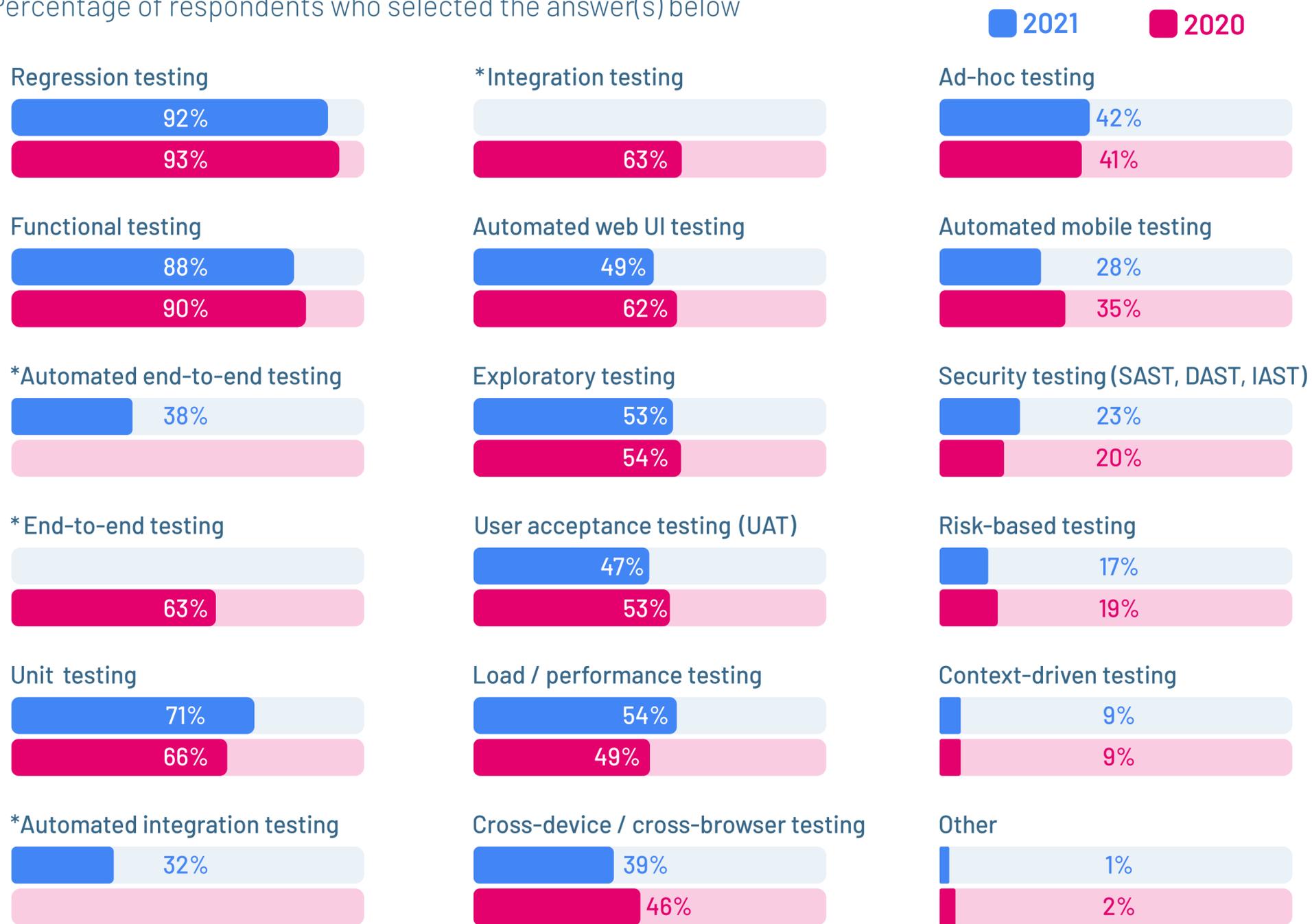
When asked about what testing types, techniques, and methodologies your team currently uses, 92% of respondents said they do some kind of regression testing, 88% reported doing functional testing, and 71% employ unit testing.

Notably, when comparing the data from responses in 2020 to 2021, there's a six-point increase from last year of respondents who do load or performance testing.

This increase in load and performance testing shows us the value put into various kinds of non-functional testing. Without good performance testing, you don't know how your system will deal with expected—or unexpected—demands. This kind of testing should be done from the start to determine viability and to build reliable performance into the design and structure of the system.

What testing types, techniques, or methodologies does your team currently use?

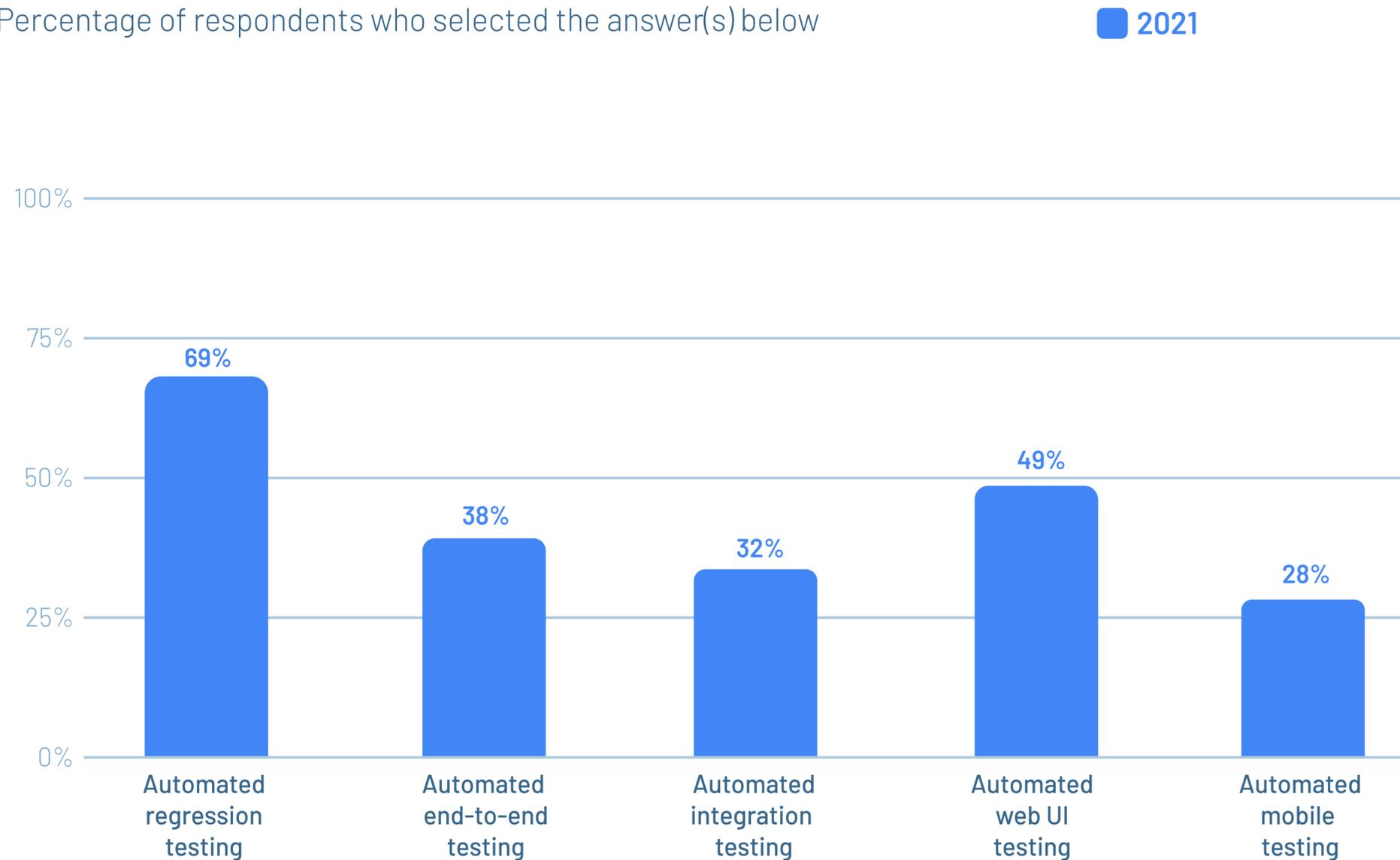
Percentage of respondents who selected the answer(s) below



*2020 and/or 2021 data not available

Automated Testing Types Drilldown

Percentage of respondents who selected the answer(s) below



Current Trends in Testing

Automated testing types

When it comes to automated testing types, regression testing appears to be the most commonly automated type of testing, with close to 70% of respondents taking advantage of it. (As a side note, 87% of respondents said they still do some manual regression testing too).

The conclusion is simple: regression testing is an indispensable part of any software development project. It can save a lot of time fixing the defects at earlier development stages and in turn, cut budget needs for a project. We're living in an era where high-quality software is no longer optional. Preventing defects is especially important because users don't always notice when an application improves, but they definitely notice when it breaks. Testers need to use an approach to regression testing that will identify defects along all paths that users could take, even the ones that aren't easy to anticipate. Whether your approach to regression testing is automated or manual, it should provide useful feedback quickly and give you the most confidence that relevant scenarios are covered.

