

A White Paper Analysis from Orasi Software

Redesigning a Failing Reporting System

Saving Tens of Millions of Dollars;
Cutting Processing Time by 83%



Contents



3
Project Summary

4
Poor System Design and Execution; Even Worse Results



5
Building a Framework for TDM

6
Analysis, Reporting and Remediation



7
Final Outcomes; Optimal Results

Lessons Learned

8
About Orasi Software, Inc.

When a global company known for its vacation destinations couldn't find and eliminate errors in a new sales reporting system—even two years after launch, the organization turned to Orasi for assistance. The results of the effort were nothing short of amazing.

Project Summary

For companies that book hundreds of millions of dollars per year in business, much of which is conducted online, even the smallest glitch in a reporting system can cause a huge margin of error. In the case of one international hospitality leader, management was unable to fully deploy a new reporting system due to millions of dollars of discrepancies between it and the legacy system it was replacing.

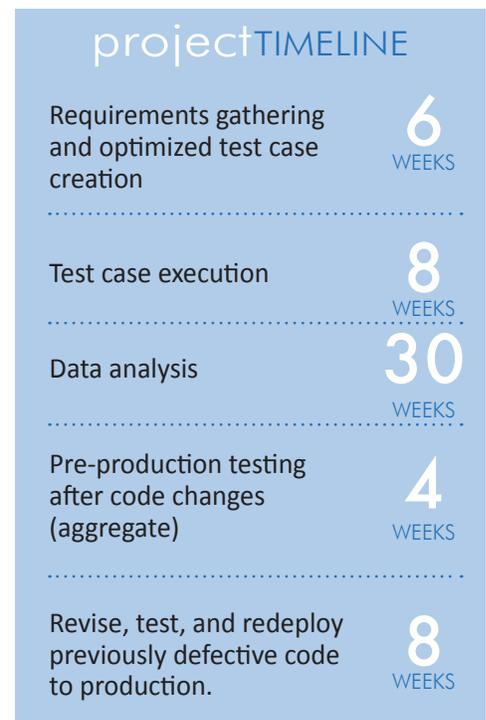
After two years of effort by in-house and third-party consulting teams, errors that had been identified had not been effectively remediated, and attempts at database integration between the two systems had been unsuccessful. Database test performance was dismal—simply loading test data took three months.

In two phases spanning six months, Orasi consultants evaluated the current systems and built, optimized, executed, and analyzed nearly 1400 test cases. The Orasi Test Data Management (TDM) team then remediated the most significant code flaws and deployed the new code to production. The team also made recommendations for best practices and properly configured an existing production monitoring solution to promote quality assurance (QA) in the future.

Parallel to these activities, the Orasi Performance Team conducted test database performance optimization, not only identifying and eliminating processing bottlenecks but also deploying and configuring hardware upgrades to maximize processing capabilities.

By the end of the project, errors in system logic had been identified and corrected, historical data processing had been cut from four months to 21 days; and the firm was positioned to recapture millions of dollars, yearly, that was being lost due to flawed calculations and bad data.

This project success story will outline the role of the Orasi TDM and Performance teams in achieving these dramatic results.



Poor System Design and Execution; Even Worse Results

When client management changed how it was reporting, it discovered its old system didn't have the capability, and leadership decided to build a new reporting system. During comparative testing after development, teams discovered that the two systems were not reporting the same results, even with identical data.

Given that the old system had worked consistently, if somewhat inefficiently, for years, the most obvious culprit was the new system. Yet, neither the consultants originally hired to support the reporting system nor the in-house teams at the client firm could determine the cause of the errors. After two years and tens of millions of dollars in expenditures, the company determined it needed fresh eyes to analyze the systems and navigate a path to issue resolution. To achieve this goal, the client organization engaged Orasi Software, which recommended a comprehensive TDM effort that included both performance optimization (discussed in a separate technical paper) and testing optimization and data analysis.



The Orasi team uncovered not only major, specific reporting issues but also widespread, endemic practice flaws, including risky team practices and ineffective use of QA tools.



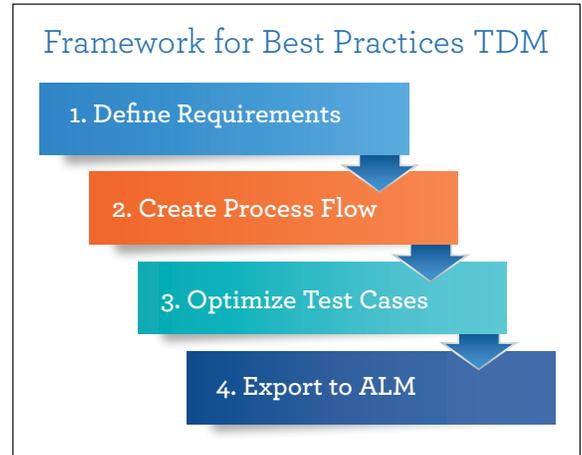
During its analysis, the Orasi team uncovered not only major, specific reporting issues but also widespread, endemic practice flaws. In addition to the reporting issues, in which both systems were implicated, the effort was being significantly impacted by risky team practices and ineffective use of QA tools. Highlights of discoveries included:

