

# OPTIMIZING AND AUTOMATING COMPLEX WORKLOADS FOR GREATER IT EFFICIENCY

// CASE STUDY



HCL OneTest



HCL SOFTWARE

# TABLE OF CONTENTS

- 2 // Introduction
- 3 // Challenge
- 4 // Strategy
- 5 // Key Findings
- 6 // Solution
- 7 // Benefits
- 8 // HCL OneTest Highlights
- 9 // About This Case Study



## INTRODUCTION

The Workload Automation product enables more than 2,000 leading organizations around the world to gain complete visibility and control over attended and unattended workloads.

From a single point of control, it supports multiple platforms, provides advanced integration with enterprise applications like SAP, Oracle, and Salesforce, and offers more than 35 out-of-the-box plugins, including plugins for FTP and Hadoop activities.

Additionally, Workload Automation allows scheduling dependencies between disparate systems, and helps increase availability due to its true fault-tolerant architecture.



## CHALLENGE

In 2018, the Workload Automation test team began to prepare for the 9.5 release in March 2019.

This major release brought about a complete redesign of the primary user interfaces. Such radical change meant that none of the existing test automation assets were usable, and that the effort to re-engineer them was significant.

The re-factoring process for existing scripts (based on manual interactions with a browser) would have been very time consuming, and since the tests were coded, they were also brittle. Additionally, the text-based development environment was not very intuitive for the testers, who adopted the framework, and it was difficult to document going forward.

Therefore, a new solution had to be found.

The team evaluated updates to their existing Selenium Robot framework while also looking at Katalon and HCL OneTest, primarily focusing on effort, ease of use and maintainability as the success criteria.

**SUCH RADICAL CHANGE MEANT THAT NONE OF THE EXISTING TEST AUTOMATION ASSETS WERE USABLE, AND THE THAT THE EFFORT TO RE-ENGINEER THEM WAS SIGNIFICANT.**



## STRATEGY

The team was unhappy with its existing framework due to daily instabilities in execution, resulting in tests hanging or producing false positive outcomes and leading to significant manual effort post execution.

Also, the Selenium Robot framework was generally considered too brittle and a coded solution was not ideal for the team. Katalon was carefully considered due to its strong out of the box capabilities. However the team noticed several problems when creating scripts, while finding the maintenance effort to be high.

Support provided by the community and even third-party vendors, was not enough to satisfy the business-critical requirement for the automation.

Hence, HCL OneTest was selected for a pilot program due to its offering of a codeless solution with the added benefits of simple script maintenance and the ability to leverage API and database actions in test scripts.

**SUPPORT PROVIDED BY THE COMMUNITY AND EVEN THIRD-PARTY VENDORS, WAS NOT ENOUGH TO SATISFY THE BUSINESS-CRITICAL REQUIREMENT FOR THE AUTOMATION.**

## KEY FINDINGS

At the conclusion of the four-month pilot program, the following were the key findings:

- HCL OneTest proved to be a convincing solution, capable not only of overcoming the limits of the previous framework, but also adding significant value
- Natural language scripts are auto-generated from recording sessions, and can be augmented with additional steps in the IDE while the application is offline
- Recording interactions with the web page under test and associating each action with a navigable screenshot of the page and DOM became a natural and intuitive way of working code-free
- Intelligent object recognition during playback made scripts resilient to changes and easy to maintain
- The ability to re-use UI tests during performance testing leads to efficiencies in script creation and earlier results
- Tests can be built that combine the browser interactions with steps to cleanup data and validate data on the back-end systems providing richer and more robust workflows





## SOLUTION

The team quickly created a modular framework of atomic UI actions, which were chained into Compound Tests to form user workflows.

Each of these workflows validates a range of functional and non-functional outcomes including verification of the UI controls and values as well as verifying parameters that are only accessible via APIs and database queries.

The entire test set of 174 end-to-end workflows (comprised hundreds of test modules) is decoupled from the test data which drives much of the logic of the test. This meant some test maintenance tasks are as simple as editing a .CSV file.

Tests are executed as part of the build pipeline through Jenkins and execute against four different environments in parallel.

With one set of scripts, users can verify that the Workload Automation core application functions correctly on Windows and Linux, as well as, each of the four supported database configurations: Oracle, DB2, Informix and MSSQL.

**TESTS ARE EXECUTED AS PART OF THE BUILD PIPELINE THROUGH JENKINS AND EXECUTE AGAINST FOUR DIFFERENT ENVIRONMENTS IN PARALLEL.**





## BENEFITS

- Ported the entire Selenium test suite (which had taken several years to create)
- Created more than 600 test cases to create a stable library of basic operations
- Gained confidence that new easily maintainable tests can be built within hours (not days as before)
- Achieved a 35% increase in coverage with new tests and the ability to bridge front and back end components of the application
- Configured the entire test suite as a Jenkins pipeline activity, exercising all deployment configurations of the application
- Highlighted more than 70 defects during the development cycle of the Workload Automation product – approximately 5% of the total defects found
- Option to run containerized test engines simplifies CI/CD process

**THE BENEFITS WERE ACHIEVED IN LESS THAN FOUR MONTHS.**







## HCL OneTest HIGHLIGHTS

- Script maintenance effort conservatively calculated at 50% to 75% less than the other tools evaluated
- Test coverage increased by 35% due to the ease of use and depth/breadth of technology coverage
- Found 70 additional defects - 5% fewer for customers to find
- Re-use of test assets increased due to script modularity and compatibility of across the different test disciplines: UI, API and Performance
- “Beyond the Contract” support via HCL’s Client Advocacy Team accelerated the project and provided a direct channel to product developers and product managers





## ABOUT THIS CASE STUDY

HCL OneTest provides UI, API, and performance testing, as well as service virtualization and synthetic data fabrication to support testers throughout a project lifecycle. It features a scriptless, wizard-driven test authoring environment and support for more than 100 technologies and protocols.

HCL OneTest belongs to the DevSecOps product domain of HCL Software which is a division of HCL Technologies (HCL) that operates its primary software business. It develops, markets, sells and supports more than 20 product families in the areas of DevSecOps, Automation, Digital Solutions, Data Management, Marketing and Commerce, and Mainframes.

For more information, visit [hcltechsw.com/onetest](https://hcltechsw.com/onetest).

Copyright ©2020. All rights reserved. No materials from this case study can be duplicated, copied, republished or reused without the written permission from HCL Software. The information and insights contained in this case study reflect research and observations made by HCL Software.