Current Trends in Testing

Adopting testing types

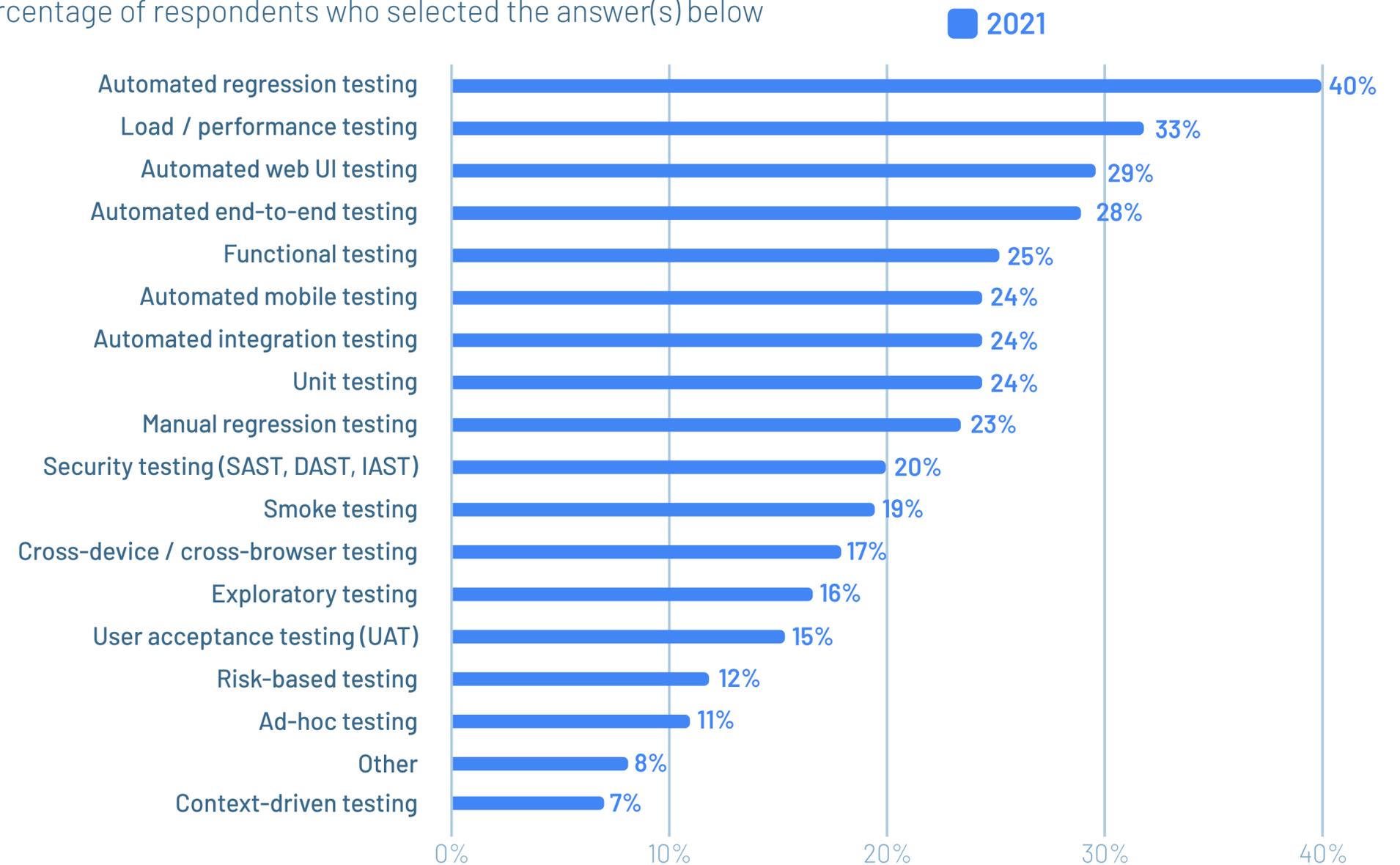
Test automation continues to be a dominant software testing trend. It has become the key enabler for many modern software development and deployment practices and is especially critical for companies that decide to implement agile and DevOps methodologies.

This trend is exemplified by the strong interest among respondents in automating more regression testing (40% of respondents) and adopting other types of automation testing such as automated end-to-end, web UI, and mobile testing over the next year. Moreover, one-third of respondents said they're considering adopting load and performance testing in the coming year.

Regression testing continues to be recognized as one of the most important testing methods to use to maintain the level of quality of your product. This importance—combined with the need to deliver quality at speed, the rise of agile development methodologies, and increases in the accuracy and efficiency of automated testing frameworks—has meant that many teams have shifted or are planning to shift from manual testing to automation. Fortunately, due to its repetitive and predictable nature, regression testing is an ideal candidate for automation. By automating regression tests, it is possible to achieve several desired outcomes that enhance team agility and product quality, like giving the team more time to focus on higher-value exploratory testing and introducing other types of testing.

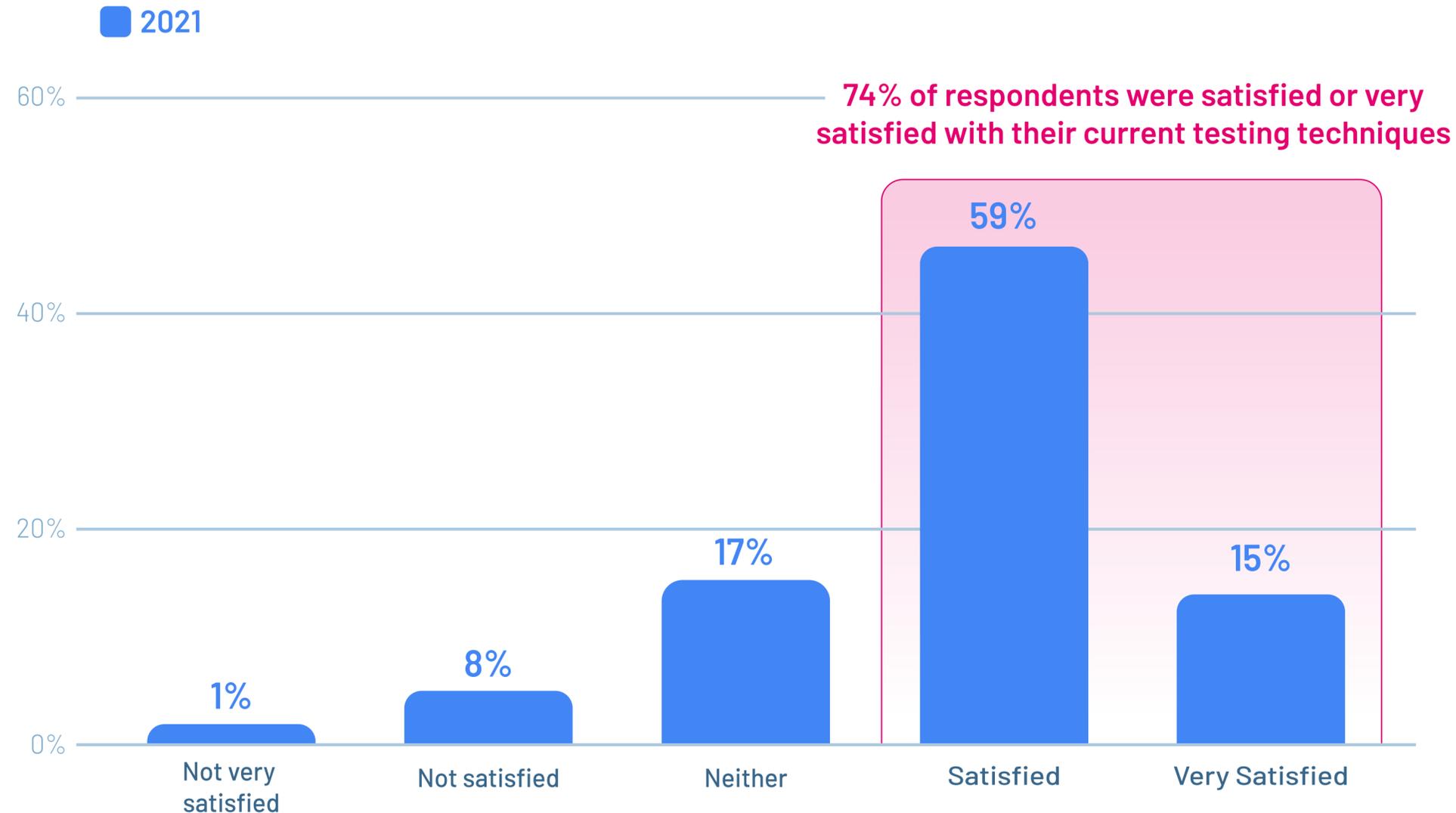
Is your team considering adopting any of the following kinds of testing in the next 12 months?

Percentage of respondents who selected the answer(s) below



How satisfied are you with your current testing techniques?

Percentage of respondents who selected the answer(s) below



Current Trends in Testing

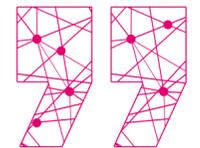
Testing techniques

Compared to the 81% of respondents who were satisfied or very satisfied with their development methodologies and techniques, 74% had the same assessment of the testing methodologies they have in place (see: [Current Trends in Development: Adopting Development Methodologies and Techniques - Page 7](#)). This 7% drop-off might be explained by common themes revolving around wanting to increase automation. Teams are experiencing an increasing pressure to do more with less and for many teams, increasing automation is necessary to keep pace with agile development.



“Manual testing is fine but we need to be extensive on the automation side as well.”

“There is not enough time spent on automation because of the lack of training and skill sets.”

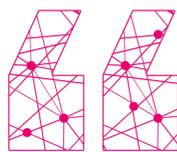


Current Trends in Testing

Current testing process

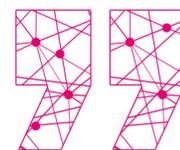
When asked about satisfaction with current testing processes, 73% of respondents said they are satisfied or very satisfied, which is in line with the satisfaction respondents reported about their testing techniques. Compared to the 81% of respondents who felt they were satisfied or very satisfied with their development processes, there is an eight-point drop-off.

Here are some common insights from the respondents regarding their satisfaction levels with their current testing process.



“Our business goals are unclear and there is too much responsibility on testers to derive requirements.”

