



The State of

Quality Report

2022

TABLE OF CONTENTS

Executive Summary

Quality at Speed: Quality Assurance Techniques and Challenges

- Common testing techniques
- The most widely used QA techniques being the most effective ones
- QA teams and challenges in high software quality
- Level of satisfaction in software quality, QA practices and tools

Test Automation

- The continued decline in open-source frameworks and tools' popularity
- Race to adopt test automation
- Persistent obstacles in applying automation tools
- Factors to ensure positive ROI

Artificial Intelligence in Test Automation

Looking Ahead to 2022

Case Study Analysis

Appendix

- Methodology
- Demographics



Coty Rosenblath

CTO at Katalon, Inc.

“ In today’s rapidly changing world, organizations have to look both inward and outward for improvements in their software quality (QA) practices. They need to collect and analyze information and data generated from their practices to determine their weaknesses and strengths in delivering software quality at speed. At the same time, they have to keep updated on the current status and trends of their respective industries.

Understanding how data analytics can assist software teams, since 2018 we have worked with our partners to conduct annual studies to discover and report insights into the current state of software quality. These annual reports also offer recommendations on what areas and how software teams can improve their QA practices and processes.

From this year's report, we learned that a vast majority of software professionals consider test automation the most effective QA technique. However, with only half of them applying test automation to their projects, teams are likely to experience difficulties when implementing this technique.

Frequent requirement changes and the lack of time are the main challenges to adopting test automation and ensuring quality. Still, we see more testing activities using automation. Besides regression testing, teams today use automation tools to analyze test results, generate test data, perform performance tests, and other time-consuming activities.

The survey also shows that, despite the great potential of artificial intelligence, AI for test automation is still in its initial stages. Teams are more concerned with employing QA practices and tools effectively rather than improving the application of AI to their projects.

At Katalon, our mission is to help teams deliver better software faster. Our test automation platforms allow easy automation of necessary but time-consuming activities, such as test generation and execution. We also incorporate advanced AI capabilities in our products, so that teams can improve the effectiveness of test automation.”



Executive Summary

In today's rapidly changing world, organizations have to look both inward and outward for improvements in their software quality (QA) practices. They need to collect and analyze information and data generated from their practices to determine their weaknesses and strengths in delivering software quality at speed. At the same time, they must look for the current status and trends of their respective industries.

Based on a survey of **3,000+** respondents, this report captures a snapshot of the current state of software quality, presents insights, and predicts trends in QA techniques, practices, and tools. As our study surveyed software engineers, QA engineers, analysts, and managers, the findings reflect the opinions and voices of people who are directly involved in ensuring the quality of their software.

Findings

1. Although it is the most popular QA technique, automated system testing is not as widely used as people would expect.

Software teams today employ a combination of QA techniques during software development. Apart from manual testing, which is performed by virtually all software teams, automated system testing is the most widely used technique among all 11 techniques surveyed.

Still, just over a half (**51%**) of respondents applied automation for system and integration testing, which is rather low given the fact that it has become crucial for Agile and DevOps practices.

51% applied automation for system and integration testing

2. Monitoring and testing in production emerged as an answer for balancing between quality and speed.

To achieve quality at speed, teams need to find the right balance between ensuring the quality of software and reducing the time needed to perform QA activities. They cannot sacrifice valuable time to make error-free software.

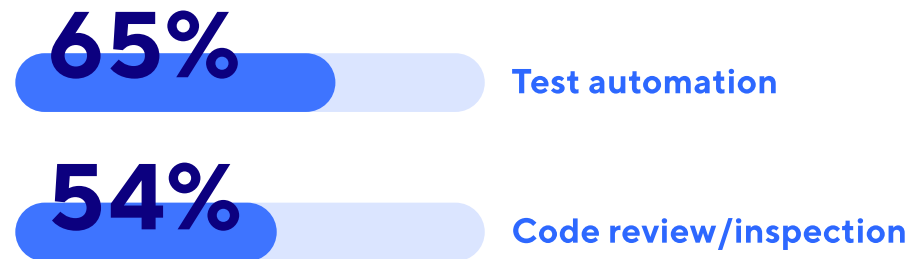
This survey found that monitoring and testing in production is the third most popular QA technique which was applied by **38%** of respondents teams.

This finding suggests that software teams embrace the possibility of errors in production and use the technique as a way to strike a balance between quality and speed.

38% used monitoring and testing in production, making it the third popular QA technique.

3. Automated testing and code review/inspection are the two most effective QA techniques.

Teams tend to use a QA technique when they find that it is effective. This survey showed that the two most widely used QA techniques are also the most effective ones. They are test automation (for integration, system, and acceptance testing) and code review/inspection, which were recognized by 65% and 54% of respondents, respectively.



4. Frequent requirement change and the lack of time to ensure quality are the most cited challenges in delivering high-quality software.

Like our surveys in recent years, this survey found the same two most common challenges in achieving high-quality software: frequent requirement changes reported by 46% of respondents and the lack of time to ensure quality by 39%.

Frequent changes in requirements make the application of automation difficult as teams have to spend time revising their automation tests to reflect the changes. With a lack of time, teams tend to perform ad hoc manual testing instead of investing resources in more effective QA techniques such as test automation.

5. Professionals are satisfied with their software quality upon delivery, but they are not equally satisfied with their QA practices and tools.

When asked to offer their levels of satisfaction with the quality upon delivery, almost 8 out of 10 respondents rated their software quality as *good* or *better*.



Whereas, only 5 out of 10 respondents were *satisfied* or *very satisfied* with their QA practices and tools.

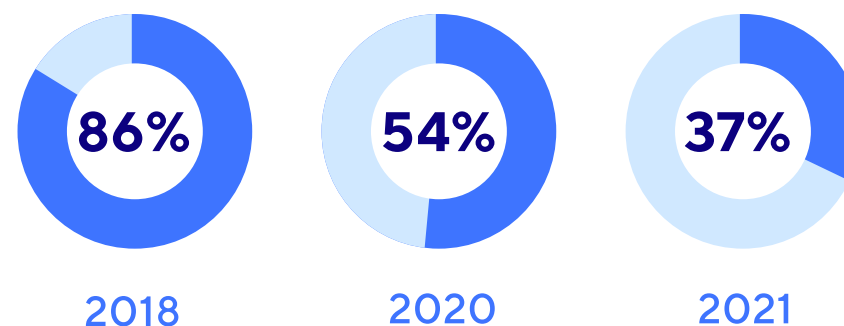


When it comes to applying test automation, many respondents were unhappy with their automation tools and practices, so they had to resort to manual testing instead.

6. The popularity of free open-source automation tools continues to decline as commercial tools are more powerful.

Tools play a key role in the success of test automation practices. Since 2018, our surveys have tracked the popularity of common tools for test automation.

One trend is clear: Selenium is the most widely used tool, but its popularity has declined sharply over the last four years, going from 86% in 2018 to 54% in 2020, and 37% this year.



This survey also shows that tools and frameworks such as *Katalon* and *Perfecto* have become more powerful. Commercial tools such as these eliminate the need to build automation frameworks based on Selenium and Appium for test automation.

7. Test automation is increasingly applied to more testing activities.

Another trend found in this survey is that teams applied automation to more testing activities. While a majority of respondents applied automation to regression testing (53%), they also applied automation to analyzing test results (38%), generating test data (36%), and performing performance tests (34%)

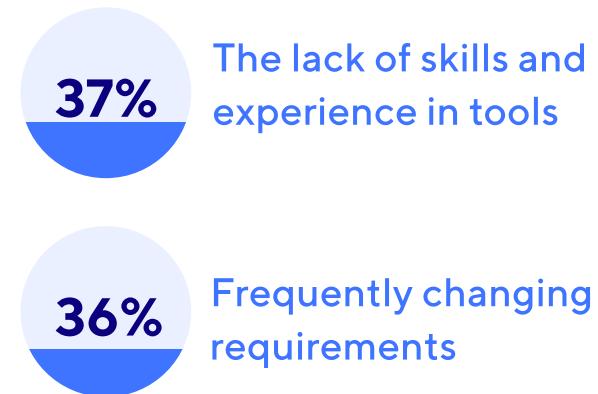
It is possible that software teams have more mature and powerful automation tools to perform more testing activities automatically.



8. The most common challenges in applying automation tools continue to persist.

This year's survey found that the most common challenges with automation tools remain the same as those reported in 2020.

They include:



These challenges are not necessarily essential as automation practices and tools have the potential to eliminate the need for advanced skills required for automation and to cope with changing requirements.

9. Most teams have achieved a high ROI with test automation.

Return on investment (ROI) is an important metric to measure the benefits of test automation. This survey asked software professionals to rate their ROI in terms of the cost/time-saving of their investment in test automation.



The good news is...

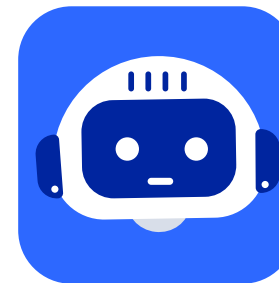
most respondents have seen a positive ROI with the majority of respondents (63%) having experienced 20% or more cost/time-saving in their test automation investment while only 6% reported a loss and 9% reported no saving.

Higher ROI is achieved when organizations provide sufficient training, choose the right tools and tasks to automate, and demonstrate persistence in applying tools.

10. AI for test automation is still in its infancy.

Artificial intelligence (AI) offers great potential for test automation. It can reduce or even eliminate the need for human involvement in time-consuming activities such as test design and generation, data preparation, test execution, verification, and test maintenance.

Still, our survey found that less than 50% of respondents use some AI capabilities for their test automation. They mainly used AI for generating test cases and scripts, test data generation, defect detection, and test prioritization and selection. This finding suggests that organizations, especially automation tool vendors, continue to improve the adoption of AI for test automation.



< 50%


of respondents use some AI capabilities for their test automation.



Quality at Speed: Quality Assurance Techniques and Challenges

There are many QA techniques that a software team can apply during the development life cycle. They include, for example, manual testing, automated testing, code review, code analysis, test-driven development, and pair programming. As these techniques are different in terms of the application context and effectiveness, using the right techniques is crucial for the success of achieving quality at speed.

This section reports the findings on the popularity of QA techniques, their effectiveness, and challenges in achieving quality through using these techniques.



Common testing techniques

Automated system testing is not as widely used as people would expect.

To ensure software quality, teams can apply many QA techniques ranging from code review and pair programming to automated testing. In this survey, we asked participants which QA techniques their teams applied during development.

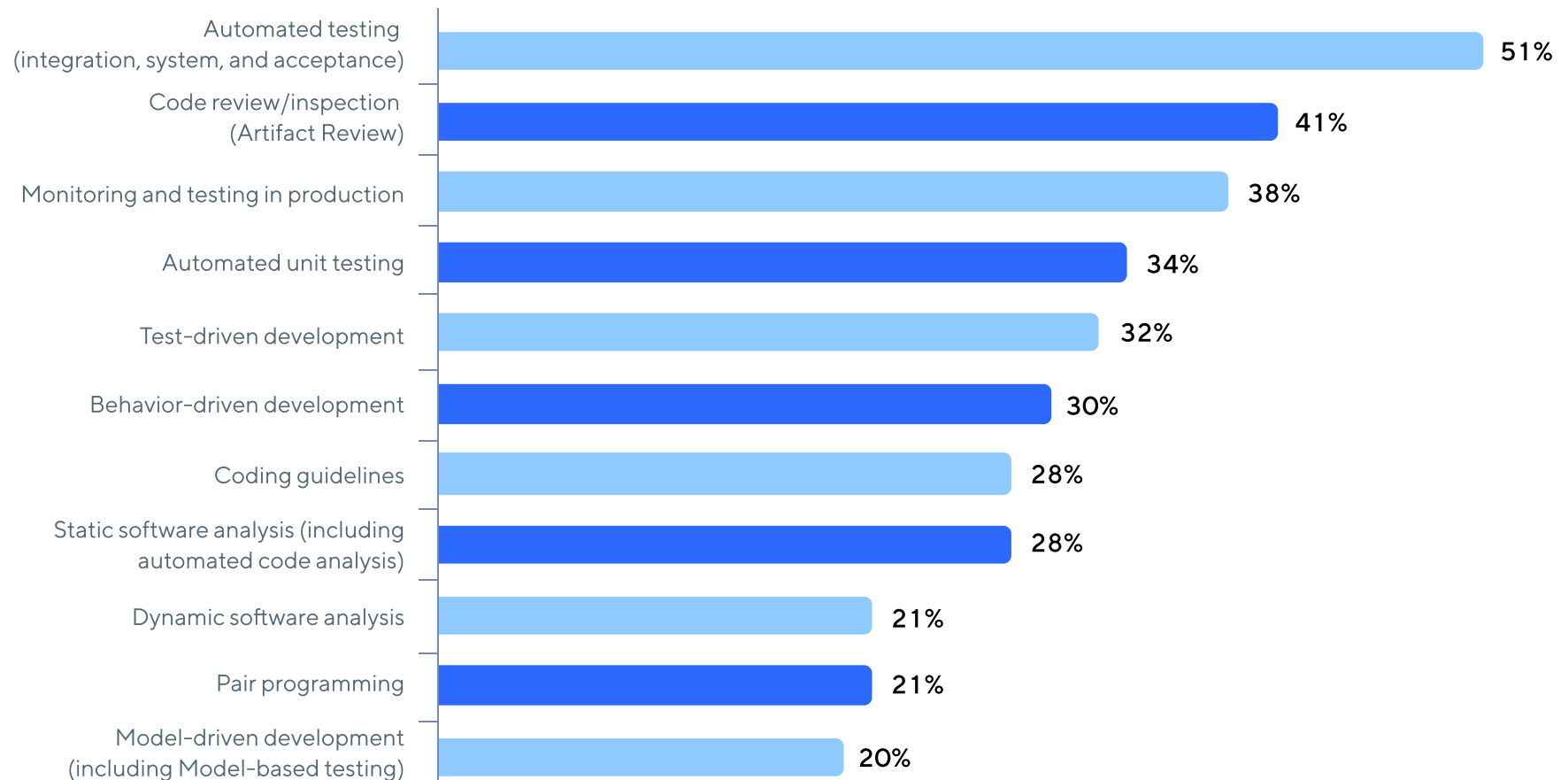
The most commonly used techniques are automated testing (51%), including integration, system, and acceptance, and excluding automated unit testing, code review (41%), monitoring and testing in production (38%), and automated unit testing (34%).



