Taming Unruly Data to Improve Compliance

亦有中文简体译本
www.isaca.org/currentissue

A banking institution based in Toronto, Canada, with approximately CA$18 billion in annual revenue found itself facing a big data problem: A multipurpose proprietary software application, used across the organization for more than a decade, was out of sync with regulatory requirements. The discovery led to a 10-month journey of assessing and solving the problem.

The issue surfaced after the Canadian Office of the Superintendent of Financial Institutions (OSFI) and the Financial Transactions and Reports Analysis Centre of Canada (FINTRAC) established a new set of compliance norms governing a variety of anti-money laundering (AML) and anti-terrorist financing (ATF) regulatory reports. In 2018, Bhavani Shankar Mudigonda, a senior consultant in application development, found that the bank was noncompliant with the new standards.

MICK BRADY

Is a freelance technology communicator with more than 20 years of experience editing and writing for technology-focused publications.

An internal investigation revealed that due to the existing data management procedures, recycled customer identification numbers (IDs) were being attached to multiple deposit and investment accounts. Consequently, there was a risk of exposing customer information to unauthorized parties and compromising the true account holders' privacy.

This was a serious issue for a financial institution that serves retail and commercial clients throughout Canada, including capital markets, with a full range of financial products and wealth management functions. To prevent potential legal, financial and reputational consequences, the bank wanted to solve the problem expeditiously, both to safeguard customer accounts and to achieve compliance before any external audits were conducted.

Assessing the Challenge

The first step was to identify the root cause of the errors. The practice of creating a unique ID (i.e., a digital identity) is a normal procedure for retail and commercial customer-servicing organizations from banks to telecommunications companies. It is necessary for tracking purposes. At the bank, an in-house application assigned a unique ID every time a client visited a banking center, made a digital transaction or telephoned a customer service representative.

"Any application tends to have a certain amount of space in the database," Mudigonda noted. "So, we do housekeeping, and then we repurpose some of the existing account numbers." The bank's system recycled client IDs after accounts were either closed or purged.

Mudigonda also discovered that throughout the application's lengthy application development life cycle, multiple records had been created for testing and other purposes. The result was that thousands of fictitious records existed alongside the bank's valid data.

Because fictitious records remained in the system even as unique client IDs were being repurposed,

there was a risk of the same account number getting assigned to multiple clients, which would allow Customer #1 to view the account information of Customer #2.

"This is a large application, and it is the book of record for client information. It is used by many groups across the organization for various purposes that include any kind of manipulation of client data," Mudigonda explained. The challenge was to keep the genuine data intact while identifying and eliminating the fictitious records to protect customer privacy and fully comply with current regulatory requirements.

Once the scope of the problem was understood, a core team was assembled to correct it. The team's mission was to investigate possible solutions, determine the best approach and, ultimately, develop, implement and oversee ongoing operations.

Developing a Solution Strategy

The core team included a software designer who held brainstorming sessions with the other team members to consider a variety of approaches. The team concluded that applying a patch to the bank's in-house application and running purge jobs multiple times each night would solve the problem.

A developer worked on the coding. Another team member conducted tests. Mudigonda oversaw the entire project, circling back with the bank's business controls team to make sure the new features would bring the application into compliance.

Certain business rules could be coded into the main application to categorize a record as fictitious, for example:

- The client's first name, middle name, surname, street address, city, province or postal code consisted of three consecutive characters (e.g., AAA, 111).
- The client's name was a very generic name (e.g., John Doe).
- The client's first name, middle name, surname, street address, city or province had the text string "test" or "testing" or "test record."
- There was a negative response to a real-time address verification API call to Canada Post.

Many of the proposed rules that surfaced during the brainstorming sessions were rejected. For instance, using OR for the business rules would have resulted in the purging of legitimate records. "The solution that we implemented uses AND, which makes it robust," Mudigonda explained.

The team concluded that applying a patch to the bank's in-house application and running purge jobs multiple times each night would solve the problem.