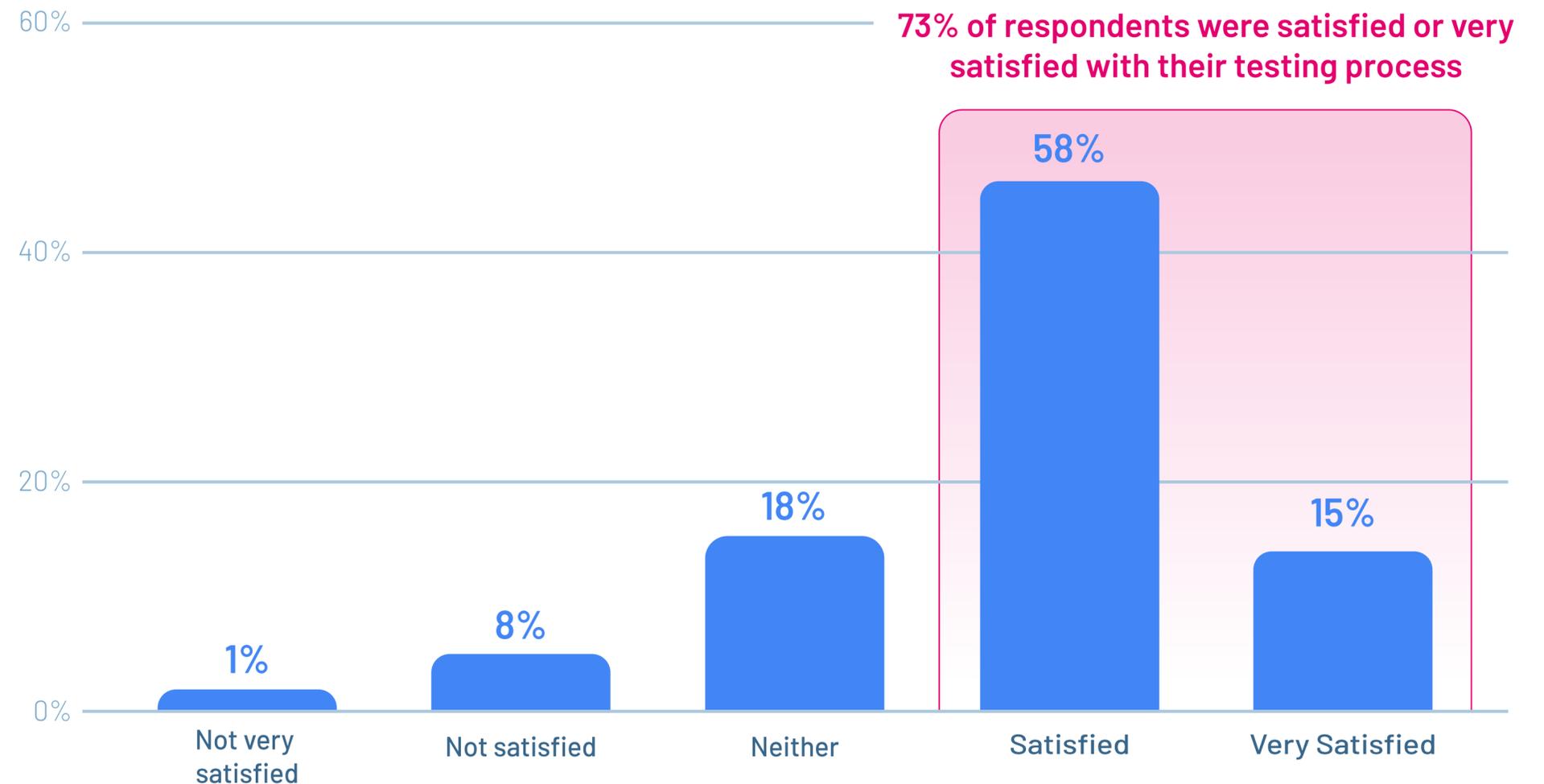
“We are working out of a google spreadsheet so there are no details in scenarios and a lot of confusion around who is doing what.”



How satisfied are you with your current testing process?

Percentage of respondents who selected the answer(s) below

2021



Current Trends in Testing

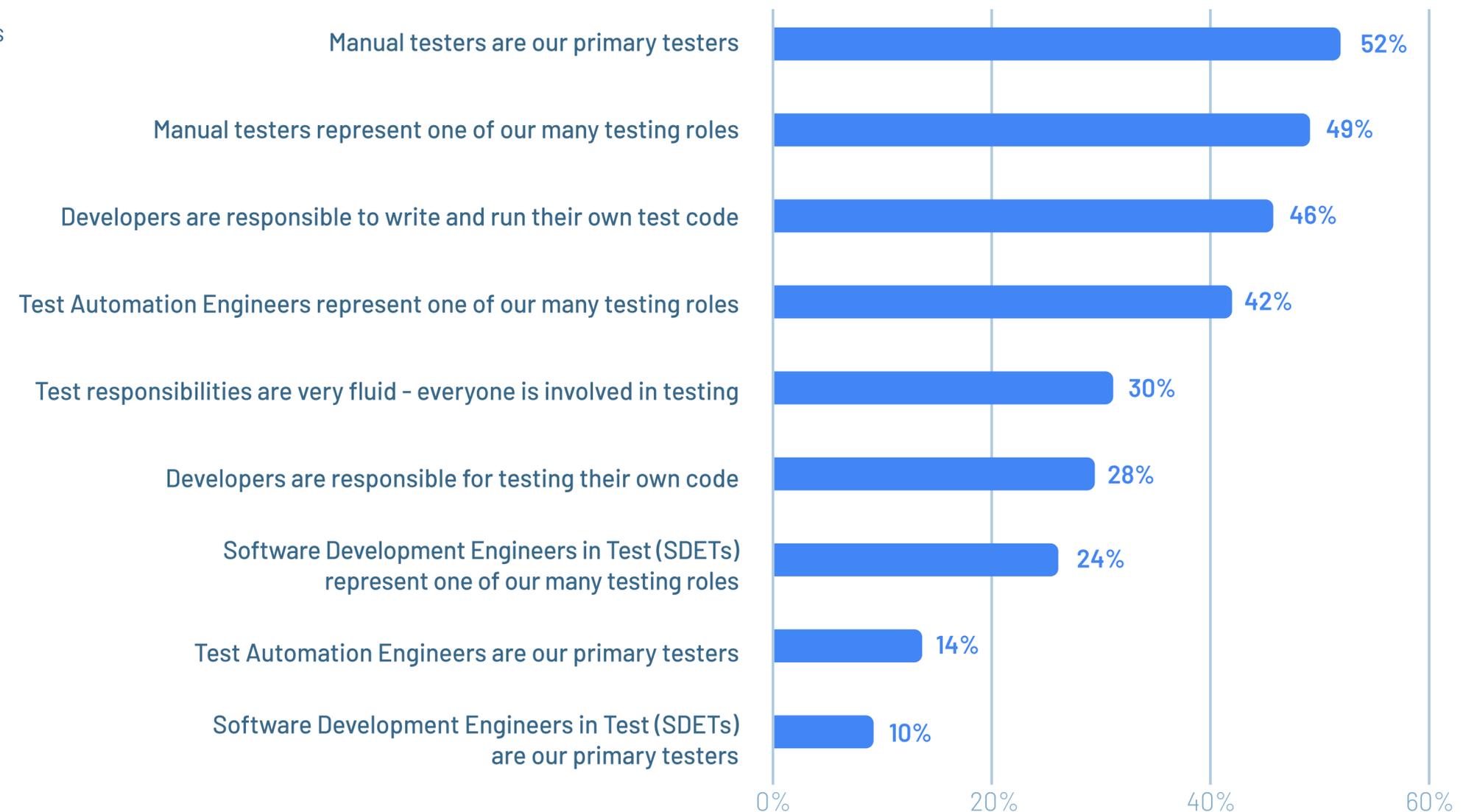
Testing responsibilities

When asked to describe quality testing responsibilities in your organization, almost half (46%) of respondents said that developers on their team are responsible for running their own test code. In comparison, nearly a third of developers are responsible for manually testing their own code (28%). This fact isn't necessarily surprising considering the growth in the number of respondents who reported their team using unit testing.

More than half (52%) of respondents said that their manual testers are their primary testers. Primary test roles are now at about two manual testers to every one automation or SDET.

