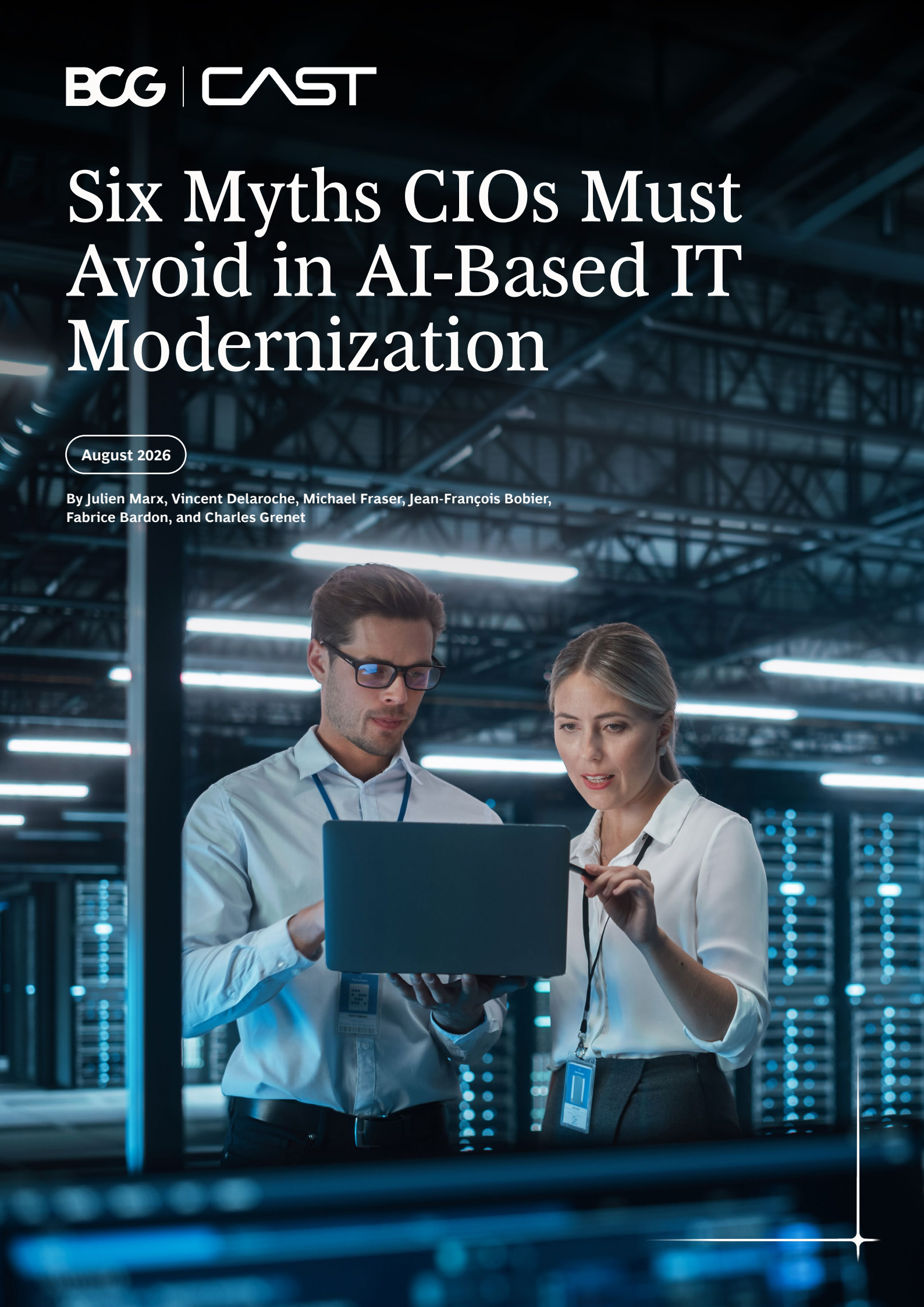
BCG | CAST

Six Myths CIOs Must Avoid in AI-Based IT Modernization

August 2026

By Julien Marx, Vincent Delaroche, Michael Fraser, Jean-François Bobier, Fabrice Bardon, and Charles Grenet



About Boston Consulting Group

Boston Consulting Group bridges the gap between ambition and outcomes for the world's leading companies and organizations. We are built for this era of unprecedented change—bringing strategic clarity rooted in over 60 years of deep domain knowledge, combined with applied AI shaped by our practitioners. BCG works shoulder-to-shoulder with CEOs across industries and geographies to deliver transformative impact at scale: stronger returns, transferred capabilities, and change that sticks. For more information, visit bcg.com.

