

# Key Success Factors for Testing Core Banking Transformation

Viswanathan Sankaran  
Head, Banking Solutions Group  
[viswanathans@quinnox.com](mailto:viswanathans@quinnox.com)

# Introduction

In the early days of business technology, the banking industry computerized their back-office accounting requirements, later bringing that same advancement to their front office, covering functions such as teller transactions, as well as liability, asset, and trade products. These were isolated implementations at the branch level.

But core banking solutions and business technology evolved, combining the back office and front office functionalities and accounting, enabling to operate from a single platform and connect all branches in the corporate structure. Customers became customers of the bank rather than of branches. Implementation would predominantly cover retail banking and corporate banking business requirements.

However, with the advent of new technologies and a marked change in communication infrastructure, specialized applications, channels, and payment systems progressed. This enabled banks to go beyond internal requirements and provide customer-focused electronic services.

But, as with any technology product, banking solutions are also prone to obsolescence. Technology can also be weakened further with added layers of updates, more and more tweaking and frequent enhancements. In addition, the maturity of the core banking solution vendors improved, bringing in sweeping changes in their offering that facilitated:

- ▶ The introduction of competitive business products
- ▶ Cost reduction efforts (transaction cost, product cost, service cost)
- ▶ The critical need to meet stringent regulatory and compliance requirements
- ▶ A new focus on customer convenience and access to their banking needs

In simple terms, technology has become a huge driving factor for the banking industry. Today's financial institutions realize and acknowledge the need for changing their legacy core banking solution and other specialized peripheral systems to help meet evolving and more demanding customer needs.

The change can be an upgrade to a higher version from the existing vendor, or migrating to a new core banking solution, along with acquiring other specialized applications/products. In either case, it's vital that banks integrate new solutions with other peripheral systems, channels and interfaces. In essence, these efforts are not merely the implementation of a core banking solution, but a transformation interconnecting multiple systems and accesses.

Obviously such a transformation needs to undergo a rigorous testing to ensure that it:

- ▶ Is fit for the purpose so that it meets end-to-end business flow requirements
- ▶ Provides operational ease for data input and processing in the business operations environment
- ▶ Complies with the regulatory requirements

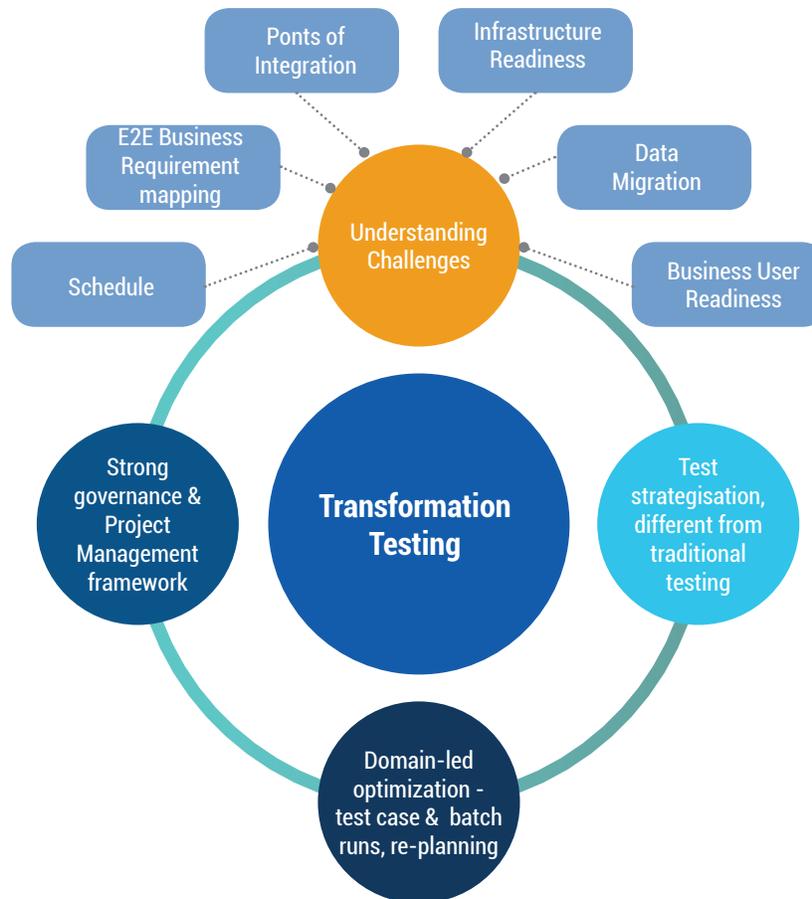
## Implementation Testing Vs. Transformation Testing

Core Banking Transformation Testing is an end-to-end business flow/functionality testing solution that covers the complex implementation of specialized commercial off-the-shelf products, channels, customization and integrations, along with retail banking and corporate banking applications.

