The benefits of **automated, risk-based testing**

A scriptless, no-code approach to test automation improves the efficiency and security of software development

John Phillips

Tricentis

Government agencies are embracing digital transformation as a way to boost speed and innovation while increasing efficiency. They are moving applications to the cloud, adopting more agile methods of software development, and embracing continuous delivery of modern front-end interfaces and back-end enterprise applications. A fast, flexible approach to software delivery makes application testing more critical — and more challenging — than ever.

In part, that's because many of the government's testing tools lag behind modern practices. Trying to fit such technologies into agile and DevSecOps toolchains increases risk exposure, drives up testing costs and delays the delivery of quality feedback.

Test case management involves governing access to vast amounts of test data and tracing changes to the micro level. However, doing that work successfully in an agile or DevSecOps environment requires automation.

Scanning for risks before and after deployment

Agencies must be able to quickly identify vulnerabilities and mitigate any risks in their applications. Adding static application security testing (SAST) and dynamic application security testing (DAST) to software development workflows can help.

SAST, also called white box testing, involves scanning an application for security vulnerabilities before the code is compiled. Those vulnerabilities include SQL injection, cryptographic failures, security misconfigurations and others in the Open Web Application Security Project's list of the top 10 security risks. DAST, also known as black box testing, is used to identify certain vulnerabilities while an application is running in a production environment.

By automating risk-based SAST or DAST, agencies can obtain coverage of more than 85% of their risks with only 20% of the effort. The approach allows agencies to de-risk their core business upgrades and updates, boost release readiness, and deliver better quality.

Increasing productivity and accuracy

The Tricentis Tosca platform is an automated, script-less, no-code approach to accelerating end-to-end testing, and it is driven by artificial intelligence. With Tosca, highly accurate testing can happen in minutes or hours instead of days or weeks.

By tapping our native support for more than 160 applications and technologies — including on-premises, clouds, mainframes, custom apps, packaged apps, APIs and microservices — an agency can achieve resilient, end-to-end test automation across any architecture or application stack. That automation covers the integrity of the data fed into an agency's system, the verification of report logic, and the accuracy of integrations, transformations and migrations.

Ren Ran

"**A fast, flexible approach to software delivery** makes application testing more critical — and more challenging — than ever."

Tosca offers a centralized area to design and update tests, and anyone — even nontechnical users — can easily create and maintain test automation without programming or scripting. This unique approach allows an agency to reduce costs and effort while achieving resilient test automation for any use case and even the most complex apps.

In addition, the approach boosts collaboration within software delivery teams. Those teams can increase productivity by identifying and sharing reusable test assets and can comply with traceability requirements for automated and manual testing while gaining "single-pane-of-glass" visibility into testing of software and hardware.

Our company's fundamentally different approach to software testing accelerates DevSecOps, cloud migration and digital transformation as a whole. Tricentis helps the government track, organize and report on testing activities across entire agencies — no matter the testing strategy — so that high-quality software can be delivered at speed and scale. ■

John Phillips is vice president of public sector at Tricentis.