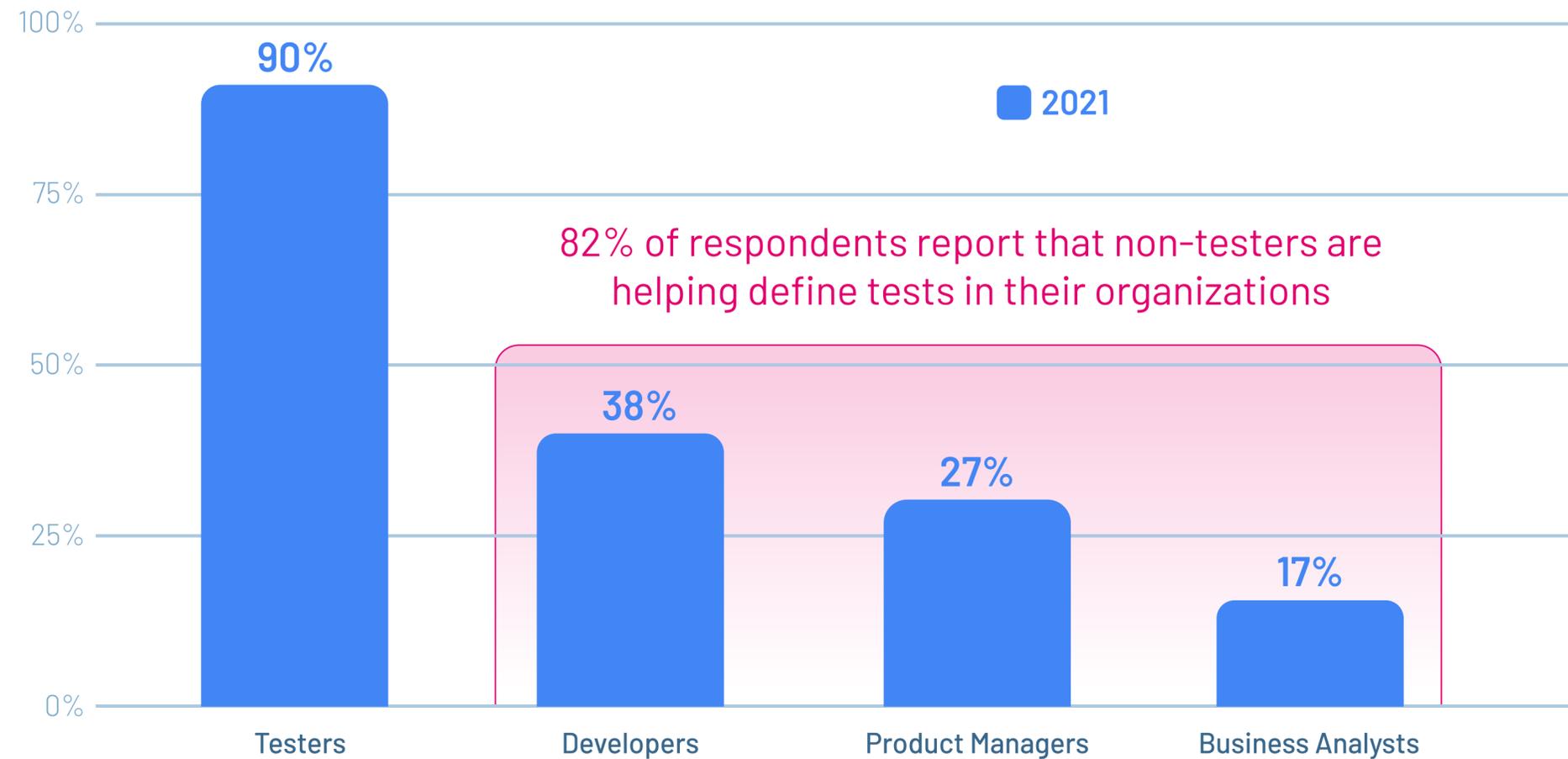
In your organization, who is primarily responsible for testing?

Percentage of respondents who selected the answer(s) below ■ 2021



Which roles define tests in your organization?

Percentage of respondents who selected the answer(s) below



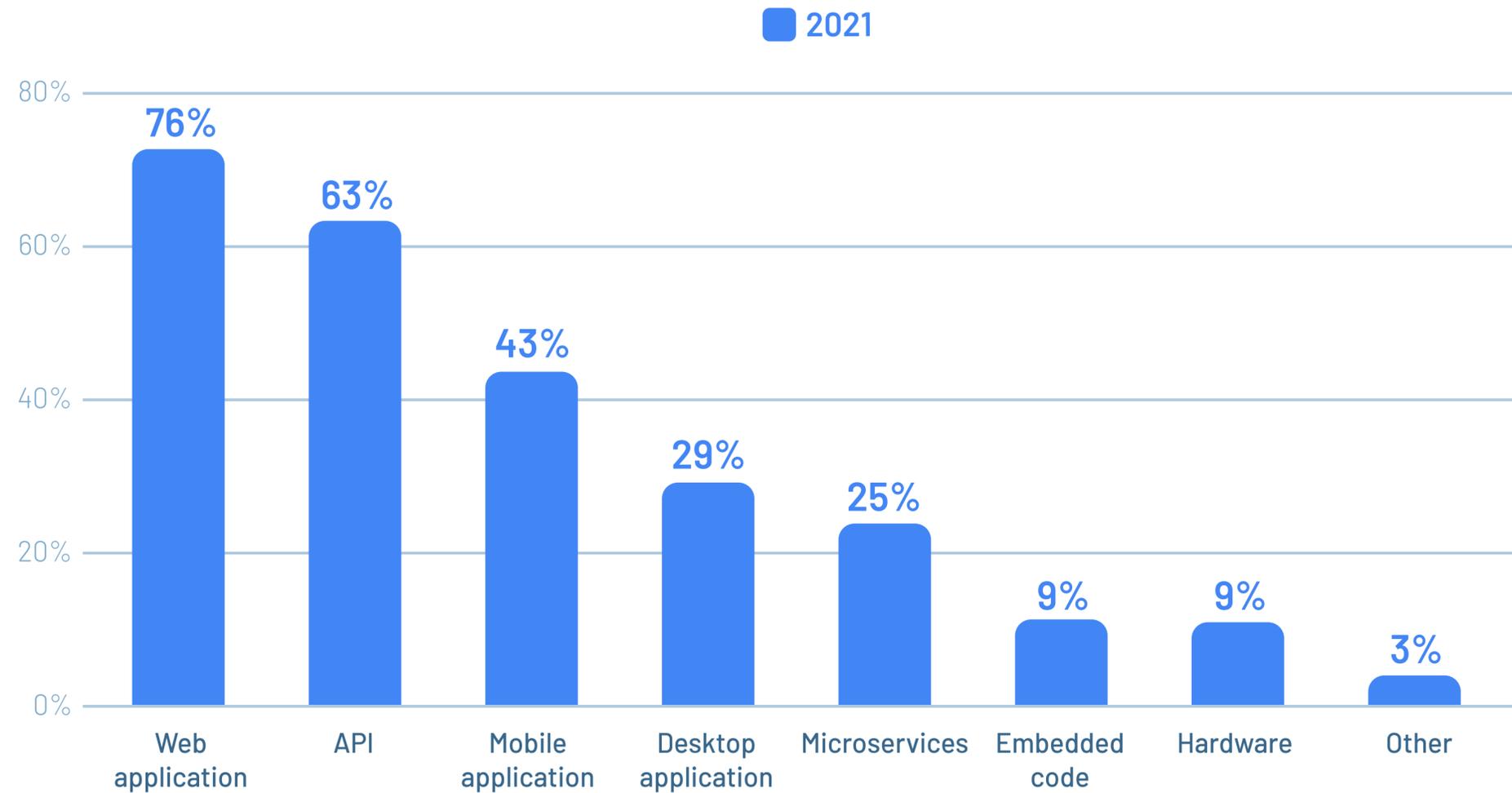
Current Trends in Testing

Roles

90% of respondents said testers define tests in their organizations, which isn't surprising. Interestingly, 82% of respondents also report non-tester roles like developers, product managers, and business analysts helping to define tests in their organizations, including 38% of respondents who said developers help define tests.

Which technologies do you test with automation?

Percentage of respondents who selected the answer(s) below



Current Trends in Testing

Test Automation

Most respondents said they use automation to test web applications (76%) and APIs (63%). Meanwhile, 43% of respondents are automating their mobile application testing, 29% of respondents are doing the same with desktop applications, and 25% of respondents said they use automation to test microservices.

By automating web application tests, testers can save time and effort on monotonous tasks because automated tests can be run continuously or scheduled at intervals. By offloading these time-consuming tasks from testers, they can focus on exploratory testing or other high-value tests that require a human perspective.

Automation should not be seen as just a plug-and-play system that requires no human intervention. Effective automation requires testers to have a thorough knowledge of the software under test, as well as an “automation first” mindset.

