DATABASE TESTING – SUCCESS STORY

Client

Our client is the world's largest enterprise software company with its position in the Fortune 100 companies. They offer a comprehensive and fully integrated stack of cloud applications, platform services, and engineered systems.

Requirement

Client was on a hunt for an independent QA vendor who has prior experience in working with Fortune 100 companies. Secondly, client was looking for vendor with proven database testing expertise and knowledge of Java scripting language to work on client's custom-built testing tool. Last but not the least, they wanted to replicate their QA process by setting up a dedicated offshore test lab and transfer it over a period of time to their captive center in India in a Build, Operate and Transfer (**BOT**) model.

Key Highlights

Key Success:
Our Automation Framework

Domain:
Hi-Tech

Duration:
8 years

QA Team:
27

Technology:
Developed using C, C++, and Java & Scripting Language: XML, Perl

Application Overview

It is command line application that facilitates online/ offline data replication across heterogeneous database systems.

It supports more than 6 databases currently (Oracle, Sybase, SQL, MySQL, Teradata, TimesTen) and Operating System (Windows, Linux, HP-UX, Solaris, IBM AIX). This application is predominantly used by banks.

Testing Challenges

The frequent change requests for the product due to agile software development, demanded continuous **regression testing** across all code branches. Some of the challenges were performing regression testing across various platforms, automating test case using custom automation tool, UI automation using Selenium, Build and release engineering.

Indium to benchmark the performance of database middleware application with SQL native replication, SQL server native replication, MySQL native replication, MSSQL server native replication.

Business Challenges

Cost of QA started to increase and became a huge overhead for the client. Company's reputation was at stake due to inconsistent behavior of their product across various platforms. Other challenge was to identify a trusted vendor to safeguard client's intellectual property.

Our Approach

Indium devised an approach to meet both the 'Testing' and 'Business' challenges faced by our client. Initial goal is to train our team on custom automation tool developed by the client and swiftly make them adapt to the client's testing process and methodologies.

To ensure the test coverage on our environment, which is similar to the client environment.

Build a team with thorough QA expertise in DB skills, OS level skills and Java scripting.

Suggest and implement the right test automation tool.

To Learn more about our Database Testing Expertise

Click Here

Our Solutions

The testing services we offered are listed below

- Functional Testing, Regression testing, UI automation using Selenium, & Performance benchmarking.
- Created a dedicated test lab.
- Ensured maximum test coverage across environments.
    - **Operating Systems**: Linux_x64/x86, Solaris_Sparc 9/10/11, Solaris_x86_64, AIX 5.1/5.2/6.1, HPUX_IA, HPUX_PARISC, Windows_x64/x86. Windows 2000 / 2003 / 2008 / 2008R2 /2012.
    - **Databases**: Oracle 9i/10g/11g/12c and SQL Server 2000 / 2005 / 2008 / 2008R2 / 2012 and DB2 91195197, Sybase 15/15.5, MySQL.
- Our team performed end-to-end regression regardless of the new functionality to ensure the product quality is not compromised.
- We suggested to implement test automation tool Selenium & ROBOT framework for UI and ATS and for Bug tracking / Defect management tool – Bugsmart / BugDB / JIRA. Indium's Customized Java based Macro (Generates Reports in Excel format) was used as the reporting Tool for Performance Benchmarking.
- We conducted performance benchmarking for the applications response time. We offered services that include test strategies, test frameworks, test scripts, test executions and production of measurable test reports.

Metrics

A total of 2400 test cases were automated. 97% of the test cases have been automated. Execution time was reduced by 30%

SQL Server Dev to Release Cycle

Month	June	July	August	September	Release
Regression Bug	19	11	4	3	0
No of Test Case Failures	72	34	4	4	0

Our Value Adds

Business Level

We brought down the automation testing cost by 60% by implementing Selenium automation Tool ROBOT framework for UI. We ensured that there was no downtime of the application under test, which in turn reduced escalation from our client's end customers'.

We hired resources as per our client's specific requirements. Using Build-Operate-Stabilize-Transfer model (BOST), we were able to quickly start the operations without any significant capital investment and increased the testing efficiency & productivity. After 8 years, we seamlessly transferred the ownership and also continued to direct the overall quality effort.

Our client established certain security policies with regard to its computer systems, networks and IP resources. So, we banned the usage of mobile phones, tablets, portable hard drives, dongles, etc. inside the office premises. We ensured that we followed the policies resulting in a higher degree of client trust. We maintained copyright infringement for more than 8 years

Delivery Level

We started the regression testing and performance benchmarking in the year 2006 and supported till October 2013. 24/7 testing support were provided from our end to give a successful productivity. 9% of the resources work even during holidays based on the client requirement.

Over the years, our team grew from 2 to 27 and became the trusted vendor to take care of all the upgrades and patches pertaining to the application. We supported every minor, major, core, main builds and custom builds. 10% investment was made on the shadow resources.

We developed a custom reporting tool for performance benchmarking as add on value and phenomenally gained client's confidence.

The performance comparison at the end showed that the testing efficiency for Functional Regression testing was 98%, which was earlier 70%. There was an increase in the scalability of capture rate for every build. For instance, the capture rate of one build had increased to 7% i.e. from 398 GB/Hour to 424 GB/Hour in two days.

Productivity of our resources:

Functional Testing – 110% (24/7 support with 2 Dedicated resources to fix EBF testing – emergency bug fixing)

Performance Testing – 96% (Our resources worked round the clock in rotational shifts)