

Transformation of core applications on the path toward digitalization

Axel Rupp

Partner
Consulting
Deloitte

Dirk Guttzeit

Partner FSI
Consulting
Deloitte

Dr. Karsten Ballüder

Director
Consulting
Deloitte

Björn Langmack

Managing Director
Consulting
Deloitte



Most financial institutions understand that digitalization is critical to meet the demands of tech-savvy customers. They also know that a fully digital business can enable them to create new products and services, and even broaden their business models.

At the heart of digitalization are customer relationships and their loyalty to the financial institution. And that's why digitalization can be fraught with risk. According to a recent Deloitte study on FinTech in the insurance sector, insurance companies often lack an adequate understanding of customer loyalty, provide inadequate value-added digital services, and may jeopardize one-to-one customer relationships.¹ Banks face a comparable scenario.

IT plays a crucial role in digitalization, albeit one that is hindered by a portfolio of applications whose technology stack dates back several decades. Even two-speed IT, which aims to integrate legacy applications with advanced technologies, doesn't go far enough. To be successful, a digital transformation strategy must encompass the entire application portfolio, organization, and processes in a multi speed model. ➤



¹ Deloitte LLP, "A catalyst for change: How fintech has sparked a revolution in insurance," 2018, <https://www2.deloitte.com/content/dam/Deloitte/global/Documents/Financial-Services/gx-fsi-catalyst-for-change.pdf>



In particular, companies should address digital sustainability among core back-end applications that have helped generate profits. Many do—Deloitte’s Tech Trends 2018 shows that organizations across industries have made significant investments in mission-critical back-end systems over the decades.² Investment in back-end systems is particularly important for financial institutions.

The reason? Banks and insurance companies often use back-end technology that dates from the 1960s and 1970s. Many core applications have been written with legacy programming languages like COBOL or PL/1, and often access assembler subroutines that are even older.

Modern front-end systems typically use middleware technology developed in the 1990s. These core applications are usually run on mainframe infrastructures that are updated with new functionality in several large release cycles per year.

The modernization strategy

In the financial services industry, the CIO has the mission-critical task of implementing a fundamental IT transformation.

The financial industry is rife with initiatives for outsourcing IT infrastructure and application maintenance. The rapid improvement of encryption technologies and enhanced performance of APIs are persuading more firms to implement cloud solutions to varying degrees. Typically, they first move noncritical business processes to the cloud.

Transactional processes, such as those used in finance or compliance, are often outsourced without the CIO’s control. This can externalize the associated applications and infrastructure components. While standard IT tasks are increasingly outsourced to third parties, legacy applications present distinct challenges. These applications are typically incompatible with modernized functionality, technology, and architecture.

Some of the issues driving modernization in the financial industry are as follows:

- Traditional ESB models are being replaced by highly flexible microservice structures. Modern front-end systems—mobile apps in particular—need these atomic services to achieve the required speed and dynamics.
- Organizations are designing more flexible core financial systems that enable products to be quickly adapted to market requirements. As a result, we’re likely to see quantum leaps in time-to-market, ranging from agile forms of organization to fully automated DevOps.
- Customer centricity also brings the expectation that a great deal of information will be compiled and made available in near-real time. On the other hand, some processes are traditionally processed in batch mode.
- The development of analytics will help organizations better manage the business right down to individual customer engagement. The emerging

field of artificial intelligence is likely to deliver additional customer-centric advantages.

In addition to these drivers, evolving workplace issues and regulation such as the following have made modernization of legacy applications a business priority:

- Modernization can enable organizations to either retire or reduce their loads on mainframes, which can result in sizable cost savings that can be reinvested in more differentiating technologies. The savings can be significant—research firm Gartner estimates that application modernization can reduce costs on hardware, software, and human resources by as much as 74 percent.³
- Developers and system programmers who maintain and continue to develop legacy applications are retiring and will continue to do so. This is a risk because these employees will take with them years of knowledge about the development of legacy applications, which is often undocumented.
- Increasingly stringent and pressing regulatory and security requirements can increase the operational risk of legacy applications. It is not unusual, in fact, for organizations to call back retired coders to help solve development issues.

As part of a comprehensive digitalization strategy, each CIO should prioritize these drivers, and define and implement an integrated modernization program for the entire application portfolio.

Paths toward modernization

Modernization is typically implemented using one of four methods:

New construction of applications

(custom development): this approach to modernizing core applications has become increasingly rare in recent years. Custom development typically entails high risk, immense costs, and a very long development phase, all of which lessens its appeal. The advantage is a fundamental

The rapid improvement of encryption technologies and enhanced performance of APIs are persuading more firms to implement cloud solutions to varying degrees.

redesign of functionality, architecture, and technology. It is worth noting, however, that custom development can fail due to the lack of specialist resources and the long period between specifications and implementation.

Use of standard software (COTS/

MOTS): the successful use of standard software requires business processes to be easily mapped to the application. We very often see the use of standard software, despite large functional gaps. This can often require additional development that can nullify the advantages of standardization. Among global financial institutions, standard software solutions typically do not address all national and regional requirements.

