

Automating Innovation

The rise and impact of automated testing tools and technology

Christopher Eyhorn
Executive Vice President
Testing Division, Telerik

The year was 1962 and the world was catching its first glimpse of the daring and colorful wonders that tomorrow might bring. It was a space-age vision of a technology-filled

future—flying cars that could fold down into an attaché case, wisecracking robotic maids, and miraculous machines that could do anything from washing the dog to cooking a three-course gourmet meal, all at the touch of a button. It was a utopian society made cleaner, better, and more prosperous through the auspicious use of technology and automation.

This description, of course, refers to *The Jetsons*, that iconic animated family, with their pneumatic people-mover tubes and moving sidewalks. Technology and automation were a key factor in improving the lives of the denizens of Orbit City, where George Jetson lived and worked as a full-time engineer...full-time being a whopping nine hours per week, thanks to a multitude of labor- and time-saving devices that were an everyday part of life for George and his family. Although our modern society hasn't reached the same utopian apex as Orbit City, there are disruptive shifts in technology automation that have changed the face of our world today and continue to have an indelible impact on businesses and consumers alike.

Take for example, the software development landscape. Facing a challenging – and frequently unruly – economic

climate, individual software developers and project teams are often tasked with doing more yet having less to do it with. In Agile environments, that need is even more pressing; minimizing waste and risk, while maximizing productivity and velocity are essential for sustained success. How to achieve better results with fewer resources might seem an impossible dilemma, however, for those willing to take a chance on today's innovative, emerging technologies, there are indeed solutions that make sound technical and financial sense. And one such solution is that of automated testing tools.

Putting Agility to the Test

B. Bereza-Jarocinski of the Global Association for Software Quality once said that “introducing test automation is sometimes like a romance: stormy, emotional, and resulting in either a spectacular flop or a spectacular success.” The metaphor is a fitting one – it is true that interleaving automated tools within the test environment can be a tumultuous process. Introducing test automation is an event that can bring with it tangible benefits and the opportunity for achieving greater success. But it can also result in

subject failure, thwarting project goals, frustrating teams, and dragging down ROI.

Scoring a favorable outcome when adding automated technologies to any testing environment depends on untold numbers of variables – testers' willingness to try and acceptance of test automation solutions, team composition, organizational priorities, budget and schedule constraints, and the quality of the tools being introduced. However, since their debut in the 1980s, automated testing tools have evolved rapidly, migrating swiftly from their core functional testing capabilities into sleek, sophisticated, and intelligent integrated suites that raise testing accuracy and team productivity. Today's cutting-edge test automation technologies have resulted in a generation of robust products that make automating tests faster, easier, and more cost-effective than at any time ever before.

With the ubiquitous adoption of Agile development methodologies, any enabling technology that permits teams to produce high-quality deliverables on-time and on-budget is a valuable resource. In the results-oriented Agile environment, the benefits that test automation offers cannot be understated:

- **Accelerated development cycles** – Short sprints require features to be fully developed, tested, and debugged in mere weeks, if not days. With this increased emphasis on Agile, the need for automated testing has risen, as well. Unlike manual practices that can result in lengthy, drawn-out test cycles, automated testing approaches allow scenarios to be performed swiftly and repeatedly without the need for manual intervention. By optimizing test processes for the utmost efficiency and speed, teams can successfully achieve and sustain durable velocity levels. Lastly, test automation introduces greater scalability into the test environment. In allowing tests to be run simultaneously rather than serially across any number of machines within the testing environment, automation shortens total execution times.
- **Improved resource utilization and productivity** – In contrast to earlier generations of testing tools that were cumbersome, overly complex, and had steep learning curves, current test automation technologies

are an evolutionary leap forward. Intuitive, intelligent, and easy-to-use, today's automated testing toolsets improve the allocation and use of limited project resources. For example, with a wider array of options, functionality, and capabilities, innovations within test automation are enabling non-technical personnel to function effectively within the test environment. In turn, developers can be relieved of some testing responsibilities, permitting them to instead focus on other critical work at hand, elevating overall project productivity.

- **Enhanced team dynamics** – As McDonald's founder Ray Kroc once noted, "None of us is as good as all of us." Collaboration is a keystone of Agile; without a tightly integrated team pulling together towards a common goal, velocity decelerates, impeding success. Agile teams often employ conventional testing practices, which can yield less than optimum results. Relying on such traditional methods can silo developers and QA personnel by putting them at odds, stifling collaboration, and blinkering project velocity. Introducing automation allows teams to rethink their testing approach and reset perceptions of how and where testing fits into the development process. By leveraging automated testing tools, silo walls can be removed, improving team collaboration and dynamics. The outcome is high-powered performance delivered by teams better functioning as cohesive, tightly integrated, organic units.
- With Agile as the new reality for many developers, project teams, and organizations, finding conduits to greater productivity and sustainable velocity is mission-critical. By availing themselves of the out-of-the-box benefits and value that test automation offers, testers and developers alike will be well prepared to meet the challenge of achieving their objectives faster, easier, and with fewer resources.

A Look Behind the Curtain

Having examined the benefits that test automation offers, it might be helpful to take an in-depth look at how test practitioners can put automated testing tools to use in real-world situations. No two teams are ever exactly alike – what works for one may spell trouble for another. How-

ever, with a better understanding of test automation mechanics, it becomes possible to see how it can successfully be applied in any development environment.

