



Re-engineering the bank: From programs to continuous delivery



Key takeaways

- Bankers have an ambition to be digital and AI-powered. You can hear it in how they talk about themselves on earnings calls, with mentions of artificial intelligence more than tripling from the year before.
- But to realize that ambition demands re-engineering operating patterns within the bank around a delivery model that focuses on outcomes instead of programs.
- This new delivery model requires executives to make four structural changes: Commit to change the business, not only technology; restructure the budget to fund value; redesign governance for speed; and locate engineers within their assigned business domain.

Every major bank is pursuing similar ambitions: faster time to market, real-time decisioning, AI-native operations. The strategic intent is modernization and flexible operations, and it's usually well-funded. The friction, however, is in the delivery model.

As so many transformation-weary executives know, assumptions made about system requirements, competitors, and business needs at the kickoff are usually out of date when final delivery occurs, often years later. Doing business today demands what we call a continuous delivery model, rather than episodic technology releases.

Continuous delivery model of banking transformation

**Organize around outcomes,
rather than systems**

**Govern progress by business value,
instead of milestones**

Moving your organization from old ways of executing a modernization plan to new ways—what we call moving from programs to continuous delivery—requires four structural changes. These changes take aim at how banks organize, fund, govern, and staff their plans for re-engineering and transformation so they achieve the improvements that are needed to thrive in today's environment.

Four operational mind-shifts bank executives should be considering

1 Commit to change the business, not just technology. Measure transformation by what changes for customers, risk, and operations—not just by what ships from engineering.	2 Restructure the budget to fund value, instead of programs. Move from multiyear programs to domain-level funding tied to quarterly outcomes to speed up the cadence of change.
3 Redesign governance for transformational speed. Embed control partners within delivery cycles to accelerate continuous delivery rather than positioning them as approval gates at the end.	4 Locate engineering within the business domain. Place product, risk, operations, and engineering in shared teams organized around business domains—rather than in separate towers connected by handoffs.

The onus is on the executive committee; are they committed to making these changes?

- For the CFO, it is whether the budget model enables or constrains value.
- For the CRO and control partners, it is how governance operates at the speed of continuous delivery.
- For the CTO, it is what the delivery model looks like in practice—how engineering, product, and data teams work within domains to produce measurable progress.

The next three sections build the case for why these changes are necessary and then we look at what these shifts look like in practice.

Table of contents

01	The case for change: Digital ambitions	04
02	The case for change: A volatile operating environment	06
03	The case for change: Bank modernization mechanisms are too slow	07
04	Four leadership decisions to enable a continuous delivery model	09
05	The continuous delivery model in practice	13
06	The cost of waiting	17



01

The case for change: Digital ambitions

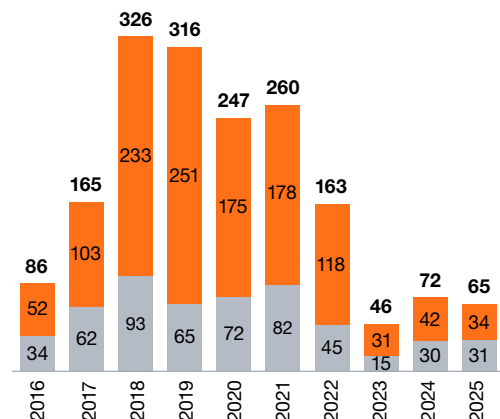
The changing language banks use to describe themselves publicly reflects their new digital ambitions. Over the past decade, “digital transformation” dominated executive commentary as institutions focused on rebuilding platforms and channels. Recently, there’s a sharp rise in the use of “artificial intelligence,” which has more than tripled in earnings call mentions compared to the prior year.

The industry narrative is moving from constructing systems to delivering intelligence; from developing channels and systems to building intelligent, controlled, real-time operating capability.

Bank leadership language reflects a shift from digital front ends to platform modernization to real-time, AI-enabled outcomes