About CAST

Businesses move faster using CAST to understand, improve, and transform their software. Through semantic analysis of source code, CAST generates dashboards and 3D maps for executives, technologists, and AI to navigate inside individual applications and across entire portfolios. This intelligence enables companies to steer, speed, and report on initiatives such as technical debt, modernization, and cloud. As the pioneer of the software intelligence field, CAST is trusted by the world's leading companies and governments, their consultancies and cloud providers.

Introduction

IT modernization has returned to the top of CIO agendas—and for good reason. Many core IT systems, from home-grown applications to decades-old COBOL code, have become strategic liabilities.

They slow time to market, complicate integration, limit data mobility, and increase operational and cyber risk. And the upkeep that these legacy systems require can consume significant IT capacity, leaving too little room for innovation.

Generative and agentic AI have prompted new optimism regarding IT modernization. Leaders hear that AI can read code, write new code, and test everything—capabilities that promise to make modernization faster, cheaper, and safer. However, although AI can unlock sizable value in an IT modernization, it is not a magic wand. Failures may occur if AI is applied too generically.

IT modernization involves multiple steps, from understanding business intent and analyzing how systems actually work to designing architecture, coding, and testing. Each step requires different types of intelligence and tailored AI support.

Drawing on real-world experience across industries and regions, we have developed a practical method for success in AI-driven modernization. Its guiding principle is simple: use the right AI in the right context for the right task. Organizations that adopt this method can achieve cost reductions of 25% to 35% and reach implementation 30% to 40% faster.

The Six Myths

To capture full value at scale, organizations must first recognize and resist six misconceptions that can derail AI-based modernization.

Myth 1: IT Modernization Delivers ROI on Its Own

IT modernization is often assumed to be inherently valuable, but the business case may be weak if its supporters justify it purely on technical grounds.

Why It Leads Organizations Astray

Many legacy systems are already amortized, remain functional, and cost relatively little to run. In contrast, the execution risk of replacing them can be quite high. Large-scale IT transformation programs have a high risk of failure, with budgets that sometimes expand two to four times and timelines that stretch close to a decade. And in many cases, the organization has already made the most accessible improvements, such as front-end and back-end decoupling, leaving only more complex and costly work unaddressed.

What to Do Instead: Anchor Modernization in Business Value

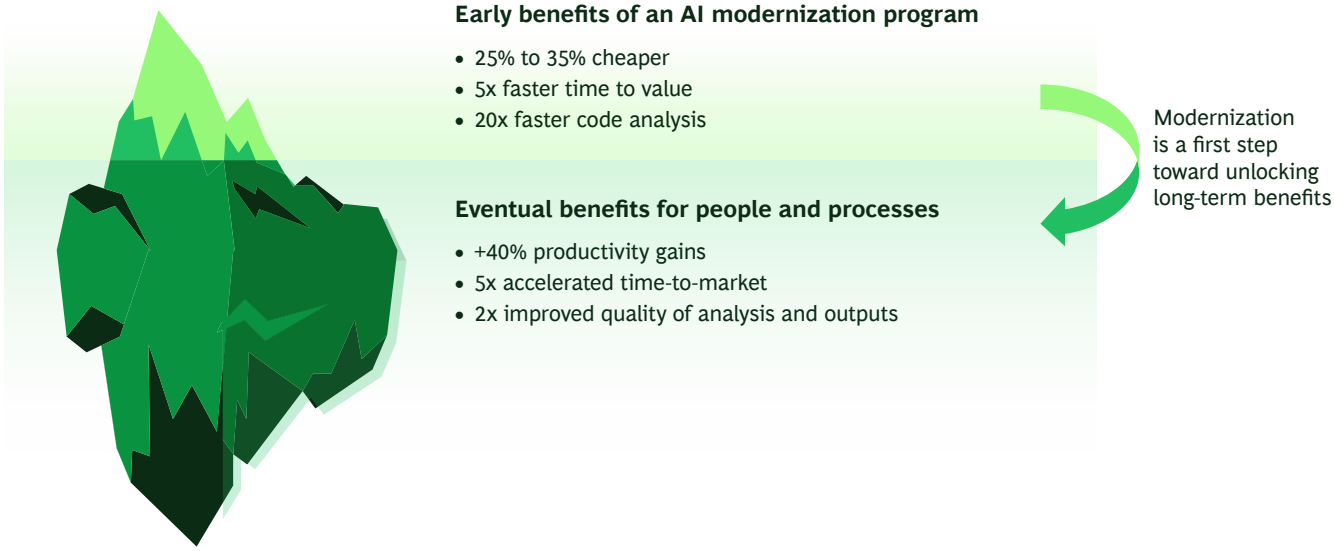
IT modernization delivers the greatest value when the effort meets three conditions:

