



# Qualitest Speeds Up Time-to-Market of Embedded Security Software with Custom Test Automation



## Challenge

Reduce Time-to-Market of new releases to keep pace with emerging security threats.

Software embedded in bespoke hardware was unapproachable.



## Solution

Implemented test automation on top of automatable test environment.

Integrated into CI/CD systems and created pipelines to boost efficiency.



## Results

Achieved shorter testing cycles through faster regression testing at lower costs.

Client now has a robust, reusable testing suite for future development.



## Client Overview

Our Client is an innovative Israeli security systems technology leader. Their systems cover multiple levels of security needs that include entry control and monitoring, using bespoke hardware and software integrated with control systems and tactical information systems.

## Five Goals Focused on Speed

The Client uses software embedded in a hardware system that is connected to multiple end points and sensors. Because the software is security-oriented, the Client had a crucial need to shorten time-to-market of new features as breaches emerge and are exploited. Defects increase false positives, locking user actions.

At the same time, the Client required the software to be tested automatically on the target hardware itself, instead of virtually.

The project had multiple goals:

- Automate existing tests to allow greater test effectiveness.
- Integrate the automated tests with CI/CD tools for fast delivery of updates and new features.
- Reduce TTM by shortening test cycles while maintaining coverage and quality.
- Lower test effectiveness erosion due to high manual repetition.
- Build and infrastructure that would allow reusable test materials.

**“Because the software is security-oriented, the Client had a crucial need to shorten time-to-market of new features as breaches emerge and are exploited. Defects increase false positives, locking user actions.”**

## An Innovative Solution in Multiple Stages

For the first stage, we built a test environment that could run automatic tests by simulating real-world actions, using a simple electronic board to run a Linux-based application. This allowed the testing tools to trigger tests and inject data into the relevant components, which had been previously unapproachable.

The next stage was to initiate two activities that would work together:

- Start implementation of the test plan, using Python on top of the setup test environment.
- Set up the CI/CD systems and build pipelines to integrate the implemented tests into them.

While tests and environments were being added, the last part of the puzzle was put in place in the form of a reporting system.



We designed the system to generate automatic reports for each execution cycle. The reports would determine Pass/Fail criteria for each test batch and for the complete cycle. This would allow for a Go/No-Go decision at each stage and let the development team review errors and dive in on problems found by the automated test cycle.

## Continuous, One-Click Visibility Is Key.

From then on, Qualitest and the Client marked the test effort and weekly priorities according to the development pressure and feature roadmap at weekly meetings and bi-weekly progress reviews. This enabled continuous visibility of automation scripts and delivery status, and we incorporated a monitoring tool that made up-to-date information accessible with one click. Our weekly progress reviews highlighted main achievements, issues, needs or blockers, as well as key tasks planned for the next stage.

The integration into the CI/CD systems allowed developers to trigger testing from the merge and build processes, while keeping the ability to trigger tests as part of a complete test cycle or manual trigger.

## Key Benefits

With the help of automated functionalities, the Client can now perform regression testing faster and lower the cost of changes before the main releases.

- The regression tests implemented are running in 50% of the time, and with zero human involvement.
- The regression pipeline is run automatically from the developer build command, lowering costs of change.
- Manual testers' work is now aimed only at new features, with little to no regression effort.
- The Client had clear visibility of the version status at "one click" distance.
- Developers have access to testing resources.

**“The regression tests implemented are running in 50% of the time, with zero human involvement.”**

# QUALITEST

## Connect with Us

[www.qualitestgroup.com](http://www.qualitestgroup.com)

<https://www.linkedin.com/company/qualitest>

<https://www.instagram.com/lifeatqualitest>

<https://twitter.com/Qualitest>

<https://www.facebook.com/Qualitestgroup>