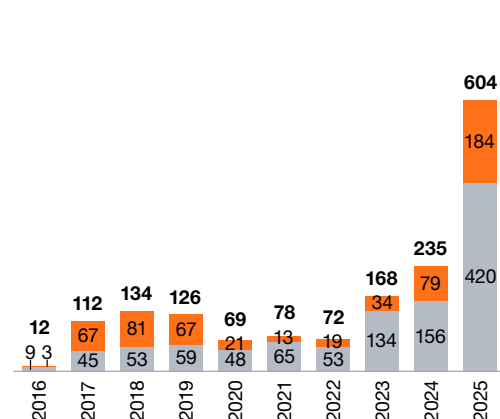
Digital transformation

Sum of mentions over time



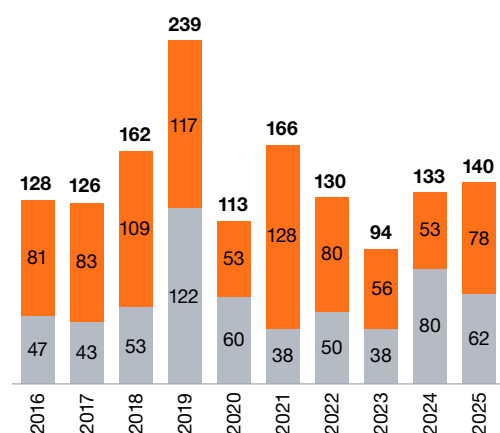
Artificial intelligence

Sum of mentions over time



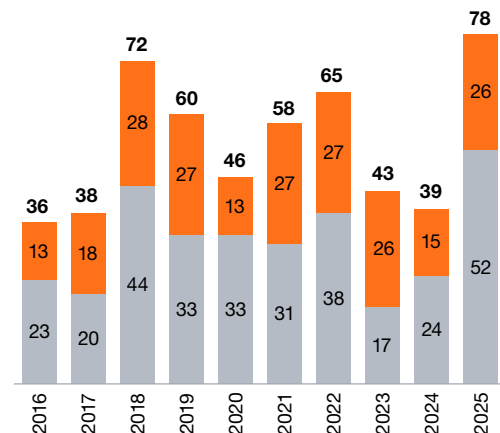
Real time

Sum of hits/mentions over time



Personalization

Sum of mentions over time



Regionals and super regionals Large bank

Source: PwC analysis. Selected Q1–Q4 earnings calls and conference transcripts for the years 2016–2025, inclusive, via AlphaSense, Inc. Methodology: AlphaSense searches were run across 2016–2025 earnings and conference transcripts for selected US large banks and regional/super-regional banks. Results show the sum of AlphaSense transcript “hits” and should be read as directional discussion intensity, not as a measure of investment, implementation, or business impact.

- Digital transformation, Artificial intelligence, and Real time used open AlphaSense keyword searches, which may include related variants, synonyms, word forms, and contextual matches.
- Personalization used a fixed exact-term search set: personalization, personalized, tailored, bespoke, customized, customer-specific, relevant offers, personalized insights, individualized, hyper-personalized, and personalizes.

AI reshapes not only back-office operations but the customer interface itself. Generative AI as a channel—conversational interfaces, intelligent advisors, and agent-driven workflows—is moving out of the novelty phase and going mainstream. Said differently, banks are no longer just building with AI; they are presenting themselves through it.



02

The case for change: A volatile operating environment

Transformation plans that once assumed relative stability over multiyear horizons now occur in an environment where key assumptions underlying the program can change within a single planning cycle. Five-year roadmaps typically can't survive amid the changes that happen within a twelve-month time span.

Multiyear planning doesn't work in a volatile banking environment

Regulations

Expectations shift based on changing government personnel, shifting macroeconomics, new sanctions policies, or emerging sector risks.

Competition

New entrants can come from anywhere. They could be rival financial services firms, FS-adjacent, fintechs, or non-FS companies.

Customers

Habits and behaviors change as tech advances across the economy, altering expectations of what's possible and how service is delivered.

AI

The advances in models, cybersecurity risks, and new financial services offerings are being measured in weeks to months, not years.