- **A clear business case is in place.** Value comes from simplifying processes and improving outcomes, not from rewriting systems one-to-one. Therefore, the organization should pair modernization with business redesign.
- **The IT system is core to the enterprise architecture.** Transformation focuses on critical IT systems; meanwhile, standard solutions can replace peripheral components.
- **The organization actively manages costs and risks.** When applied in a targeted way, AI reduces effort and execution risk. In the absence of such attention, programs may become prohibitively expensive.

AI-driven modernization should take the form of a selective, value-driven transformation rather than a blanket technical upgrade. AI delivers substantial near-term gains in modernization programs—lowering costs, speeding delivery, and reducing risk—but its greater value in the long term comes from reshaping how teams work, improving productivity, and embedding new ways of operating across IT and the business. When managed properly, AI-driven modernization can increase productivity by more than 40%, accelerate time to market up to fivefold, and significantly improve the quality of analysis and outputs. (See Exhibit 1.)

EXHIBIT 1

AI Modernization Unlocks Short-Term and Long-Term Benefits



Source: BCG analysis.

Myth 2: A Single LLM or AI Agent Can Modernize the Entire System

The idea that a single LLM or AI agent can modernize an entire IT system has powerful allure, but it often leads to disappointment.

Why It Leads Organizations Astray

AI models excel at specific tasks, such as summarizing documents, generating code, or proposing designs, but IT modernization demands multiple distinct forms of intelligence. AI must interpret business objectives, understand how systems behave, design a target state, execute safely within constraints, control risk, and verify outcomes. Often, applying one model or agent to all of these tasks produces fast but brittle results. Outputs may be inconsistent, difficult to validate, and hard to operationalize, and issues may emerge late in the integration or cutover.

What to Do Instead: Treat IT Modernization as an Orchestrated Toolchain

Successful modernization programs apply different AI capabilities to different stages of the process:

- Deterministic software intelligence can map dependencies, transactions, and data access patterns to establish a reliable view of the current system.
- Evidence-backed generative AI can help interpret specifications, explain system behavior, and generate documentation.
- Architecture support tools can evaluate design alternatives, guide migration choices, and optimize sequencing.
- Engineering AI agents can accelerate implementation within defined architectural, security, and quality constraints.
- Automated testing and analytics can expand test coverage, validate outcomes, and detect regressions early.

The key is to adopt a coordinated approach that uses the right tools at each stage, rather than expecting one tool to do everything. (See the sidebar, “Matching AI Capabilities to Modernization Needs.”)

Matching AI Capabilities to Modernization Needs

AI can dramatically accelerate IT modernization, but success depends on matching the right AI capabilities to the right modernization tasks. (See the exhibit.)

