

URL slug: tech-debt-reduces-roi-mvp-development

SEO title: Technical Debt Costs Enterprises 29% in AI ROI

SEO description: IBM data ties technical debt to 29% lower AI ROI. Unico Connect breaks down where debt starts and what enterprises must fix before programs stall.

Tech Debt Reduces ROI by 29%, and Many Teams Create It During MVP Development

Unico Connect's production experience with enterprise AI shows why early architecture shortcuts are the most expensive technical debt to fix.

Enterprises that account for technical debt in their AI business cases **project 29% higher ROI**, according to IBM's [Tech Debt Reckoning report](#).

Those that ignore it **risk ROI declines of 18% to 29%**.

EMBED

https://www.linkedin.com/posts/martin-chester-ibm_a-practical-approach-to-boosting-your-ai-share-7396502179332849664-4bX-/

Technical debt is the accumulated cost of shortcuts taken during development that must eventually be addressed. Most organizations don't recognize it until they're already paying for it.

The issue starts at the Minimum Viable Product (MVP) stage, the earliest, leanest version of a product, where speed is the right priority and architecture is the afterthought.

Most Teams Build Debt Into Their First Version

The same IBM report found **81% of executives say technical debt is already constraining AI success**, and **69% believe it will render some AI initiatives financially untenable**.

The Business Cost of Technical Debt

Stat	Finding
81%	Executives say technical debt is already constraining AI success
69%	Believe technical debt will render some AI initiatives financially untenable
15–22%	Longer implementation schedules due to technical debt
29%	Higher ROI projected by enterprises that account for technical debt in their AI business cases

Source: IBM Institute for Business Value



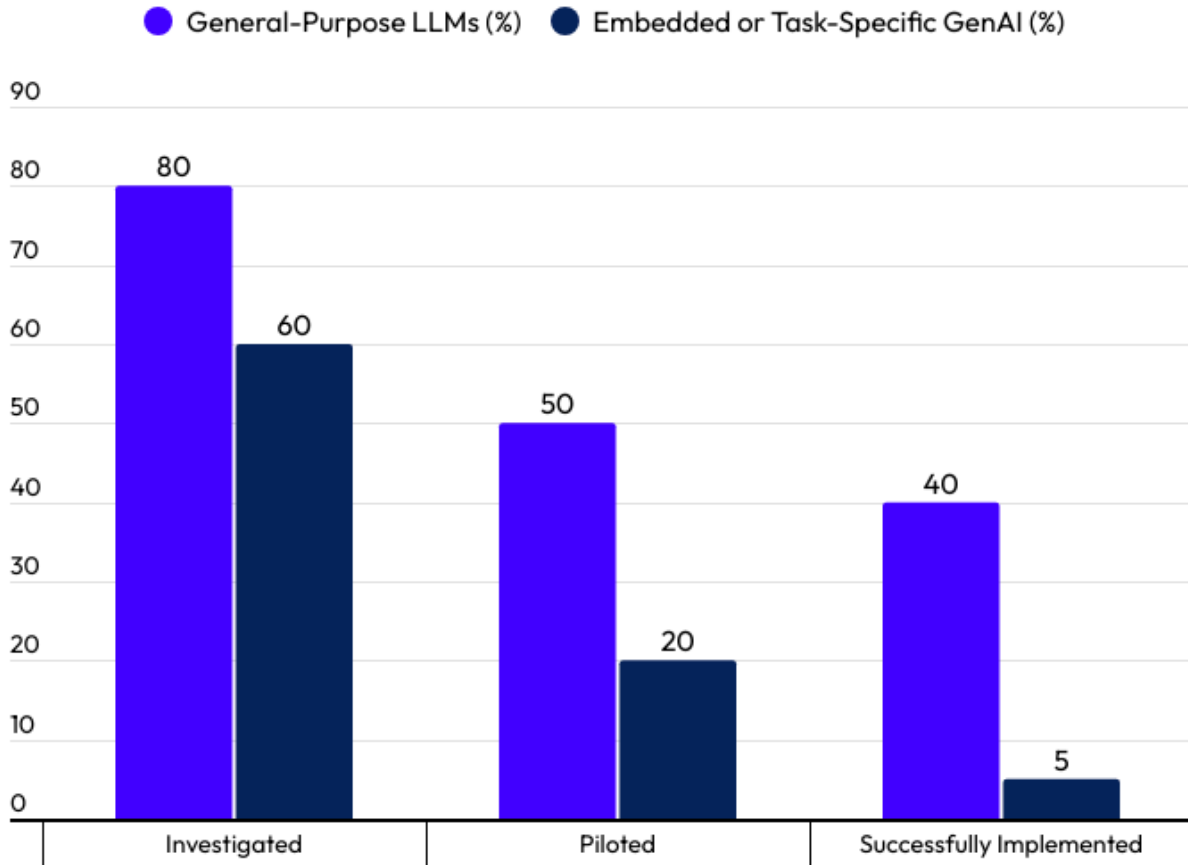
The research projects **technical debt will add 15% to 22%** to implementation schedules. A 30-month program becomes a 36-month one.

