

A helping hand: using robots as testing resources



Software testing is a fundamental element for the success of software releases. Without thoroughly tested code, entire IT infrastructures and software projects can easily fall apart due to bugs and errors. Many organizations, however, still rely on manual testing routines or complex test automation solutions that are time-consuming to configure. In addition, available time and resources are often scarce. Consequently, this can result in poor test coverage and malfunctioning releases that have to be reversed or repaired under high pressure. In order to ease this burden, free up resources and further improve software quality, Robotic Test Automation (RTA) can be brought into play. This whitepaper sheds light on the advantages and disadvantages of different approaches of test automation¹, and how software robots can serve as helping hands to accelerate one's software testing capability.

A hybrid approach

Over the past two decades, software testing has gained increasing attention and evolved from being a mere debugging method to an all-encompassing approach of quality assurance. Software testing can be defined as the process in which overall software quality is examined through validation and verification, and is used to inspect potential risks before software can be released. In order to make rational decisions with regard to the use of automation in software testing, organizations first need to fully comprehend the benefits and limitations of the various methods used.

In manual testing, features such as flexibility and adaptability are highly valued. Projects that involve a high rate of exploratory testing or ad hoc testing are therefore well-suited for manual test cases. However, due to high labor costs and the increased likelihood of human error, manual testing is less appropriate for larger projects (i.e. implementation or migration projects) or repetitive CI/CD testing tasks.

Automated software testing, on the other hand, allows organizations to achieve higher efficiency and greater test coverage. Unlike manual testing, automated software testing tools are designed to execute repetitive and high-volume test cases. Nevertheless, due to test maintenance, which increases inflexibility, as well as high initial costs and extensive employee

training, automated software testing is a rather costly and inflexible solution for (smaller) projects with ever-changing requirements and conditions.

To mitigate the drawbacks of implementing automated solutions, a 'technological and social movement', also known as low-code (Gartner, 2021), has emerged. *Low-code technologies*, such as *Robotic Process Automation (RPA)* and *Low-Code BPM*, typically have less time-consuming implementations, lower initial costs compared to traditional IT solutions, and support citizen development. RPA, in particular, allows organizations to deploy software agents (hereafter 'robots') generally within four to six weeks, pay per license, and eliminates the need for extensive employee training. Similarly, robotics can help organizations overcome challenges that regular automated testing software is not capable of.

What is Robotic Process Automation (RPA)?

RPA is considered to be a software solution which is capable of executing repetitive, standardized and rule-based actions across applications, that otherwise would have been done manually. In the last couple of years **Gartner** has proclaimed RPA to be the fastest growing type of business software. Scan the QR code for the complete introduction to RPA.



¹ Given the focus of this paper, testing approaches are described in an abstract and simplified manner.

The helping hand

Robotics within software testing, also known as Robotic Test Automation (RTA), is a combination of RPA and software testing. With RTA, organizations can automate and centralize testing projects to ensure the quality of applications before they go live, without the workload of manual testing or the development of complex automated testing scripts. It is built around the same philosophy that applies to RPA; it uses robots to carry out processes and operations.

This form of automation is best suited for high-volume, rule-based processes where human actions can be easily replicated. Like humans, robots work

with *Graphical User Interfaces (GUI)* to work their way through applications, but they are also capable of handling *Application Programming Interface (API)* connections, allowing organizations to cover both front- and back-end focused test cases. Although humans and robots have much in common, robots exhibit superior performance when it comes to consistent and accurate task execution. Because robots are taught to exhibit certain behaviors, they do not deviate from their given instructions unless told otherwise. Especially within DevOps and software testing teams, these characteristics, among other factors, are considered extremely valuable.



Reduce burden on IT/ Test capacity

One of the many reasons to implement automated testing software is to ease the burden on IT teams. Software quality cannot be guaranteed when testers are unable to cover most of the code. RTA provides organizations with the tools to create digital employees who are highly efficient, work seven days a week, and can act *autonomously*. Thanks to its model-driven tool design, characterized by *modularity* and *reusability*, RTA is easily scalable². All things considered, RTA supports testing teams in their efforts to reduce errors due to time constraints and improve overall software quality.

RTA provides testers with a low-code testing platform that allows them to create test scripts in a simplified manner, without overly complicated manuals or extensive training. Accordingly, emulating the user experience is made possible, as both testers and other internal stakeholders (i.e. business users) can now be leveraged to create test scripts, also known as citizen development. By doing so, some of the developer bias is taken out of the equation, as business users can help developers write automated test scripts. Hence, the overall test quality improves.



Simplified testing



Reliable and better software for end users

As mentioned, a robot is able to run unattended and put at use 24/7. Consequently, smoke tests can be run each night after code development, allowing for immediate feedback the next day. Consequently, software testing is fully integrated into an organization's CI/CD cycle, thus supporting continuous innovation. Furthermore, through integral testing of web, desktop and third-party applications, both front- and back-end, RTA proves to be a viable solution to increase the overall reliability of (business) software.

² A synergistic effect is created when organizations already use RPA to automatize their processes, as reusable components can easily be exchanged.

Due to the ongoing scarcity of human resources, organizations are often unable to guarantee the quality of their software, with all the consequences that entails. RTA supports maintaining an acceptable level of test coverage and accelerates the shift of the workforce towards more complex and value-adding tasks. Moreover, RTA can offer organizations greater accuracy

and consistency than manual testing, while also being more flexible and user-friendly than regular automated testing solutions. These benefits, combined with lower initial costs and faster implementation thanks to the Low-Code platform, make RTA a viable solution for organizations that are focused on continuous innovation and, hence, staying ahead of the curve.

Figure 1:
The spectrum of test automation³

Level of automation (from low to high)	Benefits	Limitations
Manual Testing	- Flexibility - Emulation of user experience - Complex test cases - Cost-effective within small projects	- Lower accuracy and consistency, compared to automation - High labor costs in larger projects - Relatively low test coverage
Robotic Test Automation	- Flexibility/reusability - High accuracy and consistency - Unattended execution - High test coverage - Low initial costs - Citizen development - Effortless integration with existing RPA infrastructure	- Tool selection - Test maintenance
Automated software testing	- Reusability - High accuracy and consistency - Unattended execution - High test coverage	- More extensive training required - High initial costs - Tool selection - Test maintenance

³ Simplified representation of automation levels within the field of software testing.

More information

Do you want to know what KPMG can do for your organization? Or would you like more information on how to implement Robotic Test Automation? Then please contact one of our specialists or leave your email address on the [Digital Process Excellence website](#) to stay informed about the upcoming client use case describing the successful implementation of RTA.

Contact

Dimitri Sloof
Partner
Digital Process Excellence
sloof.dimitri@kpmg.nl

Tjeerd Schmit Jongbloed
Senior Consultant
Digital Process Excellence
schmitjongbloed.tjeerd@kpmg.nl

Cas Weltevreden
Consultant
Digital Process Excellence
weltevreden.cas@kpmg.nl

KPMG.nl/DigitalProcessExcellence

home.kpmg/socialmedia



The information contained herein is of a general nature and is not intended to address the circumstances of any particular individual or entity. Although we endeavor to provide accurate and timely information, there can be no guarantee that such information is accurate as of the date it is received or that it will continue to be accurate in the future. No one should act on such information without appropriate professional advice after a thorough examination of the particular situation.

© 2022 KPMG Advisory N.V., a Dutch limited liability company and a member firm of the KPMG global organization of independent member firms affiliated with KPMG International Limited, a private English company limited by guarantee.
All rights reserved.