Where AI Helps, What Inputs It Needs, and How to Apply It

		Right task (why AI can help)	Right context (what input AI needs)	Right AI pattern (how to use AI)
Myth 1	IT modernization delivers ROI on its own	• Case for change	• Business value • Criticality of IT system • Transformation costs	• Draft ROI assessment
Myth 2	A single LLM or AI agent can modernize the entire system	• End-to-end modernization orchestration	• Varies by stage • Deterministic analysis (for example)	• Multiple models • Stage-specific validation
Myth 3	AI can explain everything from the code base	• Retro-documentation • Impact analysis • Business rule discovery	• Dependencies • Code semantics • Specifications • Operational signals	• Retrieval-based AI • Synthesis grounded in evidence • Coherence checks • Human review
Myth 4	AI can design the architecture based on specifications	• Target architecture • Domain decomposition • Migration sequencing	• Dependency graphs • Data lineage • Nonfunctional requirements • Operating constraints	• Option generation • Constraint evaluation • Scenario comparison
Myth 5	AI-assisted coding is enough to accelerate modernization	• Implementation acceleration	• Architecture blueprint • Standards • Security policies • API contracts	• Code generation • Automated reviews • Static analysis • Quality gates
Myth 6	AI reduces the need for testing and de-risks cutover	• Test generation • Regression assurance • Data reconciliation • Cutover readiness	• Behavioral baselines • Production traces • Acceptance criteria • Coverage gaps • Reconciliation rules	• Test generation • Execution analytics • Anomaly detection • Gated promotion

Source: BCG analysis.

Case Study

A large organization set out to modernize a portfolio consisting of thousands of applications that use AI, assuming that the model itself could drive the transformation—but early results were disappointing. Over the first 12 months, AI-generated code changes delivered an accuracy rate of only about 30%, and they often created more work than they saved. The breakthrough came when the team stopped relying on the AI model alone and instead built an automated, fact-based map detailing how each application actually worked, including dependencies, data flows, and end-to-end behavior. When they fed that information into the AI tool, the tool's accuracy rose to 85%.

Myth 3: AI Can Explain Everything from the Code Base

Large language models can analyze any code base and produce plausible explanations. Unfortunately, home-grown systems are rarely easy to interpret on the basis of code alone.

Why It Leads Organizations Astray

Decades of coding changes will introduce hidden dependencies, configuration-driven behavior, undocumented integrations, and environment-specific logic. Unless grounded in reliable analysis and supporting documentation, AI can deliver convincing narratives that are not entirely accurate.

Because LLM-based tools and AI agents operate iteratively and probabilistically, embedded inaccuracies can compound as AI analyzes a large, complex codebase. This snowballing effect creates real risk in IT modernization programs. When discovery yields interpretations rather than verifiable evidence, risk and uncertainty may emerge later in the process, when addressing them is more costly and disruptive. In addition, security, compliance, operability, integration patterns, and performance characteristics are easy to lose in translation during migration.

What to Do Instead: Ground AI in Evidence

It is essential to establish an evidence-based understanding of the IT system by mapping call graphs, dependency structures, transaction flows, data access patterns, and architectural violations. Static and semantic analysis provides a structural view of the code base, while runtime telemetry—such as logs, traces, and operational signals—reveals which pathways are actually executed in production.

When it can draw on this evidence base, AI becomes significantly more effective. It can explain system flows, synthesize documentation, identify business rules, and perform impact analysis with greater accuracy and context. Conversely, if an LLM or AI agent attempts to modernize a code base without requesting additional context, it signals a critical limitation. Today's leading models, when prompted to modernize systems, should request architectural documentation and specifications to improve output quality.

Case Study

The most successful IT modernization efforts arrange for AI to work from a verified architectural map rather than raw code alone. A telecom operator team applied this approach across 60 applications and roughly 10 million lines of code. AI helped reconstruct the technical and functional steps underlying business processes, while a deterministic system map validated how components interacted. In comparison to a fully manual effort, this method accelerated documentation by up to 80% and improved accuracy by around 60%.

Myth 4: AI Can Design the Architecture Based on Specifications

Specifications capture intent—what the business expects the system to do. But intentions can grow murky in a setting where multiple interventions have altered the original code.

