AI or obsolete?

Maia by matillion

Why the best CDOs **move before they're ready** (and how they win)

Table of contents

Introduction - 03

01 It was never “if” - 06
Why standing still is the riskiest play you can make

02 What good looks like - 15
Inside the data teams already pulling ahead

03 Proof in 90 days - 20
From intent to a business case that already works

04 Built to withstand scrutiny - 29
De-risking the move before your board asks

05 The compounding cost - 38
Why every day of hesitation costs more than the last

06 Your move - 42
The guide ends and your advantage begins

Maia by matillion

The mandate for the modern data leader has *changed*

It’s no longer about if or when you adopt AI; it’s how fast you can scale it before your competitors do. The cost of waiting doesn’t arrive all at once. It compounds until the lead your competitors built is out of reach.

Maia
by matillion

AI or
obsolete

Introduction

The pressure to scale AI creates its own trap. The fear of a public failure, or of a governance breach you'd have to answer for, locks too many leaders into analysis paralysis, waiting for an ecosystem that will never feel sound enough to trust. But waiting is the risk. The most successful CDOs have stopped waiting for perfection. They move before they're ready, and they win because of it.

Here's what they've understood that most haven't: AI doesn't create advantage by being deployed. It creates advantage by being supplied. The enterprises pulling ahead don't have better models — they've industrialized the supply of governed, AI-ready data into the systems that consume it.

Acting on it takes two moves: adoption of autonomous, end-to-end AI agents that build, govern, and evolve data products at scale, and shifting teams to directing and reviewing while agents execute. Both run on context: agents grounded in a strict, peer-tested understanding of the business, its security, and its governance.

This guide shows you what those moves really look like, and how to get there in *90 days*



Maia by matillion

Built from real adoptions, *not theory*

We studied real AI adoptions at leading enterprises to understand what separates the teams pulling ahead from the ones still stalled.

10
Enterprise customers studied

100+
Conversations reviewed

~30
Discussions analyzed in depth

8 Industries

- Financial services
- Banking
- Logistics
- Cybersecurity
- Food & facilities
- Travel
- Pharmaceuticals
- Manufacturing

Maia by matillion

01

It was never *“if”*

Why standing still is the riskiest play you can make



Maia
by matillion

The question *changed* while you were answering the old one

For a decade, the data leader’s job was to keep data flowing: standing up the pipelines, governing the warehouse, keeping the lights on. When a new technology arrived, the question was whether to adopt it at all. That call was yours to make, and “not yet” was a defensible answer.

That question is gone. The one that replaced it is harder: how fast can you scale AI before a competitor does?



70% of CDAOs are responsible for their org’s AI strategy and operating model

36% report directly to the CEO, up from 21% in 2024

75% of CDAOs not seen as essential to AI success will lose their C-level title by 2027

Source: Gartner, 2025.

That last number reframes the whole decision. The risk was never that an AI initiative fails. The risk is being the leader nobody considered essential to one.

The pressure comes from above now, and it’s *impatient*

The politics inverted. Data leaders used to spend their energy persuading a skeptical board that data infrastructure deserved investment. Now leadership is pushing, and asking why it isn’t further along.



Three questions your board is *actually* asking

Thomas Mazzaferro | 4X Top 100 CDO

Source

01 How are you democratizing AI with guardrails?

02 How are you defending against AI threats?

03 What are the big AI bets, and what’s the ROI?

A board asking these has stopped deciding whether to fund AI. It’s auditing your progress.

And it wants proof. At one travel and hospitality company, a new enterprise architecture team began reviewing the entire tooling estate, asking “why do we use that, why do we use this?”. Their Director of Data Engineering read it correctly: every product his team ran now needed a story and demonstrable value, because all of it was going to be questioned.

AI-scale demand ***cannot*** be met with ***human-scale*** data work



Maia by matillion

Underneath the board pressure sits arithmetic that worsens every quarter. Demand for governed, AI-ready data climbs. Team size stays flat.