“ When it comes to ensuring the quality of the application, code review and automation testing play a major role. Because doing effective code review itself can ensure whether both functional and non-functional requirements are effectively met. In addition to implementing automation for smoke, sanity, and regression, organizations can save a lot of time in terms of quality build delivery.”

– Vishnu VS, QA Manager, Xilligence. –

Which QA techniques does your team apply during development?

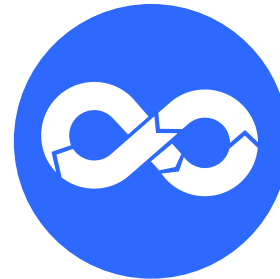


We omitted manual testing from the answers as it is presumably a QA technique performed by all software teams. If including automated unit testing, 63% of respondents reported applying test automation in their projects. Besides these techniques, other widely used QA techniques include test-driven development (32%), coding guidelines (30%), behavior-driven development (28%), and static software analysis (28%).

Surprisingly, monitoring and testing in production is the **third most popular QA technique** used by 38% of respondents. It is an emerging technique that involves monitoring and testing software in real environments with real users and real data.

As a crucial part of DevOps and continuous delivery practices, it has become a popular QA technique as more teams adopt these practices. To speed up time to market, teams test their software builds early and frequently in production-like environments.

Still, these environments cannot fully replicate the true production environment in which users, data, and usage scenarios are real.



Because of these inevitable gaps between the testing and production environments, teams have to perform monitoring and testing in production.

The shift to this QA technique also serves as a way to balance between quality and time to market: *teams speed up the release of software builds and continue testing them in production rather than trying to produce perfect and error-free builds.*

The need to deliver quality software at speed has created several trends in the software development landscape. One is the increase in shift-left testing adoption through employing various QA techniques early in the software development lifecycle.

Our correlation analysis revealed that there are strong positive relationships between QA techniques, which means that teams tend to apply a number of QA techniques together.

Automated testing is likely used together with:

- 1 Automated unit testing
- 2 Code review
- 3 Coding guidelines
- 4 Monitoring and testing in production

Two-thirds of respondents who performed automated unit testing **also applied automation in integration, system, and acceptance testing.**

About 61% of those using code review also applied automated testing. Our survey also showed that monitoring and testing in production is used in conjunction with most QA techniques, which is reasonable since there is no other alternative QA techniques used in production.

61% of those using code review also applied automated testing.



Recommendations:

This finding suggests teams embrace monitoring and testing in production as an approach to achieve **quality at speed**. Regardless of how much they test pre-release, there are inevitably bugs in production.

Thus, instead of wasting valuable time to make bug-free software and sacrificing time to market, teams need to prioritize monitoring and testing in production. It is an important complement to pre-production testing. At the same time, teams can build mechanisms to recover software from failure, making the practice of discovery and recovery from failure an important part of DevOps.

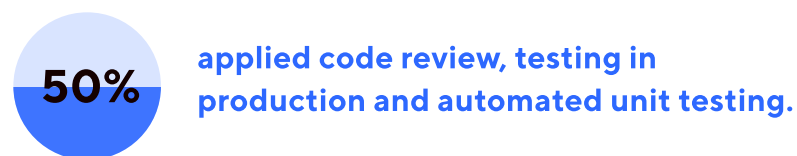
The most widely used QA techniques being the most effective ones

Automated testing is the most effective QA technique, followed by code review/inspection.

Software teams use a QA technique because of its effectiveness. It is true as our survey revealed that the most popular technique, automation for integration, system, and acceptance testing, is also the most effective, reported by 65% of respondents whose teams used this technique.



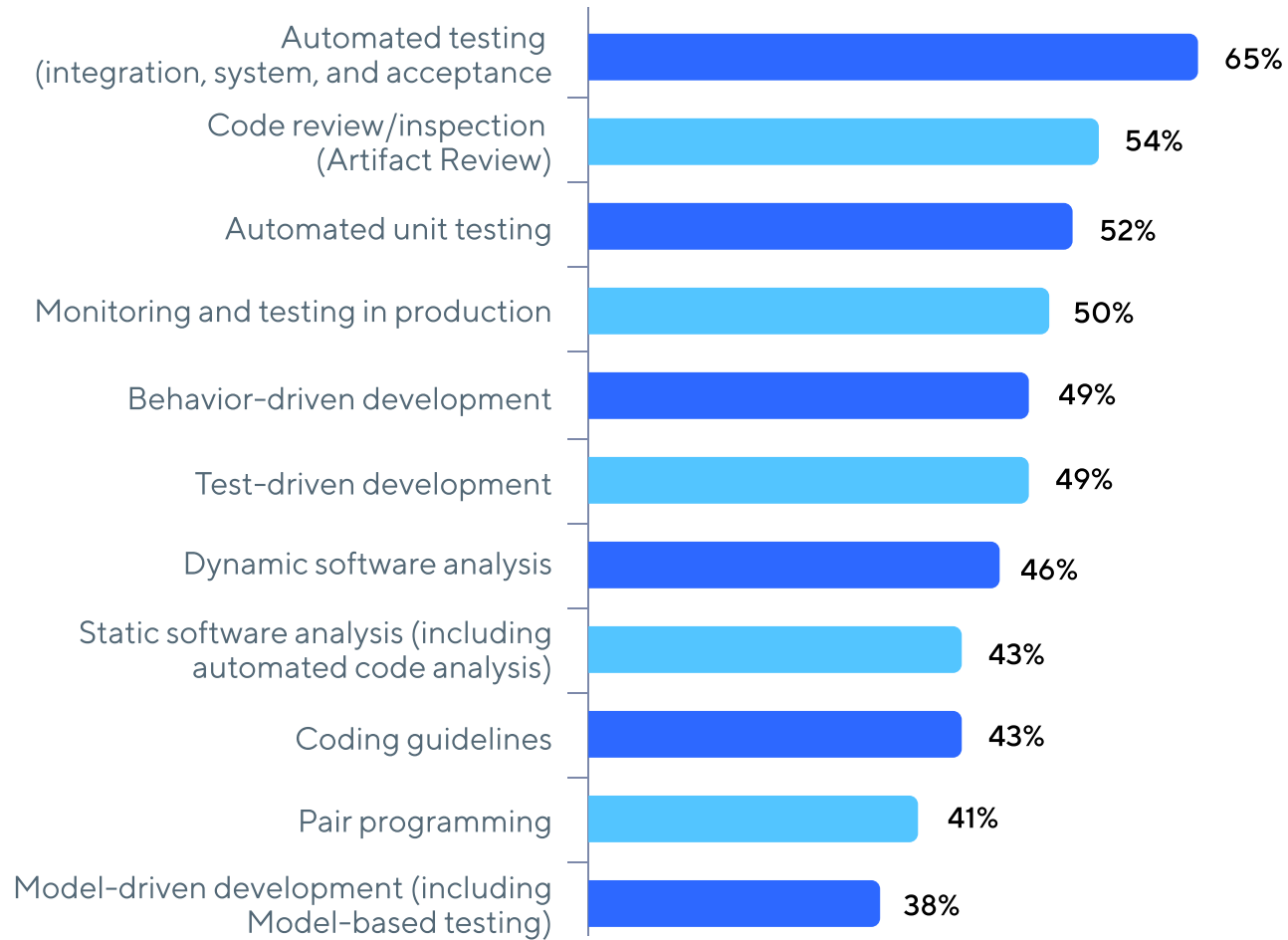
Code review, monitoring and testing in production, and automated unit testing are other most effective techniques, which were applied by more than 50% of respondents. As we can see in this result, respondents generally viewed that the most widely used QA techniques are also the most effective.



Noticeably, users of a technique tend to view it more favorably than those who do not use it. For example, only 17% of all respondents vouched for pair programming, while 41% of users of the technique identified it as the most effective.

These results seem to suggest that it is worth evaluating less popular techniques such as dynamic software analysis, pair programming, and model-driven development if they are suitable for your project's requirements and environments. They could potentially turn out to be effective for your projects.

Most Effective Techniques



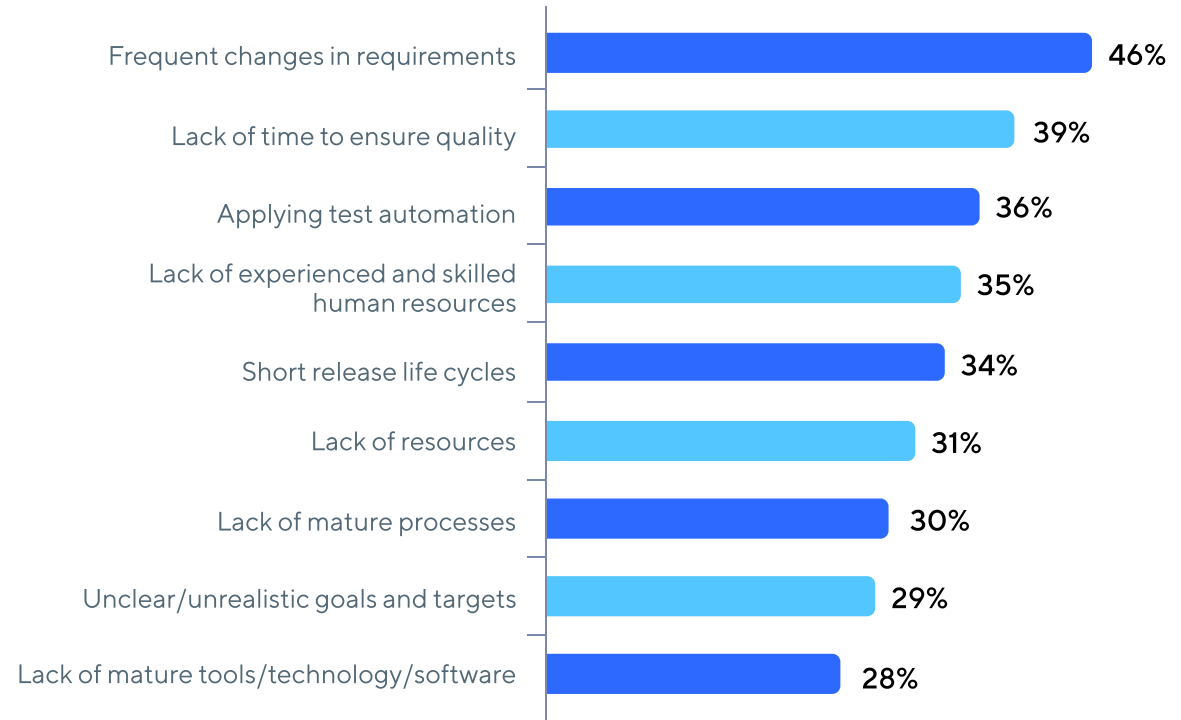
QA teams and challenges in high software quality

Frequent requirement changes and the lack of time to ensure quality are the most cited challenges in delivering high-quality software.

