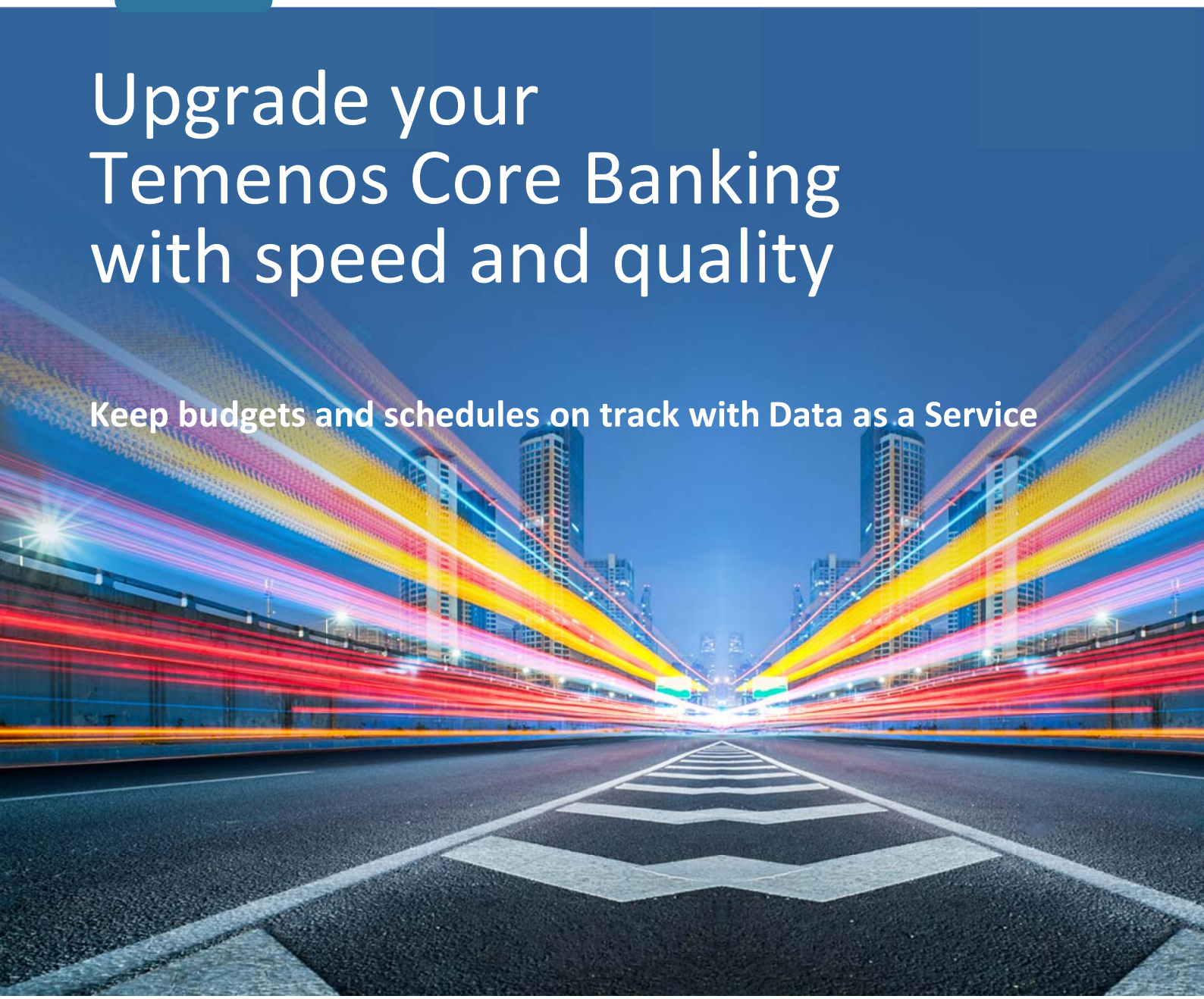
e-Book

Upgrade your Temenos Core Banking with speed and quality



Keep budgets and schedules on track with Data as a Service

Validata
New Thinking | Instant Answers

by ***Vaios Vaitsis***
Founder and CEO of Validata Group

Confidentiality Statement

This document is the copyright and the property of Validata Group and it contains Validata proprietary information. It is supplied in confidence and may not be used for any purpose other than for which it is supplied.