- **Legacy System Woes.** The legacy reporting system was an off-the-shelf software solution that had been deeply customized but inadequately tested, resulting in flaws being introduced over time. The client had deployed a patch to fix them, but when Orasi teams ran optimized test cases, they discovered the issues had not been resolved, which meant both systems were producing inaccurate results.
- **System Parity.** The two systems reported in different ways and therefore were recognizing and processing identical data differently. Differences in calculations for items such as sales taxes were causing miniscule discrepancies that, when replicated over hundreds of millions of records, resulted in million-plus dollar losses, yearly.
- **Quality Assurance.** The third-party firm hired to develop the new system had not been following best practices, making problem resolution more difficult and, in some cases, exacerbating the problem. For example, the team had been deploying code changes straight to production without proper testing or quality control, introducing additional flaws into the system.
- **Lack of Production Visibility.** At the time of the engagement, the organization had no mechanism for pinpointing errors in production. Even though the client was running AppDynamics, it had not been optimally configured.

Building a Framework for TDM

During its situation analysis, the Orasi team had determined that existing best practices. Developers were performing ad-hoc code inspections and code testing was inadequate and incomplete. Furthermore, testing teams were also missing opportunities for QA, such as not optimizing their test cases.

Orasi recommended creating and optimizing test cases from scratch, with sufficient depth and breadth to provide accurate results.



The project leader persuaded client management that the effort would pay off in the end—Orasi would be able to figure out what was wrong and the effort would provide a “proof of concept” to validate teams continuing with sound QA practices, moving forward. With client management on board, the effort—to identify the source of reporting discrepancies and provide the detail to enable their resolution—began.

Steps to achieve the goal, aligned with Orasi’s recommended best practices for an end-to-end TDM engagement, follow.

1. Define requirements for the data.
2. Create the process flow.
3. Map out and optimize the test cases.
4. Export test cases to Micro Focus Application Lifecycle Management (ALM).

With these steps accomplished, the team could run the optimized test cases to generate meaningful data for analysis.

Although all steps were essential to the process, the most extensive effort—and an invaluable component to achieving such unqualified success—was Step 3. To achieve this goal, the team used CA Agile Requirements Manager to develop a full spectrum of test cases, including cases for full testing, all-pairs testing, in-and-out edges, and other recommended methodologies. The effort revealed that reporting data was flowing from five different databases, necessitating an unusually large number of tests.

Concurrent with the TDM effort (Steps 1 through 3), the Orasi team engaged in multiple meetings with the client to obtain system functionality details. It also reconfigured and optimized the existing AppDynamics deployment. The optimized platform not only provided vital support for identifying application problems during Step 4 (running test cases); it also gave the client a functional production monitoring tool, moving forward. (This effort is detailed in Production Hits Its Stride, below.)

Production Hits Its Stride

The client had previously installed performance monitoring solution AppDynamics into all environments including, production, stage, testing, and development. Orasi was brought in to assist with configuring AppDynamics to help the client resolve a problem. During our investigation into their installation, we discovered that not all nodes had been installed with AppDynamics and that the possible source of the issue was one of the nodes where it had not been installed. We also found other areas that we could assist them in tuning AppDynamics for troubleshooting their issue: Instrumentation of systems pertinent to the project where AppDynamics was previously not installed.

- Data Collector and Information Point configuration.
- Dashboard creation for different teams across the company.
- War room creation.
- Custom Flow Map creation.
- Tuning their alerting and health rules.
- Business transaction configuration streamlining.

Analysis, Reporting and Remediation

With test cases run, the team began analyzing the resulting data. Analysis included both manual and SQL queries to identify the anomalies that would indicate where issues existed. The team reported problems to the client as it found them and the client company performed its own verification. Client management determined whether issues were important to business goals and ordered them by priority.