Current Trends in Testing

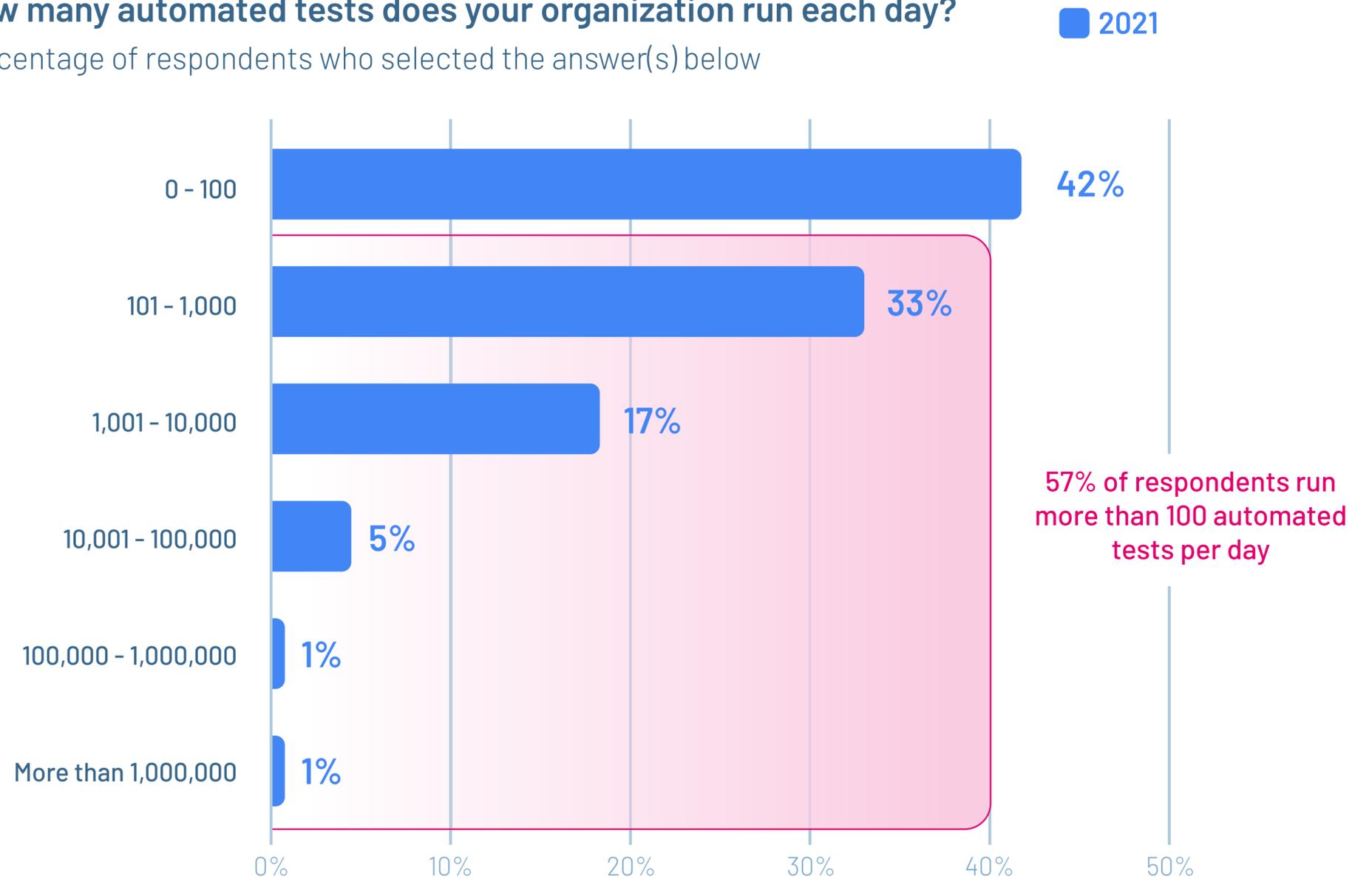
Automated tests

For those who are automating tests, when asked how many automated tests your organization runs each day, 57% of respondents said that they run more than 100 automated tests each day in their organizations.

As your application under test becomes more complicated, it becomes even more imperative to start automating some testing. But most times, your strategy around test automation is more important than the pure number of tests you've automated. The larger your automated test suite gets, the more time it will take to maintain with each new release. Focusing on automating tests of critical-path workflows or high priority scenarios can help you bring a more risk-based approach to prioritizing which tests to automate.

How many automated tests does your organization run each day?

Percentage of respondents who selected the answer(s) below



Current Trends in Testing

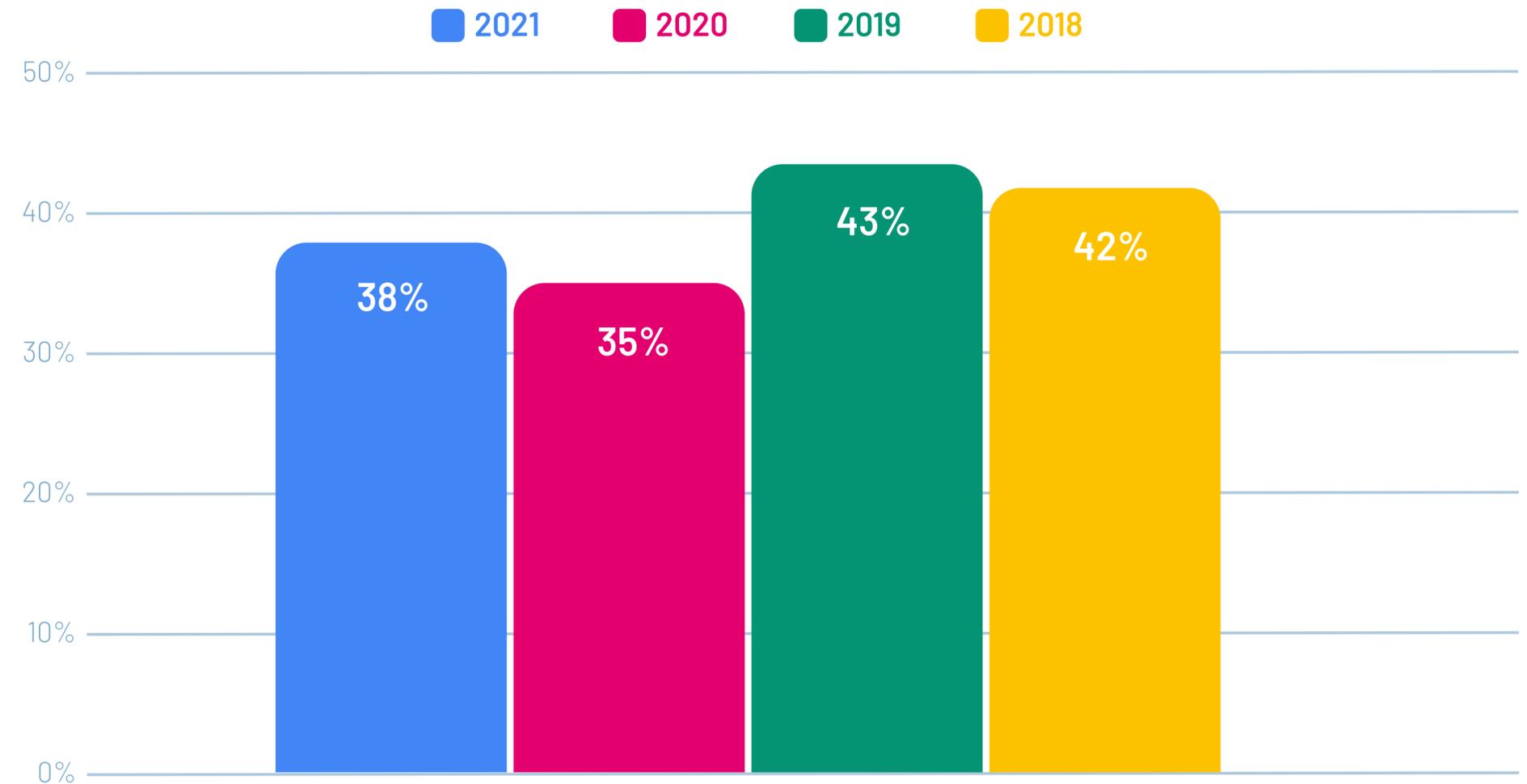
Percentage of automated tests

The rise of software development methodologies that require a continuous and streamlined workflow, such as agile and DevOps, have pushed teams to automate as much as possible. That said, even though we are seeing a slight increase from last year, we're consistently seeing about 40% percent of all tests run by respondents' teams being automated.

These responses tell us that even though automated testing is trending back upwards, this does not mean that manual testing will disappear in the near future. In fact, each type of testing serves a different function in the QA process, and together they complement each other.

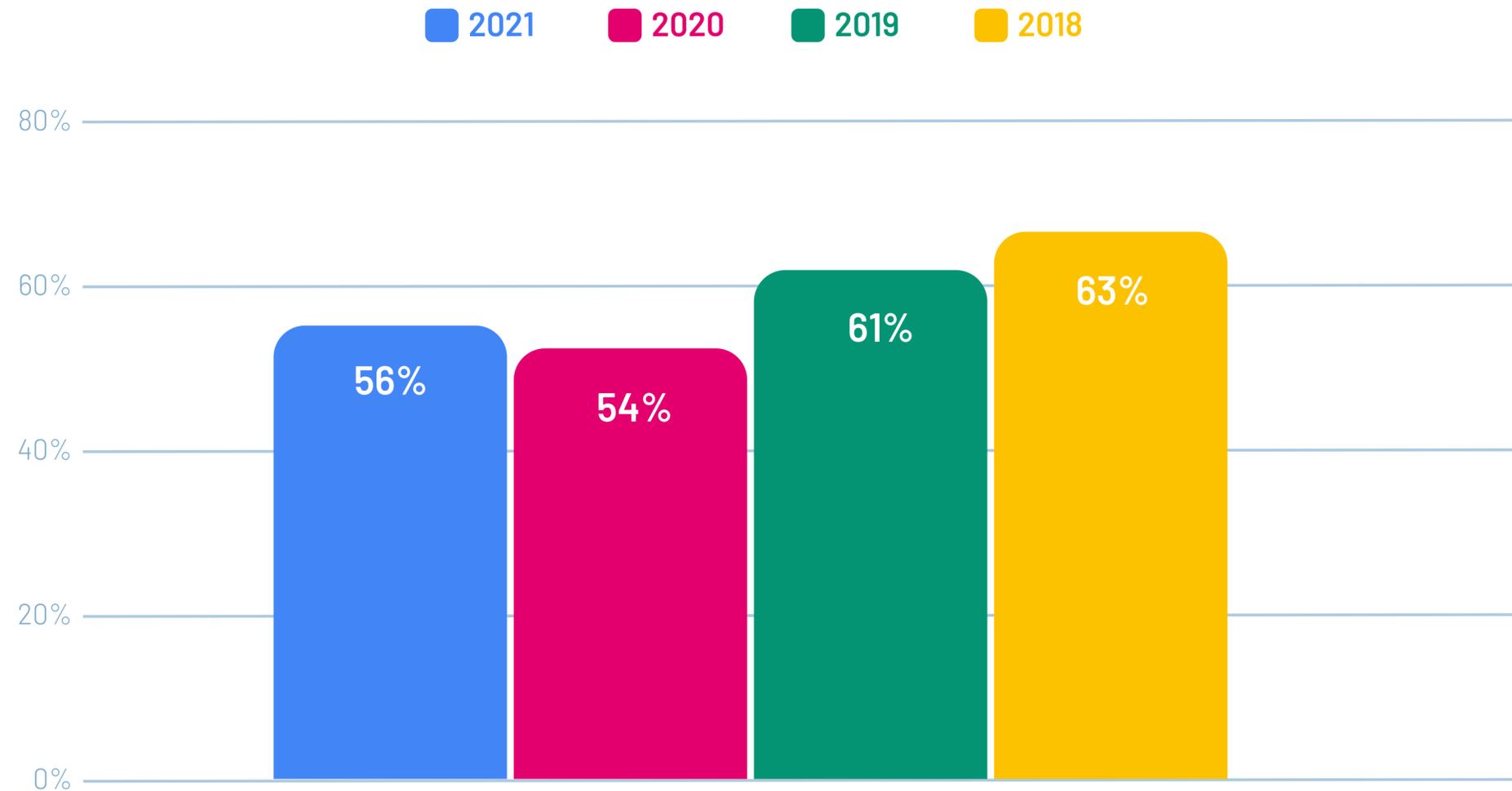
What percentage of your test are currently automated currently?