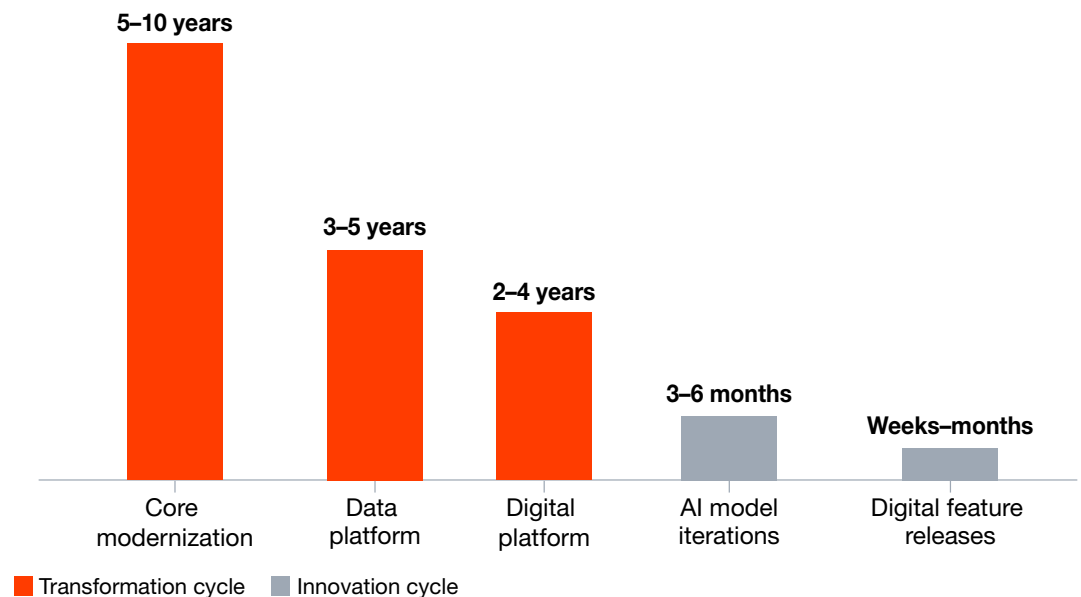
03

The case for change: Bank modernization mechanisms are too slow

While ambition accelerates and the environment becomes more dynamic, what's lagging behind are the mechanisms banks rely on to deliver technology to realize their strategic vision.

Transformation mainly still happens through large, multiyear programs—replacing platforms, consolidating systems, or migrating environments in pursuit of a defined future state. These programs can last up to five years from planning to completion. That's far too long considering the pace of innovation. What executives are realizing is that the gap between the cycles of innovation and transformation need to narrow.

Innovation cycles move faster than the timeline for typical bank transformation projects



Source: PwC. For illustrative purposes only.

Closing the gap between transformation and innovation cycles requires discretionary spending capacity—but technology budgets are structured to prevent that. Non-discretionary spend dominates: maintaining the existing estate, meeting regulatory mandates, servicing technical debt.

A reliable way to create discretionary spending capacity is to retire material complexity. But the program operating model rarely prioritizes decommissioning. It adds; it replaces; it rarely removes.

Such unreformed complexity generates its own costs in the form of fragmented data producing errors, redundant systems requiring parallel maintenance, or integrations with legacy systems that require manual workarounds. When banks layer AI onto all this without simplifying it, they add operational fragility and non-discretionary spend, further eroding the capacity they need to affect change.



04

Four leadership decisions to enable a continuous delivery model

In light of the operating environment and the current mechanisms of transformation, we see the following considerations for bank executives not as implementation steps, but as shifts in leadership focus that enable a continuous delivery model. These are structural decisions—each owned by a different part of the leadership team—that help your bank’s operational speed get in sync with your ambitious strategic plans.

The executive suite’s mandates in a continuous delivery model

CEO

A continuous delivery operating model requires that every investment be defined by what changes for the customer, for the risk profile, or for operational performance. And that must be defined at the top of the house.

CFO

Fund business domains rather than projects. Each domain receives a capacity envelope and demonstrates outcomes quarterly. Outcomes inform the next cycle’s investment. And every cycle should decommission to free financial capacity.

CRO and CCO

In today’s environment where new technology impacts the entire enterprise, and where vulnerabilities affect the business, the controls function should be a critical partner during transformation and should be shaping it in flight.

CTO and CIO

Continuous delivery requires domain teams where engineers are part of the business rather than temporary teams. The goal is not more engineers but better-placed engineers who are empowered with direct ownership of business outcomes.

Executive committee and business line leaders Commit to change in the business, not just the technology

The most common failure in transformation is delivering technology without changing the business that operates it. A new platform doesn't automatically mean a new outcome. A migrated system is not necessarily a simplified process.

A continuous delivery operating model requires that every investment be defined by what changes for the customer, for the risk profile, or for operational performance—not just by what ships from engineering. When we say “value,” we mean:

- **Transformation targets are expressed as business metrics.** Faster onboarding, improved pricing precision, reduced losses, and better customer engagement can be key metrics.
- **Business leads co-own delivery.** Product, risk, and operations are accountable alongside IT—rather than waiting to receive output from programmers and engineers.
- **Redefine what accomplishments look like.** ‘Done’ is defined by adoption and by measured impact, not by deployment.

A payments business seeking to modernize its rails doesn't deliver a ‘new payments platform.’ It delivers faster settlement, lower exception rates, expanded reach, and measures success by transaction throughput, cost-per-payment and client adoption of real-time capabilities.