The difference between implementation testing and testing of a complex transformation program is compared here.

Implementation Testing	Transformation Testing
Independent (Silo) testing of individual components of monolithic system	Testing across multiple systems, channels, interfaces
End-to-end application flow testing/product Lifecycle Testing	End-to-end business flows testing
Simple batch run schedule – simple impact	Complex batch run schedule – High impact
Negligible dependency	Inter-dependency on other systems – for flows, uptime, quick defect fix
Simple migration	Possibly complex multi-migrations

## Key Success Factors for Transformation Testing



## Challenges in Transformation Testing

Every project has its unique challenges. However, there are some common challenges to transformation testing that can be more easily overcome by approaching them from a different angle for effective planning. Here are some vital areas to consider, along with questions that may help provide a fresh perspective to these challenges:

**Scheduling:** Any transformation project can be very time consuming and require significant amounts of effort and resources. It may also be prone to delays for many reasons. However, compromising on the testing window and expecting to salvage other delays can adversely affect the outcome. Two rounds of testing followed by a regression testing ideal. Is it a catch 22 situation? Or is there a way to balance the right amount of time devoted to testing with your deployment deadlines?



**End-to-end business requirement mapping:** The business requirements from a heterogeneous group of specialist business users, operations, finance, risk and compliance have to be synthesised. They have to be interwoven according to business flows, cover multiple systems and set the stage for ensuring the fit purpose. This is a humongous task. Implied requirements at any point adds fuel to the fire and would also prove to be costly. Who can bridge the heterogeneous group or functions at each level?

**Points of integration:** Multiple systems and a host of integrations, internal as well as external, need to be unified. Deep functional and technical understanding and expertise is needed to identify where integration needs to intersect in order to have meaningful testing. The high level architecture of the transformation and a clearly spelled out interface register is essential to ensure that the interoperable solutions are working and reliable. So where are those critical integrations points?

**Infrastructure readiness:** Dedicated test environments are required for integration, user acceptance, migration, performance, security and infrastructure testing phases and training. Concurrent execution in an environment might conflict the respective purpose. If not planned at the initiation stage itself, this will hike the cost of transformation. Continuous monitoring is required to ensure availability at the right time and at the right place. So how do you contain cost without compromising on quality?

**Data migration:** Another area of concern is the possibility of multiple migrations from legacy systems. Technical data mapping should be carried out by the development team through code development. Functional verification, validation and reconciliation of static, semi-static and dynamic data of one-to-one, one-to-many and many-to-one migrations should be carried out by the testing team. Duplicating the coding of migration elements and migration rules by the testing team is best avoided. Yet, the development and testing teams need to be working in tandem and collaborating well. Then, how do you get to a point where their minds marry so that the needs of the bank are met?

**Business user readiness:** Legacy systems commonly have a loyalty among a longstanding IT culture, due to its prolonged used, a well-entrenched comfort level, and a tendency to resist change. Each application is developed differently, but a mind-set to compare legacy system with new technologies feature-by-feature can become an obstacle to new implementations. Loyalists flaying and discounting innovation can be a real challenge to successful testing. How do we break their mind-sets and make them live the change? How do we empower legacy hardliners to soften their stance, prepare for this change, and ultimately embrace it from an end-to-end business and application perspective?

## Testing Strategy: Different from Traditional Testing

A testing strategy is a vital component of every project, and should be a part of testing processes and templates. How does it vary from transformation testing? Non-functional testing may not make much difference in a transformation program. However, due to the complexities of testing, more stress is required on the functional testing level.

Hence, a testing strategy has to address how to bring together skilled resources to integrate the business flows, as well as how to direct and organize them to achieve a shared goal. This collaborative, coordinated and cooperative orchestration needs different skills than a traditional testing.

Test execution is like a relay race, where the baton is transferred to the next participant smoothly and quickly. Multiple testers are needed to test an end-to-end flow. This dependency on other testers necessitates a coordinated movement of test case execution among all parties. The baton change has to be well planned to ensure that the dependency does not slow down, impacting the testing window. It should also be remembered that when the business flow is tested, it does not always flow sequentially in one direction.

Further, the risks and dependencies have to be identified and brought to the attention of all stakeholders. Responsibilities for deliverables and reviews have to be defined.

