

Test Automation Success using Open Source tools

Client

Our client is a leading insurance and financial solution provider for schools, community colleges and health care organizations.

Requirements

Our client was looking for an independent software testing services provider who understood their domain very well. They wanted their vendor to take care of end to end testing of their applications that managed property and casualty including the compliance with the US state regulations.

Key Highlights:

Key Successes:

- Our domain knowledge
- Our test strategy and planning (risk analysis and mitigation)
- Our test automation expertise

Industry Vertical: Non-life insurance

Domain: Property & Casualty

Duration: 3 years

Team size: 5

Technology:

Developed using Java EE, Liferay Portal, JBoss and Oracle DB

Platforms: Windows platform

Automation Tools: Selenium, JMeter

Our Challenges

- We started in a multi-vendor environment where the development vendor had little domain knowledge.
- There were frequent change requests that were reported and managed within the ongoing sprints due to multiple stakeholders.
- To ensure that the development partner followed sprint plans for releasing builds to us for testing was a huge challenge.
- Lack of clear acceptance criteria for features such as user dashboards and reports that can predict risks and exposures caused ambiguity while resolving defects.

Our Approach:

- We ensured that agile development methodology with adherence to biweekly sprints were followed by product stakeholders and vendors.
- We requested documenting user stories and uploading in a tool that was accessible by all vendors before the start of the sprint. We ensured that the change requests were recorded along with the user stories.
- We advocated Test Driven Development as the other vendor team lacked domain knowledge. We created exhaustive test scenarios and test cases and reviewed with the development team during every sprint.
- To benchmark performance of the product, we published JMeter based load test reports for every sprint.
- We manage test cases using a Test Management tool.
- As the number of test cases was nearing a thousand, we motivated the client to approve test automation using Selenium.
- We have automated sanity test cases and about 90% of the regression test cases.

Metrics

- The page load times have been brought down from over 20 seconds to 3 - 4 seconds.
- Automating sanity testing and regression testing has brought about the below saving in execution time:

TYPE	# Test cases	Manual Execution time	Automation run
Sanity	120	3 man-hours	30 mins
Regression	1500	80 man-hours	5 hours
New	300	40 man-hours	Under development

Return on Investment

- We deployed a team of 3 automation engineers to automate the bulk of test cases for 3 months. We now have 1 test engineer maintaining and adding new test cases to the automation suite.
- Cost of initial development was about \$30,000 besides maintenance and new development is \$3500 per month.
- There are 5 – 6 releases in a year that requires us to run all regression test cases at least twice and we run the sanity test cases at least 3 times in a week.
- Since the script development we have saved 4 releases x 2 cycles x 75 hours (regression) and 3 times x 52 weeks x 2.5 hours (sanity) totalling 1000 man-hours in the last year.
- Our saving has been \$20,000 for less than a year. ROI has been > 60%.

Work with us!

Shivashankari | Head, Software Services

✉ shivashankari.n@focusite.com

☎ +1-630-243-4454

🌐 <https://in.linkedin.com/in/shivashankari>

🌐 <http://services.focusinfotech.com/>



DIGITAL BUSINESS
& TRANSFORMATION