Tricentis

How artificial intelligence adds real value to software test automation

INTRODUCTION

Innovation never sleeps. Even the most groundbreaking technologies will continue to evolve substantially beyond their introduction. Once we see what an invention can accomplish, it ignites our imagination and inspires us to push technical boundaries even further—leading to advances that were previously unfathomable.

Consider air travel. The world changed in 1903 when the Wright brothers took flight. Suddenly, a whole new world of opportunity opened up and continued to evolve at an amazing pace. In just another 5 short years, the first passenger flight was achieved. 50 years later, jet passenger planes were introduced—fundamentally shrinking our world by enabling us to travel vast distances in hours. Still, innovation did not stop. In just another 16 short years, the Concorde supersonic jet took flight—not only breaking the sound barrier but also breaking through past notions of what was possible in terms of speed (flying from New York to London in just 3.5 hours). New supersonic and low orbit jets are being trialed now and future air travel will certainly be very different than what we can imagine today, even in our wildest dreams.

THE FIRST ERA OF TEST AUTOMATION

The first era of software test automation was also extraordinary in its time, but it was just the first step in terms of what is possible today. In the 1970s, script-based testing tools allowed you to program the automation of the then state-of-the-art terminal systems running on mainframes. Then, with the shift to microcomputers and early desktop GUI interfaces, record and replay tools emerged—allowing tests to be created much more rapidly, without needing to code every line. The 1990s brought open-source tools and community-driven frameworks—lowering the cost of entry to automation and giving users the ability to automate more than ever before.

Each of these iterative steps, building on the past, evolved the test automation process to relieve humans from a considerable amount of mind-numbing, repetitive clicking and checking that didn't actually require human intelligence. But this relief came at a cost: it also opened the door to new types of drudgery. Automated tests became a chore to create and proved extremely brittle. They failed frequently, and each so-called "false positive" required someone to review the result, confirm that it was indeed the symptom of a faulty test, fix the test, then repeat that entire process until the test executed as expected.

Granted, it was less mind-numbing than the repetitive checking that these early test automation tools replaced, but doing it well required specialized skills including scripting/programming expertise. Testers started to reach a point where they spent all their time maintaining existing tests—not writing any new tests—and the testing efforts became parallel development efforts. Given all the extra work that it required, the automation did not ultimately free up much more time for software testers to dedicate to actual software testing (the creative, analytical work, not mechanical checking).

Moving towards automation was undeniably the right idea, but we had to address the frustration that this previous era of test automation tooling created in order for test automation to reach more of its full potential.

THE SECOND ERA OF TEST AUTOMATION: MODEL-BASED TEST AUTOMATION

In the early 2000s, a new type of testing emerged: model-based test automation (MBTA). MBTA helps testers devote more time to the essential elements of software testing: the creative, highly analytical work that automation can't replace and risk-averse businesses can't live without.

By providing a higher level of abstraction, MBTA enables testers to work with Lego-like "building blocks" that can be combined and reused to create tests. If the application changes (e.g., a field is added or removed), just update the appropriate model, and the change is automatically propagated to all impacted tests. With a clear separation between automation details (e.g., "steering"), test logic, and test data, MBTA limits the impact of each change – drastically reducing the test maintenance required over the previous generation of test automation approaches. The test logic and test data are injected into the automation model at runtime—guaranteeing that tests never use old versions of test data or access outdated technical definitions.

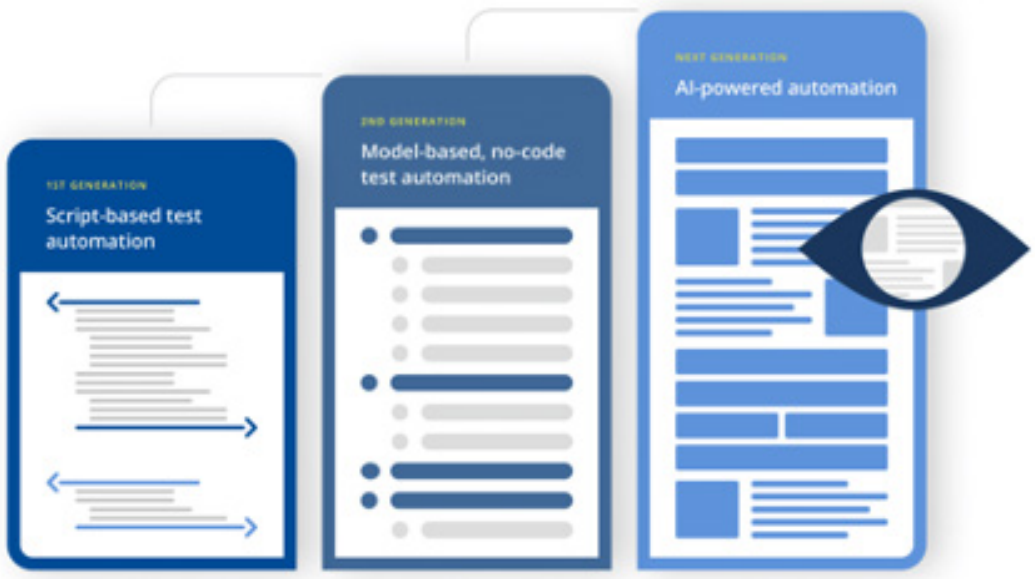
Non-programmers can rapidly create and manage sophisticated end-to-end tests, without waiting on "technical" resources. The separation of test steering, test logic, and test data allows each of these domains to focus on just that area, keeping the overall complexity as low as possible. As a result, tests are easier to understand are easier to maintain if/when needed. Moreover, false positives are substantially reduced because the MBTA approach allows the creation and management of test elements in a way that avoids the previous era's pitfalls.

