

THE MODERNIZATION EQUATION

The case for progressive, in-situ modernization in banking

DXC Public

THE MODERNIZATION EQUATION

Maximize business value while minimizing risk. This paper sets out how financial enterprises can transform mission-critical cores in place, optimizing processing and code through blueprinting, accelerating time to market with AI across the software development life cycle (SDLC), building API-led frictionless journeys, and augmenting the core by externalizing functions into modern platforms.

Guidance for CIOs, CTOs and enterprise architects in banking

Contents

Executive summary	1
The Modernization Equation	1
The case for keeping and transforming the core	3
Lever 1: Optimize through blueprinting	4
Lever 2: Accelerate time to market with AI in the SDLC	5
Lever 3: API-led frictionless journeys	6
Lever 4: Augment the core by externalizing functions	7
The DXC approach in practice	8
Proof and outcomes	9
A phased roadmap	9
Conclusion: Solving the equation	10

EXECUTIVE SUMMARY

For most banks, the systems that matter most still run on the mainframe, and they do so with a level of resilience, throughput and security that is difficult to replicate. Yet these same platforms are under pressure: a retiring COBOL and RPG workforce, rising run costs, and business demands for digital experiences the core was never designed to deliver directly.

The instinct to rip and replace is understandable, but dangerous. Big-bang core replacement is a multiyear, high-risk program with a binary outcome. **There is a better equation.**

DXC Technology advocates **progressive, in-situ modernization:** Transform the estate incrementally, keeping the business running while the future is built beneath it, so each milestone delivers measurable value before the next begins.

THE MODERNIZATION EQUATION

Every modernization decision is, at heart, a trade of value against risk. The reason so many core programs disappoint is that they attack the modern capability while ignoring, or even inflating, the foundational architecture. Progressive, in-situ modernization is the discipline of improving both, at once.

THE MODERNIZATION EQUATION

$$\text{Business Value} = \frac{[\text{Optimized Core} + \text{AI-Accelerated Delivery} + \text{API-Led Experience} + \text{Externalized Functions}]}{\text{Risk}}$$

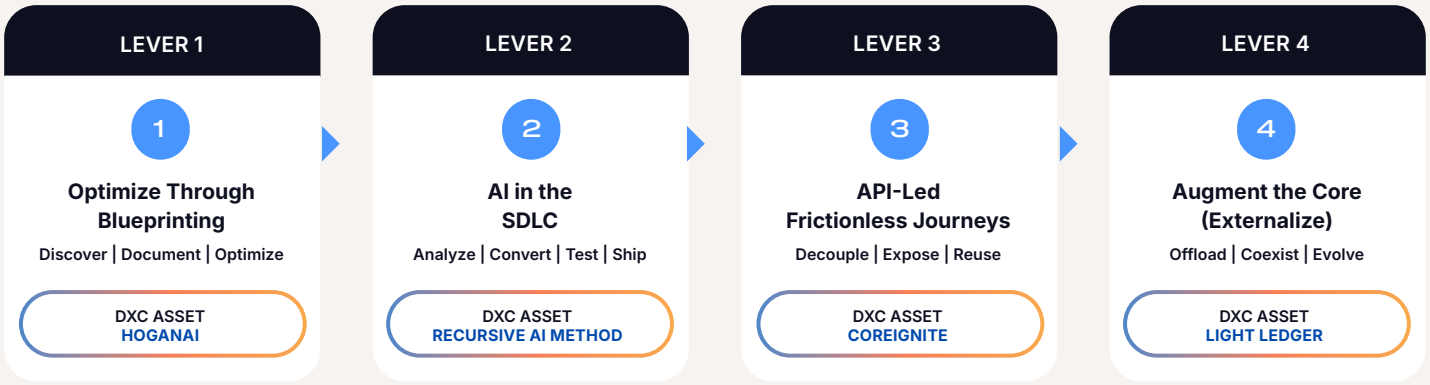
In-situ, progressive delivery raises the numerator while systematically driving down the denominator.

The equation, in four levers

The Modernization Equation comprises four levers that change the risk/reward math. Delivered as modernization as a service (MaaS) — outcome-based, phased and commercially structured — the four levers turn modernization from a cost of keeping the lights on into a driver of new revenue.