By accepting this Document the recipient agrees to keep permanently confidential all information which it contains. No part of this document shall be reproduced, published or disclosed to any third party without prior written permission of Validata Group.

Validata
New Thinking | Instant Answers

Table of Contents

Validata
New Thinking | Instant Answers

Introduction

The core banking system is the central processing unit of a bank. It is a customer accounting and transaction processing engine for high-volume back-office transactions. Upgrading it is definitely not an easy task. Upgrades can end up being really tough, and only one simple mistake can end up costing way more than expected.
A core banking system will undergo a series of upgrades over its lifecycle, with each upgrade consuming millions of dollars and months, or even years, of implementation time.

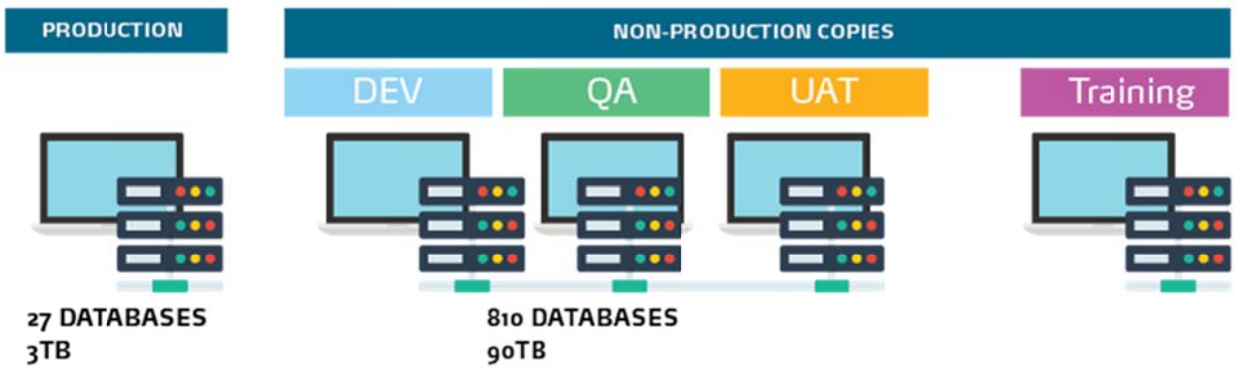
Too often, data stands in the way of executing those projects at speed. To roll out new products, business processes, and modules, professionals need data for development and testing environments but are operationally constrained by the slow, inefficient process of extracting, copying, and moving data. Legacy approaches can take days or weeks to deliver or refresh environments, making data the critical bottleneck to completing Temenos upgrade projects on time and on budget.

Validata
New Thinking | Instant Answers

Upgrades take too long

Extensive changes to packaged code – layered on top of customizations already introduced to the Temenos Core Banking system – mean that testing for an upgrade project rarely unfolds in a standardized, predictable manner.

- **Custom and Packaged Code Changes:** Temenos core banking implementations almost always include customizations that need to be rolled forward and tested thoroughly. The developers who created the custom code may no longer be available, making testing even more critical as new developers learn the details of the code they are migrating.
- **New Tuning Requirements:** As the code and underlying infrastructure of the Temenos system changes, previous performance tuning may no longer apply so it has to go through significant re-testing to ensure adequate response levels.
- **Changes in Data access**: As the data model of the Temenos system changes, applications may require new data objects fed from different systems. Data access from new sources drives considerable testing, to ensure both transaction and reporting accuracy.
- **Development and Testing Environments Provisioning:** Teams must approve, provision, and support additional dev/test environments, a process that can take weeks or months at many organizations.
- **Database Patch Management**: An upgraded Temenos system might require new patches for the underlying database. This requires additional testing of the database layer, with even more test cycles for the application to determine the root cause of errors during upgrade.