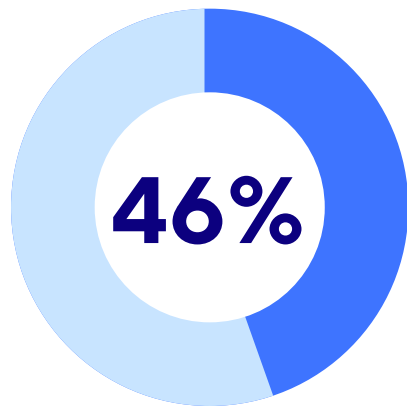
The need to deliver quality at speed creates a shift in QA approaches in which software teams have to employ faster release cycles, and test early and often in production-like environments.

However, frequent requirement changes and the lack of time directly go against implementing this shift.

Common challenges you face when delivering high software quality



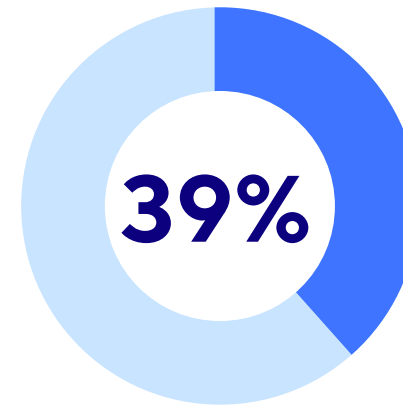
These are the two most common challenges reported by respondents in our survey with 46% citing frequent requirement changes and 39% citing the lack of time as obstacles for delivering high-quality software.



Frequent requirement changes



Frequent changes in requirements force teams to revise their tests for regression testing, which reduces the effectiveness of regression testing that is usually applied during automation. They also have to perform more manual testing instead, which slows down the process of testing and delivery overall.



Lack of time



When the team lacks time, they are unable to invest resources upfront for test automation, and instead resort to manual testing. With the lack of automation, it becomes more challenging to pursue short and rapid release cycles, leading to a vicious cycle of inefficiency.

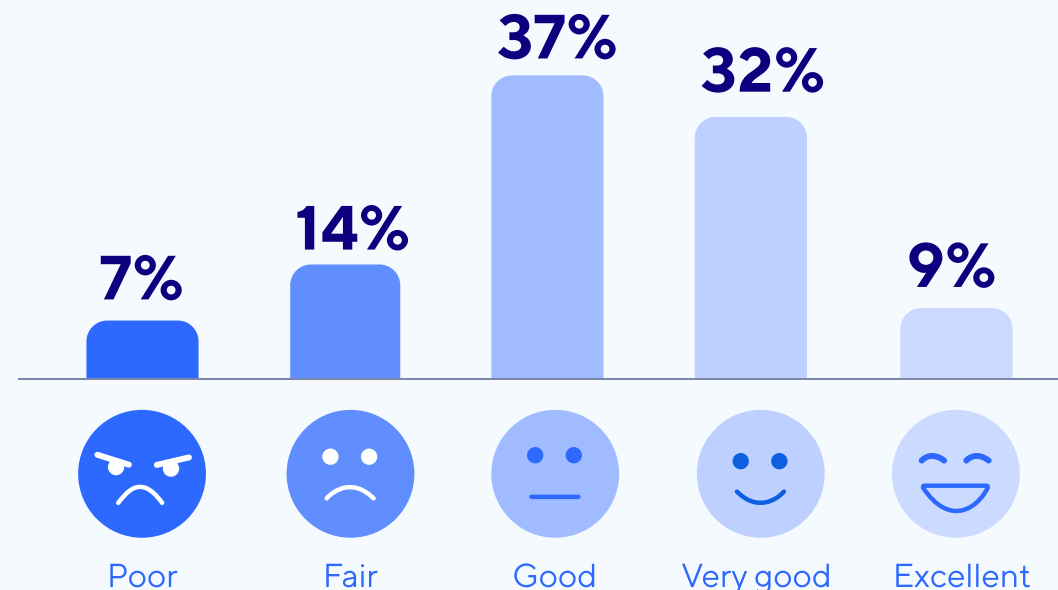
Level of satisfaction in software quality, QA practices and tools

Although people are satisfied with their software quality upon delivery, they are not equally satisfied with their QA practices and tools.

The level of satisfaction related to software quality upon delivery is probably one of the most important measures of success for software teams. Our survey asked respondents to offer their perceived levels of satisfaction with software quality upon delivery, QA practices, and tools.

Despite challenges in delivering high-quality software, most of the surveyed individuals are satisfied with their software quality. An overwhelming 78% of respondents rated their software quality upon delivery as good or better while just 7% and 14% of them reported poor and fair software quality upon delivery, respectively.

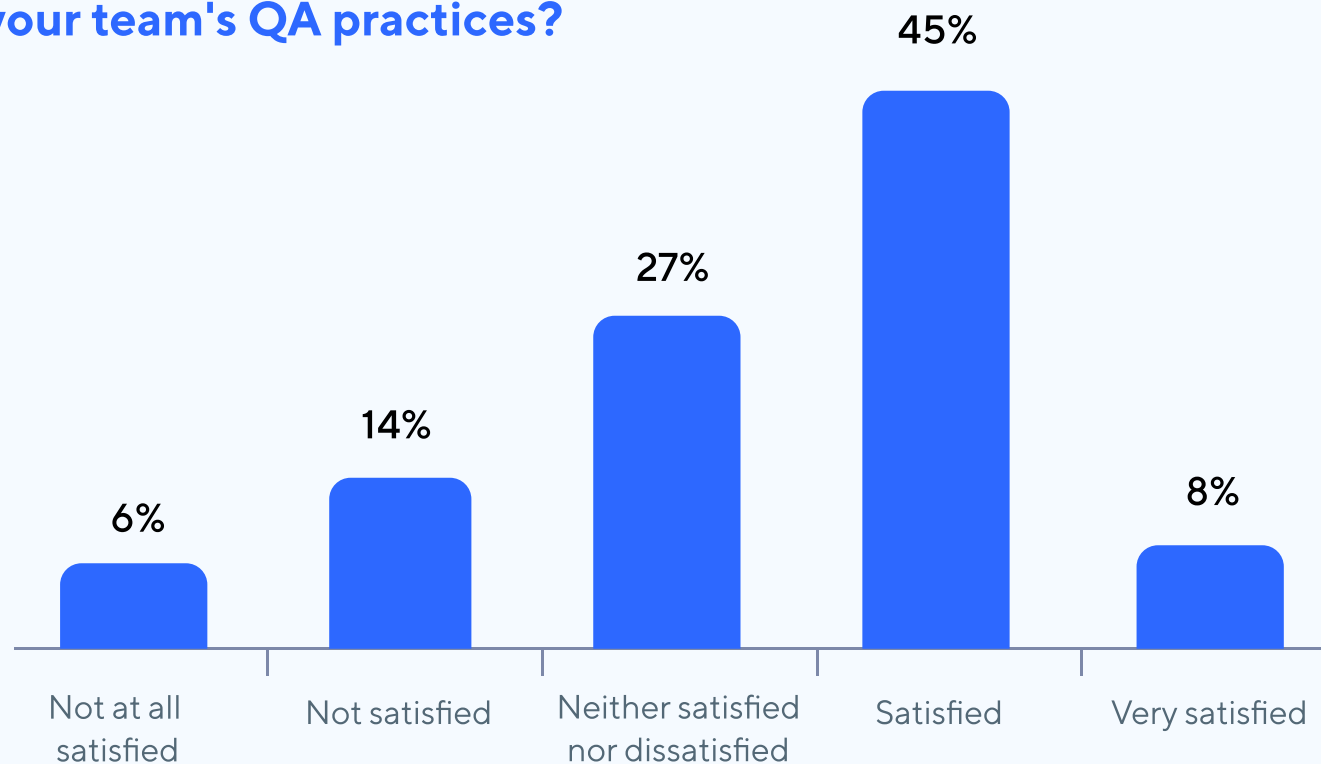
What is the level of your software upon delivery?



However, when asked whether they were satisfied with the QA practices and tools used for their projects, their satisfaction levels are not equally high compared to those of software quality. About half of the respondents said they were either satisfied or very satisfied with their QA practices (53%) and tools (50%).

Roughly 20% and 22% of them were not satisfied with the QA practices and tools used for their projects, respectively. Many participants in our follow-up interviews noted that they were not very happy with automation tools and had to rely on manual testing instead.

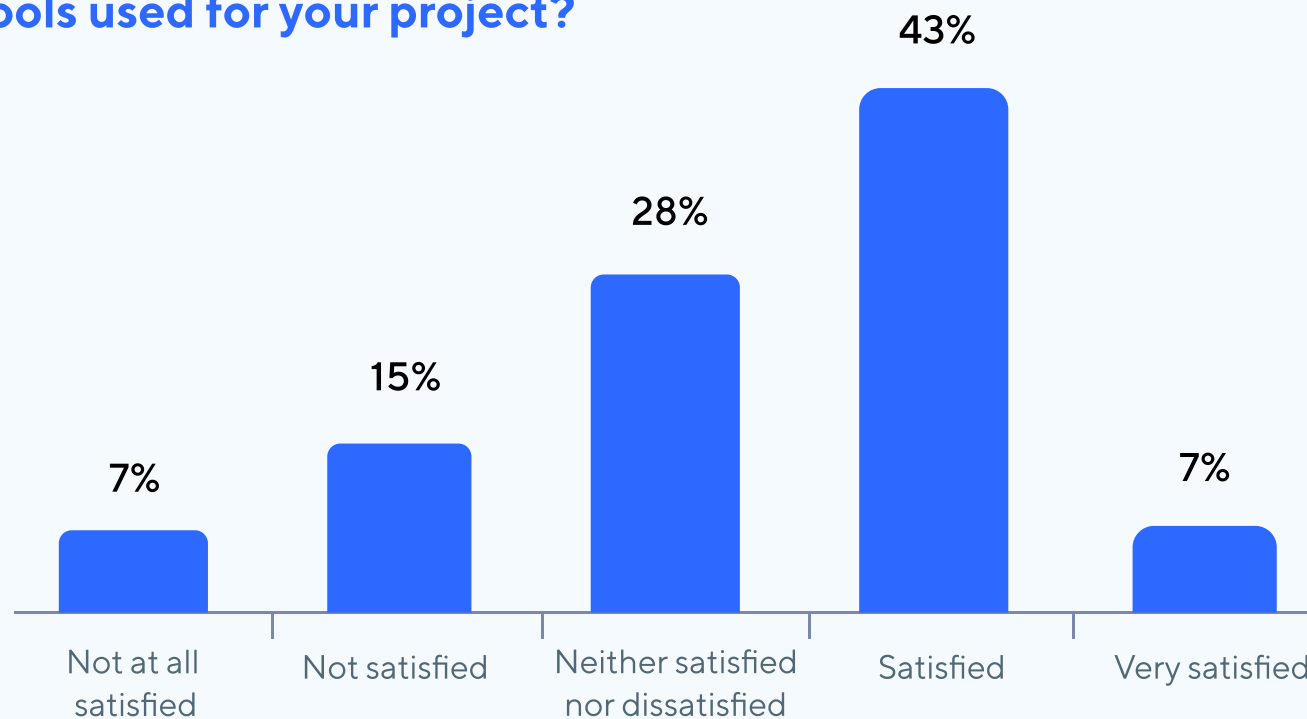
How would you rate your satisfaction with your team's QA practices?



These results showed that while people are happy with their software quality upon delivery, they expect more from their QA practices and tools. At present, they can overcome limitations in QA practices and tools to deliver high-quality software...

...but they look forward to improving their practices and using better tools in the long run. "We are satisfied with the QA practices and tools used in our projects," said one interviewee, "but in a highly competitive market, tools with AI technology integrated seem to be the right approach for the future."

How would you rate your experience with tools used for your project?





Test Automation

Test automation is reported in this study as the most widely used and effective QA technique. It is essential for achieving quality at speed in today's software development with Agile and DevOps methodologies.

This section describes the findings related to the automation tool landscape, common challenges in applying test automation, and the return-on-investment for test automation.

“ The key benefit of automated testing (across your application layers) is feedback. Ultimately, we have to mitigate against the introduction of bad code into our software codebases.