## Domain-led Optimization

Optimization of test cases, efficient scheduling of “logical days,” and their constant planning and adjusting to meet the objectives of the plan during execution requires a high level of business acumen. Engagement of domain experts with an understanding of end-to-end functionalities can greatly drive these actions and benefit the program.

**Test cases:** Micro-level test cases prepared by different testers should be consolidated at macro-level to form the end-to-end flow. This should then be aligned to form the run (execution) plan. As testing is by sampling, optimization of test cases and ensuring coverage and end-to-end flow will help maintain the testing window. This can be achieved by controlled sampling based on driving parameter changes.

To ensure reuse, the methodology of your testware creation needs to ensure that test conditions, flow paths and data are separated, yet linked. This loose coupling allows for easier maintenance of overall testware.

**Logical days/batch runs:** All the systems under test have to be progressed to a certain period, say a year, so that occurrences of expected events in the lifecycle of various functions and business products are tested. Each desired working day (a “logical day” in testing parlance) should be simulated and tested over one or more calendar days for each logical day. Though not unique to core banking, execution of some of the test cases are possible or essential only after the end of day (EOD) batch is run (formal completion of a logical day). Thereafter the system should be moved to the next logical day.

For testing business flows across systems, there is a need for synchronized logical days across systems on specific calendar days. However, it may not be practicable due to certain system limitations, failure of one or more systems, or delayed defect fixes causing progress to stall. Further, the logical days may have to be rescheduled during test execution, due to exigencies.

The testing window can be defined by the number of logical days and associated varying calendar days required to execute all the planned test cases. This is also influenced by the time taken for batch runs. Optimizing the required logical days can help reduce a compressed test window. It is less of a science and more of an art, requiring a combination of domain and testing expertise.

**Re-planning:** Test cases should be planned for execution and a run (execution) plan produced, ordering the test cases according to calendar days and logical days, as well as assigning the responsibilities to testers and activities of other stakeholders. The execution of planned test cases may be affected for more than one reason:

- ▶ Delayed code drops
- ▶ Delayed fix for severity 1 and severity 2 defects
- ▶ Delays EOD batch runs
- ▶ Mismatched (asynchronous) logical days
- ▶ Scope creep or change
- ▶ Interface failures
- ▶ High downtime

The may have an impact on the schedule, scope (including deferral of requirements) or resources. In such a situation, there should be prompt changes and adjustments made to the run schedule or run (execution) plan.

## Strong Governance and Test Management Framework

Once domain expertise has been brought in and testing processes are implemented, a continuous monitoring and control over the final lap of the transformation plays a major role in the success of the program.

The framework for the governance and test management should encompass:

- ▶ Service execution
- ▶ Test management reviews
- ▶ Performance management review
- ▶ Strategy reviews

All these have to be defined at various levels of hierarchy between the independent testing vendor and the bank stakeholders. These tasks also need to be monitored and controlled at respective specified intervals, such as daily, weekly, monthly and quarterly.

### Confidentiality Note

This document contains confidential information developed by Quinnox, Inc. No part of this document may be reproduced, stored in retrieval form, adopted or transmitted in any form or by any means, electronic, mechanical, photographic, graphic, optic or otherwise, translated in any language or computer language, without prior written permission from Quinnox, Inc.

This document has been prepared in accordance with the accepted techniques for software engineering at Quinnox, Inc. The representations and related information contained in the document reflect our best understanding on the subject of this document.

However, Quinnox makes no representation or warranties with respect to the contents hereof and shall not be responsible for any loss or damage caused to the user by the direct or indirect use of this document and the accompanying software package. Furthermore Quinnox reserves the right to alter, modify or otherwise change in any manner the content hereof, without the obligation to notify any person of such revision or changes.

## Conclusion

Advancements in financial and business technologies, combined with greater customer demands and expectations are powerful driving forces that continue to challenge the banking industry. In order to stay competitive in an increasing crowded space, today's financial institutions must transform their legacy core banking technology and other specialized peripheral systems.

Such a transformation requires rigorous testing to ensure that end-to-end business flow requirements are met, operations continue to produce results, and compliance is met. Although testing is complex, the keys and recommendations in this white paper—along with the necessity to partner with a highly experienced testing solution provider—can help ensure both testing and core banking transformation success.

[enquiries@quinnox.com](mailto:enquiries@quinnox.com)

### Corporate Headquarters

The Wrigley Building  
400 North Michigan Avenue  
Suite 1300  
Chicago, IL 60611