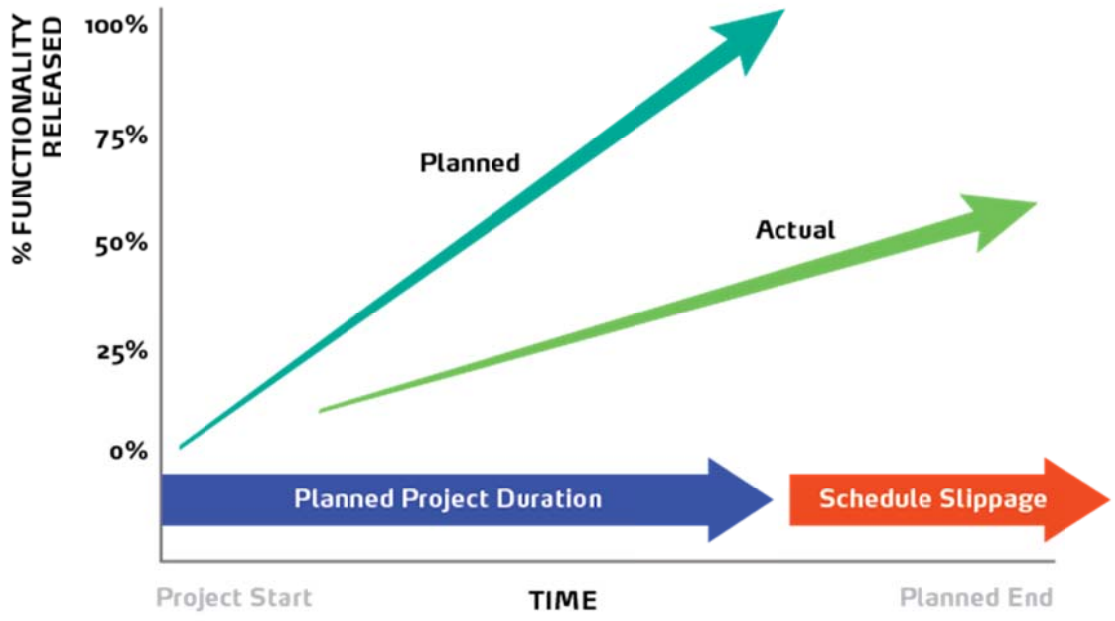
>Upgrade projects require expensive non-production environments, especially for testing

Validata
New Thinking | Instant Answers

Upgrades can cost a lot

Project delays and complexities drive costs up and result in Upgrade projects that run over budget and schedule, with some of the key contributing factors being the following:

- **Contractor Costs:** Expensive contractors and consultants, paid on a time and materials basis, sit idle while they wait for the hardware to be prepared, databases to be patched, and testing environments to be provisioned.
- **System Configuration:** Unexpected system configurations drive the need for additional QA environments, increasing operational and capital expenses.
- **Functionality Shortfalls:** Upgrades are sometimes only partially implemented due to schedule and cost overruns, delaying or even eliminating features and process improvements that could cut operational costs for the business.
- **Application Downtime:** Defects making their way into production result in costly downtimes and expensive rework efforts.



>Planned vs Actual variations of a typical Upgrade Project

Validata
New Thinking | Instant Answers

Integration and testing bottlenecks

Testing teams struggle to extract and deliver fresh, high-quality test data associated with each of the software modules that make up a given composite application. It is also nearly impossible for those teams to "synchronize" that test data: integration testing requires that data sourced from multiple repositories be synchronized to the exact same point in time.

Setting up an integrated test environment an excruciating process – and can extend projects by weeks. However, having a final integration state is worth the effort. Without this, development teams are forced to test different applications against limited copies of a few databases, and ultimately this increases the possibility that more defects will slip into production.

