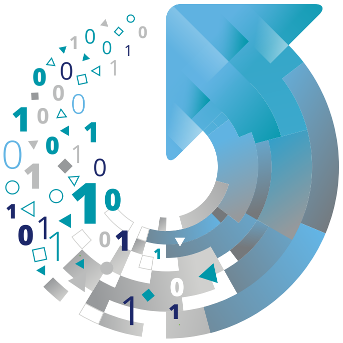
Deloitte.



Modernizing an insurance claims application for enterprise-wide compatibility

Deloitte Application Modernization

Global insurer updates legacy app with automated refactoring

The challenge

A large European insurance company with operations in more than 100 countries had embarked on an enterprise-wide transformation to help it achieve strategic growth.

But as the insurer began to plan the project, it faced a problem: Its claims-processing application could not be migrated to a flexible, open-systems platform. The software ran COBOL, a 59-year-old programming language that, while used by many large businesses, can be difficult to integrate with modern programming languages, database software, and Java environments.[1]

What's more, the insurance company was having difficulty finding IT personnel with experience in legacy code languages like COBOL, making the antiquated system increasingly costly to support. In part, that's because workers fluent in old languages like COBOL are in short supply as they begin to retire. At the same time, most universities no longer offer courses on legacy code. In many cases, this shrinking supply of skilled workers has resulted in high salaries for legacy coders and database administrators—which in turn has increased the costs to support outdated applications.

The insurer also needed to update its claims-processing application without altering the user interface and functionalities to minimize disruption for end users and help ensure that developers and database admins did not face a steep learning curve. More importantly, the company needed to retool the applications without system downtime or interruption to operations.

The application modernization approach

The insurance firm engaged Deloitte's Application Modernization services to help it plan and implement an end-to-end modernization program of its legacy claims-processing application.

The project began with a Discovery assessment that analyzed the COBOL code, mapped interfaces to other applications, and identified mutual dependencies. The Deloitte team worked with the company's IT staff to identify numerous applications that would be affected by the refactoring.

The Discovery work resulted in a defined scope, duration, and project implementation plan for the modernization project. Separately, the Deloitte team carefully reviewed the roles and responsibilities of the firm's business departments and suggested revisions that could improve process efficiency.

Based on this assessment, Deloitte's Application Modernization team recommended a migration strategy based on automated refactoring, a technique that uses software to automatically replace old code like COBOL and Natural with newer languages such as Java and .NET.

The modernization project would employ Deloitte's Transformation tools, a component of our Application Modernization powered by innoWake™ suite of solutions. The Transformation tools can convert all legacy code into modern languages with full accuracy, a process known as 100% automated refactoring. And because the new code mirrors legacy programming, both legacy and Java developers can easily read and maintain the migrated application. This approach preserves the functionality and interfaces of the existing application while building a foundation for future modernization.

> "Everything works just as it did before refactoring. But the Java environment makes it easier to integrate operational standards and align with best practices. End users did not notice any changes in the application."
>
> *— Mathias Jacobi, project manager, Deloitte Consulting LLP*

[1] The National Museum of American History, COBOL, accessed April 23, 2018

The application modernization process

The Transformation project comprised three major initiatives: migrating the COBOL code to Java, transferring the data from Adabas to Oracle, and updating the mainframe operating system (OS) to Linux from BS2000.

The Deloitte Application Modernization team used its proprietary innoWake tools to refactor the legacy code. The tools simultaneously migrated the legacy and refactored code to the mainframe and to the new target architecture. This allowed normal maintenance of code and data to proceed without a code freeze or system downtime, minimizing disruption to the business.

The next step was to fully automate the code migration and the Job Control Language (JCL) implementation. This approach helped create a smooth transition for legacy developers, who could continue to work with COBOL using the Eclipse integrated development environment (IDE).

To verify system functionality, the Application Modernization team created more than 200,000 record test cases that were used for regression and performance analysis of the new engine. This very high number of test cases not only helped speed implementation, but also delivered insights into the application and helped build confidence that the software would perform well in production.

Results

To help the insurer prepare to use, maintain, and operate the new version of its claims-processing system, the Deloitte team designed a training program that began early in the project.

Rapid implementation, minimal training

Within six months, the team conducted Java toolchain workshops for legacy developers, and by go-live, all developers had been fully coached in the use of the new code and tools. Because the interface and functionality of the application remained the same, claims processors quickly accepted the updated software, with minimal training required.

Directly after go-live, the company's IT staff assumed full management and operations of the application. Almost immediately, the insurer began developing and integrating new functionalities using Java code.

Identical functionality and interface

When the 28-month modernization project concluded, the insurance firm had a highly integrated version of its claims-processing application that provides the same interface and functionality as the legacy version.

The modernization project was completed on time and on budget, with no code freeze or downtime, a critical requirement for the insurer. "Everything works just as it did before refactoring, but the Java environment makes it easier to integrate operational standards and align with best practices," said Mathias Jacobi, project manager, Deloitte Consulting LLP. "End users did not notice any changes in the application. We promised nothing would change in interfaces and functionality, and from a user standpoint, nothing did change–even though everything has changed under the hood."

A solid foundation for the future

The refactored system is now hosted on an open-system platform using modern technologies and supporting tools. All core functions of the application backend are now available as RESTful Web services, making it easier to share business functions of the server. As a result, new functionality is easier, faster, and less complex to develop and integrate.

Finally, the modernization will decrease operating costs because the insurer will no longer have to maintain the mainframe after it is switched off next year. That's expected to save millions of dollars a year on systems maintenance and operations.

Beyond an updated claims-processing application that is compatible with the insurance firm's global open-system platform, the software modernization created a solid foundation for future enhancements to meet evolving business and technology requirements.

Refactoring by the numbers
The project refactored 1.2 million lines of COBOL code and 1,400 objects. Data refactored included 22 million claims, 2.5 billion database records, 250 tables, and 200 sequential and ISAM files.

Deloitte innoWake tools used: development, enabler, and operations

Dr. Karsten Ballüder
Director,
AppMod Centre of Excellence
Deloitte Consulting GmbH
Tel. +49 172 654 6151
Email: kballueder@deloitte.de

www2.deloitte.com/de/CoE-AppMod

About Deloitte
This publication contains general information only and is based on the experiences and research of Deloitte practitioners. Deloitte is not, by means of this publication, rendering business, financial, investment, or other professional advice or services. This publication is not a substitute for such professional advice or services, nor should it be used as a basis for any decision or action that may affect your business. Before making any decision or taking any action that may affect your business, you should consult a qualified professional advisor. Deloitte, its affiliates, and related entities shall not be responsible for any loss sustained by any person who relies on this publication.

As used in this document, "Deloitte" means Deloitte Consulting LLP, a subsidiary of Deloitte LLP. Please see www. deloitte.com/us/about for a detailed description of the legal structure of Deloitte LLP and its subsidiaries. Certain services may not be available to attest clients under the rules and regulations of public accounting.