

HOW DISPRZ SLASHED THEIR TEST CREATION TIME BY ½ AND ACHIEVED NEAR-ZERO TEST AND INFRASTRUCTURE MAINTENANCE COSTS WITH THE CLICK OF A BUTTON.

Customer Overview

Disprz is an EdTech company driven to upskill the workforce to face the demands of the future. They offer a variety of products that leverage AI and cognitive science-based learning. Through the use of psychometric testing, skills gap analysis, and digital tools, Disprz is currently helping over 350 customers achieve breakthrough performance that embeds into their business and aligns with team goals.

Let us find out what they were able to accomplish with testRigor.

Near-zero Maintenance Costs

No more explicitly investing in test infrastructure and test maintenance

2X

Effort and time reduction compared to Selenium

Manual Testers can Automate in ½ the Time

Complex test cases can be automated in ½ day by in-house manual testers

The Problem – Disprz’s Struggles with QA

While Disprz has a robust QA team, there were a couple of concerns that needed attention.

- The first one was their infrastructure, which was a time and maintenance bottleneck that could be improved. They were managing their own infrastructure, including virtual machines, code repositories, and the setup of all their testing environments. This stole precious time from the team, time that could be better spent on creating and executing test cases.
 - Manual testing was a foundational block of Disprz’s testing strategy, and they were looking to increase their reach with automation. They had dedicated manual QA and automation engineers. While this was good, they needed a better solution to make their testing efficient and cost-effective.
 - The test automation that was being done was through Selenium, which is known to be unstable. Thus, every little change in the UI meant disruption in test runs, which in turn ended up devouring the time and effort of automation engineers.
- The above pain points led to increased maintenance costs. They realized that the time and effort being spent on maintaining infrastructure, revising automated tests and performing manual testing tasks could be better utilized if manual intervention were reduced in these activities.

“We didn’t want to reinvent the wheel; we wanted plug and play and to move on.”

Prem Anand Shanmugam
Head of QA at Disprz

What Disprz Wanted:

Less Maintenance

Disprz’s QA team was stuck maintaining test infrastructure along with manually testing their products.

Cost-effective and Stable Test Automation

While they had automation engineers on their team, they needed an easier way to upgrade their existing team, both manual and automation testers, to test automation without having to incur more costs. Moreover, the solution needed to be one that was easy to adopt by their in-house QA team.

The Solution – Why Disprz Opted for testRigor

Disprz realized that testRigor could offer them a simpler way to execute tests while managing testing infrastructure. Additionally, Disprz could also utilize their in-house resources like manual QA, to automate testing as testRigor does not require coding skills, just plain English to write test steps, which eliminates the need for explicit training or skilled engineers.

The key selling points for Disprz choosing testRigor as their automation solution were setting up testing infrastructure and executing tests at the click of a button. With testRigor, they were able to address three main issues in their QA processes:

1. One-stop Testing and Infrastructure Solution: With the click of a button, Disprz was able to execute their test suites without the cumbersome manual process they previously followed to set up infrastructure, move code, and test. testRigor’s all-inclusive test platform allowed Disprz to save time and effort in their overall QA processes.
1. In-house Testers Could Contribute to Automation: Manual QA could now automate and create complex automation with plain English. This allowed part of the manual QA team to participate in test automation creation for their applications. Furthermore, they were able to increase the speed at which manual testing was done when executed in parallel with testRigor – a best of both worlds situation.

Reduced Test Maintenance: With testRigor, test maintenance was close to zero when compared to their experience with maintaining their previous Selenium scripts. Like every modern application, theirs too was revised. But incorporating these UI changes required a lot of maintenance efforts with Selenium scripts. On the contrary, testRigor has built-in capabilities that leverage generative AI to gracefully handle these situations, providing stable and reliable test runs.

“testRigor’s ‘find and replace’ helps us. It doesn't let us have any loose ends.”

Prem Anand Shanmugam
Head of QA at Disprz

The Result

Disprz tries to help their customers in an efficient way – swiftly identify skill gaps, provide effective and suitable training, and thus, solve their problem. It’s only poetic that they decided to use testRigor, which also helps its customers approach QA in a simplistic and effective way.

Here’s how Disprz benefited from adopting testRigor:

- testRigor simplified Disprz’s QA processes by eliminating the manual setup of tests and infrastructure, saving time and effort.
- Disprz was able to decrease the test creation time by ½ with manual QA, without needing an engineer with technical knowledge.
- With testRigor’s plain english test cases, manual QA were now able to contribute in the test automation process.
- Before testRigor, a regression test took 2 days to execute. After switching to testRigor, the same regression test takes half the time as before.
- Another win for Disprz is being able to run the regression tests and branch testing in parallel.

testRigor has also reduced their test maintenance to almost zero with a click of the button.

Key Achievements

- Eliminated manual setup of tests and infrastructure
- Regression testing and branch testing are running in parallel
- Creating automated tests in ½ the time with Manual QA
- More test cases with automation
- Reduced test maintenance and loose ends

Before

- Complex test cases take a day to automate
- Infrastructure setup took a lot of valuable time
- Dedicated manual QA and automation engineer teams
- Explicit test setup was required

During and After

- Complex test cases can be automated in ½ day with manual testers
- Infrastructure and testing are as easy as a click of a button
- Manual QA can now create meaningful test automation.
- Tests executed with the click of a button

testRigor’s generative AI-based no-code platform makes it easy for QA teams to quickly automate testing while spending almost no time maintaining tests. Tests are in plain English and empower everyone, be it with or without technical knowledge, to easily build and maintain tests as well as understand test coverage.

“Our initial reaction to testRigor was Wow! You write code and hit start, and it starts.”

Prem Anand Shanmugam
Head of QA at Disprz