Validata
New Thinking | Instant Answers

End-to-end Financial Reconciliation

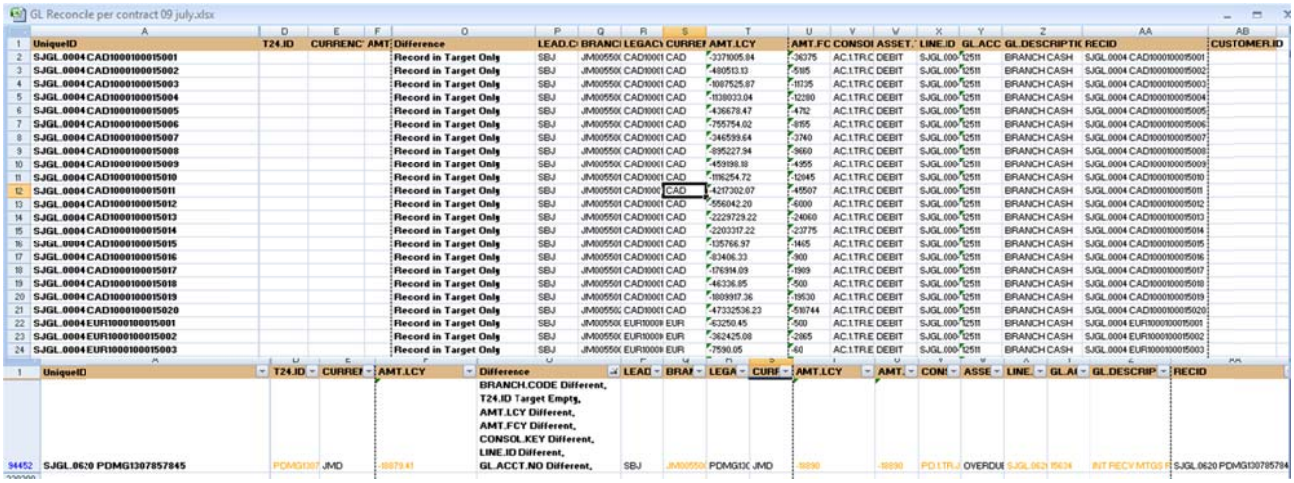
Both financial and nonfinancial data from the legacy system must be properly migrated to ensure data sanctity. With complex business rules and large volumes of data, reconciliation becomes an onerous task.
Reconciliation checks need to be performed to ensure there are no mismatches before going live. Once the target system goes live, it has to be monitored to gauge success and note any improvements required.

In terms of General Ledger (GL) system, the bank may be using:

i) built-in GL systems in the legacy / core banking application
ii) a separate GL Accounting system with the data feed from the core banking system
iii) routing the data feed from core banking system through an intermediary engine, where specific accounting entries are generated as per the country specific accounting guidelines and interfacing to GL system.

In all the cases, the existing and proposed GL accounting systems (source and target) are to be studied with reference to update of accounting entries, interest accruals, closing balances, P&L and A&L. The existing GL accounts are to be mapped to the target GL system and reconciliation template is to be designed to reflect pre and post upgrade GL balances. The accounting entries that take place during upgrade process pre and post are also to be reconciled.

Example of financial reconciliation reports:



Validata
New Thinking | Instant Answers

GL Reconciliation per Contract report can be used by the Bank to prove how migration was done from a financial point of view. It shows if there are any differences and the reason why these differences were accepted. In details, it shows:

- each contract/account in which GL Line is reported – new GL Line and Legacy ones ;
- each company and branch under which this contact is reported - new and Legacy ones;
- original currency of the contract/ account- in new system and Legacy ;
- amount of the contract/ account- in original currency and local – in new system and Legacy;
- CRF key from T24- it's in order Bank to know where is reported this contract /account;
- Asset type - debit or credit as in different system asset type is with different sign

E6 0

	A	B	C	D	E	F	G	H
1	LINE.ID	AMT.LCY			AMT.LCY		AMT.LCY	
2		T24 R14	R07	R7 GL No's	UBS	UBS GL No's	R07+UBS	Difference Column B - Column G
3	SJPL.0015	56,368,345.34	56,368,345.34	SJPL.0015|RBTTJMPL.002	0.00		56,368,345.34	0.00
4	SJPL.0025	11,522,237.42	11,522,237.42	SJPL.0025|RBTTJMPL.003	0.00		11,522,237.42	0.00
5	SJPL.0030	5,517,439.13	5,517,439.13	SJPL.0030|RBTTJMPL.001	0.00		5,517,439.13	0.00
6	SJPL.0031	183,022.66	183,022.66		0.00		183,022.66	0.00
7	SJPL.0040	742,754.60	742,754.60		0.00		742,754.60	0.00
8	SJPL.0084	299,083.42	299,083.42	SJPL.0084|RBTTJMPL.009	0.00		299,083.42	0.00
9	SJPL.0085	116,640,350.37	116,640,350.37	SJPL.0085|RBTTJMPL.008	23,298,063.98	SJPL.0085|IN2001211	139,938,414.35	-23,298,063.98
10	SJPL.0086	1,099,490,230.42	1,099,490,230.42		0.00		1,099,490,230.42	0.00
11	SJPL.0088	484,838.93	484,838.93		0.00	SJPL.0088|IN2001225	484,838.93	0.00
12	SJPL.0090	31,021,302.28	31,021,302.28		306,702.32	SJPL.0090|IN2001010	31,328,004.60	-306,702.32
13	SJPL.0092	143,920,759.85	143,920,759.85	SJPL.0092|RBTTJMPL.009	0.00		143,920,759.85	0.00
14	SJPL.0094	0.00	0.00		295,619.36	SJPL.0094|IN2001210	295,619.36	-295,619.36
15	SJPL.0096	0.00	0.00		82,978,137.27	SJPL.0096|IN2008105	82,978,137.27	-82,978,137.27
16	SJPL.0100	20,000,808.30	20,000,808.30	SJPL.0100|RBTTJMPL.012	0.00	SJPL.0100|IN2001216	20,000,808.30	0.00
17	SJPL.0104	0.00	0.00		347,345,025.06		347,345,025.06	-347,345,025.06
18	SJPL.0115	0.00	0.00		0.00	SJPL.0115|EX8202215	0.00	0.00