For a \$20 billion enterprise allocating 20% of IT spend to AI, IBM estimates that translates to more than \$120 million in hidden implementation costs annually.

MIT's [2025 State of AI in Business report](#) found 95% of enterprise GenAI pilots fail to deliver measurable business value, and only 5% of custom enterprise AI tools reach production.

GenAI Implementation Rates

General-Purpose vs. Task-Specific Tools



Source: MIT NANDA



However, the baseline cost runs deeper than AI programs alone.

Deloitte's [March 2026 analysis](#) found **technical debt accounts for 21% to 40% of IT spending** across enterprise organizations.

Nearly 60% of leaders believe **another 21% to 50% of value remains trapped** within their current tech, data, and people.

A typical organization spends just 23% of its tech budget to drive revenue. The rest goes to maintaining and fixing what already exists.

EMBED <https://www.youtube.com/shorts/sQ52JLr6d2c>

This is the problem that [Unico Connect](#), a digital product development agency that builds and deploys AI systems for enterprise clients, works to reverse before new programs get scoped.

"Speed is the right call at the MVP stage. Most teams never go back to fix what they rushed, and those shortcuts shape the architecture for everything that follows," said [Malay Parekh](#), CEO of Unico Connect.

Debt accumulated at the MVP stage is the most expensive to fix because it gets built into every system that follows.

Refactoring it later means dismantling the architecture that an entire product has grown around.

Tech Debt Follows the Same Pattern Every Time

Across codebases and industries, the same causes compound most, driven by wrong priorities applied repeatedly over time.

Rushed codebases produce functions that work under controlled conditions and fail under real ones.

Edge cases get deferred. Documentation gets skipped. What compiles today becomes the function no one wants to modify next year.

Missing development standards fragment codebases as teams grow.

When different developers solve the same problem in different ways, every future change requires understanding multiple conflicting approaches first.

EMBED

https://www.linkedin.com/posts/unico-connect_tech-debt-is-visible-ai-debt-is-silent-activity-7462849702330408960--NfE

QA treated as an afterthought means bugs get caught later and cost more to fix.

Test coverage built into the development cycle prevents the backlog that consumes sprint after sprint without visible progress.

Patchwork integrations create unmapped dependencies. One system changes and breaks another, and the team learns about it when it's already in production.

Architectural workarounds are temporary solutions that outlive their usefulness. Undocumented, unowned, and never replaced, they block the next migration.

"The first audit always tells us that speed was the priority, and no one ever went back. By the time we arrive, the debt has become a business constraint," Parekh added.

Still, IBM's research found 80% of executives agree that remediating debt in one initiative improves the ROI of related future initiatives.

Fix the Foundation Before AI Exposes It

Agents require clean data, structured integrations, and reliable system logic. They don't fail gracefully when they encounter fragmentation.

Deloitte's modeling found that infrastructure modernization alone reduces technical debt by 18% over five years.