1. **Optimize through blueprinting.** Reverse-engineer and map the core with HoganAI to establish a factual baseline, remove dead code and target MIPS and cost.
2. **Accelerate time to market with AI in the SDLC.** Apply the DXC Recursive AI Method and HoganAI to compress analysis, conversion and testing from years to quarters.
3. **Build component-led frictionless journeys.** Expose core capabilities as governed APIs and use DXC CoreIgnite to accelerate fintech and digital-asset adoption.
4. **Augment the core by externalizing functions.** Progressively move ledger, accounting and settlement onto modern platforms such as DXC's Light Ledger, HoganX.

The Modernization Equation: Four levers and their DXC assets



The business stays operational while the future is built beneath it.
Delivered as outcome-based modernization as a service (MaaS) — value during every quarter, risk contained at every step

Why in-situ, progressive modernization?

The mainframe still matters

IBM Z and IBM i remain the backbone of regulated industries because they excel where it counts: transaction integrity, throughput, availability and security. They host the systems of record for deposits, cards, policies, claims and citizen services. The goal is not to escape these platforms but to unlock the value trapped inside them — the decades of proven business logic that competitors cannot easily reproduce.

Big-bang modernization fails

Full replacement concentrates all risk at a single future cutover. Requirements drift, business logic is under-documented, and the "new" platform must reach functional parity with a system refined over 30+ years before it can carry a single live transaction. Budgets are consumed long before value is realized, and the binary outcome means a slip becomes a crisis. In summary:

- Multiyear timelines are detached from business milestones.
- Under-documented logic is reimplemented from assumptions, not evidence.
- Parallel run and cutover risk is concentrated at the end.
- Value is deferred until the very last phase.

The in-situ, progressive principle

In-situ modernization keeps the core authoritative while the estate is transformed around and beneath it. Work is sequenced by business domain; each increment is proven for functional equivalence and put into production before the next begins. This is the principle behind DXC's phased core transformation: **The business stays operational while the future is being built beneath it.**

THE CASE FOR KEEPING AND TRANSFORMING THE CORE

IBM Z and IBM i in the modern estate

The modern estate is hybrid by design: The core provides the systems of record while cloud-native platforms provide systems of engagement and differentiation. The mainframe is not a silo to be tolerated but a high-performance service provider to be integrated — exposing its capabilities through APIs and events so the rest of the enterprise can consume them safely.

The skills and knowledge cliff

More than 60% of COBOL developers are over 55. As employees with institutional knowledge retire, systems become harder to document, change and defend. Organizations that delay do not just accumulate technical debt; they lose the people who understand what the code actually does. **Capturing that knowledge as evidence now is itself a risk-reduction strategy.**

The cost of doing nothing

Inaction carries compounding costs: rising MIPS and run charges, slower time to market as change becomes riskier, and mounting exposure as the skills base thins. The question is not whether to modernize, but how to do so without destabilizing the systems the business depends on every second.

200+	370,000	5,600	99.5%
MAINFRAME MIGRATIONS	MIPS UNDER MANAGEMENT	MAINFRAME PROFESSIONALS	MODERNIZATION SUCCESS RATE

DXC provides scale as one of the world's largest independent mainframe services providers.

LEVER 1: OPTIMIZE THROUGH BLUEPRINTING

DISCOVER | DOCUMENT | OPTIMIZE

You cannot safely modernize what you do not understand or observe. Blueprinting establishes the factual baseline — the specifications, dependencies and business logic actually embodied in the core — and turns tribal knowledge into evidence. This is where **HoganAI** does its first work: reading the Hogan and IBM Z estate to reverse-engineer how it really behaves.

Automated discovery and dependency mapping

DXC uses AI-assisted tooling to reverse-engineer COBOL, PL/I, Assembler and RPG code, extract specifications and map dependencies across programs, jobs, copybooks and data. The result is a living map of the estate: what exists, what calls what, and what is genuinely used versus what is dead weight.

HoganAI accelerates this discovery on Hogan cores specifically — documenting logic, surfacing hidden coupling and producing the artifacts that a modernization program would otherwise spend months hand-crafting.