Outsourcing: fewer organizations outsource their modernization initiatives. A primary risk is that old programming languages are transferred to the service provider and ultimately are not fully addressed. In addition, the relevant contracts must be drafted to help ensure that agility, innovation, and the requirements for functionality, architecture, and technology are effectively implemented. Often, however, outsourcing focuses on cost savings, and that can incur additional costs in the future. ➔

³ Gartner, "IT Key Metrics Data 2017: Key Infrastructure Measures: Mainframe Analysis: Current Year," 2017/IT Key Metrics Data 2017: Key Infrastructure Measures: Linux Server Analysis: Current Year, Gartner, 2017



Refactoring: this approach uses fully automated software to convert user interfaces, code, and data. Once converted, the refactored applications have the same functionality and behavior as the legacy applications. Significant advantages of automated refactoring include rapid implementation, a 10 to 15 month return on investment, enhanced scalability, and reduced implementation risk. Additionally, modernization and implementation of new requirements is typically faster and more reliable with the new platform.

In addition to these methods, some organizations opt for re-hosting (maintaining code, data, and user interfaces on a modern platform). Nevertheless, given the drive for digitalization, this scenario is not recommended because it tends to shift skills and functionality problems, rather than resolve them.

Combined, these factors will require that CIOs carefully assess the entire application portfolio to understand and prioritize existing modernization drivers. These drivers comprise functional, architectural, and technological criteria, as well as risk and cost considerations.

The modernization initiative should be aligned with the defined target state of the digitalization strategy.

A better way

Based on our experience, we have found that a method that Gartner calls Continuous Modernization tends to be the most efficient, fast, and accurate path.⁴ It combines the rapid, low-risk automated refactoring and migration of the application to an open-system environment with the ability to continuously modernize applications to the desired functionality and target architecture.

The distinguishing characteristic of refactoring is its use of technology to automatically convert 100 percent of the application's components. This includes all online batch modules and interface modules, as well as the database. Refactoring tools can also fully automate test-case creation and execution, providing a further advantage.

The strength of refactoring lies in its 1:1 functional mapping of the code to the new technology. For example, automated refactoring can precisely map data types,

as well as the arithmetic comparison operations, to help ensure that the resulting application behaves identically to its legacy predecessor. What's more, developers can easily compare legacy code with the transformed target application, with the potential to greatly minimize implementation risks.

The transformation

Once the code has been refactored and migrated to an open platform, the modernization phase can begin.

Following the Continuous Modernization approach, tasks are prioritized to create an efficient path toward modernization. This matters because approximately 30 percent of the legacy codebase is typically never changed. Freezing unchanging code in the target system can allow the modernization initiative to concentrate on the remaining 70 percent of the code base.

All modernization activities should be consistently aligned with the future-state target. Automatic or semi-automatic code-mining activities can be executed on the new platform to support and accelerate integration with the target architecture.

They can also help identify and resolve friction points, as well as help prioritize microservices for mobile applications. This platform can also be used to modernize business rule externalization, workflow management, data analytics, and cloud transformation.

The advantages of Continuous Modernization can be further enhanced by the adoption of a DevOps model. DevOps can help support agile development, continuous integration, continuous delivery, and continuous monitoring. But it's important to note that it must be backed by state-of-the-art toolchains. Modernization provides CIOs with an opportunity to introduce an agile culture that, in addition to agile front-end development, helps achieve a state of right-speed IT.⁵ Legacy system developers can be integrated into DevOps teams to share knowledge of and experience with the application.

**From the frontline:
Neither make nor buy: refactor and modernize**

Business situation: an international insurance company operated its core business applications on an aging mainframe. The system had been customized over many years to meet the insurer's specific business needs. It was stable and very reliable. However, the company was having difficulty finding qualified COBOL developers, who are retiring in waves. Another significant issue was operating costs that were increasing year over year.

A stalled attempt: the insurer, working with an external vendor, had launched a project to rewrite the core application. The project duration was estimated at three years, but seven years after starting, only 30 percent of the functionality had been rewritten in Java.

Solution and impact: the insurance company engaged Deloitte to help it design and implement a code-conversion initiative that could be rapidly implemented. Our fully automated Refactoring tool enabled a smooth go-live on a Linux platform after only two years. The company then decommissioned its mainframe and began to incrementally modernize the application for future needs.

Conclusion

Financial institutions cannot afford to base their digitalization strategy solely on front-end systems. Rather, they should carefully craft a digitalization strategy that also emphasizes back-end applications and is fully integrated in the modernization strategy.

To get there, each organization will need to carefully assess its current- and future-state systems, and identify and implement a singular modernization strategy. There are no shortcuts, and there is no universal approach. Issues such as a lack of talent, the speed of implementation,

and transformation risks will affect each business in unique ways.

A combination of refactoring and subsequent modernization phases can help organizations reap the advantages of a transformational project while eliminating corresponding disadvantages. Ultimately, this approach can also help CIOs prioritize the right investments to transform legacy applications into a sustainable platform. It can also help implement the requirements for functionality, architecture, and technology in an agile organization. ●

⁵ Deloitte LLP, "Right-speed IT: Living between black and white," 24 February 2016 <https://www2.deloitte.com/insights/us/en/focus/tech-trends/2016/devops-it-optimization-speed.html>