Chris Mihalicz, CEO of Three Point Turn, calls the drag underneath it “the manual tax”.



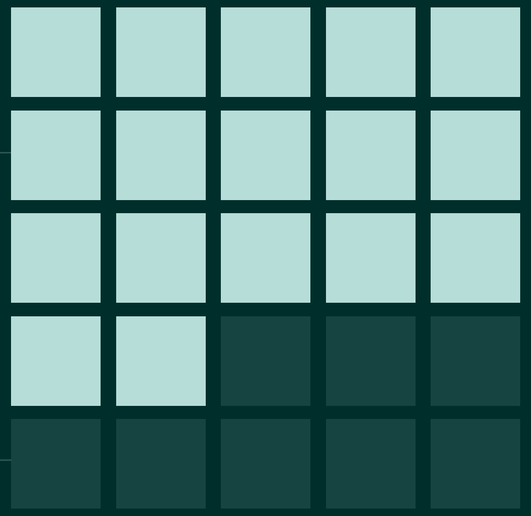
Chris Mihalicz
CEO of Three Point Turn

Bolting assistants onto a team buried in manual data work creates more surface area to manage. It does not remove the manual work itself: the brittle hand-built pipelines, the legacy ETL sprawl across Informatica, Alteryx, SSIS.

That work consumes the capacity that would otherwise supply your AI roadmap. When that capacity runs short, approved use cases don’t get canceled. They queue. And a queue is invisible at board level until someone asks why an initiative funded 18 months ago still hasn’t produced a number worth reporting. That question arrives at your desk.

The manual tax

60–70%
of a typical data engineer’s week goes to repetitive, low-value work.



30–40%
Everything the business actually asked for.



77%
of engineering teams report heavier workloads despite AI tools

MIT Technology Review / Snowflake, 2025.

Maia by matillion

How an approved AI portfolio **stops converting**

The reframe that changes the conversation. Your AI roadmap has a supply side, and how fast you can resource it determines how much of your AI investment converts into measurable value.



Ambition is approved

The board funds a portfolio. Use cases are approved on the assumption that data will be there when needed.

The supply side goes unresourced

Nobody funded the capacity to deliver governed, AI-ready data at the rate the portfolio consumes it.





Use cases stall in the queue

Approved initiatives wait on data that doesn’t exist in usable form. The delay is invisible at board level.

Value leaks

Investment is committed and never converts. The portfolio reports activity rather than outcomes.





Confidence erodes

The executive committee tightens approval, abandoning the investment at exactly the point the constraint is finally understood.

Maia by matillion

Gartner projects that 60% of AI projects unsupported by AI-ready data will be abandoned, not deprioritized, through 2026. That is the leakage chain at scale, across the market.

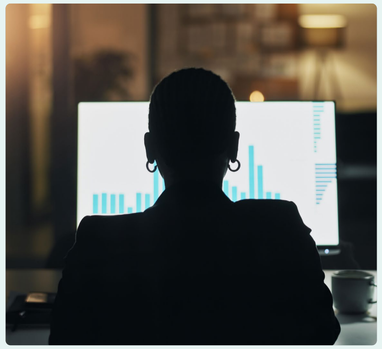
The fear is legitimate and **that’s what makes it dangerous**

Most data leaders freeze here.

95% of enterprise GenAI pilots deliver no measurable ROI (MIT, 2025)

88% of AI pilots never reach production (CIO.com)

<10% of enterprises have scaled agents to tangible value, while ~2/3 have experimented (McKinsey, 2025)



If you’re accountable for AI outcomes and those are the odds, waiting looks like prudence. Let the tooling mature. Let someone else absorb the failure rate.

That instinct is wrong, and MIT’s own research explains why. The 95% failure rate is not a verdict on model quality. MIT’s Project NANDA analysis attributes it to the organizational learning gap: the difficulty of absorbing AI into real workflows, real governance, and real teams. These pilots don’t fail because the technology can’t do the work. They fail because the organization can’t take delivery of it. None of this is your team’s failure.