When software is developed under a test-driven development paradigm, and the tests are written with feedback and information sharing as the goal, the inevitable bugs that our automated testing pipelines discover are rapidly resolved. The reason why we want tests to run quickly, and to fail quickly, is to get feedback.”

Ron Powell

CircleCI
Manager, Marketing Insights and Strategy

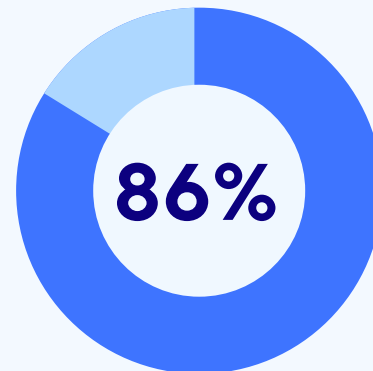


The continued decline in popularity of free open source automation tools

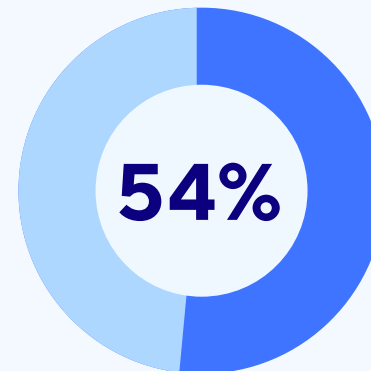
A successful application of test automation has to start with choosing the right set of tools. There is a wide variety of test automation tools that software teams can choose from, ranging from open source to commercial tools, and something in the middle.

Selenium continues to be the most popular automation tool, which is used by 37% of respondents. Yet, its popularity has declined sharply over the last four years, going from 86% in 2018 to 54% in 2020 in our previous surveys.

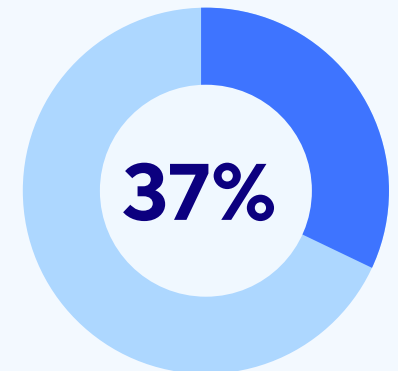
Selenium



2018



2020

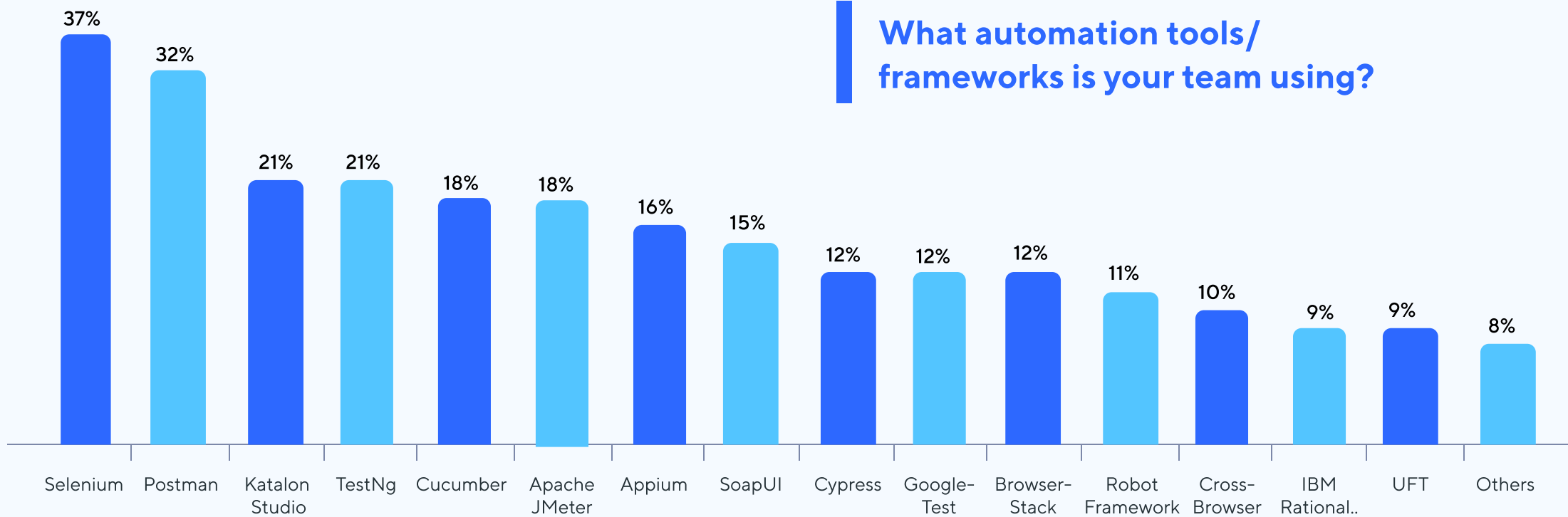


2021

One reason is that automation tools and frameworks based on Selenium such as Katalon and Perfecto have become more powerful to eliminate the need of building custom frameworks using Selenium.

This survey also found Postman as a popular alternative to Selenium for API testing and monitoring.

This trend of declining reliance on building custom frameworks using open source solutions for test automation is good news for the software community as they would spend less time and expertise on building and integrating various open-source solutions and more time focusing on what really matters – automated testing.



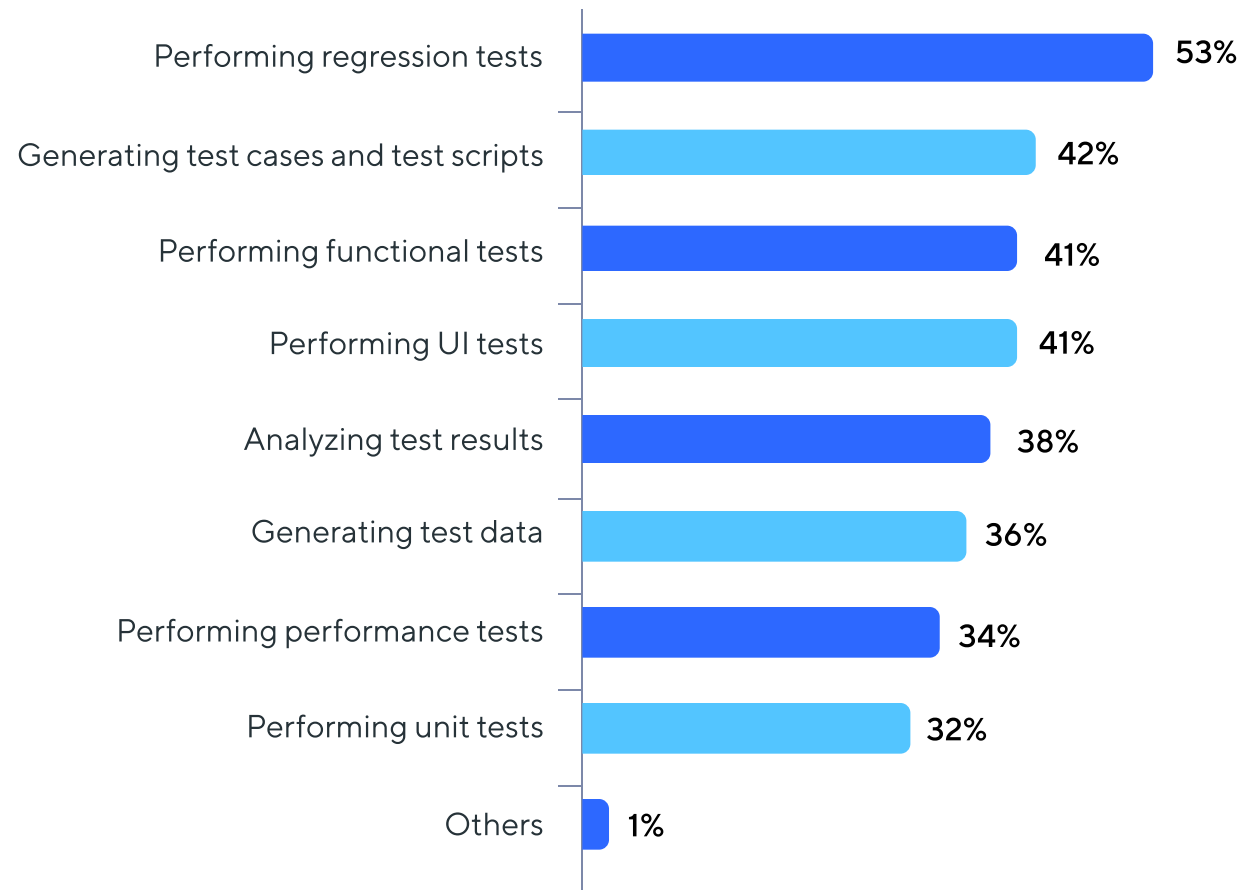
The race to adopt test automation

Test automation is increasingly applied to more testing activities

It is a fact that when teams apply test automation, regression testing is the activity they start with. This survey confirms this strategy with 53% of respondents applying automation for regression testing.

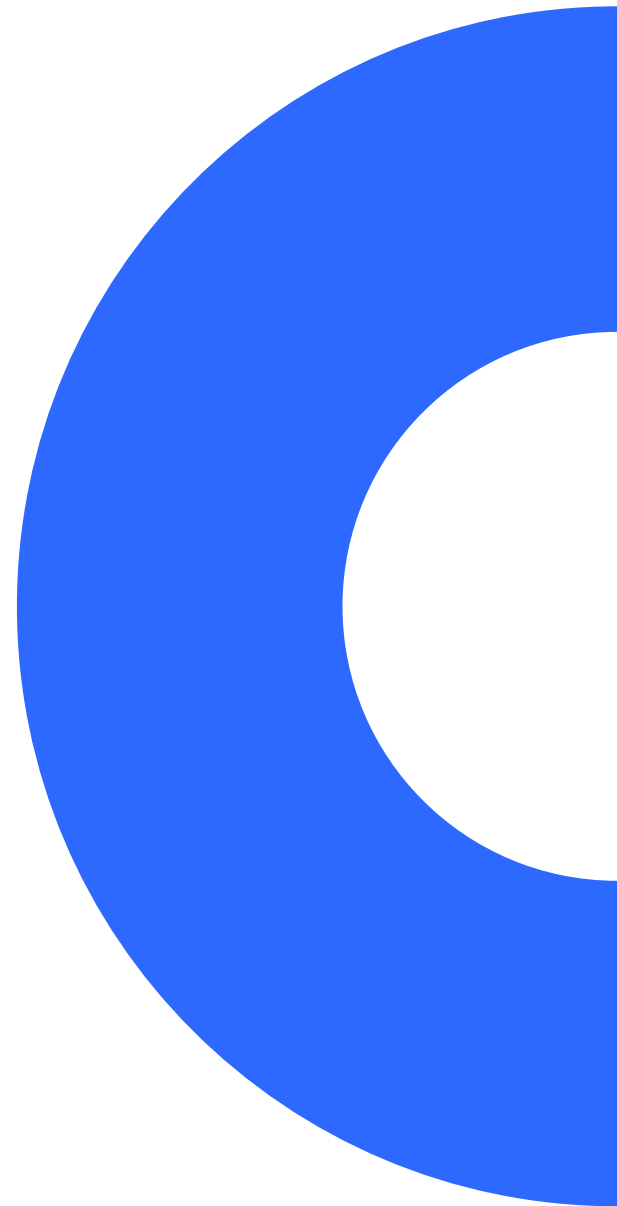
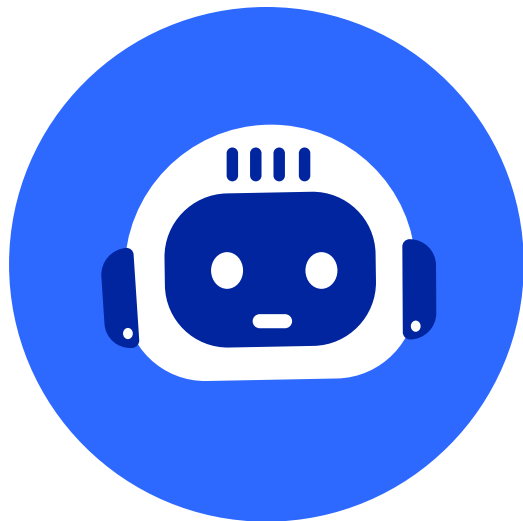
However, it is increasingly used for other activities including analyzing test results (38%), generating test data (36%), and performing performance tests (34%).

Which of the following testing activities does your team apply test automation for?