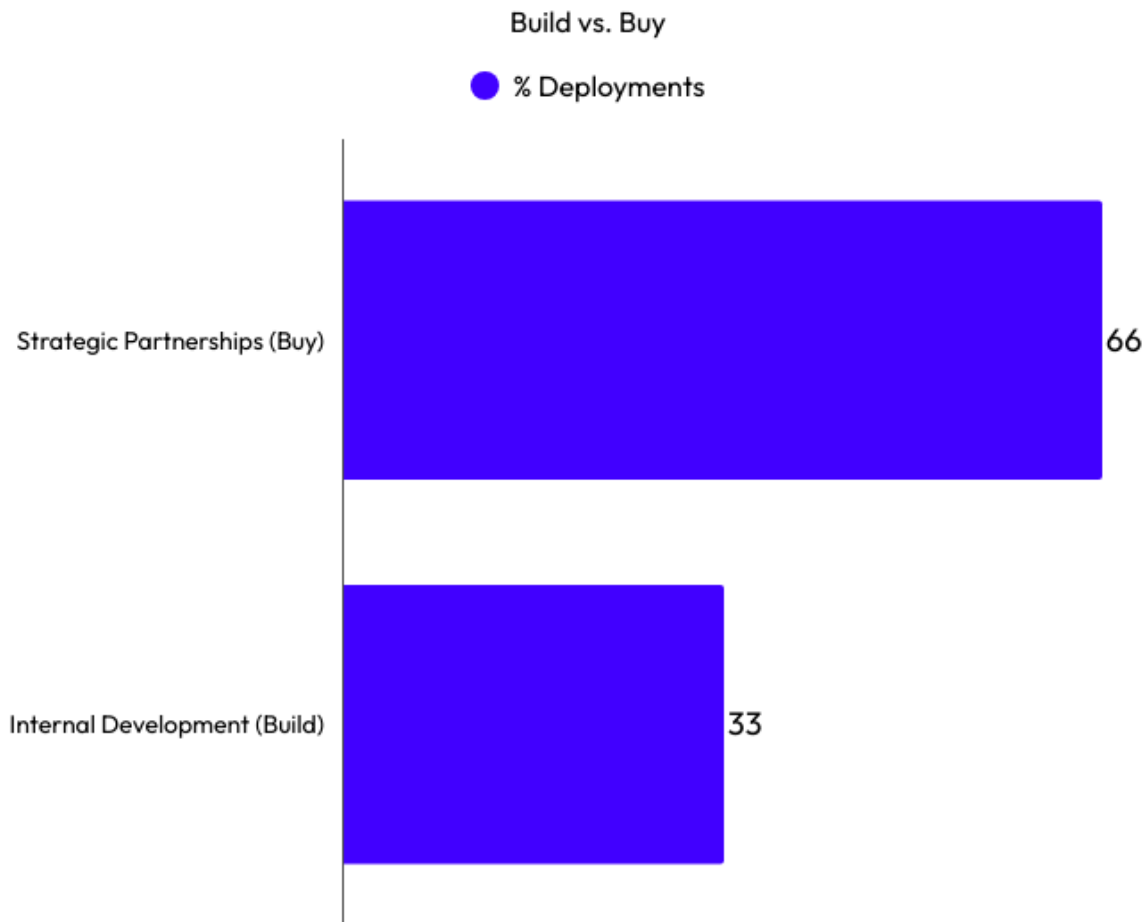
Data transformation produces a **52% improvement in latent potential** over the same period.

Likewise, as McKinsey's [2025 State of AI report](#) found, only 7% of enterprises have fully deployed AI across their organizations.

[Unico Connect](#) reviews existing architecture before scoping any new AI capability. Current systems define what AI can deliver.

"We look at what exists before we scope what we are building. That is not always what clients expect. But what the current system can support determines what the next one can be," Parekh noted.

Deployment Success Rates



Source: MIT NANDA



So what should teams address first?

For organizations earlier in this process, the recommendation is to start with three things before any new AI capability gets scoped.

- **Map dependencies before expanding them.** Fragmented systems without clear ownership are the primary source of integration failures.
- **Treat QA as part of the build.** Test coverage built into the development cycle prevents the bug backlog that slows every subsequent sprint.
- **Make architecture decisions before writing the first line.** Structural decisions made at the MVP stage are the hardest to reverse, and the cheapest to get right.

Debt addressed before an AI program begins is debt that does not become that program's first obstacle.

Governance Separates Scalars From Restarters.

Every engineering leader knows what technical debt is. The problem is that knowing does not change the sprint deadline.

AI is accelerating that moment. Teams that accumulated debt quietly for years are now deploying agents into fragmented systems.

Agents don't work around inconsistent data models or undocumented integration logic.

They fail, and the failure is operational, immediate, and visible in production rather than buried in a backlog.

Organizations that treat architecture review as a pre-deployment requirement will build on foundations that compound value.

Those that don't will spend the next two years watching pilots succeed, and programs stall while budgets disappear into cleanup work nobody planned for.

Technical debt is not a technology problem. It is a decision made in every sprint where the architecture review loses to the shipping deadline.

The teams paying 2024's shortcuts with 2027's budget already made that decision. The next sprint decides whether they keep doing it.