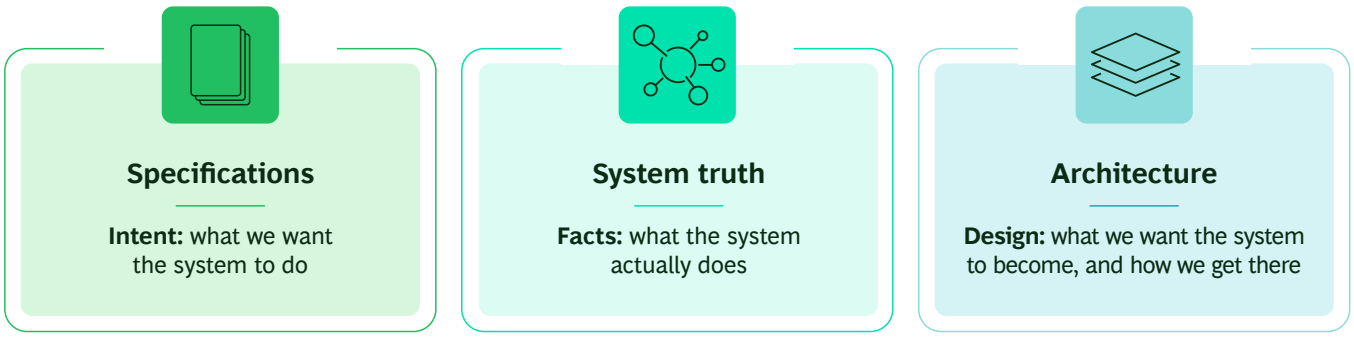
Why It Leads Organizations Astray

Over time, through patches, workarounds, and undocumented changes, systems tend to drift; and as a result, actual behavior may diverge from the original specifications. Architecture defines what the system is designed to become, including the security, compliance, operability, integration patterns, performance, and migration sequencing it needs to get there. (See Exhibit 2.)

Architecture requires a series of decisions and tradeoffs. Teams must determine what to decouple first, where to introduce seams, which interfaces must remain stable, and what components should be retired, replatformed, or rebuilt. They must also ensure that the system remains safe and operational in various intermediate states throughout the transformation. AI can rapidly produce architecture diagrams—but unless the design is anchored in real system constraints, those diagrams may not be achievable.

EXHIBIT 2

AI Can Support Decisions on IT Specifications, System Truth, and Architecture



Source: BCG and CAST analysis.

What to Do Instead: Use AI to Explore Options and Tradeoffs

A better approach is to use AI to support structured architectural reasoning. AI can generate and evaluate multiple architecture options; clarify tradeoffs across cost, risk, speed, and operability; stress-test designs against dependencies and data flows; and help develop migration roadmaps with stable intermediate states. In this model, human judgment remains central in selecting a path that will deliver the most value while keeping risk in check.

Myth 5: AI-Assisted Coding Is Enough to Accelerate IT Modernization

Organizations commonly imagine that they can accelerate IT modernization on the strength of AI-assisted coding alone. But in large IT modernization programs, coding is rarely the primary bottleneck.

Why It Leads Organizations Astray

Programs tend to break down in situations where implementation begins before the organization has resolved critical design decisions, including domain boundaries and interfaces, integration patterns, behavioral requirements (what must not change), data migration approaches, and testing and verification strategies.

Implementing AI coding agents on top of an unstable design will inevitably generate rework. Or worse, teams will deliver IT systems that function on a superficial level but are challenging to operate, secure, and scale over time.

What to Do Instead: First Validate and Then Accelerate Coding

High-performing programs make explicit design decisions long before large-scale implementation begins. A typical progression looks like this:

- Analyze the current state by documenting systems, dependencies, interfaces, and business logic.
- Design the future state through architecture, decomposition, and interface design.
- Define expected behavior by using tests, acceptance criteria, and other mechanisms that specify what must not change.
- Scale AI implementation by deploying engineering AI agents within clear guardrails, including engineering standards, security policies, interface contracts, and automated quality checks.

AI agents deliver real productivity gains only after the organization has defined the system thoroughly enough to support clear, explicit design decisions.