Every assumption of the analytics era has broken

This is a structural problem, not a performance failure by your data team.

Your capacity grows by hiring; demand grows with every use case the business can imagine, and no hiring plan closes that gap. Roughly 9 in 10 organizations now use AI somewhere; what separates them is how much converts, and conversion runs at the speed governed data can be supplied.

Waiting would help if this were a technology problem: models improve, tooling stabilizes, and a later start becomes a safer start. Organizational absorption works the other way. Your definitions, standards, and approval processes don’t converge, document, or accelerate on their own, and your engineers don’t learn to direct agents while they’re still hand-coding pipelines. None of that waits on a vendor roadmap; all of it waits on you. Waiting only delays the work that lowers your exposure to failure.

The analytics era assumed	The AI era requires	What that means for ***you***
A human interprets the data	Autonomous systems consume it directly, with no interpretive layer	Meaning has to be encoded in the data estate itself, not held in teams’ heads
Latency of days to weeks is acceptable	Agents operate continuously and consume on demand	Data supply becomes an operational service rather than a project deliverable
Demand is bounded by headcount	Demand is bounded only by the use cases your business can imagine	Linear engineering capacity cannot serve exponential demand
Governance is applied at consumption, through review	Governance has to be enforced at creation, before autonomous action	Controls move upstream into the build, or they fail
Documentation is a compliance artifact	Documentation is machine-readable context that determines output quality	Institutional knowledge becomes a productive asset with measurable value

Maia by matillion

You cannot wait for *perfect*

Jason Mulvin, Director of Enterprise Data at Sophos, gave his peers direct advice on adopting agentic AI in data and analytics:

> “At some point you need to stick a pole... If you think about where you start, where you are, and where you want to go — if you don’t pick a spot... that’s the sort of technical gap you’ve got to worry about in the end.”



Jason Mulvin
Director of Enterprise Data at Sophos
SOPHOS

We saw the same discipline everywhere: teams picked their spot on data and tooling as much as on scope. They declared what they had good enough, then proved it on one bounded use case. Perfectionism about your estate is the trap; what matters is rigor on the pilot itself.

The supply-side bottleneck sits inside your function. That makes you accountable for it. It also makes you the one person positioned to clear it.

02

What *good* looks like

Inside the data teams already pulling ahead



What repeated across every successful organization we studied wasn't the size of the team or the budget behind it. Two of the three teams below have fewer than seven people. It was three behaviors: how their teams are structured, how they operate, and how they measure success.

How their teams are structured

The shift is a change in what the job is, not a reorg.

	The old operating model	The *AI-native* operating model
Capacity	Output scales with headcount	**Output scales without it**
The engineer's day	Hand-coding pipelines, source-to-target mapping	**Directing agents, validating output, encoding context**
Where seniority goes	Building and maintaining	**Architecture and business context**
What you hire for	Coding ability	**Problem-solving and orchestration**
Where knowledge lives	In people's heads. It leaves when they leave.	**In the context layer. It stays.**
The real constraint	Engineering hours	**Quality of the business context you supply**