Properly identifying a team's goals for using automation is an important first step when starting a new project and is especially important for teams resurrecting their automation efforts for existing projects.

General considerations teams should remember to take into account:

- Get started with the automation effort as early as possible.
- Remember that it is critically important to have clear direction from leadership about how and what to automate.
- Don't fall into the tempting trap of trying to write automation framework internally, as this attempt will become just another development project competing for limited resources.
- Automate end-to-end scenarios and not just middle level scenarios.
- A test is never finished until it successfully runs in the execution environment.
- At the end of the day, automation technology can only take a team or a project so far; the overall attitude toward quality is equally important.

Key qualities to look for in a good automation tool:

- It should help everyone take ownership of quality by serving multiple roles, and allowing all stakeholders for the project to construct tests and understand their purpose.
- The ability to support a test-first approach by using a hybrid of automated and manual steps. Teams should leverage the power of automation but with manual steps for user interface that isn't finished or is changing. Doing this allows teams to build more automation faster.
- Easy integration with the same source control system that the project code base uses. This eliminates the barriers that fragmented systems create when trying to debug tests and resolve bugs in code.

- Facilitation and fostering of greater collaboration – collaboration is of the utmost importance for successful automation efforts on any Agile project.

What to automate and things to remember when introducing automation:

- User stories and dependency-free requirements. Start with user stories or requirements that do not have dependencies that can break automated tests. Before automating a scenario, teams should always analyze areas of the project that are interconnected and still under development in order to minimize the fragility of a given test.
- Important business scenarios. Always focus on testing those scenarios that make the tool valuable to the business. For example, if the organization is an airline, the first test the team should consider automating is ticket booking.
- Complexity vs. Regression value. The desire for a manual tester to find a bug often leads them to try the most obscure, random, and crazy scenarios possible. In automated testing, this is suboptimal – once this obscure bug is found, an automated test for that scenario is essentially worthless and has zero regression value. Very complex tests should always be weighed against the regression value they will bring to the project.
- Write the happy path test first and then expand to complex or edge scenarios.

“Testing is Human, Quality is Divine”

While this saying might seem to run counter to the concept of automating testing, the opposite is actually true – any tool is only as good as the person using it. Stated plainly by Aristotle, “Quality is not an act, it is a habit.” Applied to the testing arena, this concept means that no matter how automated the environment, the human touch will always be essential to achieving meaningful, measureable success. And indeed, that human element might be the most important one of all.

Once the sole domain of software developers and those experienced testers with the technical proficiency for building and executing test scenarios, the ongoing evolu-

tion and maturation of automated testing tools is reshaping the erstwhile software testing paradigm. With new test automation tools, less skilled and even non-technical test personnel are realizing productivity levels rivaling those of traditional professional testers. The suggestion that non-technical personnel can be a valuable addition to the testing discipline will likely bring cries of protest from those testers who have invested years honing their craft, yet it is an idea that should not be rejected out of hand. It should also be noted that having non-technical testers is not a novel concept, as they are commonly integral members of manual testing teams. What is new however, is the idea that they can become vital players in automated testing settings.

With the rise of intelligent tooling backed by intuitive, point-and-click interfaces, the learning curve for automated testing tools is negligible. By eliminating barriers to entry and adding simpler-to-use elements such as conditional logic support, these sleek, easier-to-use, and more powerful tools allow non-technical testers to be swiftly added to the test mix. They can now quickly learn how to efficiently and successfully write reusable automated tests without the assistance of developers or technical testers. This provides a richer project resource pool that teams can call upon to attain better overall productivity, accelerate the testing cycle, and realize greater added value and ROI.

There are other benefits that are less obvious, as well, and these can have a profound impact within Agile environments. For example, they bring with them a broader perspective uncolored by previous assumptions – they may try to use the test subject in ways developers had never intended, uncovering hidden bugs and enabling them to capture scenarios that might not otherwise have been tried. Furthermore, this lack of supposition can also pro-

vide insight into the end-user mindset, allowing for better refinement of features, functionality, and UI.

While the benefits of non-technical testers are manifold, there are considerations that must be taken into account. Before dashing over, grabbing Sally from Accounting, and ushering her into the test lab, teams should consider what traits a good non-technical tester should possess. Non-technical testers should be curious, willing to ask questions that others (like business analysts) might not, and have a desire to learn more than just the functional aspects of testing. They must be willing to take on a role as a cheerleader for product quality and not be easily discouraged. In turn, teams must rethink their opinion and treatment of non-technical testers – they should be seen as valued team members, rather than adversaries or interlopers because despite the continued advance of automated testing tools, all testers will continue to be a lynchpin for success in the software development landscape.

Improving the World One Bug at a Time

Test automation has come far since its first appearance in the software development landscape. New automated tools are elevating the profile of the testing discipline, giving it added relevance, importance, and value, particularly in Agile environments. While automated testing tools may not be as impressive as creating the world's first flying-suitcase car or usher in the age of the nine-hour workweek, they are truly essential in providing better, faster, richer, and more rewarding experiences for testers, developers, and ultimately, the end-users they serve. In the end, the truth of the matter is that test automation tools are the channel that will allow developers and teams to improve the world...one bug at a time.

Christopher Eyhorn
Executive Vice President
Testing Division, Telerik