Case Study

A major transportation operator used AI agents to rewrite legacy code, but the results were unreliable, with too many silent breaks and undocumented dependencies. To address the problem, the operator spent one week building a trusted map of how the system worked, detailing every dependency and data flow. With that foundation in place, the operator could generate credible cost estimates, consistently meet timelines, and build AI-assisted delivery on solid ground rather than guesswork.

Myth 6: AI Reduces the Need for Testing and De-risks Cutover

AI can perform tasks with impressive speed and seeming comprehensiveness, but its talents do not obviate the need for careful testing or for attention to cutover risks.

Why It Leads Organizations Astray

Testing ensures protection of business continuity, regulatory compliance, and the customer experience. A modernized IT system must perform reliably and correctly across critical user journeys and transactions, edge cases and regulatory logic, performance and resilience, and data reconciliation and reporting. AI can help draft test cases and accelerate triage, but unless it operates in an environment with clearly defined outcomes and robust regression baselines, AI-driven testing can produce a lot of activity leading to limited assurance.

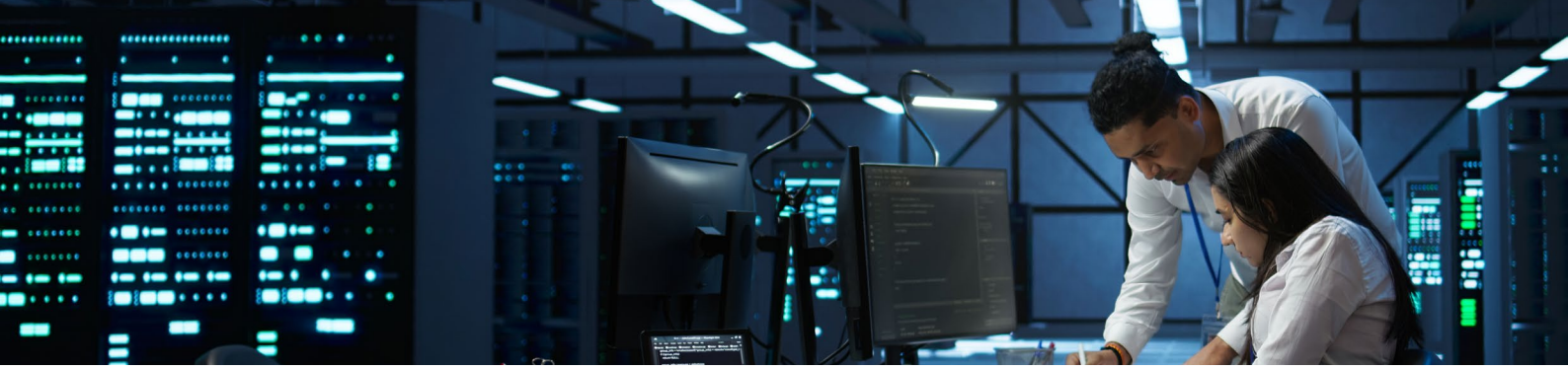
The final cutover from the old system to the new one is another common failure point. Treating it as a single, final event elevates risk. Big-bang transitions often fail because they force too many unknown interactions into one moment.

What to Do Instead: Combine AI Speed with Engineering Discipline

The most effective programs combine AI-driven acceleration with disciplined engineering practices:

- Use AI to generate test cases, data sets, and scenarios more quickly.
- Apply deterministic system intelligence to identify structural risks early in the process.
- Enforce continuous quality gates.
- Modernize incrementally, with stable intermediate states and clear rollback paths.
- Treat data migration and reconciliation as core design elements from day one.

AI can significantly accelerate verification, but it does not eliminate the need for it.