This finding suggests that as **automation tools become more mature and powerful**, they offer greater capabilities for performing more testing activities that have been traditionally done using manual testing. In other words, more testing activities and types that have been so far tested manually will be performed automatically using tools in the future.

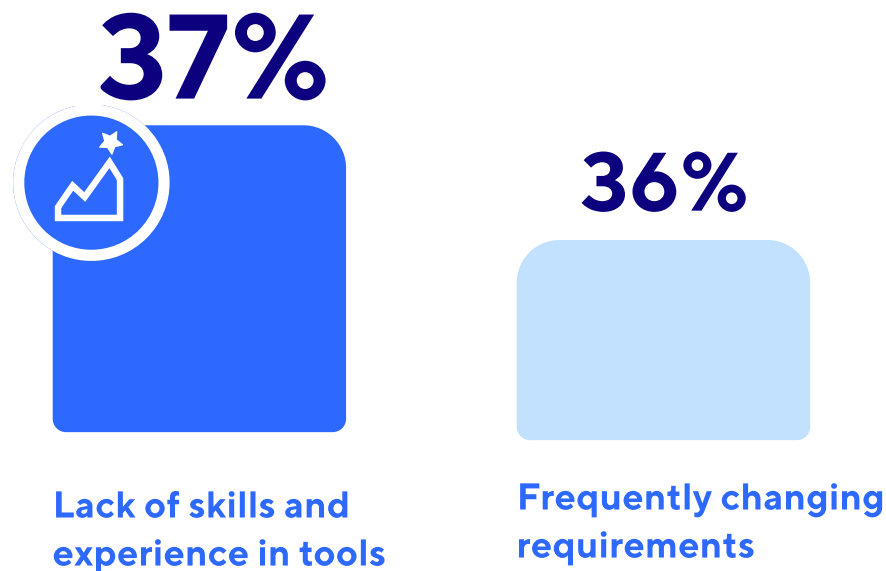
The application of AI technologies to automate more testing activities and to offer intelligent decision support for test automation is likely a trend in the advancement of tools.



Persistent obstacles in applying automation tools

The two most common challenges in applying test automation are the **lack of skills and experience in tools** (37%) and **frequently changing requirements** (36%). This finding is the same as our survey reported in 2020. QA practices and tools have changed much over the years, but these challenges continue to persist.

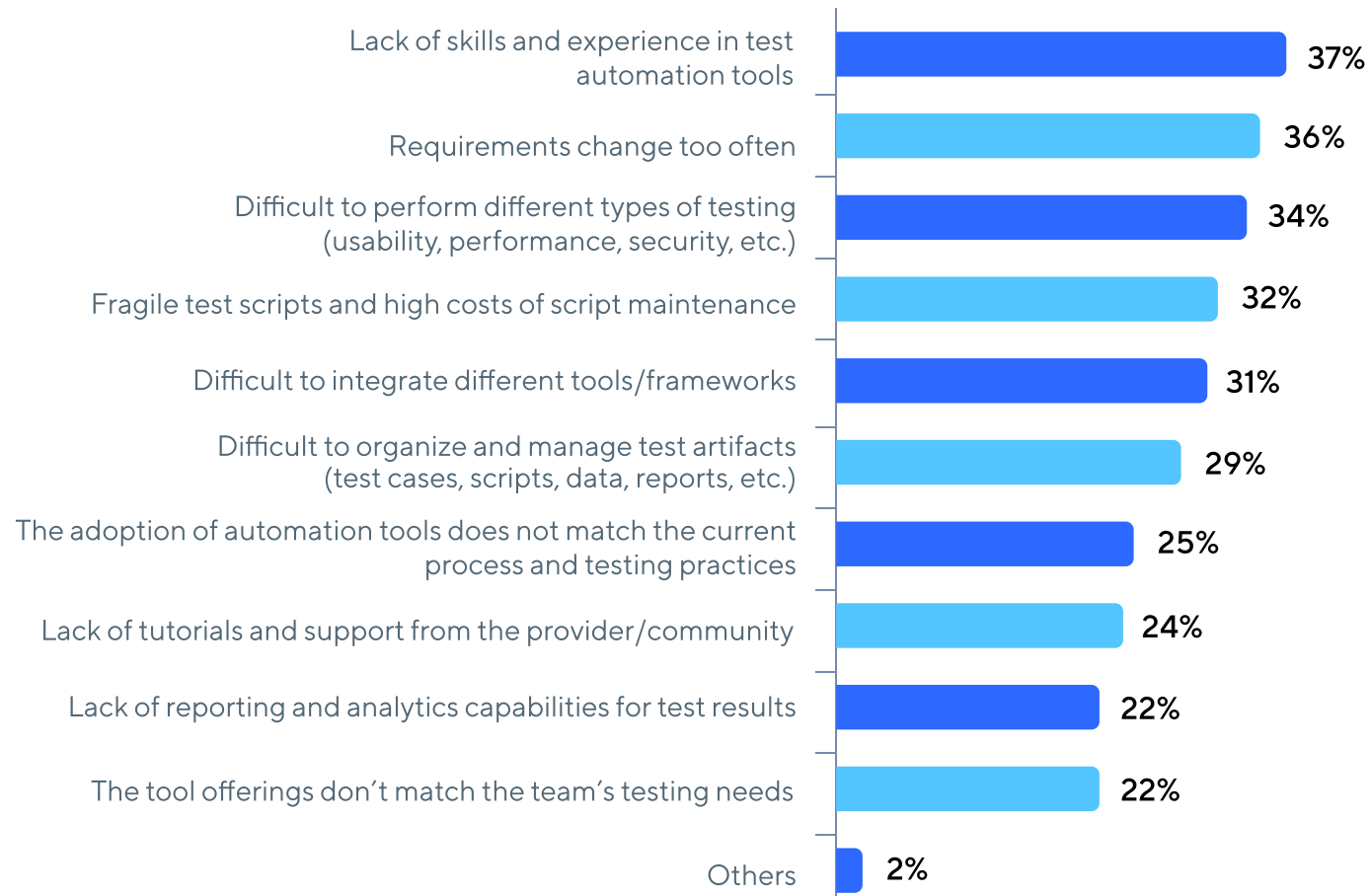
The most common challenges in applying automation tools continue to persist.



Indeed, the top five challenges remain the same as our previous survey results except that this year's study shows an increase in the level of the challenge related to script maintenance (32% and ranked 4th) compared with the finding in 2020 where the challenge was ranked relatively low (5th).

Surprisingly, this survey shows that the **tool offerings** is ranked **the lowest** although it was ranked high in our previous survey. This finding indicates that test automation tools have become more mature and powerful in satisfying testing needs.

The most common challenges that your team has when using test automation tools?



Factors to ensure positive ROI

Most people have achieved a high ROI for test automation.

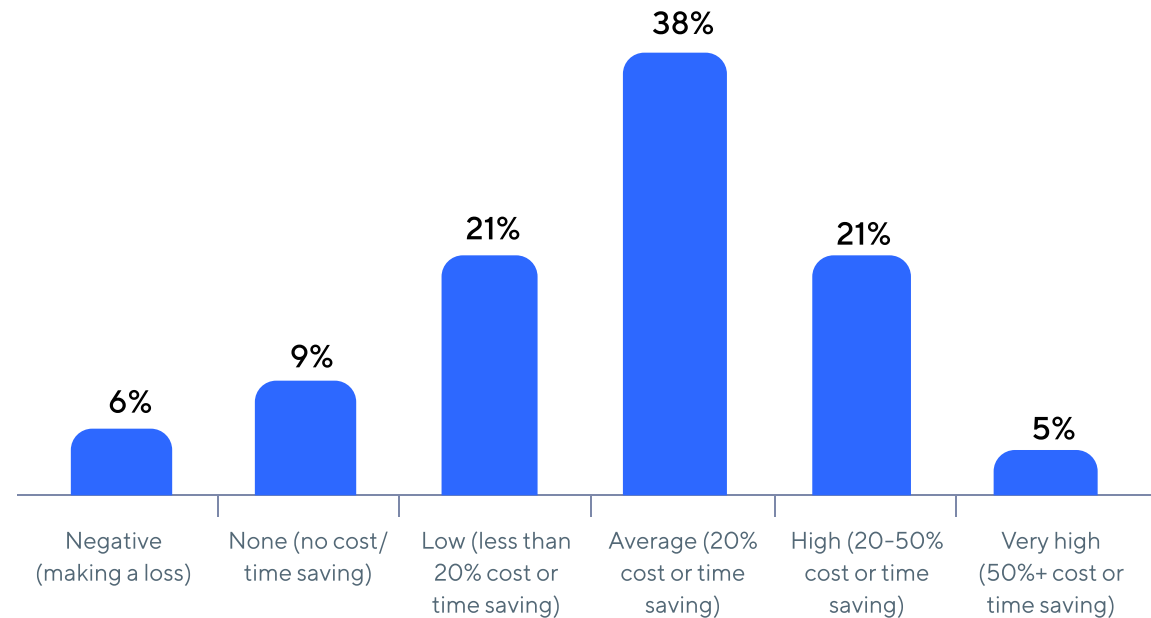
Despite challenges, the good news is that the return on investment for test automation is relatively high. Most respondents have seen a positive ROI with the majority of respondents (63%) reporting to have seen 20% or more of the cost/time-saving in their test automation investment. Only 6% of them experienced a loss, and 9% have seen no saving.

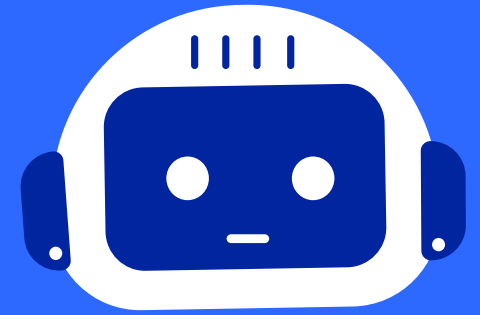
The level of ROI depends on a number of factors such as the effectiveness of training, choosing the right automation tasks, test maintainability, reusability, and persistence in applying the tool.

Organizations should provide sufficient training to their teams on practices and tools when applying automation

Automation is known to be challenging and time-consuming in the beginning, which causes many teams to give up. One of our follow-up interviews pointed out that many companies would acquire an automation tool and use it for initial days, but eventually return to their regular practices.

What is your return on investment (ROI) of test automation for your project?





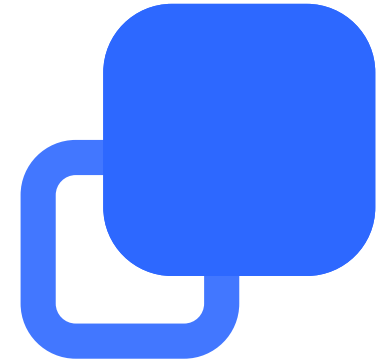
Artificial Intelligence in Test Automation

The application of artificial intelligence to test automation has become an exciting trend in recent years. The software industry has seen a sharp increase in tools that offer AI capabilities for test automation activities ranging from smart test creation and data generation to intelligent test maintenance and execution.

These advancements in the application of AI to software testing have the great potential to overcome many challenges in automation.

The advancements in AI open up opportunities to automate many testing activities that have often been done manually.

With AI-enabled tools, time-consuming activities such as test design and generation, data preparation, verification, and test maintenance can be more efficient and effective.



Example

AI-enabled functionality can learn to automatically revise test cases and scripts whenever the AUT is changed, reducing the effort spent for maintaining tests and enabling teams to speed up the regression testing process.

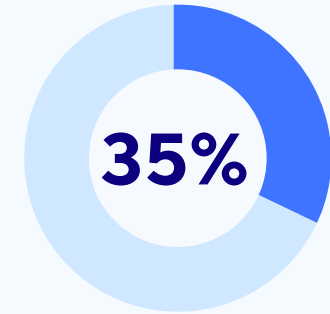
AI for test automation is still in its infancy.

Despite people's high expectations, AI for automation is still in its early stage of application. Our survey showed that less than 50% of respondents reported using some AI capabilities for their test automation. This result is surprisingly low given that AI has received a lot of attention from the test automation community in recent years.

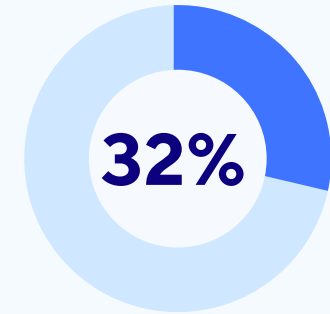
Among those who use AI capabilities with their automation tools, about a third use it for automated test cases and scripts generation (35%), test data generation (32%), automated defect detection (30%), and test prioritization and selection (30%).