Processing and code optimization (MIPS, batch, data)

With the map in hand, optimization becomes targeted rather than speculative. Blueprinting identifies MIPS hotspots, inefficient batch windows and redundant processing so effort is spent where it moves cost and performance most:

- Eliminating dead and duplicated code before any conversion, shrinking scope and risk
- Targeting MIPS-heavy paths and batch bottlenecks for refactoring or offload
- Rationalizing data access and reducing unnecessary I/O against the core

This directly addresses a pattern DXC sees across the Hogan client base: institutions already offloading analytics and channel inquiries from the core, now seeking to solve the real-time portion without destabilizing it.

Building the transformation blueprint

The output is a prioritized transformation blueprint: a domain-by-domain plan that sequences the work, quantifies the optimization opportunity, and defines clear entry and exit criteria for each increment. It is the single artifact that derisks everything downstream: Conversion, API exposure and externalization all trace back to it.

HoganAI in blueprinting

- Reverse-engineers and documents Hogan/IBM Z logic to create the baseline
- Pinpoints MIPS hotspots, dead code and coupling for targeted optimization
- Produces the evidence base that makes every later step lower-risk

LEVER 2: ACCELERATE TIME TO MARKET WITH AI IN THE SDLC

ANALYZE | CONVERT | TEST | SHIP

Modernization has historically been slow because it is knowledge-intensive. DXC changes the economics by embedding generative AI across the software development life cycle through the **DXC Recursive AI Method** — delivering a 3x acceleration of modernization timelines without compromising quality, governance or business-logic fidelity. Work that once took years of consultant-led mapping can be completed in quarters.

AI-assisted code conversion and refactoring

The Recursive AI Method combines automated and manual conversion to move legacy COBOL, PL/I, Assembler and RPG (including AS/400) to modern, cloud-native Java. This is DXC's exclusive tooling that converts mainframe and AS/400 languages to Java at up to 50% lower cost. HoganAI feeds this stage with the specifications and dependency context extracted during blueprinting, so conversion is grounded in evidence rather than assumption.

Not all code needs should be converted. Hogan Umbrellas[1] code is very domain specific and has been proven at a large scale. However, an enterprise may have some legacy COBOL or Assembler outliers where there is a strong business case to convert.

Human+ validation and functional equivalence

AI does not run unsupervised. Every increment is validated by expert engineers for **functional equivalence**, confirming the modernized code does exactly what the original did before it is handed over. This human-in-the-loop model preserves compliance, quality and business-logic fidelity, which is non-negotiable for regulated cores.

Continuous, milestone-based delivery

Delivery is continuous rather than big-bang: each domain moves through analysis, conversion, validation and release with clear entry and exit criteria. Your engineers observe progress throughout, so knowledge transfers into the organization rather than out of it:

- Analysis and documentation are accelerated by AI reading the real codebase.
- Code is converted to production-ready, cloud-native Java with functional parity.
- Automated test generation and equivalence checking are completed before handover.
- Release is done by domain — value is delivered every quarter, not every few years.

[1] Hogan Umbrella is the Hogan platform development and runtime environment — the engineering platform where new banking capabilities are built and the runtime that executes them in production.

LEVER 3: API-LED FRICTIONLESS JOURNEYS

DECOUPLE | EXPOSE | REUSE

Customers and employees do not experience the core; they experience the journeys built on top of it. Component- and API-led modernization decouples those journeys from the mainframe by exposing core capabilities as secure, governed services.

Component-led modernization decomposes the monolithic core into discrete business capabilities (payments, deposits, cards, lending, customer, etc.) that can be modernized, replaced, or re-platformed independently. It's about where the functional boundaries sit, and about tackling the estate piece by piece.

API-led modernization exposes capabilities through well-defined, layered APIs (typically, system → process → experience tiers) so consumers never touch the core directly. It's about how capabilities are accessed and composed, and about decoupling change on either side of the interface.

This is where **DXC CoreIgnite —** a cloud-native, composable, API-first financial services operating system — accelerates adoption of new capabilities, especially in digital assets.

Externalizing core services as APIs

Stable, well-governed APIs turn the core into a reusable service provider. Channels, products and partners consume standardized interfaces instead of coupling to internal programs, so new experiences can be assembled quickly without touching the systems of record. The core stays authoritative; the pace of change moves to the edge.

