

+11,00.00

Still Playing Catch-Up with **Core Banking Technology?**

by Andrea Lorenzo Solèr and Ronald Roccioletti

Contents

<u>Introduction</u>	<u>1</u>
<u>It's a Costly Business</u>	<u>2</u>
<u>Spread the Word</u>	<u>3</u>
<u>HR and KPIs</u>	<u>4</u>
<u>10 Things to Avoid</u>	<u>5</u>

Introduction

A growing number of digital commentators seem to believe that the word **“bank”** is synonymous with **“IT museum.”**

Harsh?

Not surprising though, when you consider that the financial sector (in keeping with large governmental departments, insurers, healthcare providers and most Fortune 500 companies) depends on legacy systems which can be as much as 40 years old. After all, these core banking systems were built on COBOL, a near-forgotten language which predates Coke cans, computer mice and The Beatles.

That said, 220 billion lines of COBOL are still in use today. And because of the scarcity of people versed in the language, veteran engineers have been coaxed out of retirement to fix ancient banking system software at handsome hourly rates. Which is why there's a gathering movement to train a new generation of COBOL engineers to replace the former employees who built and maintained these core banking systems the first time around.

It's a Costly Business

Typically, banks spend around **80%** of their IT budgets on maintaining legacy systems. And with something like **\$3 trillion** flowing through them daily, replacing a core banking system is more of a must-do than a might-do. But where do you start? **What are the best practices and pitfalls?**

Make Sure You Know What You're Dealing with

Familiarize yourself with the new banking software's capabilities and, possibly, exchange legacy satellite systems where possible, weighing-up the pros and cons of monolithic and distributed systems (see our [other blog](#) on this topic). Very often, new standard banking software automatically enables functionalities which were located in those old satellite systems. It's all too easy to find yourself duplicating functionalities in a new system.

Make sure you clean up the data in your legacy systems. Migrating bad data to a new system will generate poor results, however advanced the banking software and technology are – garbage in, garbage out. Check if the data in the new system differs from the original data. You could find that the functionality is also different; something you might only realize at the end of the project and that can be expensive to put right before the new system goes live.

Testing is really important, but remember, when you test functionalities, keep to the same transaction each time because, otherwise, you may not catch all the bugs in the system.

Business Wise

Standard core banking software is widely used, but all too often, brokers, custodians and counterparty banks are set up incorrectly. Not enough emphasis is placed on counterparty alignment and, invariably, that's another problem only recognized at the end of a project.

Don't forget to involve all stakeholders as early as possible. Get them on board and committed to easing your journey to the project vision. However, if they're encouraging you to cram functionalities into the last couple of days before go-live, refuse politely and change the subject. And make sure the business doesn't use the old system too much in the days running up to go-live, in case you snag in-flight transactions.

Spread the Word

Core banking implementation usually triggers massive organizational change. Whether you're building a new business unit or a completely new business, or implementing a major regulatory update, you need to have a sound strategy for communicating the changes. You'll get a ton of questions like "How are you organizing the bank with the new software?", "What are the new processes?" and "What is the new target operating model?", because everyone needs to feel part of the new system and be aware of what's coming.

Often, you'll need to share post-implementation concerns as well. For instance, someone will flag up that some checks are missing in particular processes – "we're sending reports to the client without anybody keeping an eye on them" – and this proactive move will initiate a remedy.

People Need People

People are a key element in the success of the core banking implementation, so you need to make sure everyone is properly on board and no one is left behind. Today's managers are tomorrow's leaders and plans to facilitate that should be locked in. Promising young talent should get similar treatment. However, while the current project leader and project manager are focusing on the success of the project and future obstacles and opportunities, issues that have been left behind in the run-the-bank (RTB) area can easily be overlooked instead of overseen.

HR and KPIs

It should come as no surprise to find out that decision-makers want it both ways: to keep the business running, as well as being involved in the change project to ensure the new software capabilities match business requirements. In spite of that, the bank is often reluctant to commit adequate resources to the project. One way around this is to ask HR to make involvement part of business executive KPIs. For instance, if they take part in testing, it will have a positive impact on their KPI and encourage them to loosen the purse strings a little.

Better Together

It sounds obvious, but involving people who actually understand how both the systems and the business elements work is another smart move. Business people need to have an affinity with IT and vice versa – the stronger the better. It's particularly important for positions like stream leader, people writing business specifications, user acceptance test cases, training and so on. Over the years this has proved to be an integral part of successful core banking implementation and change acceptance. Often, the consulting companies have limited access to these key resources. Therefore, it's imperative to address this point at a very early stage and even before starting a new project.

10 Things to Avoid

As well as employing the positive implementation actions, here are some negatives you'd do well to avoid:

1. A temporary solution becoming permanent
2. Not spending enough time designing new business processes for the new banking software
3. Being too ambitious – allowing scope creep, or providing Hollywood solutions to problems with a small number of transactions
4. Not maintaining standard functionality
5. Allowing the number of change requests to rise
6. Allowing change requests even if not regulatory or requested by bank policy
7. Implementing weak change management
8. Establishing unrealistic timelines guaranteed to disappoint
9. Not accounting for the time it takes to adopt a new core banking system and processes, planning project phases like "stabilization phase," followed by "enhancement and new or second priority features phase"
10. Inadequate defect management (e.g., bugs, functional/change requests, performance issues, etc.). Not focusing on go-live elements or deprioritizing/moving to second phase

Sharpening the Competitive Edge

The plain fact is that while a bank is weighed down by monolithic legacy infrastructure, it cannot hope to compete against a bustling horde of agile, innovative and funky digital competitors. However, as technology advances reduce the cost of upgrading/replacing systems, more and more former naysayers will feel able to go with the flow.

This new clarity of vision will help banks recognize the futility of patching threadbare technology with rolls of \$100 bills, and the brilliance of implementing a new core banking system.

About the Authors



Andrea Lorenzo Solèr

Vendor Solutions Core Banking DACH Delivery Head, Luxoft
andrealorenzo.soler@excelian.com

Andrea Lorenzo Solèr gained an M.A. in Business Administration from the University of Zurich, Switzerland. Since 2006, he has been working in the financial sector in various Swiss and international projects as management consultant / business analyst. He has significant cross-functional expertise in consulting, banking, advisory, project management and change management. Andrea is fluent in English, German and Italian.



Ronald Roccioletti

Global Head Core Banking, Luxoft
ronald.roccioletti@excelian.com

Ronald Roccioletti studied Electrical Engineering at the Federal Institute of Technology in Zurich (Dipl. El. Ing. ETH), later completing a Master in Advanced Studies in Management, Technology and Economics (MAS MTEC). Subsequently, he chose to work in the financial industry, gaining valuable experience as a consultant, a team lead and a project lead. He speaks German, Italian, English, French and Spanish fluently.

www.luxoft.com

Luxoft, a DXC Technology Company (NYSE: DXC), is a digital strategy and software engineering firm providing bespoke technology solutions that drive business change for customers worldwide. Luxoft uses technology to enable business transformation, enhance customer experiences, and boost operational efficiency through its strategy, consulting, and engineering services. Luxoft combines a unique blend of engineering excellence and deep industry expertise, specializing in automotive, financial services, travel and hospitality, healthcare, life sciences, media and telecommunications.