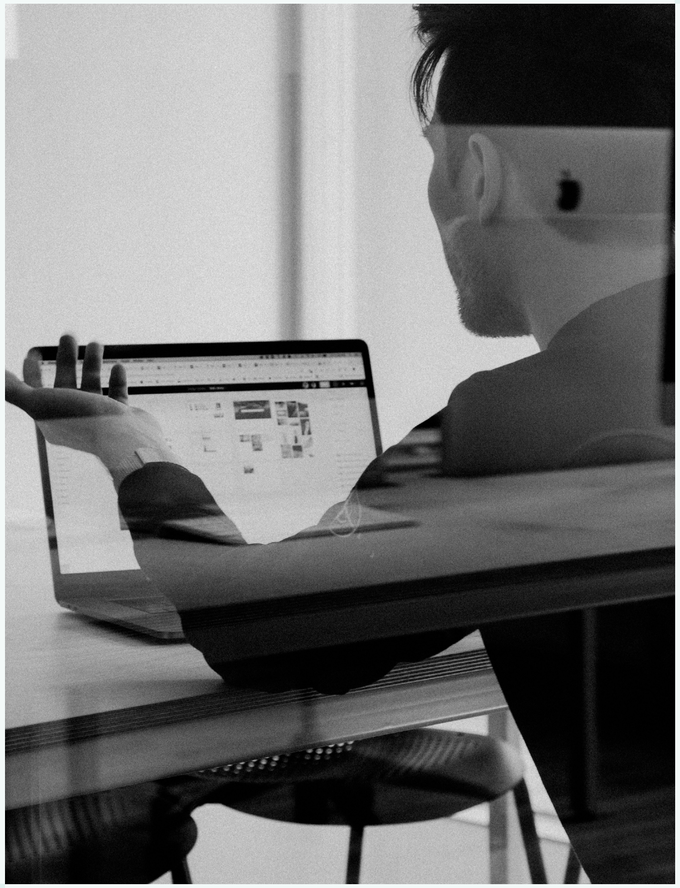
We saw this shift in action across both **commercial** and **enterprise-size** data teams

Edmund Optics supports global marketing, sales, and manufacturing across roughly 34,000 SKUs with two data engineers. Global Analytics Manager, Daniel Adams, had previously been planning to hire more purely to keep pace.

At Sophos, the same shift appears in seniority rather than headcount. Director of Enterprise Data, Jason Mulvin, described freeing principal engineers from the blocking and tackling of ETL development so they can focus on architecture and business context. In the longer term, Thomas Mazzaferro sees the data engineering role splitting in two: the domain data engineer sits in the business and writes its requirements to an agent, and the infrastructure data engineer curates, governs, and serves the context-rich data underneath. His headcount point is blunter: you no longer need 10 engineers to build a pipeline. One or two will do.

Whether or not the role divides that cleanly, the capacity shift is already here. The teams we studied are absorbing work that would previously have needed more engineers, an outside firm, or both.

The question this raises for your team is how much of their time goes to encoding business context rather than hand-coding pipelines, and which of the two you're currently hiring for.



How they **operate**

Context before build. This is where winning teams diverge most sharply from stalled ones.

> “Before you can run agents, you need clean data. Skip it and the agent amplifies whatever’s already there. You automate the confusion, and make the same mistakes you always made, much faster.”



Chris Mihalicz
CEO of Three Point Turn

Source

National Safety Apparel spent about a week and a half mapping their master tables together before building anything. Every field, what it meant across departments, one normalized name everyone agreed on. The map became the agent’s context. Their principle is worth stealing outright: map as a team, then build.

The effect is measurable. On Snowflake’s own evaluation, auto-gathering context lifted agent accuracy from 24% to 83%.

With context-aware agents doing the build, the engineer’s time shifts. Humans stay at defined gates, reviewing under version control while the agent handles scaffolding and setup. And they make peace with a different way of getting the work done. Mulvin’s position at Sophos was that good and consistent output beats output produced exactly the way we’ve always done it.

The fear is that quality slips. Sophos measured the opposite: full standards compliance and 95%+ first-time deployment success, with the quality bar moving up. Insisting the process look familiar is what stalls a rollout; judging it on the result is what gets it shipped.

How they measure *success*

Winning teams count the whole lifecycle net of review, then convert the time into money. What a CFO credits is what that time displaces: consulting and systems-integrator engagements avoided, legacy licenses retired, hires deferred, etc. Across the teams we studied, a master-tables build went from two months to two weeks. A connector two external contractors couldn't deliver was built in an afternoon, avoiding roughly $100K in services. One global energy distribution company eliminated over $300K without a systems integrator. Per-pipeline time fell roughly 4× across all six lifecycle stages, review included.