THE THIRD ERA OF TEST AUTOMATION: AI-DRIVEN TEST AUTOMATION

While model-based test automation has proven instrumental in helping leading organizations transform their testing processes, there are some use cases beyond the power of any test automation technology that ultimately operates at the technical level. For example:

- Testing needs to shift left, but UI test automation requires a completed (and stable) UI
- Extremely new/old/specialized technologies are unsupported or require extensive customization
- Virtual and remote applications are beyond reach
- Even highly resilient tests require occasional attention and updating
- App modernization = test destabilization

Enter the next generation of automation: AI-powered automation that can see and use the UI like a human would.



Human behavior is simulated using various AI and machine learning strategies — for example, deep convolutional neural networks combined with advanced heuristics — to deliver stable, self-healing, platform-agnostic UI automation.

From the tester perspective, you provide a natural language description of what actions to perform, and the engine translates that to the appropriate UI interactions. UI elements are identified based on their appearance rather than their technical properties. If some UI element is redesigned or the entire application is re-implemented using a new technology, it doesn't matter at all. Like a human, the automation will simply figure it out and adapt.

Essentially, if you can see it, this new approach can automate it. This includes anything from an app using now-deprecated technologies, to an app using emerging technologies, to apps you access remotely. You can even start building test automation from mockups or whiteboard drawings. This brings a new meaning to test-driven development.

Here are just a few of the ways this AI-driven approach adds real value to your software testing practice:

You can build automation before a UI exists.

The AI can take a simple definition — like a textual requirement definition, a mockup, or even a whiteboard drawing — and construct a running automation case based off of it. With this "shift left," in-sprint UI test automation is finally within reach.

Your tests can withstand app modernization

When you upgrade your application, this AI-based automation doesn't break. This goes far beyond minor upgrades, like a change to technical IDs. We're talking about major upgrades like moving from one technology to another or upgrading your JavaScript library from an old jQuery UI to a new Angular Material Design.

Citrix? Customizations? Old tech? new tech? No problem.

If you can see it, this AI-driven approach can automate it. It works on any visual interface. Since it's using the visual interface of the technology connection, it can work through things like Citrix and it can work on Remote Desktop. It can work when you're not even connected to the machine that you're looking at (for example, you're viewing it through an RDP interface). It works on interfaces using technology that's not available anymore — things like outdated/deprecated versions of Gupta, Silverlight, and Flash (we know they're not fashionable, but they're still out in the world today). It also works with very new technologies that aren't yet supported by (most) automation tools: for example, Flutter, Blazor, and Electron.

It's so easy, your grandmother could do it

With this AI-driven approach, test automation is really, REALLY easy. In fact, one of our guiding principles for the project was "So easy, your grandmother could do it." Since the technology sees interfaces the same way that humans do, you can define the automation just like you would describe it to another human.

CLOSING THOUGHTS

This new era of AI-driven test automation does not replace testers. It never intended to. It elevates the role of the tester by removing the need to consider the mechanics of the technical automation–enabling the tester to focus on the challenging analytical and investigative work that attracted them to the profession in the first place. It also helps the business reduce risks while accelerating application delivery. Ultimately, it's all about bringing vital information to light so teams can rapidly release amazing products that grow the business.

ABOUT TRICENTIS

Tricentis is the global leader in enterprise continuous testing, widely credited for reinventing software testing and delivery for DevOps and agile environments. The Tricentis AI-based, continuous testing platform provides automated testing and real-time business risk insight across your DevOps pipeline. This enables enterprises to accelerate their digital transformation by dramatically increasing software release speed, reducing costs, and improving software quality. Tricentis has been widely recognized as the leader by all major industry analysts, including being named the leader in Gartner's Magic Quadrant five years in a row. Tricentis has more than 1,800 customers, including the largest brands in the world, such as Accenture, Coca-Cola, Nationwide Insurance, Allianz, Telstra, Dolby, RBS, and Zappos.

To learn more, visit www.tricentis.com or follow us on LinkedIn, Twitter, and Facebook.

AMERICAS

2570 W El Camino Real,
Suite 540
Mountain View, CA 94040
Unites States of America
office@tricentis.com
+1-650-383-8329

EMEA

Leonard-Bernstein-Straße 10
1220 Vienna
Austria
office@tricentis.com
+43 1 263 24 09 – 0

APAC

2-12 Foveaux Street
Surry Hills NSW 2010,
Australia
frontdesk.apac@tricentis.com
+61 2 8458 0766

v. 0621