Percentage of respondents who selected the answer(s) below



What percentage of your tests do you expect to automate in the next year?

Percentage represents automated tests



Current Trends in Testing

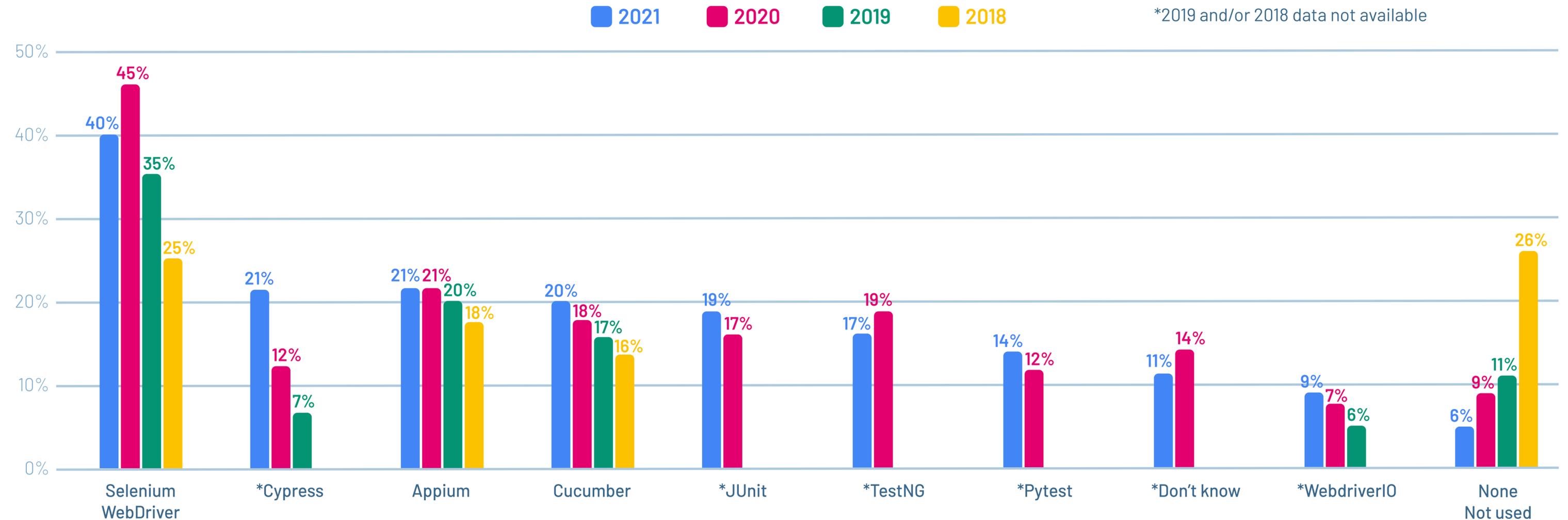
Automated tests

For the past four years, respondents have expected to automate over half of their tests. Compared to the percentage of tests that respondents have actually automated, this data indicated that they succeeded in automating approximately 20% fewer tests than expected.

Moving from manual to automated testing without impeding current development and test processes requires more than just hiring an automation engineer. Frameworks have to be built, tooling has to be implemented, employees have to be trained, and more. These things take time and effort. Doing all of that without compromising the quality of the software being released is a significant undertaking – but it's doable.

What test automation tools or frameworks do you use? Top 10 answers shown.

Percentage of respondents who selected the answer(s) below



Current Trends in Testing

Automated tools and frameworks

When asked which test automation frameworks teams used, Selenium appeared as the most popular choice even after a five-point drop from 2020. Cypress is becoming increasingly more popular with a nine-point increase from 2020, while Appium and Cucumber saw the same steady growth they've had since 2018. Other notable leaders include JUnit, TestNG, and Pytest, while the number of respondents who answered "none" or "not used" continued to drop yearly.

Current Trends in Testing

Key challenges

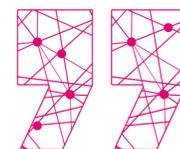
Similar to 2020, developing and increasing automation is still the biggest challenge reported by the testing community, while having enough time to complete QA tasks is second—which may be a contributing factor as to why people want to automate more. Users also reported experiencing challenges with end-to-end testing, managing testing data and environments, and driving testing earlier in the process.

Numbers speak volumes: [the global automation testing market size](#) reached \$12.6 billion in 2019, and this number is expected to grow to \$28.8 billion by 2024. The market is big and enterprises are getting on the automation rails, but many of them are not able to fully enjoy the benefits of it and even fail without properly tackling the top challenges in automated testing. Whether it's struggling to find the right framework or tool, lack of communication and collaboration among the team, not implementing a proper test automation strategy, or even setting unrealistic expectations, automation is a challenge for many teams.

When asked what key initiatives or challenges your teams are planning to focus on over the next 1-3 years, there were three common responses: **Developing more automated tests, increasing automated testing, and taking advantage of CI/CD.**

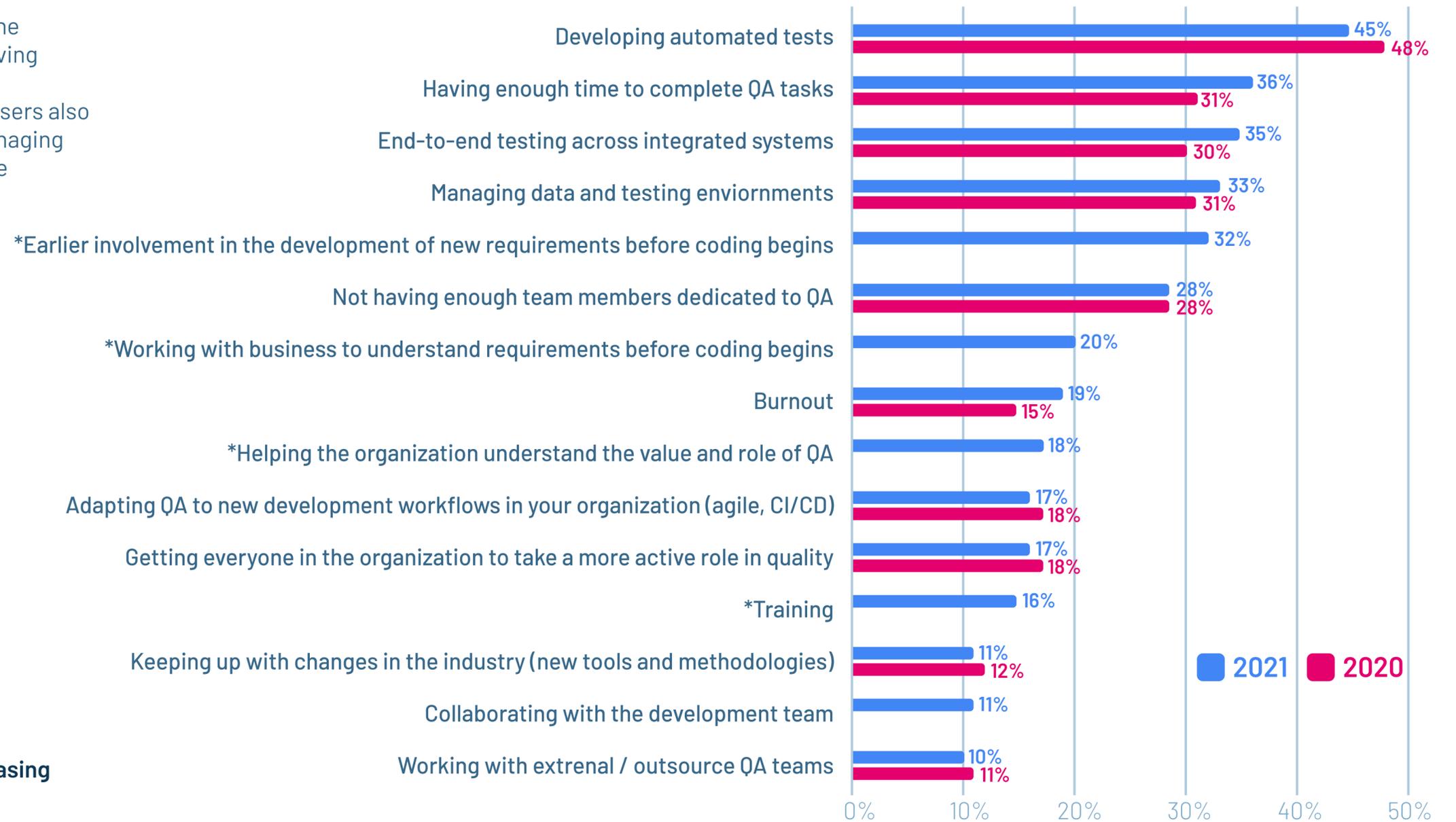


"We want to focus more on adapting the latest industry-standard technologies at all levels of the CI/CD pipeline plus development and testing. We also want to have all of the regression testing automated."



What are you/your team's top 3-5 biggest challenges? Top 15 answers shown.

Percentage of respondents who selected the top 15 answers



*2020 data not available

What Business Goals Does Testing Help Achieve?