What it adds up to

What changes when the constraint is removed

- Backlog clears without a hiring program
- Legacy ETL estates migrate in weeks, not 18 months with a GSI
- Engineers govern and validate instead of build and maintain
- AI initiatives unblock because governed data arrives continuously
- Institutional knowledge stays in the platform when engineers leave

This new operating model runs on autonomous, end-to-end data automation: agents that design, build, test, deploy, monitor, and govern data products, while people direct and review. A coding assistant makes one engineer faster. End-to-end automation absorbs an entire category of work.

None of these teams arrived here by accident, and none did it in a weekend. So how do you get there?

Maia by matillion

03

Proof in *90* days

From intent to a business case that already works



Maia
by matillion

When the clock actually *starts*

Section 2 showed the public wins. This one shows the candid questions, objections, and near-dead deals behind them. Start with the timeline, the first place these deals nearly stalled.

90 days can sound aspirational, but our fastest adopter reached quantified results in about six weeks, inside a 12-week pilot that ran start to finish.

The clock starts the moment you're internally cleared to begin. Run governance, legal, security, and funding conversations in parallel with your technical evaluation.

What 90 days actually delivers

Day 30

- Alignment
- An executive path
- The slow gates open and running

Day 60

A proven pilot on a bounded use case, quantified net of review

Day 90

- A secured funding path
- Pricing terms in motion
- A documented scaling plan

Day 0 Governance · Legal · Security · Funding path

What you have at 90 days is a proven pilot, a funded path, and a plan to scale. Not full production. Prototype-to-production averages closer to eight months industry-wide.

> "That is going to take some time, a month or so, to get the necessary approvals to enable this in our environment."
>
> **Data Engineering Leader**, Banking

Maia by matillion

Which AI adopter are *you?*

Across the businesses we studied, the same four adoption personas came up again and again: the Time-Boxed Prover racing a deadline, the Budget-Gated Adopter justifying spend, the Regulated Navigator whose first milestone is governance, and the Lean Multiplier moving on conviction. They don't change the 90 days. They change where the effort concentrates, which gate is most likely to hold you up, and the risks to watch. Each has a phase where it's hardest, and we flag whose it is as we go. Most leaders are primarily one, with elements of the others.

Regulated Navigator

Time-Boxed **Prover**

Budget-Gated Adopter

Lean Multiplier

Maia by matillion

Days 1–30

Shift the mindset, open the gates

The first 30 days are not about tooling. Adopting agentic automation is a change-management exercise before it's a technical one, and what you manage first is perception: what your team believes this change is for.

> "It's not a 'hey, I'm going to cut my team in half.' It's very much a 'hey, I'm going to have half my team spend their time doing something that's more valuable.'"
>
> **Director of Enterprise Data**
> Cybersecurity firm

Map who has to say yes while you're doing it: the economic buyer, security, legal, procurement, and the practitioners whose verdict will decide internal credibility. Days 61–90 runs on that map.

Then open the slow gates. In regulated environments the AI layer triggers a review well beyond normal vendor onboarding, and being an already-approved vendor doesn't earn a pass:

> "Since you're a vendor we're already approved to use, but we're doing a new product, I've got to figure out what the requirements are."
>
> **Data Engineering Leader**
> Consumer Financial Services

Expect to answer for how the tooling handles your data, where it runs, whether your audit trail survives it, and who reviews its output. Those are the four governance gates, and Section 4 works through each one. The teams that got through opened all four on Day 0 and ran them in parallel with the technical evaluation. You can't compress an approval body's lead time; you can only decide when it starts.

Regulated Navigator

Governance isn't a step in your timeline; it is your timeline. Expect closure to land later than the other patterns, and treat that as normal rather than failure.

Time-Boxed **Prover**

Budget-Gated Adopter

Lean Multiplier

Days 31–60

Design a pilot that reaches production

Most pilots die here, however, it’s rarely because of the model. It comes down to the use case, and that’s yours to approve. Don’t sign off on one unless it’s bounded, reversible, and visibly painful to the business: a concrete, high-volume conversion, scoped so that a bad result costs you nothing you can’t undo.

