

Case Study: Modernizing Northern Trust's Treasury Management System

Client: Northern Trust

Project: Treasury Management System Modernization

Duration: Ongoing (3.5 months since Sebsibe joined)

Creospan Consultant: Sebsibe

Month: June 2024

Business Problem and Project Overview

Business Problem:

Northern Trust's Treasury Management System was critical for the strategic management of client funds but had become increasingly challenging to maintain due to its reliance on an outdated technology stack. This 30-year-old system, utilizing Java and a JSP frontend, was integral to daily operations for both internal teams and large corporate clients. Over the years, it had accumulated a range of issues, including numerous critical bugs, complex integration challenges, and significant security vulnerabilities. The heavy dependence on offshore development teams for ongoing maintenance had compounded these issues, adding layers of complexity to system management and escalating the urgency for a modernization initiative.

User Impact:

The inefficiencies and operational disruptions stemming from the system's instability had profound impacts on both internal stakeholders and corporate clients. Persistent bugs and integration challenges, notably the protracted integration issue with the Bank of New York Mellon, had severely disrupted daily operations, leading to operational inefficiencies and a marked increase in support calls. Moreover, the unresolved security vulnerabilities represented a considerable risk, with the potential to expose sensitive financial data. The long-standing nature of some bugs, unresolved for as long as 2-3 years, underscored the critical need for a system overhaul to ensure reliability, security, and efficiency in fund management processes.

Project Overview:

The objective was to stabilize the existing system, address the critical bugs and security vulnerabilities, and prepare for a phased migration to Azure. This modernization effort aimed to enhance system stability, reduce operational risks, and minimize dependency on offshore teams.

Our Approach & Solution

1. Application Stabilization & Major Bug Fixes:

The Creospan Consulting team focused on resolving critical bugs from the outset. They addressed significant integration problems with the Bank of New York Mellon, which took two sprints to resolve. The team also fixed issues related to currency code changes and account load processes. For example, they introduced a function to handle currency code changes by writing Unix scripts, ensuring the application could detect changes such as converting USD to EUR.

- **Currency Code Changes and Account Load Processes:** The team implemented a function to handle currency code changes through Unix scripts. This upgrade ensured that the application could automatically detect and process changes in currency codes, such as converting USD to EUR. This functionality is crucial because without it, financial transactions and reports could reflect incorrect currency amounts, **leading to significant financial discrepancies and compliance issues**. Correct currency conversion is vital for accurate fund management, financial reporting, and maintaining trust with global corporate clients who operate in multiple currencies.
- **Integration Problems with Bank of New York Mellon:** Addressing these integration problems was essential to restoring and enhancing the **interoperability between Northern Trust's system and the Bank of New York Mellon's systems**. Previously, issues in this integration had caused delays and errors in processing transactions, which are critical for seamless financial operations. By resolving these issues, the team helped ensure that financial transactions involving these two institutions were executed more reliably and efficiently, thereby reducing operational risks and improving client satisfaction.

2. **Core Security Upgrades:** The consulting team successfully fixed a major SQL injection vulnerability. This fix was crucial as it served as a model for addressing other similar vulnerabilities. The system had **480 identified vulnerabilities**, categorized into different classes, and the team's work on the SQL injection vulnerability set a precedent for tackling others.
 - **SQL Injection Vulnerability Fix:** This resolution of a major SQL injection vulnerability was a critical step in safeguarding the system against unauthorized data access and potential data breaches. **SQL injection is a common attack vector** that can allow attackers to manipulate databases, steal confidential information, and even disrupt system operations. By fixing this vulnerability, the team significantly lowered the risk of data breaches, enhancing data security and protecting sensitive financial information of Northern Trust and its clients.
3. **Database Migration Planning:** Preparations began for the database migration to Azure, scheduled to start mid-June. This phase focuses on ensuring data encryption both on-premises and during transit to meet stringent financial industry requirements. The plan involves a phased migration of eight databases, with the first database expected to be transferred by November.
4. **Knowledge Transfer and Dependency Reduction:** The Creospan Consulting team actively learned the intricacies of the system to reduce reliance on offshore teams. This effort is aligned with the client's goal to minimize dependency on two senior offshore developers contracted by a different vendor (Cognizant), who had been working on the project for **15-30 years**.
 - **Reducing Dependency on Offshore Teams:** By empowering the Creospan Consulting team to take over more responsibilities previously handled by offshore teams, this initiative aimed to reduce response times for bug fixes and system updates, enhance the team's ability to manage and adapt the system internally, and decrease reliance on external contractors. This shift was vital for improving the agility and responsiveness of the maintenance team, ensuring that system updates and fixes could be implemented swiftly and efficiently. It also facilitated better control over the system's evolution, aligning development more closely with Northern Trust's strategic needs and reducing operational risks associated with external dependencies.

Impact for Northern Trust

- 1. Enhanced System Stability:** The consulting team's focused efforts led to a significant reduction in critical system bugs, thereby enhancing the overall reliability and performance of the Treasury Management System. This improvement has ensured smoother financial operations and strengthened the system's resilience against future challenges.
- 2. Reduced Operational Risks:** The critical security overhaul addressed and mitigated numerous security vulnerabilities, substantially decreasing the risk of data breaches. This proactive approach to cybersecurity has safeguarded sensitive financial data, aligning with industry best practices and regulatory compliance standards.
- 3. Built Onsite Expertise and Reduced Dependency on Offshore Teams:** With strategic knowledge transfer and skills development led by the consulting team, Northern Trust successfully built substantial onsite expertise, significantly reducing its dependency on long-term offshore contractors. This shift not only enhanced operational control and responsiveness but also ensured that critical system knowledge was retained within the company, promoting long-term sustainability and independence in system management.
- 4. Improved User Satisfaction:** The stabilization and enhancements of the system led to smoother operations and fewer disruptions, resulting in higher satisfaction levels among both internal users and corporate clients. The reliability and efficiency improvements have fostered stronger trust and confidence in Northern Trust's services, enhancing client relationships and internal morale.

Next Phase

- 1. Database Migration to Azure:** Starting mid-June, focusing on encryption and phased migration of eight databases.
- 2. Frontend and Backend Modernization:** Migrating to React and SpringBoot, expected to take about a year, ensuring a robust and modern system.

Technology Stack

1. Current Stack:

- Backend: Pure Java
- Frontend: JSP
- Servers: Unix, Mainframe with Unix scripts and Java code

2. Future Stack:

- Frontend: React
- Backend: SpringBoot
- Database: Azure