Business Goals

Metrics and KPIs

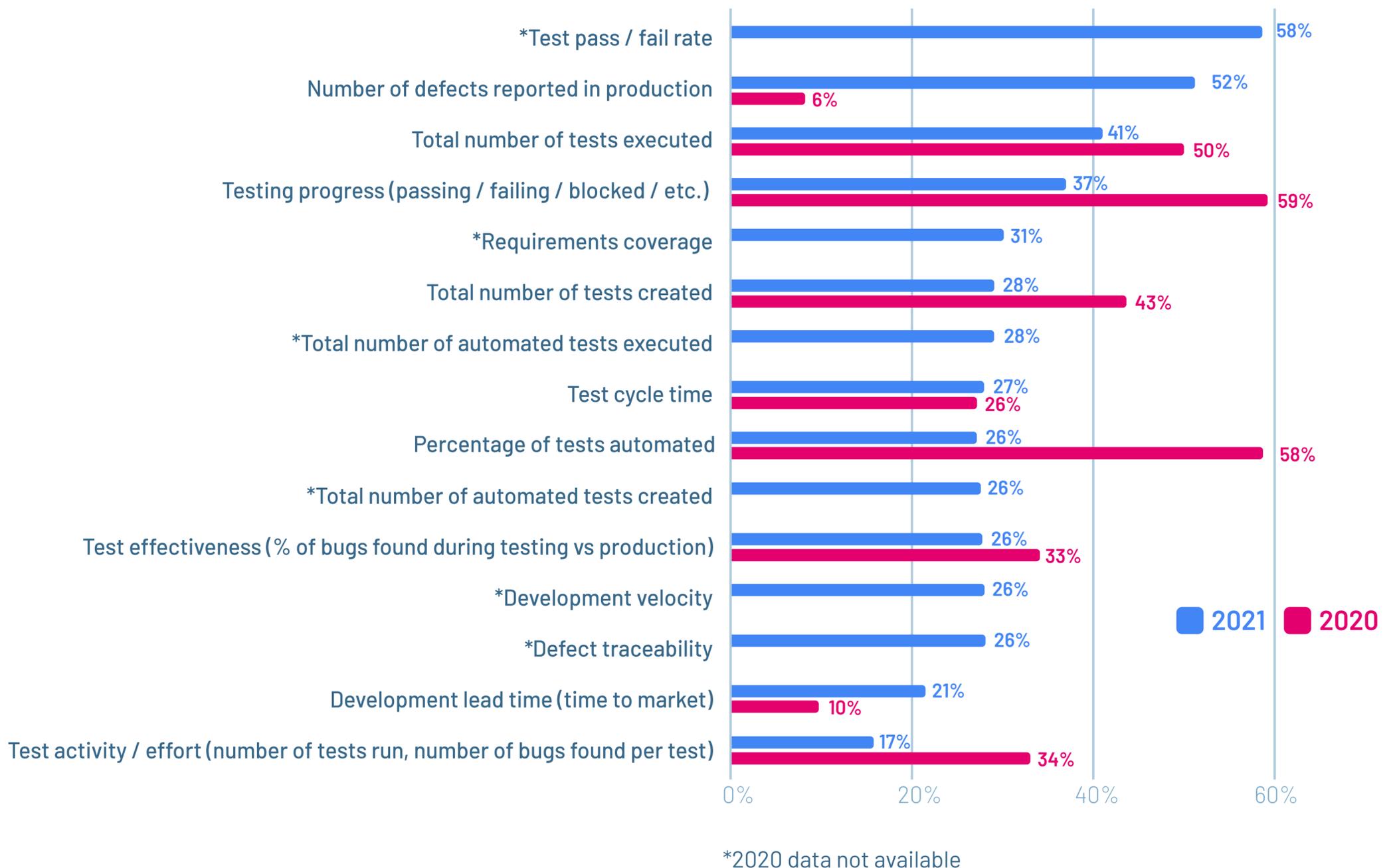
When asked which metrics or KPIs your team uses to measure the impact of your testing, almost two-thirds of respondents measure test pass/fail rate, closely followed by metrics like the number of defects reported in production and the total number of tests executed.

This year's responses suggest that there has been a shift to put more of a focus on metrics that are aligned with actual value centers and key results around quality—like the number of defects detected in production—instead of just measuring testing activity.

When looking at trend comparisons from 2020, measuring the number of defects reported and development lead time was way up. On the other hand, measuring the percentage of automated versus manual tests, test execution numbers, and bugs found were down.

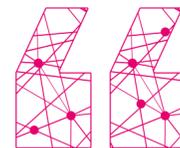
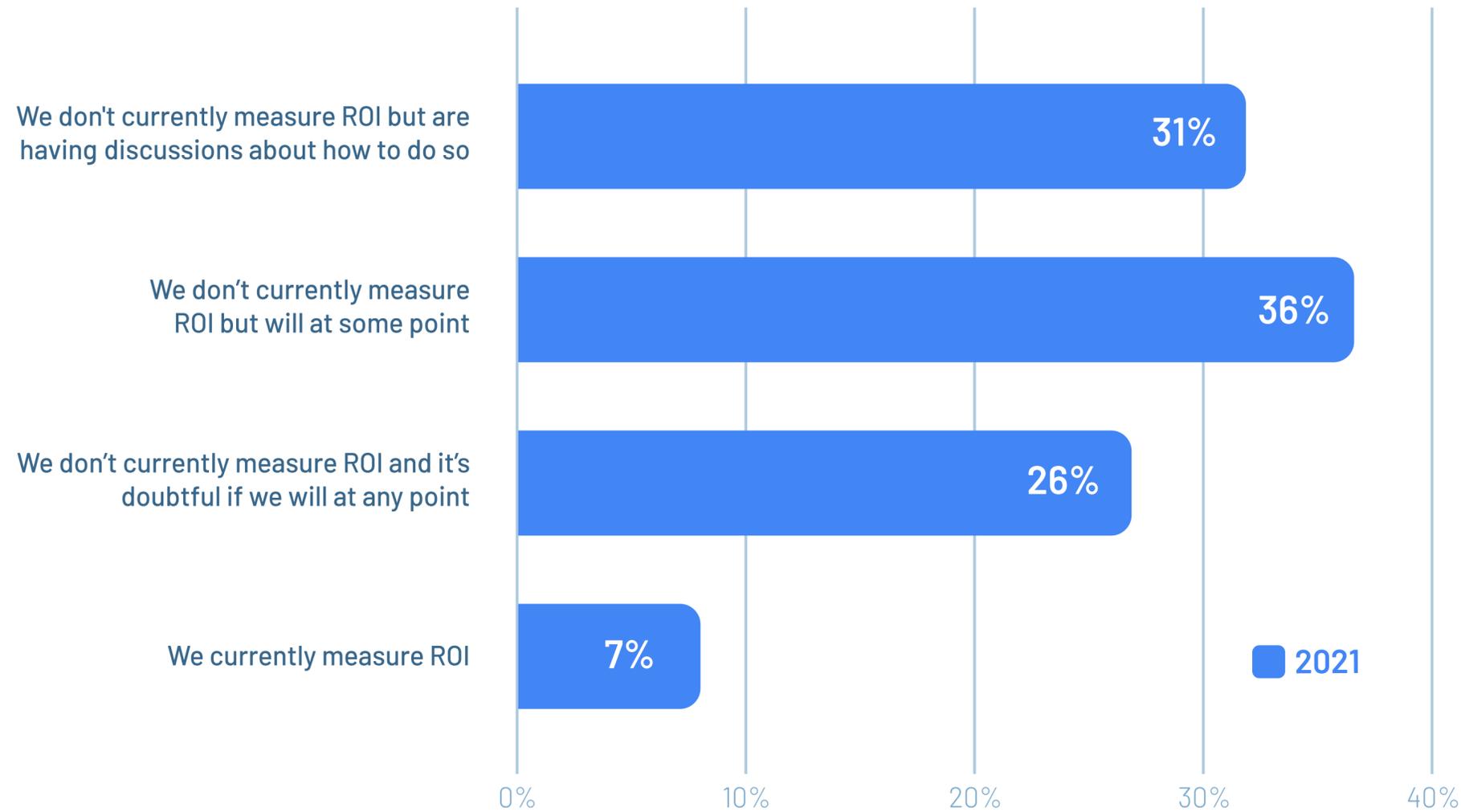
Which metrics or KPIs does your team report on? Top 15 answers shown.

Percentage of respondents who selected the answers below



How is ROI calculated for testing in your organization?

Percentage of respondents who selected the answer(s) below



"ROI is the number of bugs found during pre-production versus the number of bugs found in production. If we are seeing less than 20% production bugs, that is successful."

Business Goals

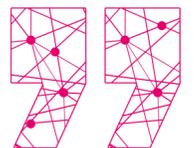
The ROI for testing

When asked how return on investment (ROI) is calculated for testing in your organization, 93% of respondents told us they don't explicitly measure the ROI of their testing efforts. However, 67% of respondents do want to calculate the ROI for their testing and are thinking about how to calculate ROI in the future (although based on the open text responses to this question, we found that there appears to be a lot of inconsistency in the definition and understanding of ROI around testing in the first place).

There is so much pressure in software development projects to reduce costs and reduce development timeframes, and testing is a common target for cuts. Oftentimes, the benefits of testing aren't understood, go unnoticed, or don't have revenue dollars associated with them. For that reason, some consider the idea of calculating ROI for software testing to be a myth. Some questions to consider are: "What can we, as a testing community, do not only to demonstrate improvements in the effectiveness of the QA process but also to quantify the actual ROI of QA? Is it worth trying to quantify ROI? Would quantifying the ROI of testing help make QA more visible to the rest of the organization?"

Some of the responses we got regarding what ROI means described calculating ROI by the ratio of manual versus automated tests, test effort versus bugs in production, and investment in cost versus revenue generated.

"ROI is measured by calculated daily revenue generated versus the cost of the project."