Where Leaders Should Start

Organizations do not need to commit immediately to a multiyear IT modernization program. The most effective starting point is a focused, time-boxed diagnostic that builds a credible path forward. Leading organizations follow a pragmatic three-stage approach (see Exhibit 3):

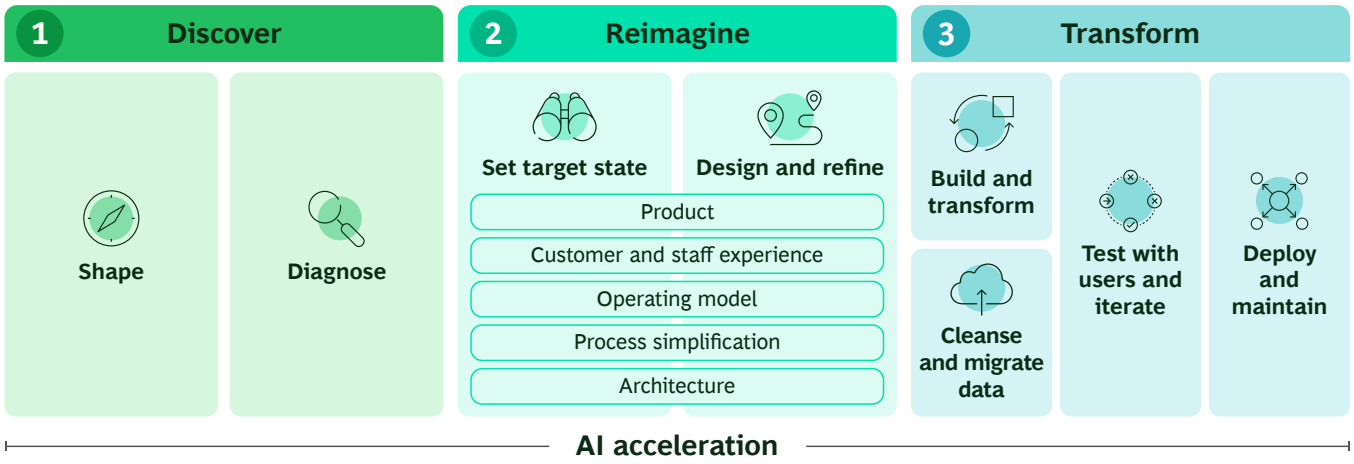
- **Discover.** First establish the transformation ambition and business case, and then diagnose the current system using deterministic analysis and AI-accelerated discovery. This early phase helps build a shared, fact-based understanding of how the system works—its dependencies, data flows, and business logic—so decisions are grounded in evidence. The resulting shared alignment and credible sequencing are strong predictors of success.
- **Reimagine.** Define the target state across products, customer experiences, operating model, processes, and architecture. Use AI to generate and compare options, decompose the system into modernization blocks, and refine a sequenced roadmap with stable intermediate states that maximize value and reduce risk. At this stage, to define how the organization will apply AI, leaders design a toolchain that aligns to each task, from specifications and architecture to coding, testing, and data.
- **Transform.** Execute the IT modernization, powered by AI agents, using standardized methods that can scale across many systems. Build and integrate components under clear guardrails, migrate and reconcile data, and continuously validate the results through testing, user feedback, and iteration. Deploy incrementally with full operational readiness to ensure that each step delivers value while maintaining system stability and enabling future scale.

Taken together, this three-phase approach connects architectural choices to business outcomes and turns IT modernization into a sequenced, credible transformation plan.

The real promise of AI in modernization comes from disciplined use of the right AI, grounded in the right context, at each stage of the journey. When executed well, this approach significantly reduces costs and accelerates delivery compared to traditional programs. It also produces higher-quality systems that are better documented, easier to maintain, and more resilient over time.

EXHIBIT 3

A Three-Stage Approach to Tech Modernization



Source: BCG and CAST analysis.

About the Authors

Julien Marx
Managing Director & Partner
Paris
marx.julien@bcg.com

Vincent Delaroche
Founder & Chairman of the Board
CAST

Michael Fraser
Managing Director & Partner
Los Angeles
fraser.michael@bcg.com

Jean-François Bobier
Partner & Vice President, Deep Tech
Paris
bobier.jean-francois@bcg.com

Fabrice Bardon
Director, BCG Platinion
Paris
bardon.fabrice@bcg.com

Charles Grenet
Architecture Director, BCG Platinion
Paris
grenet.charles@bcg.com

BCG | CAST