Generating test cases, scripts and data is a crucial process for test automation. Tools with AI capabilities can improve this process by automatically creating test artifacts faster and better than human testers. Also, the AI-enabled test prioritization and selection capabilities allow testers to reduce the number of tests needed and optimize the process of test execution.

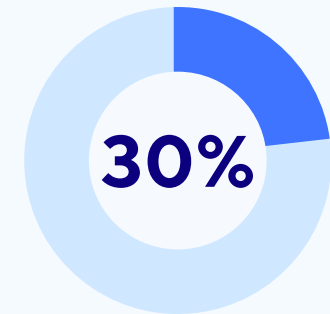
Scripts generation



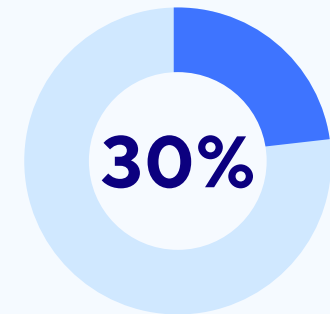
Test data generation



Automated defect detection

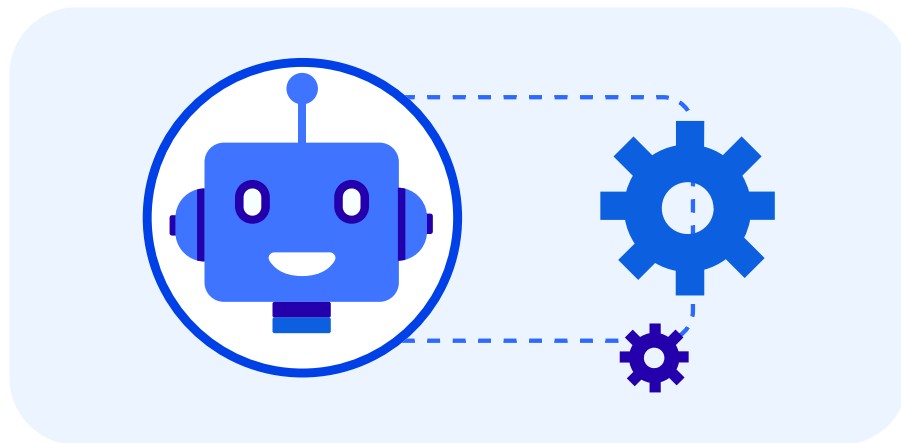


Test prioritization and selection



There are several reasons for the low adoption of AI.

1 One reason is that AI capabilities for automation are still limited and underdeveloped. They are presently used for simple and easy-to-automate activities such as generating simple test cases from requirements using natural language processing (NLP) and comparing screens using computer vision. Complex testing activities such as automated test data creation, assertions generation, and test evaluation require more mature AI technologies to handle.



2 Another reason is the lack of tools that support AI for many interleaved activities of the testing life cycle. Today many tools integrate AI to address just a few specific and narrow problems, e.g., handling test flakiness using auto-healing and reducing script coding via exploring elements on the user interface. This lack of comprehensive AI-enabled tools requires teams to adopt various tools if they want to use AI technologies.

3 The complexity of the test automation process and the challenge in introducing and integrating AI technologies to automation tools are also the reasons. AI models require a lot of data to train and validate, but software organizations are not ready to share their data.

Trends in AI for test automation

Although receiving a low level of adoption today, AI has a lot to offer for test automation to support a wide range of testing activities.

Our follow-up interview participants suggested that integrating AI capabilities in test automation tools could take the automation testing process to a different league altogether. An automation tool with AI capabilities can significantly reduce the effort for time-consuming processes such as scripting, test design, test data preparation, script maintenance, and failure analysis.

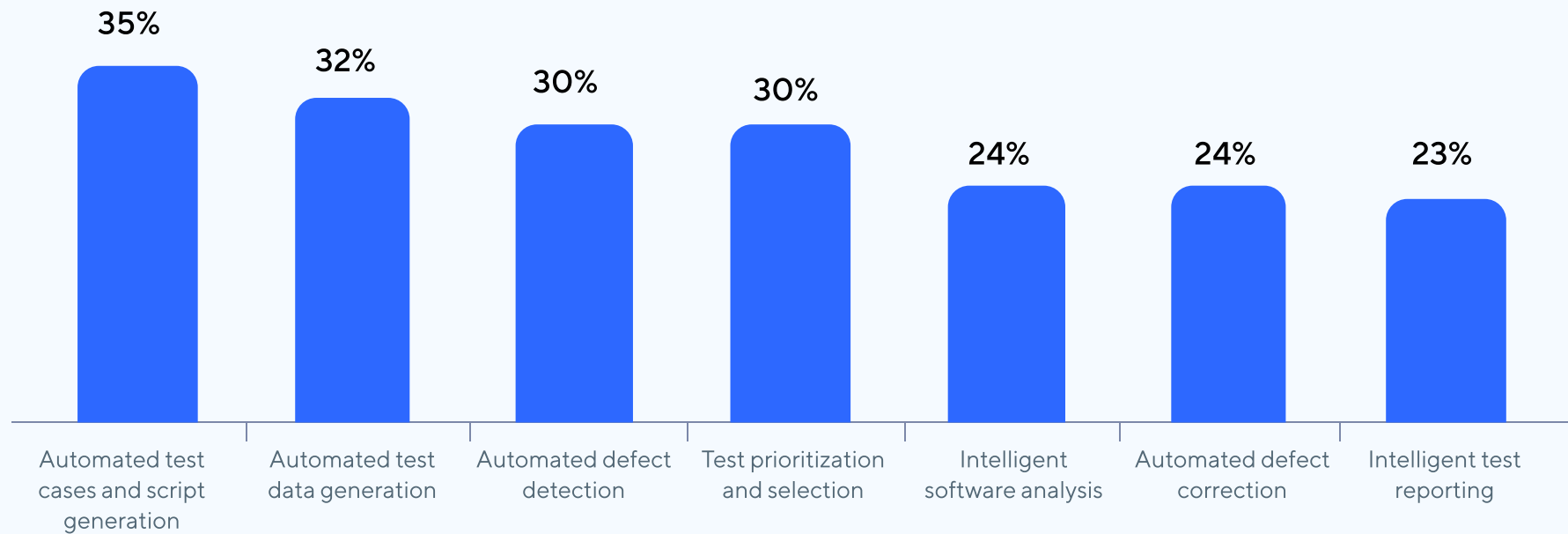
We expect more advanced AI technologies to be developed and applied for test automation. More comprehensive automation tools will leverage AI technologies to support a wide range of testing activities, resulting in more autonomous testing life cycles.



We will likely see further advancement and integration of AI technologies in the following areas of testing in the future:

1. Test Planning and Management	2. Test Generation and Maintenance	3. Test Selection and Optimization
<ul style="list-style-type: none"> • Intelligent software analysis • Test effort estimation • Intelligent test decision making • Test efficacy and robustness analysis 	<ul style="list-style-type: none"> • Automated test case, script, and data generation • Automated test migration • Test reuse analysis • Automated test repair 	<ul style="list-style-type: none"> • Test prioritization and selection • Test suite optimization • Test impact analysis • Test coverage optimization
4. Test Execution	5. Test Evaluation & Report	6. Test Monitoring
<ul style="list-style-type: none"> • Applying computer vision to automation (visual testing) • Application exploration 	<ul style="list-style-type: none"> • Automated defect detection • Intelligent test analytics • Defect prediction • Defect classification and triage 	<ul style="list-style-type: none"> • Usage pattern analysis • Performance analysis

Do you use the following AI capabilities in your test automation tools?





Looking Ahead to 2022





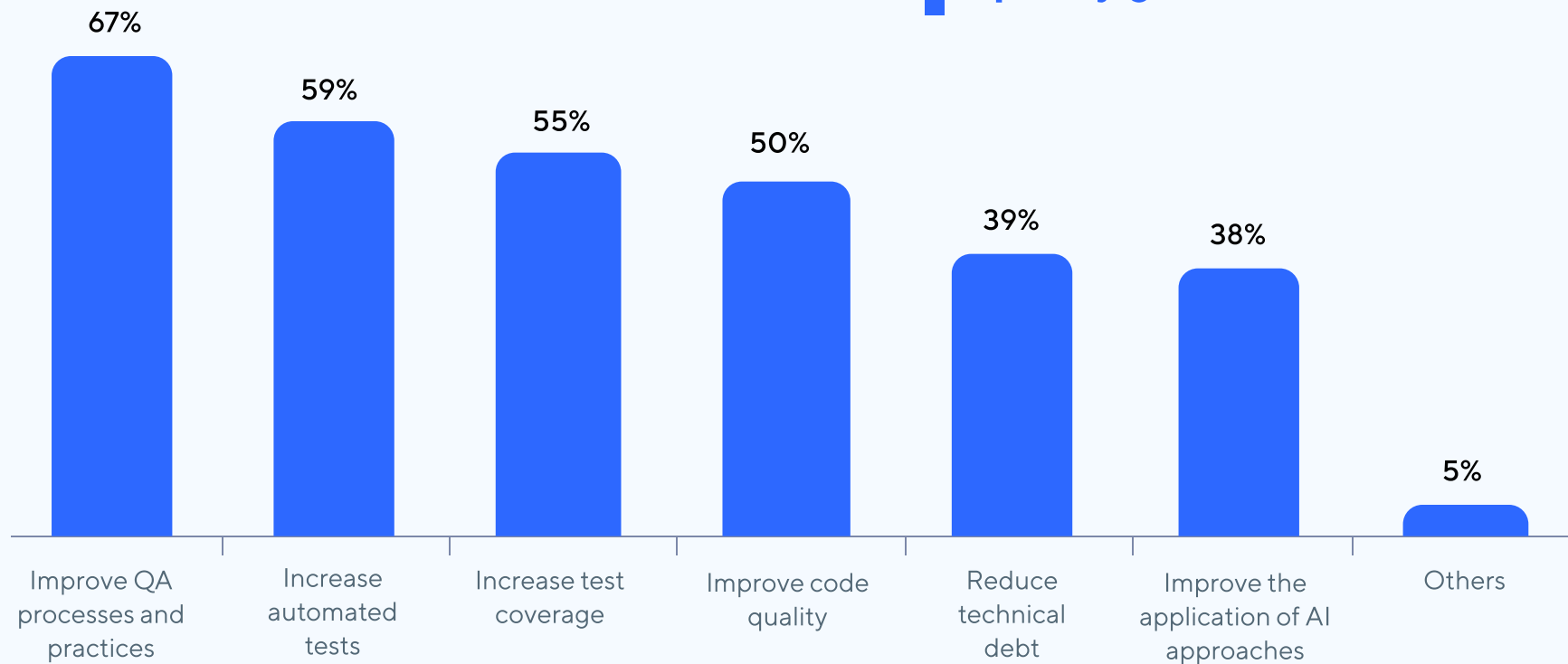
Our survey asked respondents about which QA practices they planned to improve and what quality goals they would pursue in the coming year, finding that the increase in adoption of test automation continues to be a trend as more than 60% of respondents said they are likely/very likely to improve their adoption of automation (automated unit testing, automated test case/data generation, and test execution) in the coming year.

Improvements in adopting automation should be a high-priority action for many teams to address the challenges in automation-related practices and tools that remain. Teams need to cope with the lack of skills and resources for automation and find ways to overcome the problems caused by frequently changing requirements.

The prolonged COVID-19 pandemic forces software professionals to continue working from home, affecting their well-being and mental health. Mental health issues such as stress, fatigue, exhaustion, and loss of motivation can severely influence the effectiveness of QA activities.

These issues emphasize the necessity of improving QA practices, processes, and tools employed by teams.

What are your primary software quality goals in 2022?





Case Study Analysis

**Technology Solution Provider Reduces Time to Market,
Execution, Maintenance Time, and Stress With Katalon**



Summary

A leading technology solutions provider tapped Katalon to replace its legacy test automation system. In doing so, the firm was able to significantly reduce both maintenance and execution time, allowing it to better serve its pharmaceutical company partners with powerful research that enables them to get life-saving drugs to the market.

“We selected Katalon because it gives us the ROI we wanted and it’s very stable. Now if our test script fails, we know it’s the AUT’s fault, not the platform’s,” said the QA manager. “Instead of thinking about which test to execute based on a code change, we can selectively execute those that are relevant to the impact analysis result.”



