

HOW PWC SCALED TO 1,000+ AUTOMATED TESTS WHILE REDUCING MAINTENANCE BY 60%

Customer Overview

PwC is a global professional services organization delivering assurance, consulting, tax, and technology services to organizations around the world.

As its QA automation needs evolved, the team wanted to make automation more accessible beyond highly technical automation engineers. The goal was to reduce the knowledge gap between technical and non-technical QA professionals while increasing automation development speed, reducing maintenance effort, and expanding coverage.

PwC adopted testRigor's zero-code approach and cloud-managed infrastructure to simplify automation and enable more team members to contribute.

Today, the team has 1,003 automated test cases, test maintenance effort has been reduced by 60%, and automated tests that previously took around three days to make functional can now take at most 1.5 days

60%

Reduction in test maintenance

1,003

Automated test cases

1.5 Days

Maximum time to make a test functional

9

QA automation engineers

The Problem

PwC already had an established automation practice before adopting testRigor.

The challenge was not simply to introduce automation. It was to make automation easier to create, maintain, and scale across the QA team.

Traditional automation created a knowledge gap between technical automation engineers and manual testers. PwC wanted QA professionals without extensive development experience to participate in automation without requiring significant training in programming or specialized automation languages.

- At the same time, the team wanted to:
- Increase automation development speed
 - Reduce test maintenance effort
 - Expand automation coverage
 - Make onboarding easier for new team members
 - Reduce dependence on specialized technical knowledge

What PwC Wanted

Before adopting testRigor, PwC already had an established UI automation practice, with approximately 830 automated test cases. However, creating and maintaining automation required greater technical expertise, making it harder for manual testers and less technical QA professionals to contribute effectively.

The team wanted to reduce this knowledge gap while accelerating automation development, lowering maintenance effort, and expanding automation coverage.

This created several challenges.

High Technical Barrier to Automation

Automation required technical expertise, making it difficult for manual testers and less technical QA professionals to contribute without significant training.

Time-Consuming Test Development

Depending on complexity, it could take approximately three days to make an automated test functional.

High Test Maintenance Effort

Maintaining existing automation required significant effort, taking time away from developing new tests and expanding coverage.

Slower Onboarding Into Automation

New joiners and non-technical QA professionals faced a learning curve before they could effectively contribute to automation.

Accessible Automation

PwC wanted to enable manual testers and less technical QA professionals to contribute to automation without extensive coding knowledge.

Faster Test Development

The team wanted to reduce the time required to develop and make automated tests functional.

Lower Maintenance

PwC needed to reduce the effort spent maintaining existing automated tests as applications evolved.

Greater Automation Coverage

The team wanted to expand automation coverage without increasing technical complexity or dependency on specialized automation skills.

Key Objectives

- Reduce the technical barrier to test automation
- Accelerate automated test development
- Reduce test maintenance effort
- Increase automation coverage
- Enable more QA professionals to contribute to automation
- Improve overall QA productivity

“The learning curve is easy enough for anyone to follow and start working on automation sooner rather than later, despite the person not having great technical knowledge in development languages.”

PwC QA Team

The Solution

With testRigor, PwC created an accessible automation model where QA professionals with different technical backgrounds can contribute without mastering a complex framework.

PwC evaluated other solutions before selecting testRigor.

Two capabilities stood out:

- **Zero/low-code automation** made it possible for team members without deep programming expertise to participate in automation.
- **Cloud-based infrastructure managed by testRigor** reduced the infrastructure burden associated with maintaining the automation environment.

The accessibility of testRigor also became particularly valuable for new team members. They could begin creating automation while simultaneously learning the application's workflows and project infrastructure.

The team now has 1,003 automated UI tests out of 1,060 documented test cases, compared with 830 out of 900 before testRigor.

Test development time has decreased from approximately three days to at most 1.5 days, while maintenance effort has been reduced by around 60%. Integration with Azure DevOps and GitHub has also embedded automation into PwC’s existing CI/CD workflow.

Solution Highlights

- 60% reduction in test maintenance
- 1,003 automated test cases
- Test development has been reduced from approximately 3 days to at most 1.5 days
- Automated coverage increased from 830 to 1,003 test cases
- Documented UI coverage increased from 900 to 1,060 test cases
- Zero/low-code automation makes automation accessible across QA
- Cloud infrastructure managed by testRigor
- Integration with Azure DevOps and GitHub

Before testRigor

- 830 automated test cases
- Approximately 3 days to make a test functional
- Higher test maintenance effort
- Automation required greater technical specialization

During and After testRigor

- 1,003 automated test cases
- Up to 1.5 days to make a test functional
- 60% reduction in maintenance
- QA professionals with varying technical backgrounds can contribute to automation

The Result

PwC expanded its automated suite from **830** to **1,003** test cases, while reducing test development time from approximately three days to at most 1.5 days. Test maintenance effort also decreased by approximately 60%, enhancing productivity and delivering cost savings compared with the previous tool.

testRigor has made automation more accessible across the broader QA organization by reducing the technical barrier to creating and maintaining tests. New team members and QA professionals without extensive development experience can begin contributing sooner while learning the application.

Summary

Objective 🎯	Result 📊
Reduce Test Maintenance	60% reduction
Accelerate Test Development	~3 days → up to 1.5 days
Expand Automation	830 → 1,003 automated tests
Expand Test Coverage	900 → 1,060 documented UI tests
Democratize Automation	QA professionals without deep coding expertise can contribute
Improve Productivity	Enhanced QA productivity
Integrate Testing into Delive	ADO + GitHub integration
Simplify Infrastructure	Cloud infrastructure managed by testRigor

testRigor can also help you

“testRigor offers a great solution if you have a team with multiple manual testers you would like to embed into automation support, or even if you have a team with no automation test support whatsoever.”

PwC QA Team

By reducing the technical barriers to automation, testRigor helps manual testers and QA professionals build and maintain automated tests in plain English without deep programming expertise. Supported by documentation, guides, and direct Slack assistance, teams can expand participation and coverage, accelerate test development, reduce maintenance, and focus more on product quality. testRigor’s capabilities make automation simple across web, mobile, desktop, APIs, databases, AI features, and mainframe applications.