



OPCITO TECHNOLOGIES

# QA Automation Using Pytest

## About The Customer

The client is a leading software provider for delivering enterprise applications in data centers and cloud. The client has services and products around load balancing, application analytics, and security for data centers and the cloud.

## Business Challenge

The client approached Opcito to convert their existing test suites, which were written in Robot Framework, into pytest. The main objective was to convert these test suites by utilizing the advantages from pytest, like parameterization and a wide range of plugins available with minimum effort.

## How Opcito Helped

The initial challenge was to design the test infrastructure that could reuse existing functionality/infrastructure written for Robot Framework to reduce efforts during the conversion. In Robot Framework, we usually have a separate test case for each scenario we want to automate. In Robot Framework test suites, we found that most test cases were testing the same functionality/scenario using different inputs. Opcito reduced the code for a number of test cases using `pytest.mark.parametrize` decorator, which enabled the parameterization of arguments for a test function.

Opcito leveraged pytest fixtures to replace the setup and teardown functionality by setting its scope to the module or using `setup()` and `teardown()` functions. After the yield statement in the fixture, all code serves as a teardown. Similarly, the test setup and teardown were replaced by setting the scope of the fixture to function.

We could have also used pytest fixtures as function arguments in this case. There was no need to change most of the functionality/infra that was written in Python. We generated the reports in HTML formats by using pytest HTML plugin.



This document is proprietary and confidential. No part of this document may be disclosed in any manner to a third party without the prior written consent of Opcito Technologies.

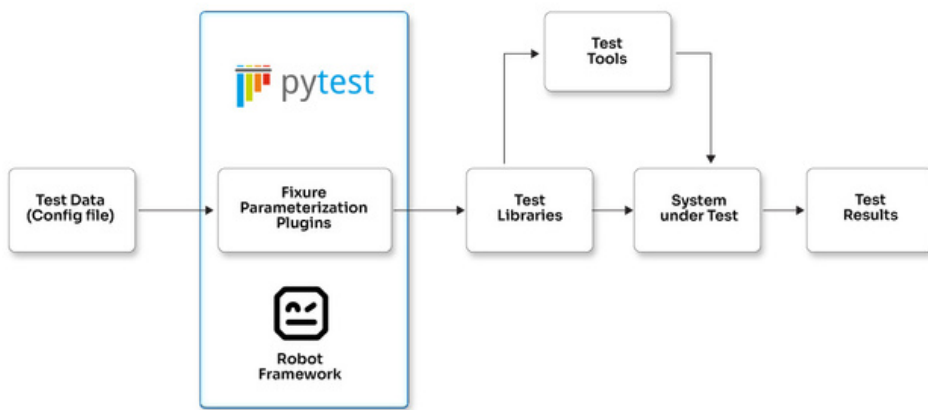
India office +91 (20) 6712 4100

US office +1 (650) 772 4442

## Technologies, Tools, and Platforms used

PYTEST

ROBOT



## Benefits

### MODULAR DESIGN

Modular fixtures make it easy to modify or add assertions because test cases are reduced in the pytest functions compared to Robot Framework. By taking advantage of `pytest.mark.parametrize` decorator, adding an extra scenario is just a matter of adding the additional input tuple

### CUSTOM ASSERTTS

Pytest allows the use of standard Python `assert` to verify expectations and values in Python tests. User can also custom assertion using the `pytest_assertrepr_compare` hook

### EXECUTION TIME

Execution time reduced drastically (35 to 40%) compared to the Robot Framework test suite

### PLUGINS

Available plugins and the provision to design own plugins

## About Opcito

At Opcito, we believe in designing transformational solutions for our customers, start-ups, and enterprises, with our ability to unify quality, reliability, and cost-effectiveness at any scale. Our core work culture focuses on adding material value to your products by leveraging best practices in DevOps, like continuous integration, continuous delivery, and automation, coupled with disruptive technologies like containers, serverless computing, and microservice-based architectures. We also believe in high standards for quality with a zero-bug policy and zero downtime deployment approach.