What the first move looked like varied by situation. Three real pilots from our research:

01 A cybersecurity team regenerated already-trusted pipelines and legacy mappings on a 12-week clock, diffed against known-good output and run entirely inside their own perimeter, then scaled it into an 800+ pipeline migration.

02 An energy and utilities team converted a sample of Alteryx workflows and measured build time against the manual baseline, producing the efficiency figure their funding case needed.

03 A banking team, under a regulatory finding, converted marketing-campaign data marts off ungoverned SaaS code onto governed pipelines in a two-week pilot, then scaled it into a month-long rollout as the team’s new governed-pipeline standard.

Regulated Navigator

Time-Boxed **Prover**

A deadline makes the context work look like the obvious thing to cut, but it’s the one thing that determines whether the pilot produces a result worth showing.

Budget-Gated Adopter

Lean Multiplier

Three things then separated the pilots that scaled from the ones that stalled.

01 First, the strongest teams prioritized funding and time for the context work. Teams that skipped it got weaker results and blamed the tooling. The strongest teams pointed the tooling at their existing pipeline metadata and let it draft their standards from how they already build; several bootstrapped their entire context layer that way.

02 Second, the CDO, not the evaluation team, set success criteria up front, in financial terms. “Better than what we have today” is not a criterion, and neither is a raw generation-speed percentage. The bar was the full lifecycle, measured net of review and translated into money.

03 Third, the daily user’s verdict was weighed as heavily as the champion’s. The two rarely experience the tool the same way, and it’s the daily user’s read that decides whether adoption holds.

Regulated Navigator

Time-Boxed **Prover**

A deadline makes the context work look like the obvious thing to cut, but it’s the one thing that determines whether the pilot produces a result worth showing.

Budget-Gated Adopter

Lean Multiplier

Days 61–90

Procure, fund, and plan to scale

A successful pilot is necessary, but not sufficient on its own. In our research, funding the AI layer was a separate hurdle, and frequently a harder one than the technical validation. This phase is almost entirely your work rather than your team's.

Prove it

Every successful team ran a structured playback against the baseline set at the start. One cybersecurity team held theirs roughly six weeks in and recorded the session rather than reschedule around two Out of Office colleagues. Waiting on diaries costs more than the attendance is worth.

Carry it up

What the leading organizations had in common was collateral their sponsor could use without rebuilding it. The pieces varied — a summary, a demo, a set of numbers — but the one that consistently decided the outcome was the ROI framing, already expressed in terms Finance recognized.

In one financial services organization, that packaging is what moved the case out of the data team and in front of the Chief Data & AI Officer who owned the budget, and the deal closed only after a direct conversation between executives on roadmap and commercial terms. When you're committing to a platform your AI roadmap will depend on, that conversation is reasonable to require. You also shouldn't be the only voice behind it when it gets there. One bank ran its evaluation with a pilot lead, a procurement lead, and a quality team that set the validation standards. You still own it. Each function just answers for its own area of expertise.

Fund it

Anchor the cost to something finance already sizes rather than a line item they have to price from scratch. Most often that's a fully-loaded senior data engineer, which is the natural comparison when what you're buying is a digital data team. If you're retiring a legacy estate, the retired license cost is the stronger anchor. Then work the calendar as hard as the case:

Regulated Navigator

Time-Boxed Prover

Budget-Gated Adopter

Lining the funding path up with the budget cycle makes it easier, and it isn't a precondition. A case strong enough to be worth funding gets funded.

Lean Multiplier

“Honestly, it’s not necessarily hard to get funding, but the problem is we’re mid-budget. If it’s something that’s not a line item…”

Data Engineering Leader
Logistics

Mid-cycle rigidity is real, but it’s almost never the reason a good case fails. The same accounts that couldn’t find a line item found the money once they expressed the return in terms finance already tracks. Budget cycles bend when an executive believes in the return — a productivity percentage alone rarely moves them. Where the money genuinely isn’t there this cycle, the route around it is usually procurement rather than finance, and several teams in our research funded through existing commercial arrangements rather than opening a new line. Ask your vendor what’s available before you assume the answer is next fiscal year.