Customer and employee experience patterns

CoreIgnite accelerates fintech adoption by making advanced capabilities consumable through APIs and a partner ecosystem, including Ripple, Euronet, Splitit and Aptys. In **digital assets**, it brings embedded finance, buy now pay later (BNPL), stablecoins and payments orchestration (ACH, RTP, FedNow, wire and card) within reach of the existing core — enabling banks to offer new services to customers and streamlined tooling to employees without a core rewrite:

- Frictionless customer journeys: real-time payments, digital-asset custody and buy/sell/hold, cross-border flows
- Frictionless employee journeys: unified servicing and exception handling instead of green-screen navigation
- Reuse across lines of business: one API, many products and channels

Governance, security and reuse

- Externalizing the core raises the bar on governance. API access is secured, versioned and audited; entitlements are enforced centrally; and a managed catalog drives reuse so each capability is built once and consumed everywhere. The result is faster time to market with stronger, more demonstrable control.

DXC CoreIgnite for digital assets

Cloud-native, composable, API-first financial services operating system

Ecosystem: Ripple, Euronet, Splitit, Aptys — embedded finance, BNPL, stablecoin

Payments orchestration across ACH, RTP, FedNow, wire and card — without a core rewrite

LEVER 4: AUGMENT THE CORE BY EXTERNALIZING FUNCTIONS

OFFLOAD | COEXIST | EVOLVE

The final lever moves selected functions off the core and onto modern platforms — progressively, so the mainframe is relieved rather than ruptured. A modern core ledger, **Light Ledger**, anchors this step by externalizing core accounting, balances and settlement into a cloud-native subledger while the mainframe remains the trusted system of record during transition.

The strangler pattern in practice

Rather than replacing the core in one move, the strangler-fig pattern grows modern capabilities around the core and gradually redirects traffic from it. New functionality is built on modern platforms, existing functionality is migrated domain by domain, and the core is quietly relieved of load until its remaining role is clear and small. At every step the system is whole and in production.

Choosing what to externalize

The blueprint from Lever 1 decides sequence. Good early candidates are functions that are costly to run on the core, change frequently, or are needed in real time by digital channels — analytics, channel inquiry, pricing and digital-asset ledgering. Systems of record with heavy regulatory weight are externalized last, and only after equivalence is proven.

Light Ledger is a natural first externalization for balances and settlement: It provides a modern, scalable ledger/subledger that offloads MIPS-heavy accounting from the core, supports real-time posting for digital channels and digital-asset flows, and coexists with the mainframe until cutover is a low-risk formality.

Data, integration and coexistence

Coexistence is an engineering discipline. Event streaming and change-data-capture keep the core and modern platforms consistent; APIs (Lever 3) provide synchronous access; and reconciliation proves parity continuously. This lets externalized functions run in real time without sacrificing the integrity guarantees the core is trusted for.

Light Ledger benefits

- Modern cloud-native core ledger/subledger — externalizes accounting and balances off the mainframe
- Offloads MIPS-heavy processing, reducing core run cost and freeing capacity
- Real-time posting and settlement for digital channels and digital-asset flows
- Coexists with IBM Z/IBM i as system of record, enabling low-risk, progressive cutover

THE DXC APPROACH IN PRACTICE

The four levers are delivered through a single, commercially aligned operating model. It is designed so that risk sits with DXC and value arrives early and continuously.

Modernization as a service

Modernization as a service (MaaS) combines AI-assisted code conversion, cloud-native rearchitecture and outcome-based delivery into a managed model that helps you generate revenue, not just retire technical debt. Modernization is phased, continuous and commercially structured so each milestone delivers measurable value before the next begins — a faster path to modern architecture, with lower program risk.

Recursive AI Method

The DXC Recursive AI Method is the engine beneath Lever 2: generative AI with human-in-the-loop validation, delivering 3x acceleration without compromising quality, governance or business-logic fidelity. Combined with HoganAI's discovery, it turns years of consultant-led mapping into quarters of evidence-based delivery.