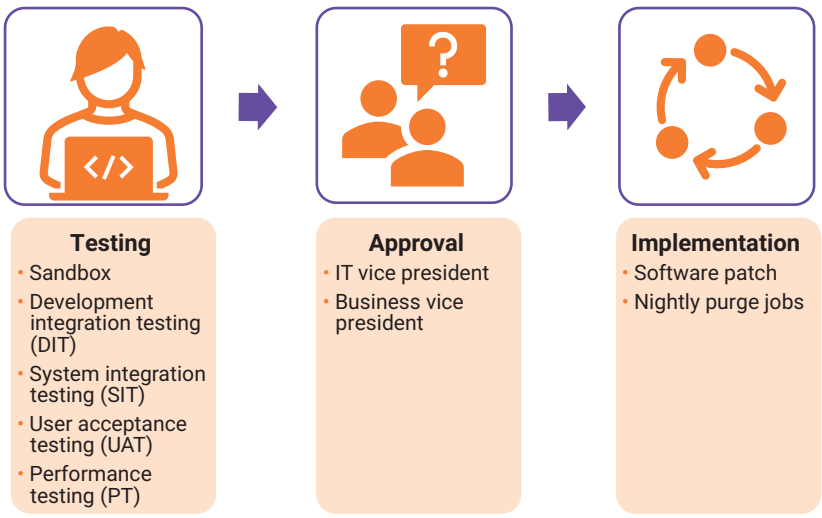
Ultimately, approximately 24 specific business rules were framed by an internal controls team composed of subject matter experts responsible for internal audit, according to Mudigonda. A record was categorized as fictitious only if it satisfied all of the 20-plus criteria established in the rules.

In addition to the software patch coded into the application, the solution called for running purge jobs several times each night. For that, the team was able to reuse the jobs scheduler it had, which ran many other jobs within the application.

Mudigonda provided an overview of the project's development, from the sandbox to implementation (**figure 1**).

FIGURE 1

Fictitious Records Purge Development Process



Testing, More Testing and Retesting

In keeping with the bank's practice of following a complete software development life cycle, several testing phases of prototypes took place prior to the deployment of an integrated solution.

In the Sandbox

The team first developed the code in a local machine (i.e., a sandbox machine) and performed one round of unit testing.

"I did a test on my own," Mudigonda said. "My system was not connected to anything. It was just a stand-alone box. Once I, as a developer, thought that everything was fine, then I promoted the same code to the DIT environment."

> Because the system used a shared platform, the bank was able to automatically provide regulators with corrected data in real time.

DIT Environment

The DIT environment adds functional integrations. When the code was promoted to DIT, developers were able to test the patch's functionality with other integrated systems.

With approximately 200 dependencies—about 100 upstream applications and 100 downstream applications—a great deal of testing was required. "In the DIT environment, all the integrations were available," Mudigonda pointed out. "For example, our patch was the ABC application and there was another application called XYZ. In the DIT environment, ABC was connected to XYZ. However, in the sandbox, ABC was just a stand-alone machine."

Once satisfied that the code worked properly when the dependent systems were integrated, the software patch was promoted to the next level of testing: the SIT environment.

SIT Environment

In the SIT environment, the quality assurance (QA) department took over the testing process.

"Since they provide specialized QA services, they write their own test cases," noted Mudigonda. "Based on the test cases they wrote, they tested the piece of code we deployed in the SIT environment."

After completion of the SIT environment tests, the code was promoted to the UAT environment.

UAT Environment

In the UAT environment, a cross-section of users, primarily from the bank's business side, worked with the new application. The project team provided access to the environment to a sample of users who tested it in the context of their work routines and provided signoff. Once it was approved in the UAT environment, the software was promoted to the PT environment.

PT Environment

In the PT environment, a dedicated team within the QA department retested the code from the performance perspective. The testers considered a range of issues, such as whether the central processing unit (CPU) was being overtaxed because of the code, or whether the code used too much memory.