Expect the remaining objections to be commercial rather than technical. In our research the blockers were cost predictability and long-term roadmap rather than capability, and each was resolved through the commercial conversation. Raise them early, because they take longer to work through than a technical question does.

Regulated Navigator

Time-Boxed Prover

Budget-Gated Adopter

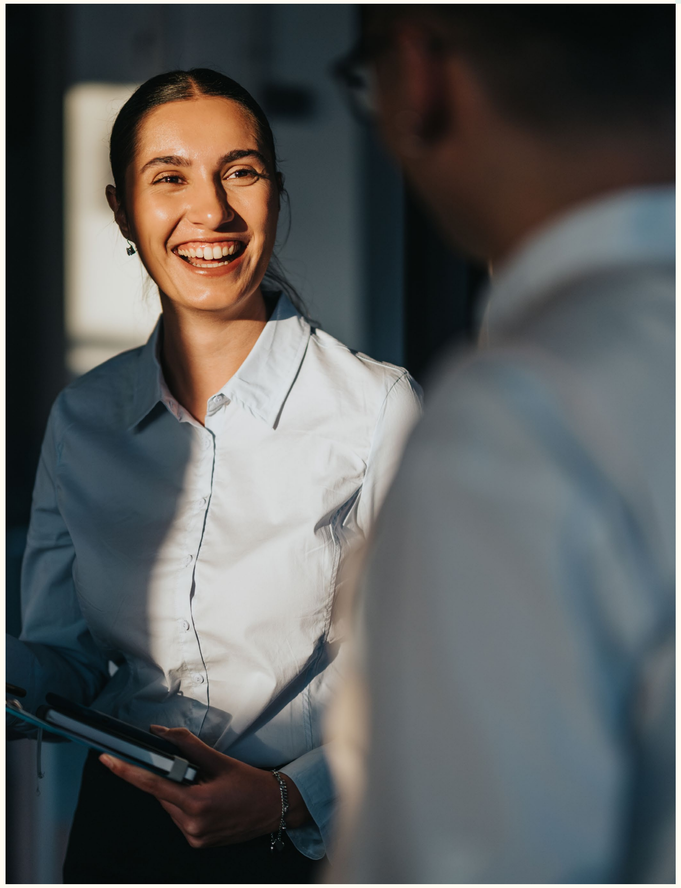
Lining the funding path up with the budget cycle makes it easier, and it isn’t a precondition. A case strong enough to be worth funding gets funded.

Lean Multiplier

Beyond 90

Operationalize and scale

More pilots die between proof and full adoption than anywhere else. Scale what already worked rather than starting something new: the same use case to more users, then adjacent workloads, then the next team. Expand autonomy on the same curve, earned in increments rather than granted by policy: you set where the human review gates sit, and you decide when to move them. Each step costs less than the last, because the context you built for the pilot carries forward.



Regulated Navigator

Time-Boxed **Prover**

Budget-Gated Adopter

Lean Multiplier

You moved fast and cleared governance early. The blocker waiting for you is infrastructure and network friction, so plan for the infra gate rather than the one you've already passed.

04

Built to withstand scrutiny

De-risking the move before your board asks



Maia
by matillion

“Is our context ready?”

Context is the single biggest controllable variable between a strong result and a disappointing one. It’s also the part that sounds like the opposite of moving before you’re ready.

Context-ready doesn’t mean your whole estate is clean. It means the data behind your targeted initial use cases is ready.

The path from raw data to something an agent can use runs through five stages.

01 **Raw**

Captured in the shape the source system needed, not the shape the business means.

02 **Integrated**

Moved and joined. Technically available, but meaning is still supplied by whoever happens to query it.

03 **Governed**

Access, quality and policy applied. Defensible, and still requires a human to interpret what a number means.