Validata
New Thinking | Instant Answers

Data as a Service Reduces Upgrade Costs and Complexities

Data as a Service (DaaS) eliminates significant infrastructure costs for core banking upgrades and delivers data to project teams faster, while significantly accelerates the pace of development and testing. It addresses the need for better data orchestration, delivering application data with the scalability, automation, and self-service capabilities demanded by fast-moving project teams. It enables organisations to move away from legacy systems and practices, helping them cut application project schedules in half and reduce infrastructure costs.

DaaS bring the benefits of virtualization to upgrade projects by:

- Capturing application data—including ongoing changes—in production systems
- Versioning and managing data across the full application lifecycle
- Delivering virtual copies of data to non-production systems, including test environments for upgrade projects

DaaS will deliver three main benefits to banks that are planning a Temenos core banking upgrade:

- **Increased productivity:** DaaS solutions deliver full test environments in minutes, so project teams can work instead of sit and wait.
- **Reduced Costs:** DaaS decreases the expense of intermediate storage for upgrade projects. Instead of duplicating the physical storage environment of a production system, DaaS solutions share common data blocks across non-production environments.
- **Optimised Risks:** Better testing ensures user acceptance of the upgraded system, with higher quality and functionality. DaaS provides an easy means for recovering old data whenever needed.

Validata
New Thinking | Instant Answers

Less Defects, Faster Upgrades

DaaS provides functionality that speeds application projects, including Temenos Core Banking upgrades:

- **Self-Service Creation of Dev/Test Environments:** With minimal storage costs or process delays required to create virtual data copies, teams can create as many test environments as needed.
- **Avoid Recode with Early Error Detection:** With the ability to quickly deliver refreshed test environments, teams can do more testing earlier so they can identify bugs when they are less expensive to remedy.
- **Easier Database Patch Testing:** By making virtual copies of databases, test teams can easily apply patches and test effects before migrating patched databases to the target system.
- **Database Rollback and forward:** By making multiple virtual database copies at different points in time allows better testing and root-cause analysis.
- **Team Sharing and Collaboration:** Leveraging DaaS, developers can easily share copies of a database for unit testing, integration testing, certification, or cutover.
- **Multi-Source Integrated Provisioning:** With DaaS, developers and testers can provision multiple environments from different sources as of the same point in time, allowing effective and repeatable tests of code integration across applications.

About Validata Group

Validata Group is the leader in **Continuous Testing, Legacy Migrations and Release Automation**, helping Temenos clients **accelerate application delivery**, optimize business risks and reduce costs.

Until now organisations have been forced to bolt together a number of specialised tools from different suppliers in order to achieve anything close to Application Lifecycle Management (ALM). Validata's Agile ALM platform is a robust solution built from the ground up, to bring together requirements, testing, defects, planning, resources, development and deployment, capable of integrating with external systems such as HP QC, CA Clarity, IBM RTM, Atlassian Jira etc, delivering a **'single version of the truth'** and enabling **continuous delivery, integration and monitoring**.

Validata's **model-based approach** enables the automatic generation of test cases and eliminates the high maintenance costs of traditional script-based test automation. By **automating the impact analysis** process and through the automatically generated scripts, it focuses your efforts on what to test and what scenario to test it with.

For more information

visit www.validata-software.com

or call us at +44 020 7698 2731

Validata
New Thinking | Instant Answers