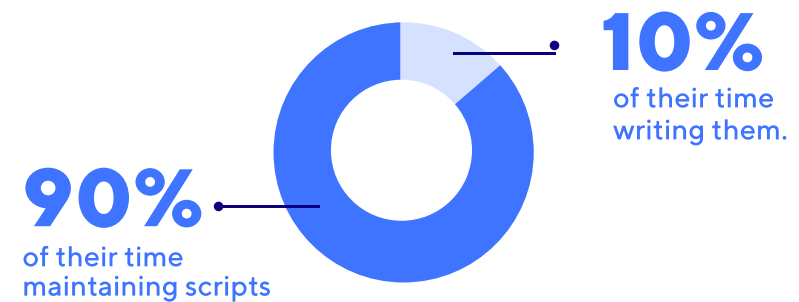
The Challenge

The client, a leading technology solutions company for clinical trial innovation and evidence, aims to empower its partners to transform lives by parlaying deep scientific expertise at a global scale into the broadest possible endpoint technology platform.

The company is committed to innovating the future of clinical trials through technology and having a positive impact on global healthcare.

As the client began to rapidly scale its test automation operations, it found that its developers were spending about **90%** of their time maintaining scripts and only **10%** of time writing them.

The company needed to streamline and simplify its software testing automation so that its engineers could spend less time maintaining scripts and more time analyzing them to identify problems and more rapidly iron out product defects.



Consequently, the company could scale faster and better serve its clients and partners, who are mainly large pharmaceutical companies.

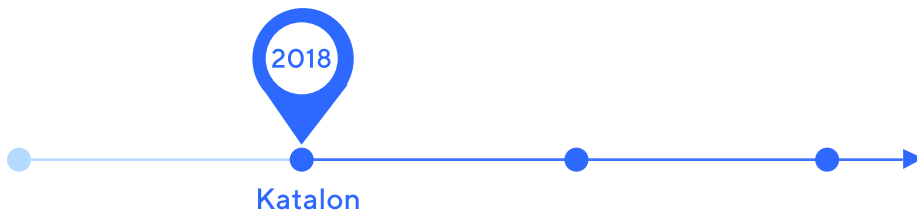
The software tested would be used to analyze data from large clinical studies. And since the outcome could potentially contribute to life-saving drugs, engineers needed to be certain their testing automation was as efficient and effective as possible.

“We have a system where we deliver studies to the customers, and every study is different,” said the manager. “Before we release those studies into production to users, we need to test them, and it becomes impossible with the increasing volume of studies.

Scaling horizontally by delivering more resources to deliver more studies was not a viable solution. So we started looking at automation so that we didn’t have to test every study again and again.”

The Solution

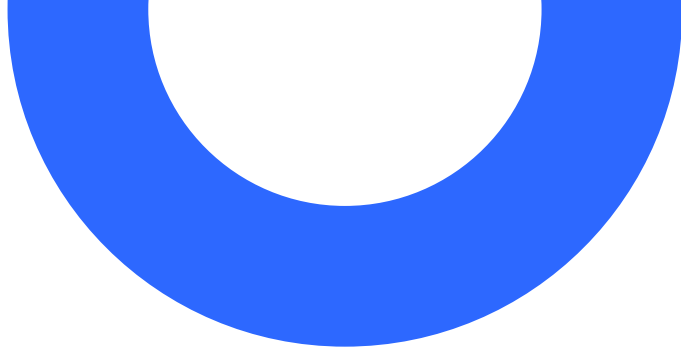
After conducting a thorough ROI analysis of several test automation tools on the market, the client chose Katalon—specifically, Studio Enterprise and Runtime Engine—for its testing automation solution, and began using Katalon in 2018 to test web (user interface functionality) and native mobile applications (both iOS and Android), as well as APIs.



The company had done manual testing for the last 11 years before switching to test automation. After a number of years employing Selenium-based frameworks, namely Robot Framework, Protractor and Selenium itself, the client found that it was too programming-centric and required too much maintenance.

“Sometimes they worked, sometimes they didn’t,” said the manager.

“And the manual testers in my team found it very challenging to start scripting right away, while still having to constantly fix the old tests and have less effort to develop newer scripts.”



“ Our work is in a heavily regulated industry so we need a lot of documentation around whatever we do. We need to document our test steps with evidence because we are subject to 70 to 80 audits a year. So documentation was a key part of it.”

The QA manager and his team eventually built a framework around the mentioned open-source test automation framework to help the engineers automate it and to give the auditors confidence in the designs.

“But what we found over time was that the maintenance became a nightmare for us,” he said.

“We integrated the testing into our pipeline and we never had a pipeline that was clean-never. We were always executing a test on a local machine and saying, ‘here it works, there it doesn’t, and we don’t know why,’ so we wanted to move on and wanted to do something that was more stable.”

Seeking more stability and faster releases, the company began using Katalon for web and mobile application testing. “Everybody was immediately happy with it,” the manager said.

Using GitLab as a CI/CD system, the client has **50** test suites running in tandem with about **150** scripts in total and executes them using Kubernetes with one dynamic port for each suite.

Test engineers have integrated their tests into pipelines and execute the tests automatically with every code build.

From validation, staging to production environments, there was a set of different profiles for their automated test scripts to run against.

Additionally, the support for Docker images in Katalon Studio has also made everything much easier and quicker with lower memory consumption of their team’s machines.



The Results

► More Efficient Use of Internal Resources

After a six-month-long process of rewriting their legacy scripts to migrate to Katalon Studio, their team immediately saw the results they wanted, especially in optimizing internal resources.

“It was definitely worth the effort. With the same team, I was able to handle more business,” the manager said. “Not that we let people go – we took on more work with the same team and level of productivity.”

► Significantly Reduced Execution And Maintenance Time

Before, running through hundreds of test scripts, one after another, took the firm weeks. As a solution, the firm implemented some new key test designing concepts: limiting the execution time to only 15 minutes per script and breaking those that exceed the set time into multiple scripts to run parallelly.

“Our execution time did actually come down from several hours to minutes and executed all our test suites within 15 minutes – and that’s for hundreds of scripts. Also, our maintenance time has gone down by at least 60%. Katalon’s AI capabilities such as Self-healing and Smart Wait help reduce maintenance time and test flakiness.”, said the QA manager



15 mins

-60% maintenance time

► Transformation Of DevOps Infrastructure

“It was very easy to plug test automation into our pipeline, at any stage, with Katalon,” an engineer said.

“Every test suite becomes a test automation stage for us. Also, the biggest challenge we had with every other tool was maintaining compatibility between the browser version and the driver version.

With Dockerization, we don't have this problem and Katalon makes Dockerization very easy. We can use any version of Chrome we want.

And also, one test suite issue doesn't have an impact on another one. So if a node goes down, no problem, the test will still get executed.”

► Productive Failure Troubleshooting

“Katalon has introduced a lot of features for debugging that are really helpful. If our test fails at a particular point in time we can start our execution from the same point again.

We don't have to go back 50 steps and re-execute. We can put breakpoints in between and do better debugging.”

► Enhanced Product Quality

“We’ve reduced our production defects from a large number to lower single-digit numbers for each product,” the manager said.

“A key part of our mission is delivering high-quality products to our customers. Our work is also measured on what percentage of our requirements are covered through automation tests.

That is a key metric that I need to report to executive leadership every month. And Katalon has helped me with that.”

► Improved Time To Market and Reduced Stress Levels

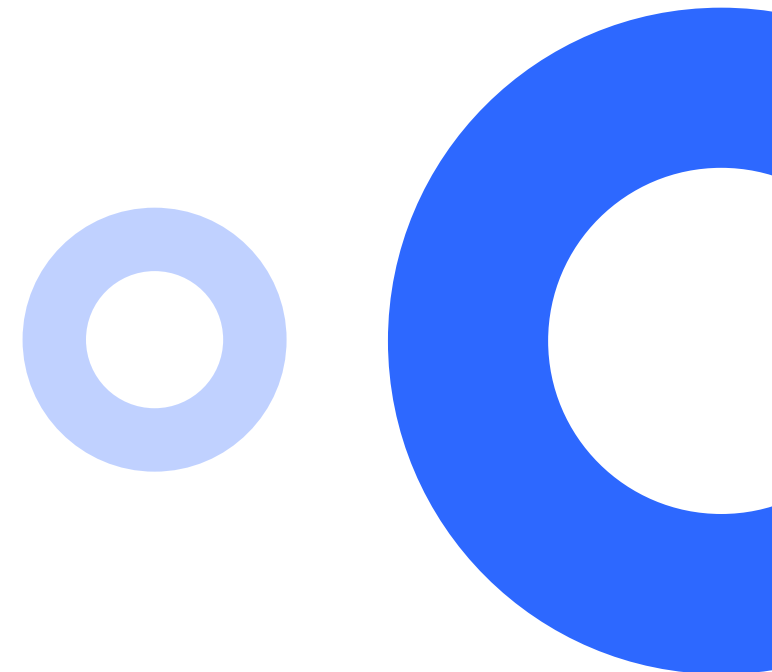
“Switching to Katalon has allowed us to not miss a single committed timeline in the last three years,” the manager said.

“And that’s something that has never happened before. We used to say, ‘Well, we can’t release it now because we haven’t completed testing.’ That case doesn’t happen anymore.”

Overall, the fear of unfinished tests and delivery dates nearing no longer were piling on the pressure for their QA engineers.

► Security Compliance

Operating in a highly regulated environment, not having to work with additional third-party software was key for the client’s data to not be at risk. Test reports, in particular, were generated automatically and locally stored in Katalon Studio.



Thank you for sticking with us till the end and becoming a part of “**The State of Quality Report 2022.**” We hope the report's insights have been helpful to both you and your customers to thrive better.

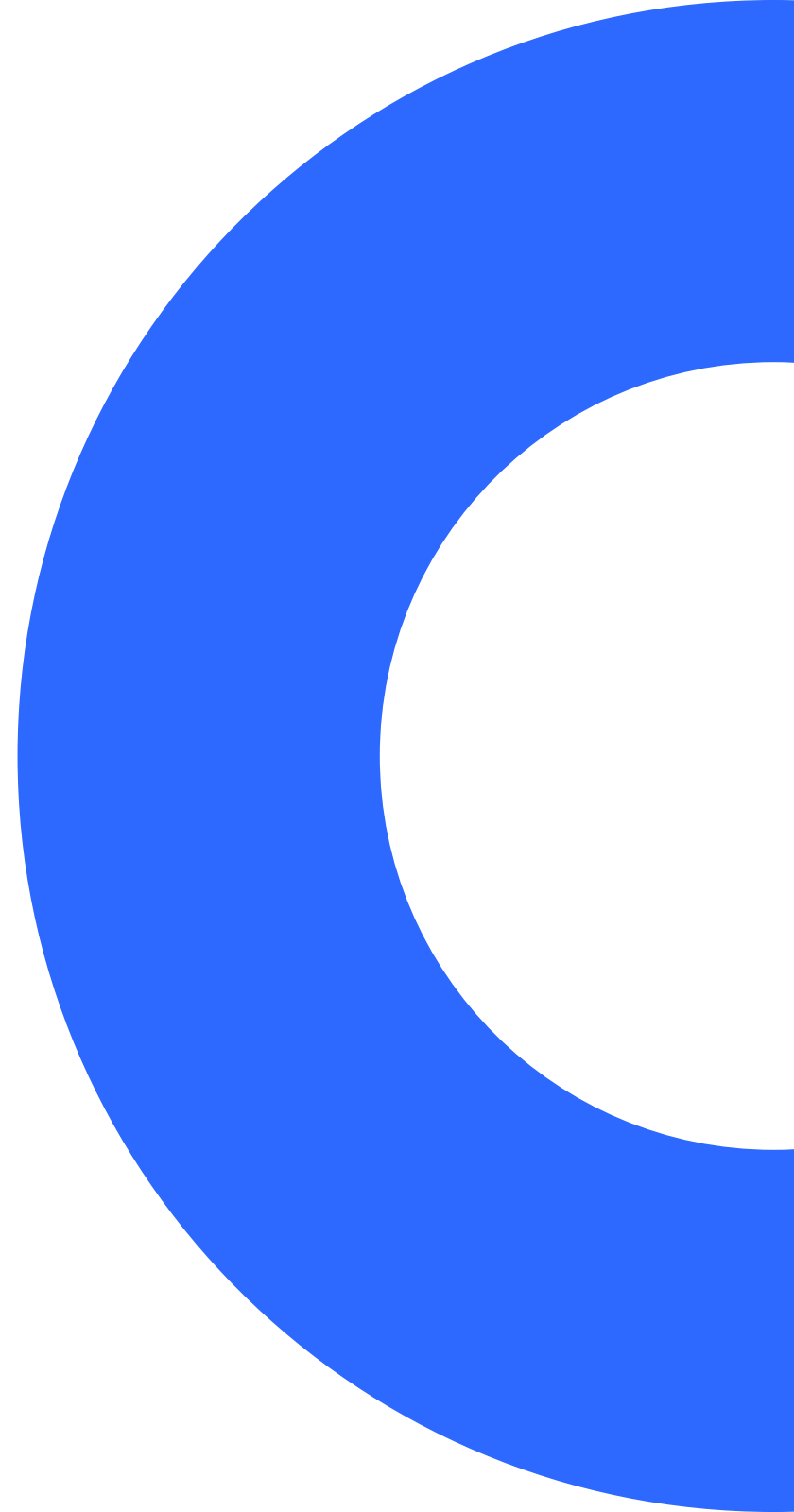
At Katalon, we always keep a close eye on every slightest change in the test automation space to see where we can help steer the industry forward. And of course, this mission is forever incomplete without your comments. Yes, yours.