Gaining Executive Approval

After the code was approved in each testing environment, the team presented its proposed solution to the IT and business decision makers who could give its deployment a green light.

"On the IT side, we had to get it signed off by our vice president," recalled Mudigonda. "We had to first develop a prototype and then demonstrate the prototype to the VP and get his blessing. And, at the same time, we had to also conduct a demonstration for the VP on the business side and get approval from him before we could actually implement the product."

Those executives approved the solution, and the next project phase began.

Implementing the Solution

By autumn 2019, the business rules were coded into the software and the solution was ready for use. The system took .CSV files from the landing platform (folder) where files were shared with the regulatory agencies and applied the rules. Any records in the files that satisfied all the rules were categorized as fictitious. All the fictitious records were then combined in a file in the database and inputted to

a purge job. The purge process moved the records from the core application to the history application.

"This purge job was scheduled to run five times per night," Mudigonda said. "Gradually, over a period of about 10 weeks, we got rid of 95 percent of fictitious records, cleansing the system and making the regulatory reports that we produced free of fictitious records."

Because the system used a shared platform, the bank was able to automatically provide regulators with corrected data in real time without the need to engage in multiple rounds of follow-up. The error-filled reports could be corrected prior to the performance of regularly scheduled audit activities.

"Reports were compared with the production database and reconciled with the regulatory teams," Mudigonda noted, so there was no possibility of the new rules identifying any legitimate records as fictitious.

With the system now a regular part of its data management procedures, the bank is "100 percent confident" in the accuracy of its results, said Mudigonda. "Post-implementation, we verified client records that got purged using the robust audit tables in our database."

The database has audit tables that create a unique transaction ID for each event, allowing developers to trace the flow of each transaction end-to-end, he explained.

Calculating the Benefits

For the bank, the business rules innovation amounted to much more than a one-time solution for cleaning up the thousands of fictitious records that were identified in its system. It became a long-term strategic solution for more robust database maintenance. The bank now routinely schedules an automated job to analyze client identifiers five times a night and purge records identified as fictitious.

As for return on investment, "the total cost of the solution—including designing, developing, coordinating with business, and production deployment—was approximately CA$50,000," Mudigonda said.

It is reasonable to assume that any audit-related penalties the bank might have incurred if it failed to achieve compliance with regulatory requirements could have been much higher. "I really cannot exactly quantify the number, but usually it is in millions of [Canadian] dollars," Mudigonda said.

Yet producing trustworthy and credible reports is, in a sense, a priceless benefit due to the potential far-reaching ripple effect of mismanaged data. "There are many millions of client identifiers in the system," Mudigonda said. "It is important for this system to be cleansed of fictitious records since many other product systems rely on this data, as do many regulatory agencies that publish reports."

Producing trustworthy and credible reports is, in a sense, a priceless benefit due to the potential far-reaching ripple effect of mismanaged data.

The entire bank relies on customer data for everything from routine day-to-day banking to mortgage lending, commercial banking, wealth management, investment products, and enterprise trading in capital markets. The software innovation helped the bank alleviate privacy violations that could result in reputational damage if not addressed, Mudigonda pointed out.

There are compliance standards in every business sector, he noted, and the organizations operating in them can run into similar issues. "This is actually not just applicable to financial institutions—it could be applicable to any organization which is serving customers," Mudigonda said. "That could be a telecom operator, or it could be an airline, or it could be a retail store—or it could be any store which is basically serving customers. For example, if you have a membership at [the US-based big-box chain] Costco, then every time you go you are probably given a unique ID. So, any organization which creates that unique ID for each client—this problem can happen to them."

Through a very modest investment and reliance on the ingenuity of its problem-solving team, the bank prevented its fictitious records problem from morphing into a big data disaster.

ENJOYING THIS ARTICLE?

- Learn more about, discuss and collaborate on risk management in ISACA's Online Forums. *https://engage.isaca.org/onlineforums*