Business Goals

Primary objectives of QA

QA is an activity that aims to evaluate the quality of a software product and improve it by identifying defects and potential risks present in the software before it gets shipped to end-users. Clearly defining the objectives of your testing not only helps communicate the value and impact of QA to the rest of the organization but also ensures effective testing.

When asked about the primary objectives for your team's software development and QA teams, the top three responses you selected are tied directly to quality, namely, increasing test coverage, reducing bugs in production, and automating more tests. The following two most commonly selected objectives address speed to deployment, while the following four priorities revolve around driving efficiency in testing.

What are your primary objectives for your team's software development and QA teams around quality right now?

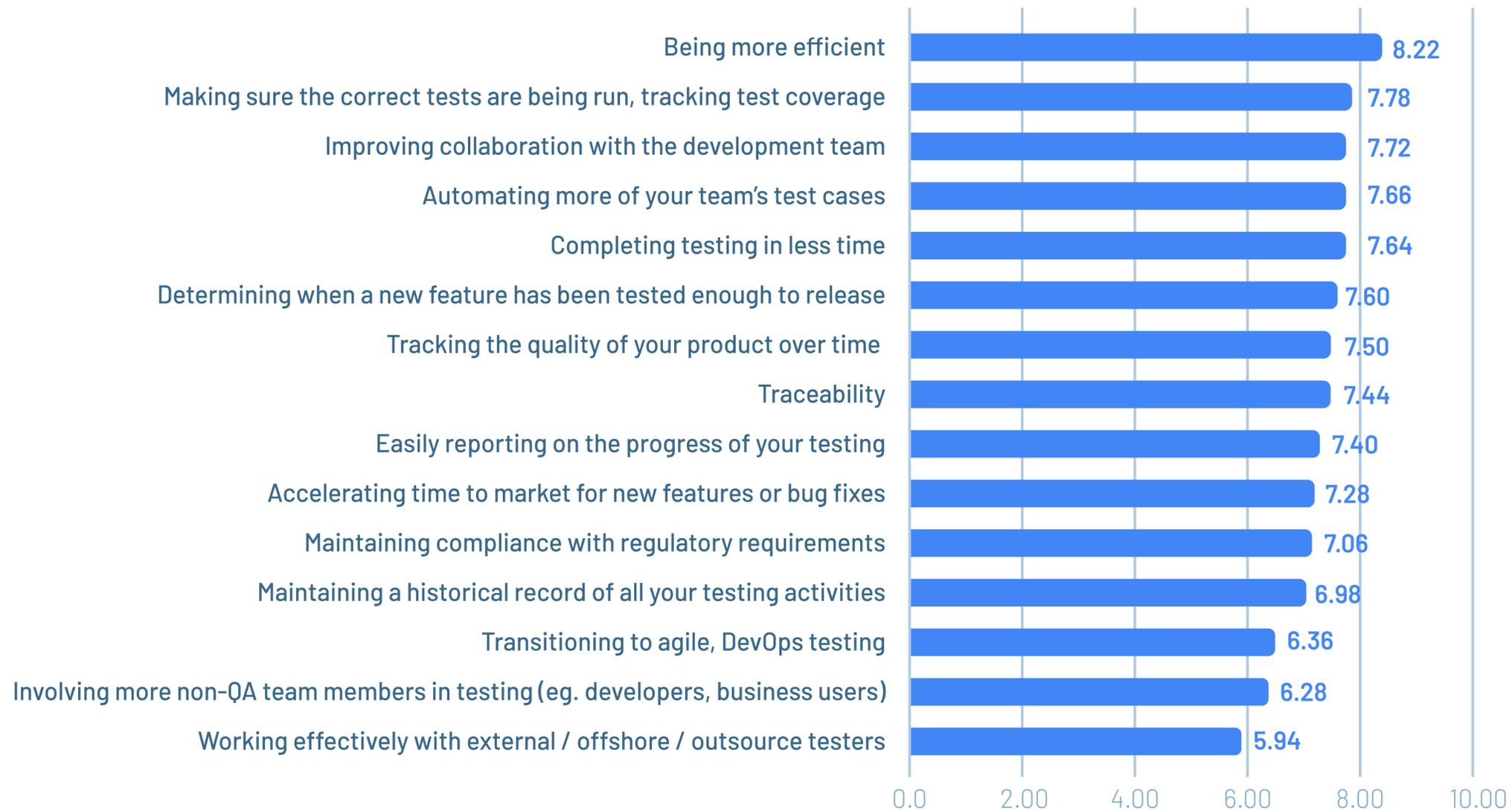
Average rank 1-10, 10 being highest priority



Rate how important each of the following activities is to your success

Average rank 1-10, 10 being highest priority

2021



Business Goals

Successful habits of a QA team

When asked to rate how important various activities are to your success, respondents reported that "being more efficient" was the top activity driving their success. Other top priorities like "making sure the correct tests are being run", "tracking test coverage", "improving collaboration with the development team", and "automating more of your team's test cases" may be areas where respondents are looking to "be more efficient" to achieve overall success.

These responses show us that people are looking at efficiency and better collaboration to help them achieve their goals of increased test coverage and achieving better quality overall.

Closing Thoughts

So now that you have read through this report, let's revisit our initial question:

What's going on in the world of testing and quality assurance?

With 81% of respondents using agile, it is clear that agile is here to stay. However, this does not mean it will remain static. Since their inception in 2001, agile practices have continued to evolve. It was first about shifting from the waterfall model of software development to focusing on iterative development using scrum, XP, and other methodologies, thereby removing many bottlenecks organizations had with more waterfall-style development delivery. Agile is now about optimizing operations within the development process, using different practices and tools, and making the entire process more efficient. It is about starting the testing process right from the beginning of the development lifecycle to catch defects early (i.e. "shift left") and distilling insights from the data collected from each stage of the development process.

The agile methodology quickly gained widespread adoption and became the industry standard. Yet, in a few years, a critical oversight arose: in its initial iterations, agile left out the processes and requirements of the operations team that deployed and managed software products. This, along with evolutions in technology and new distributed approaches to development, led to DevOps, an approach that aligns development and operations teams around the core tenants of shipping high quality software early and often. As we found in this year's report, the adoption of DevOps continues its steady increase, with almost half of respondents now reporting that their teams employ at least some form of DevOps practices. DevOps has been a trend for the last several years, but it is here to stay and will continue to be a big focus for organizations moving into the rest of 2022 and 2023.

DevOps enables teams to build, test, and release software faster and more reliably both by incorporating agile principles and by streamlining and accelerating certain processes with increased automation, including test automation. In 2022, test automation is at the forefront of users' minds and larger organizations are executing more automated tests on a daily basis. This trend is exemplified by the strong interest among respondents in adopting more automated regression testing (40% of respondents) and other automated testing methods such as automated web UI testing (29% of respondents) and automated end-to-end testing (28% of respondents) over the next year (none of which are necessarily exclusive of each other).

Along with increasing automation, most respondents identified increasing test coverage and reducing bugs in production as primary business goals, while efficiency, coverage tracking, and better collaboration with developers were called out as key drivers of success. We know that high-quality products are not the result of chance or random luck. Setting goals, communicating well, and tracking KPIs are all part of the concerted effort to create a culture of excellence in QA and software testing.

"Quality assurance" is often assumed to be the step that happens between writing code and getting it deployed to production. But software quality doesn't happen at just one point in time; it encompasses more than just your bug count. QA is a mindset and a culture that your entire team, company, and industry should be involved in.

We look forward to reporting how these trends continue to develop next year. Thanks again to everyone on the team that collaborated on this report!

TestRail

TestRail is a test management platform built to help testing, QA, and software development teams ship rock-solid products faster. TestRail makes it easy to create organized, repeatable, and streamlined processes so you can scale test coverage, plan test executions more strategically, and spend more time actually testing.

TestRail can be used in waterfall or agile projects and is flexible enough to be used in any type of QA process.

- Manage test cases, plans, and runs
- Increase test coverage
- Get real time insights into your QA progress
- Integrate with issue trackers and test automation
- [Explore Enterprise for advanced features and security](#)

Integrate with all of the tools in your stack to build quality into every stage of your software development lifecycle. More than 10,000 organizations like NASA, Apple, Microsoft, Activision Blizzard, and Amazon trust TestRail to power their QA and test management processes.

[Visit our website](#) for more information or get started today!

Try TestRail for free

The screenshot displays the TestRail interface for a specific test run. At the top, there's a navigation bar with tabs for OVERVIEW, TODO, MILESTONES, TEST RUNS & RESULTS (active), TEST CASES, REPORTS, and ADMINISTRATION. Below the navigation, the test run details are shown: R270, Version 4.2, Build 183, Android (Chrome). There are buttons for Defects, Reports, and Edit. A summary dashboard features a pie chart and a large '73% passed' indicator. The pie chart shows: 43 Passed (73% set to Passed), 2 Blocked (3% set to Blocked), 3 Retest (5% set to Retest), and 2 Failed (3% set to Failed). A secondary box shows '9 / 59 untested (15%)'. Below the dashboard, there are controls for sorting (Section) and filtering (None), along with buttons for Add Results, Assign To, and Columns. A section titled 'Login' shows a progress bar and a table of test results:

ID	Title	Priority	Type	Defects	Status
T24484	Login page loads from homescreen	Medium	Smoke		Passed
T24488	Login page loads when app-switching	Medium	Smoke	TRM-60	Retest
T24485	Username and password text boxes allow f...	Medium	Smoke		Passed

The right sidebar contains a 'Tests & Results' section with 'Activity', 'Progress', and 'Defects' sub-sections. Below this is a tree view of test cases, including folders like Login, Download and Installation, Certification, Contacts, Messaging, Image Messages, Reactions, Replies, Voice Message, Hardware Integration, Security, and Performance.

Image: Track all of your test activities and quality metrics in a single platform to improve collaboration, evaluate risk, and release higher-quality software.