So why not [LEND US YOUR THOUGHTS](#) and win a **\$30** E-gift card?

May we ask just 4 simple questions to improve our report?

Yes, sure

HAPPY TESTING!



Appendix

Methodology

This study was carried out in two phases. In the first phase, we conducted an online survey of **3,000+** respondents with direct involvement in quality practices of software development projects.

The survey consisted of **28 questions** concerning software quality techniques, test automation, and the adoption of AI. Our survey was distributed via various channels of software development and testing communities around the world.

In the second phase, we performed follow-up interviews with experts and experienced professionals to gather their insights and opinions about the results we obtained in the first phase. The results obtained from the first phase and the prepared questions were shared with the interviewees during the interview.

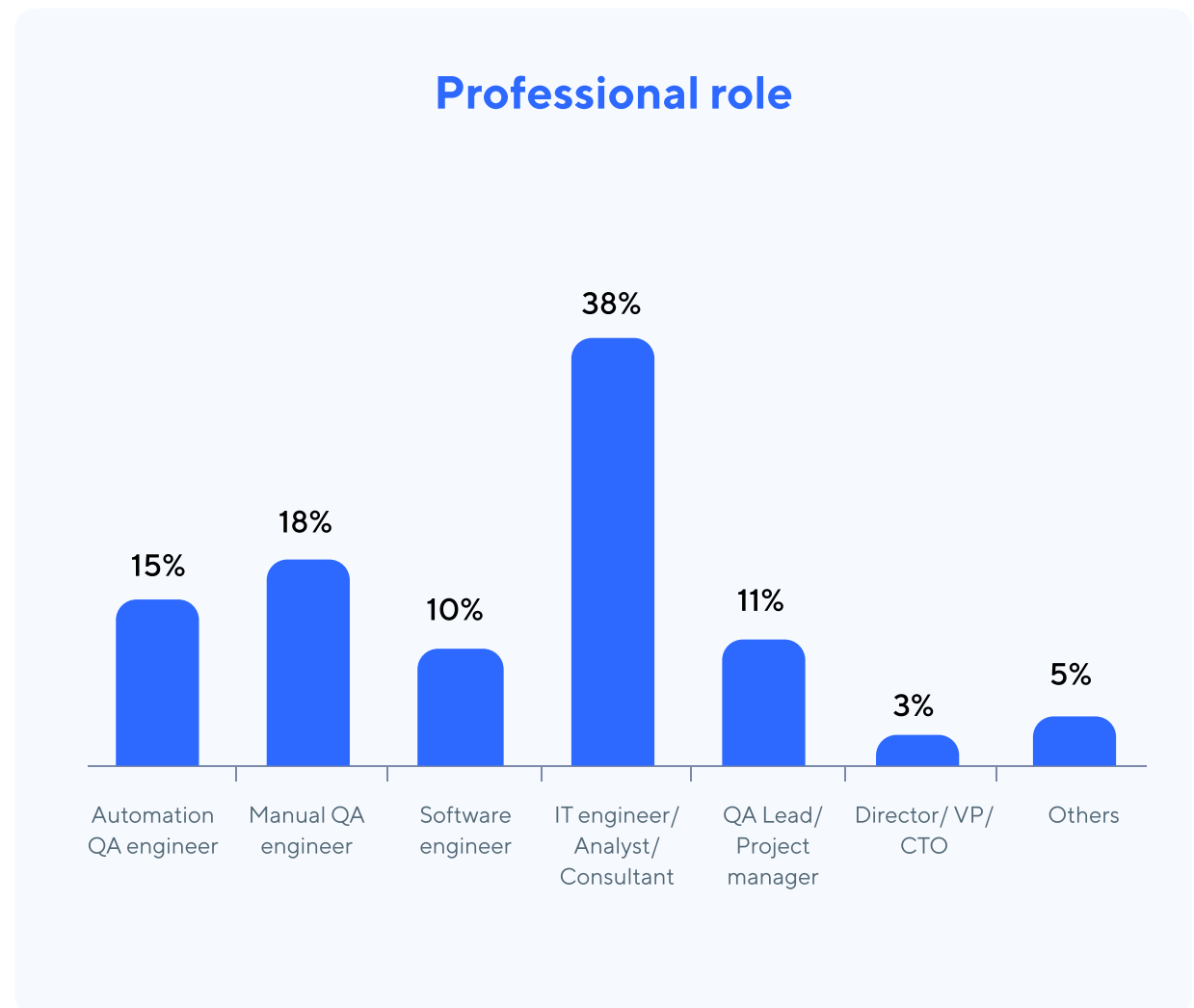
Although we conducted the study by following a thorough process of designing the questionnaire, choosing audiences, and analyzing the results, we do not claim it to be fully scientific. All online surveys reflect the opinions of audiences in online communities rather than the overall population of software development and testing communities.



Demographics

Of over 3,000 respondents who answered our survey, QA engineers and managers make up the largest group with a total of 44%, including 18% engineers for manual testing, 15% engineers for automation testing, and the remaining 15% managers.

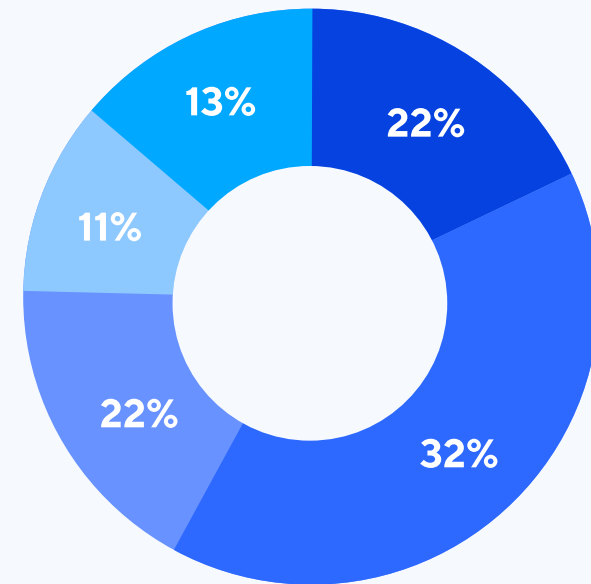
The second largest group consists of IT engineers, analysts, and consultants who were involved in various QA and development activities in their projects. The participants in this survey also include software engineers (10%) and a small group of senior managers such as directors, vice presidents, and CTO (3%).



As this survey is focused on team experience, it included only the respondents who worked with at least one teammate. As the results show, the majority of respondents worked in teams of 6 - 10 members (32%).

Twenty-two percent of respondents belonged to teams of 2 - 5 members, the same as those belonging to teams of 11 - 20 members. The remaining (24%) of respondents worked in teams of 21 members or more.

Team size

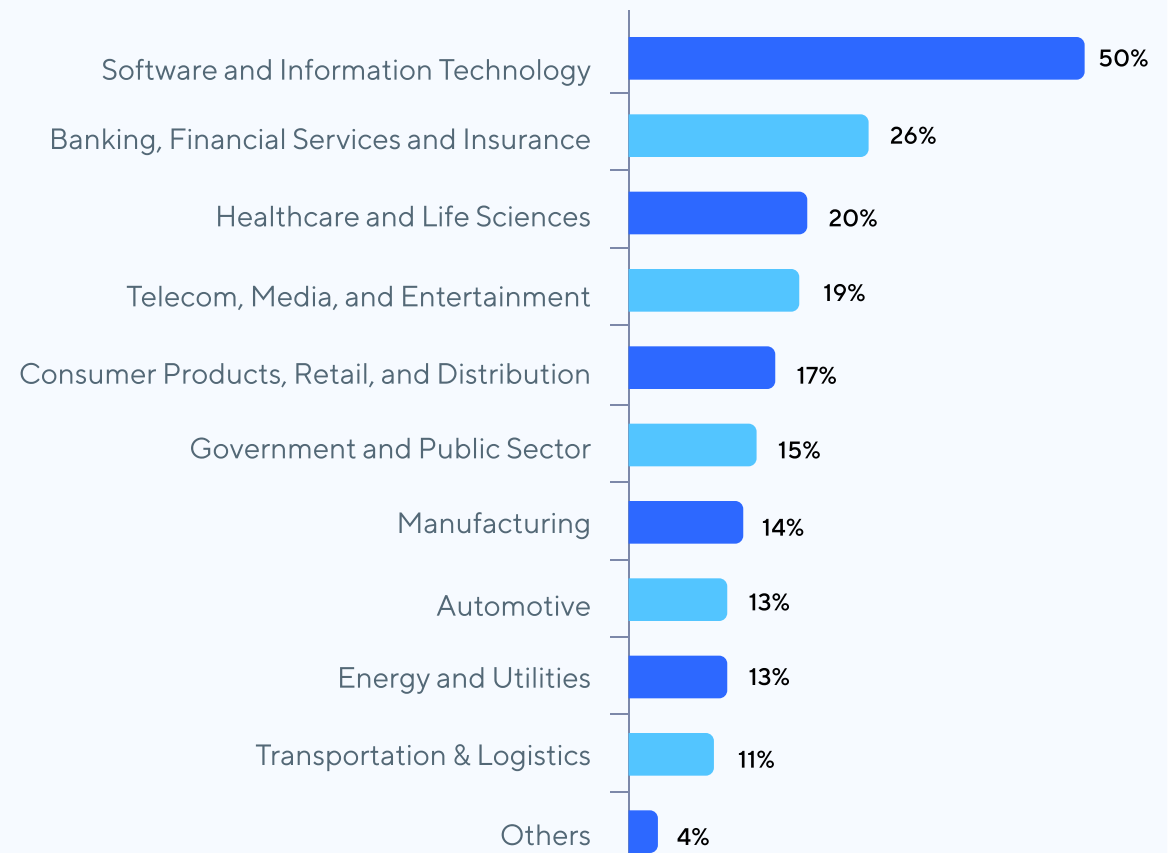


- 2-5 members
- 6-10 members
- 11-20 members
- 21-30 members
- More than 30 members

Half of the participants in this survey worked in the software and information technology sector. The data shows that the majority of them developed software applications for other sectors. A quarter of participants come from the Banking, Financial Services, and Insurance sector.

The survey also consists of participants from other industries: Healthcare and Life Sciences (20%); Telecom, Media, and Entertainment (19%); Consumer Products, Retail, and Distribution (17%); and Government and Public Sector (15%).

What are your company's industries?



Acknowledgements

Katalon Team would like to thank...

- **the 3,000+ respondents** for their time and contribution by taking part in the research study this year,
- **the testing communities** who have graciously helped our survey reached and informed more people,
- and our media partner, **Xilligence, Testbits, Nihilent, CircleCI, Trigent, BMP Technologies, Syone, QAlified** who has helped us distribute the survey to a wider audience.

The identity of the participants in the research study and their responses shall remain confidential.





Shape your automation testing future with us!

Katalon is the leading provider of software test automation solutions. The company offers a flexible platform for web, API, mobile, and desktop testing that fits teams and projects of any size, for any purpose from test creation, execution, reporting to seamless integration with popular CI/CD systems.

Widely used by a global network of 100K enterprises and 850K users across 160+ countries, Katalon is high-ranking on trustworthy review sites such as Gartner, Capterra, and IT Central Station. Katalon solutions include **Katalon Studio**, **Katalon TestOps**, **Katalon Recorder**, **Katalon TestCloud** and **Katalium**.

For more information, visit www.katalon.com.

General & Legal inquiries: info@katalon.com

License & Purchase inquiries: business@katalon.com

Partnership inquiries: partner@katalon.com

Our products:



Follow us:    