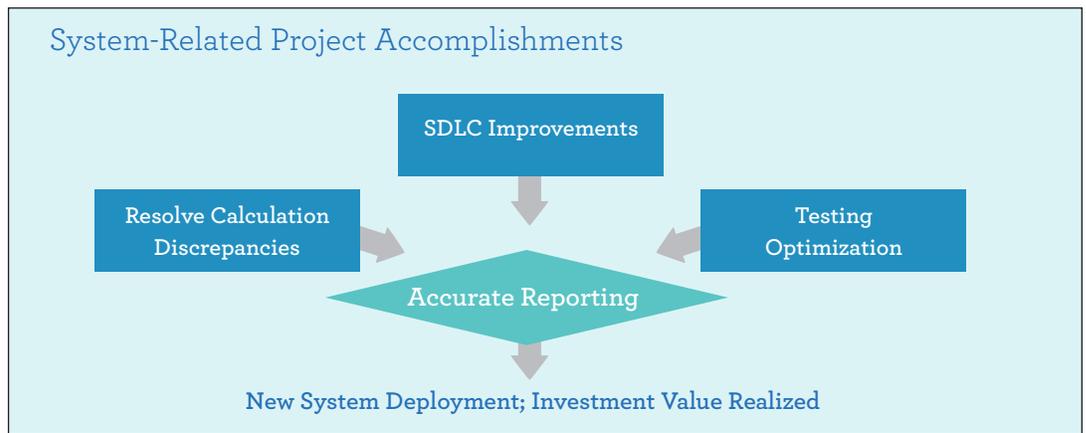
Of the issues reported, client management determined that 13 were major reporting issues. Four were deemed critical, but all were important enough to be put into the production fix. The four critical issues were:

- **Hard-coded Calculation Discrepancies.** One source system was coded to calculate values using a particular method, while another source system was calculating them using a different method, resulting in incompatible results.
- **Varied Tax Calculations.** Calculation of tax exemptions was not consistent across all systems.
- **Improper Data Grouping.** Data was placed in one grouping when it should have been in another.
- **Rounding.** Fractions were being rounded differently from one system to the next.

Many of the fixes would require significant code updates and system adjustments, so the next step was to find resources to perform the work. In order for changes to be meaningful, code rewrites would need to be accompanied by appropriate QA testing, possibly with additional adjustments, before the code updates went live.

Although the client had developer resources to make the code updates, the previous method of allowing developers to identify and correct their own defects was counter-intuitive to success. Client management did not have the ability or will to acquire or manage trained testing resources, so Orasi testing teams assumed that role. Over a period of eight weeks, all defective code was revised, properly tested and deployed to production.

From a broader perspective, rectifying the discrepancies and allowing the client to move forward on its new reporting system will enable better, or accurate reporting and provide elevated, trustworthy business intelligence for effective, executive decision making.



Final Outcomes; Optimal Results

Every optimization engagement eventually reaches a point of diminishing returns after. For this project, although the Orasi team recommended a reconciliation to resolve all the issues it had found, the client did not see value in continuing to the recommended level of defect remediation and process refinement.

Nevertheless, project results were noteworthy. Specific to QA, the Orasi team developed and put into practice a reusable testing environment:

- The new reporting system—into which the client had invested millions of dollars—is in operation and can achieve stakeholders’ original objectives.
- The Orasi team’s analysis pinpointed calculation errors that will enable the client to save hundreds of millions of dollars annually.
- The team identified an unacceptable level of instability in the production environment and helped the firm implement AppDynamics to pinpoint production-side issues.
- The testing environment which was previously not being used was improved to have data flowing from the source systems and again used for testing, as it should have been, all along.
- Data refreshes have gone from three months to one.

Lessons Learned

Although the Orasi team accomplishments were impressive, the client firm could have avoided two years of pain and millions of dollars in losses if it had followed a QA-focused model—or hired a software QA firm such as Orasi to help them implement one if they lacked the capability.

Even though the scope of the problem—and the project—were unusually large due to the size of the company, the underlying issues that triggered it were not remotely unusual. At the heart of the predicament was failure to adhere to best practices across the board; especially in the areas of testing and QA.

Had the in-house teams and their initial consulting firm followed best practices for system development—from creating proper requirements to testing thoroughly across the development cycle—the resulting code would have been much cleaner from the outset. From that point, ongoing production monitoring and support would have kept the systems operating more efficiently and accurately. Instead, a lack of support merely exacerbated an already serious problem.

“

Although the Orasi team accomplishments were impressive, the client firm could have avoided two years of pain and millions of dollars in losses if it had followed a QA-focused model—or hired a software QA firm such as Orasi to help them implement one if they lacked the capability.

”

About Orasi Software, Inc.

Orasi is an award-winning software reseller and provider of software training, support and professional services. To help companies focus on a complete software quality lifecycle, Orasi offers market-leading consulting services and solutions to support automated testing, application performance management/intelligence, mobile technologies, DevOps pipeline efficiency, and operational excellence. Orasi continues to expand its offerings across the entire software delivery spectrum, from data analytics to continuous delivery and open source tooling. Orasi maintains strategic partnerships with Micro Focus (formerly HPE), Chef, Delphix, SAP, XebiaLabs, and others. For more information, please visit www.orasi.com.



For more information, contact Orasi today.
www.orasi.com | 678.819.5300

© 2018 Orasi Software, Inc. The information contained herein is subject to change without notice.

Orasi is a trademark of Orasi Software, Inc. All other product and company names are trademarks or registered trademarks of their respective owners.