CFO and finance organizations Restructure the budget to fund value, not programs

Technology budgets in banking are increasingly constrained by non-discretionary spend. Items such as maintaining existing systems, meeting regulatory mandates, and servicing accumulated technical debt consume much of available budget capacity. The problem is especially acute at mid-sized institutions which have less room to invest in simplification or innovation compared with larger banks.

The traditional budget model reinforces non-discretionary spend: multiyear program envelopes allocating capital at initiation, measuring progress by spend rate, and resisting reallocation when conditions change. The CFO governs cost categories—“run” and “build”—instead of governing value trajectories.

Continuous delivery requires a different funding model.

- **Domain-level allocation.** Fund business domains (e.g. deposits, lending, payments) rather than projects. Each domain receives a capacity envelope and demonstrates outcomes quarterly.
- **Quarterly reprioritization.** Outcomes from the prior cycle inform the next cycle's investment—not a static three-year plan written before the first learning was captured.
- **Retire to reinvest.** Every cycle should decommission something—a redundant interface, a legacy integration, an unused capability. Retirement isn't optional; it is how discretionary capacity is created. The banks that execute this well have spoken publicly about relentless decommissioning. Their operating and financial results suggest that this approach compounds savings over time.

CRO, CCO, and control partners

Redesign governance for speed

Control functions—risk, compliance, audit—are currently structured to review completed work. Governance boards convene monthly or quarterly, assessing changes after the fact and creating approval gates that can extend delivery timelines. This cadence is insufficient.

In an environment where new technology has widespread impact on the enterprise, and where vulnerabilities also affect the business, governance is quickly becoming a critical partner during transformation and should be shaping it in flight.

Continuous delivery requires that governance operates in real time.

- **Embed control partners within delivery cycles.** Risk and compliance participate in sprint-level decisions, not end-stage reviews. This requires upskilling control partners to understand engineering workflows—code, infrastructure, data pipelines—so they can assess change when and where it happens.
- **Automate the repeatable.** Procedural controls—code documentation, policy checks, configuration validation—are executed by AI agents and automated tooling. Human judgment is reserved for novel risk, rather than routine compliance.
- **Govern for outcomes.** A quarterly business review replaces the steering committee. Business leads, technology, risk, and finance jointly assess what each domain delivered, at what cost, and what the next cycle targets.

The CRO's challenge is not about ceding control. It's about whether governance designed for annual cycles protects an institution operating on a quarterly cadence.

CTO and CIO

Locate engineers within the business domain

In many banks, engineering is organized by function—a platform team, a data team, a channels team—and connected to the business through required handoffs and ticketing systems. High-value work passes through multiple teams before reaching production.

Continuous delivery requires a different structure:

- **Create domain teams.** Engineers, product managers, data scientists, and control partners form stable teams around business domains rather than temporary project teams assembled for a project and disbanded when it's over.
- **Rethink what goes offshore.** Maintenance, monitoring, and operational support can be distributed. Re-engineering and innovation cannot be distributed without accepting significant velocity loss from time zone, culture, and communication distance.
- **Resize for ownership.** The goal is not more engineers but better-placed engineers who are empowered with direct ownership of business outcomes.

The CTO's decision is structural, focusing on reporting lines, team composition, and geographic placement. How those teams actually deliver day-to-day is described below.



05

The continuous delivery model in practice

The four shifts above establish the conditions. Let's now examine the mechanics: how engineering, product, and data teams actually interact under a continuous delivery model.

Three concepts define the continuous delivery model in practice

Domain-oriented planning

Each domain is a coherent stack where experience, decisioning, product logic, and data are owned by a persistent team and improved incrementally. It changes who owns what and how budgets flow.

Embedded engineering

In a domain-oriented model, engineering operates inside the problem, sharing context on observed problems with product managers, risk specialists, and operations leads through co-located work.

"Value drops"

A "value drop" is unit of progress in a continuous delivery model where a clearly defined improvement is typically delivered in a single quarter that produces a measurable business outcome.

Plan by domain, not system

Traditional transformation organizes work around systems: replace the core, migrate the warehouse, rebuild the channel. But customers and risks do not operate by system boundaries.

Under continuous delivery, work is organized around business domains, where customer journeys, decisions, and products intersect. Deposits. Lending. Payments. Onboarding. Each domain becomes a coherent stack: experience, decisioning, product logic, and data are owned by a persistent team and improved incrementally.