Modernization Express: A 90-day start

Not every program needs to begin with a multiyear roadmap. Modernization Express is a fixed-price, fixed-scope engagement that delivers a defined outcome in 90 days across four stages: scope and secure access; conversion via the Recursive AI Method with your engineers observing; validation for functional equivalence; and handover of production-ready, cloud-native code. Every stage has clear entry and exit criteria.

Outcome-based, phased delivery

Milestones are tied to business results, not billable hours, with transparent line-of-code pricing. DXC takes on the complexity and risk so you move forward with predictable costs, minimized disruption and a clear path to transformation.

How the levers map to delivery

- Blueprinting (Lever 1) → HoganAI discovery + MaaS assessment
- AI in the SDLC (Lever 2) → Recursive AI Method, human-in-the-loop
- API-led journeys (Lever 3) → CoreIgnite APIs + fintech/digital-asset ecosystem
- Externalize functions (Lever 4) → Light Ledger + strangler-pattern coexistence

PROOF AND OUTCOMES

Scale and reliability

DXC is one of the world's largest independent mainframe services providers, named a Leader in the ISG Provider Lens 2025 report for Mainframes — Services and Solutions. We have the scale to enable modernization in place.

3x	50%	90	99.5%
FASTER TIMELINES (RECURSIVE AI METHOD)	LOWER-COST JAVA CONVERSION	DAYS TO FIRST OUTCOME	SUCCESS RATE

Representative program metrics across DXC mainframe modernization engagements

Representative outcomes

- A leading Spanish bank reduced online transaction response time by over 50%, speeding customer transactions and improving satisfaction and loyalty.
- A multinational automotive manufacturer integrated mainframe data with modern systems to improve operational efficiency and supply-chain visibility.
- Across the Hogan client base, institutions offloading analytics and channel inquiry are extending to real-time externalization without destabilizing the core.

A PHASED ROADMAP

A pragmatic sequence turns the equation into a plan. Each phase delivers value on its own and derisks the next:

1. **Assess and blueprint.** Use HoganAI and a MaaS assessment to map the estate, quantify MIPS and cost opportunities, and produce the prioritized blueprint (Lever 1).
2. **Prove value in 90 days.** Run Modernization Express on a contained domain to demonstrate AI-accelerated conversion and functional equivalence (Lever 2).
3. **Expose and experience.** Stand up governed APIs and use CoreIgnite to launch new customer and employee journeys, including digital-asset capabilities (Lever 3).
4. **Externalize and offload.** Move ledger, balances and settlement to Light Ledger under the strangler pattern, relieving the core while it remains system of record (Lever 4).
5. **Expand and operate.** Repeat by domain under outcome-based delivery, operating the modernized estate with continuous optimization.

Guiding principle

The business stays operational while the future is built beneath it, so value is delivered in every quarter and risk is contained at every step.

CONCLUSION: SOLVING THE EQUATION

Progressive, in-situ modernization resolves the tension that defeats big-bang programs: It raises business value while systematically reducing risk.

Blueprinting with **HoganAI** establishes the evidence; the **DXC Recursive AI Method** accelerates delivery; **CoreIgnite** turns the core into frictionless, API-led journeys and fintech/digital-asset capability; and **Light Ledger** augments the core by externalizing functions onto a modern ledger — all delivered as outcome-based MaaS.

The result is not a leap of faith but a sequence of proven steps, each paying its way. The core keeps running the business while the future is built beneath it.

Start with a blueprint

Begin with a Modernization Express assessment: a fixed-price, 90-day engagement that maps your estate, proves AI-accelerated conversion and sets the roadmap — with predictable costs and clear entry and exit criteria.

Learn more at

dxc.com

DXC Technology (NYSE: DXC) is a leading enterprise technology and innovation partner delivering software, services, and solutions to global enterprises and public sector organizations — helping them harness AI to drive outcomes at a time of exponential change with speed. With deep expertise in Managed Infrastructure Services, Application Modernization, and Industry-Specific Software Solutions, DXC modernizes, secures, and operates some of the world's most complex technology estates. Learn more on **dxc.com**.

LinkedIn | Instagram | TikTok | YouTube

© 2026 DXC Technology Company. For discussion purposes. Product names and capabilities referenced are DXC offerings and partner solutions within the CoreIgnite ecosystem. DG_1740a-27