This has two practical consequences:

- **Dependencies shrink.** A domain team controls its own stack. It does not wait for a platform team to deliver infrastructure, a data team to produce feeds, or a channel team to expose capabilities. Coordination happens within the team, not between programs.
- **Improvements compound.** Each cycle strengthens the domain's architecture. Shared capabilities emerge, such as a pricing engine, an orchestration layer, a decisioning service, and that reduces the cost and effort of the next improvement.

This is not cosmetic. It changes who owns what, how budgets flow, and what 'done' means for engineering.

Embedded engineering

With domain teams in place, the question becomes: how do they actually work?

In the traditional model, engineering receives requirements, builds to those specifications, and delivers months later. Feedback arrives too late to shape the work. In a domain-oriented engineering model, engineers operate inside the problem: product managers, risk specialists, operations leads, and engineers share context daily through shared work on observed problems.

What changes in practice:

- **Engineers shape scope, not just execute it.** The team identifies the highest-value improvement in the domain, defines the minimum capability required, and builds it informed by direct exposure to operational data, customer behavior, and control requirements. This is occurring because AI compresses engineering—code generation, testing, refactoring—from hours to seconds. But engineers are skilled at applying judgment to imprecise requirements by interpreting ambiguity and filling gaps. AI cannot safely do the same. As engineering velocity becomes commoditized, the metrics that matter shift to clarity around requirements, decision latency, and complete specifications.
- **Learning cycles collapse.** When the team that built a capability also observes its performance, the next improvement starts from evidence, not from refreshed requirements six months later.
- **Control partners participate in real time.** Risk and compliance are in the room as design decisions are made, not reviewing them after the fact.
- **Product management becomes orchestration.** As organizational boundaries between P&L owners, operations, and engineering blur, product managers emerge as the connective tissue orchestrating across pricing, operations, and engineering to translate business intent into deliverable scope.

Success is measured by outcomes: faster onboarding, improved pricing precision, reduced losses, better customer engagement.

“Value drops”

A “value drop” is a clearly defined improvement, typically delivered in a single quarter that produces a measurable business outcome and strengthens the architecture for what comes next.

Value drops are not projects in miniature. And they aren’t sprints within a larger program. They are the unit of progress in a continuous delivery model.

Each value drop:

- Delivers a clear business outcome (not a technical milestone)
- Creates or extends a reusable capability
- Reduces the effort required for the next improvement
- Retires or simplifies something in the existing estate

Getting started: Value drop cycle in a single domain

Weeks	Activity
1–2	Define the outcome, ownership, and scope. What business metric improves? Who owns adoption? What is explicitly out of scope?
3–6	Build or adapt the capability required. This may be net-new, or it may be refactoring an existing service into a reusable layer.
7–10	Integrate with operational systems and data. Connect the capability to live environments, validate with real inputs, stress-test with control partners.
11–13	Deploy, measure, and capture reusable patterns. Record what was built, what was retired, and what the next cycle should target.

Real-life examples:

A deposit franchise seeking faster response to market conditions introduces a dynamic pricing capability that integrates market signals, customer behavior, and liquidity conditions. Instead of embedding pricing logic in multiple product systems, this capability operates as a shared decision layer accessible to both legacy and modern systems. The outcome measured is not ‘system delivered’ but margin improvement and time-to-market for rate changes.

An onboarding improvement takes the form of a shared orchestration capability that coordinates identity verification, document capture, credit checks, and approvals. This layer interacts with existing systems through defined interfaces, without requiring full platform replacement. The outcome measured is time-to-account and drop-off rate—not project completion.

The compounding effect:

Each value drop delivers measurable benefit. Each also creates infrastructure that can be reused. Over time, reuse compounds value, meaning the third value drop in a domain is faster and cheaper than the first because it builds upon shared capabilities.

Transformation becomes a pattern of progress rather than a single, coordinated leap.



06

The cost of waiting

The gap between your bank's ambition and engineering delivery isn't static. The gap compounds over time. Every quarter your bank operates under the old model means that complexity is likely to grow, capacity shrink, and more nimble competitors move further ahead.

The institutions that commit to continuous delivery are not merely getting ahead of the competition. Their operational gears are moving faster as each cycle builds reusable capability that reduces the cost of the next cycle. Banks employing a continuous delivery model are creating the conditions for significant improvement.



Contact us

Dan Goerlich

Partner, Banking & Capital Markets Leader, PwC US

Ryan Battles,

Partner, PwC US

Chris O'Hara

Partner, Data & Analytics, PwC US