04 **Contextualized**

Authoritative definitions, policy constraints, lineage and institutional knowledge travel with the data.

05 **AI-ready data product**

Owned, documented, versioned, service-levelled and machine-consumable. An agent can reason against it without a human intermediary. Organizations with successful AI initiatives invest up to 4× more of their revenue in foundations: data quality, governance, and change management. *(Gartner, 2026)*

Maia by matillion

Most enterprises stop at stage three, where a human still has to interpret what a number means. One test exposes it: if an AI system needed your definition of a profitable customer, where would it find it? For almost every enterprise, the answer is a person, not a governed table, a defined metric, or a single authoritative record an agent could query. Stages four and five move that definition out of someone's head and into somewhere an agent can reach, and skipping them has a measurable cost.



Thomas Mazzaferro
4X Top 100 CDO
Source

Thomas Mazzaferro estimates roughly 80% of AI POCs never reach production because they lack a semantic knowledge layer, and that only about 17% of AI-generated answers to business questions are accurate enough to act on. The cause is data missing the metadata and business context to make it usable, not model hallucination.

So what does being context-ready *actually* take?

- Documented naming conventions and data types
- The architectural patterns your team already follows
- Exportable pipeline metadata
- Agreed definitions where departments disagree
- The tribal knowledge nobody has written down

Context earns you a stronger pilot. The next question decides whether you get to run one at all.

“Is it safe?”

In regulated organizations, this question gets answered before anything else does. Governance stopped being something you apply to the platform. It became part of the platform. As Baris Gultekin, VP of AI at Snowflake, puts it:

> “Governance cannot be bolt-on. We think of AI next to data.”



Baris Gultekin
VP of AI
at Snowflake
Source

Thomas Mazzaferro frames it operationally: the layer that tracks what data you’re using, how it’s used, and who’s using it is the data foundation now.

Open all four questions on Day 0:

01

Does our data train the vendor’s model?

The most common legal trigger in our dataset. Get the contractual answer in writing before the pilot, not during procurement.

02

Where does execution happen?

In security-forward organizations, keeping data inside your own perimeter is the difference between a pilot and a dead end. Sophos cleared their AI council’s restriction on third-party AI by running execution as a Snowflake Native App, so data never left their environment. Jason Mulvin called it a huge win.

03

Does the audit trail survive the new architecture?

Confirm your compliance controls hold before you scale, not after. One food and facilities team found a years-old audit method break under a new platform’s concurrency model, with SOX compliance for financial reporting attached to the fix.

04

Who reviews what, and when?

Document your human-in-the-loop policy. It’s a genuine control, and political cover when someone asks who approved an agent’s output.

Maia by matillion

Regulated organizations go one further. A review gate can't keep pace with autonomous execution, so enforce governance at build time, structurally, so policy stops being a document and becomes a constraint. Gartner expects half of agent deployment failures by 2030 to stem from insufficient runtime governance enforcement.

> "Any new AI capability has to clear our AI council and a vendor security assessment before it goes wider."

Enterprise architect
Global Industrial Manufacturer

Once governance is settled, the conversation shifts to *numbers*

“How will we measure the return?”

The strongest framing here isn’t a number at all. A CFO cares less about what this costs than about how much of the AI investment they’ve already approved is converting into a result, and what improving that rate would be worth. Framed that way, it stops being a new line item and becomes the thing that makes already-committed money pay off.

Everything below is how you deliver that argument in numbers a *CFO can check*

Convert time into money

Apply the percentage to the engineers it affects and price it at their loaded cost. Even at the low end of a productivity range, the recovered capacity covers the platform on its own. That’s before you count consulting avoided, hires deferred, or roadmap items shipped.

Lead with the hardest number you have

If you’re carrying a legacy estate, that’s a retired license: an existing cost line disappearing on a date you can name, where recovered capacity is only ever a projection. Without one, avoided spend does the same job — the consulting engagement you no longer need, the hire you can defer — because both are decisions someone can point to rather than forecasts.

Measure net of review, across the full lifecycle

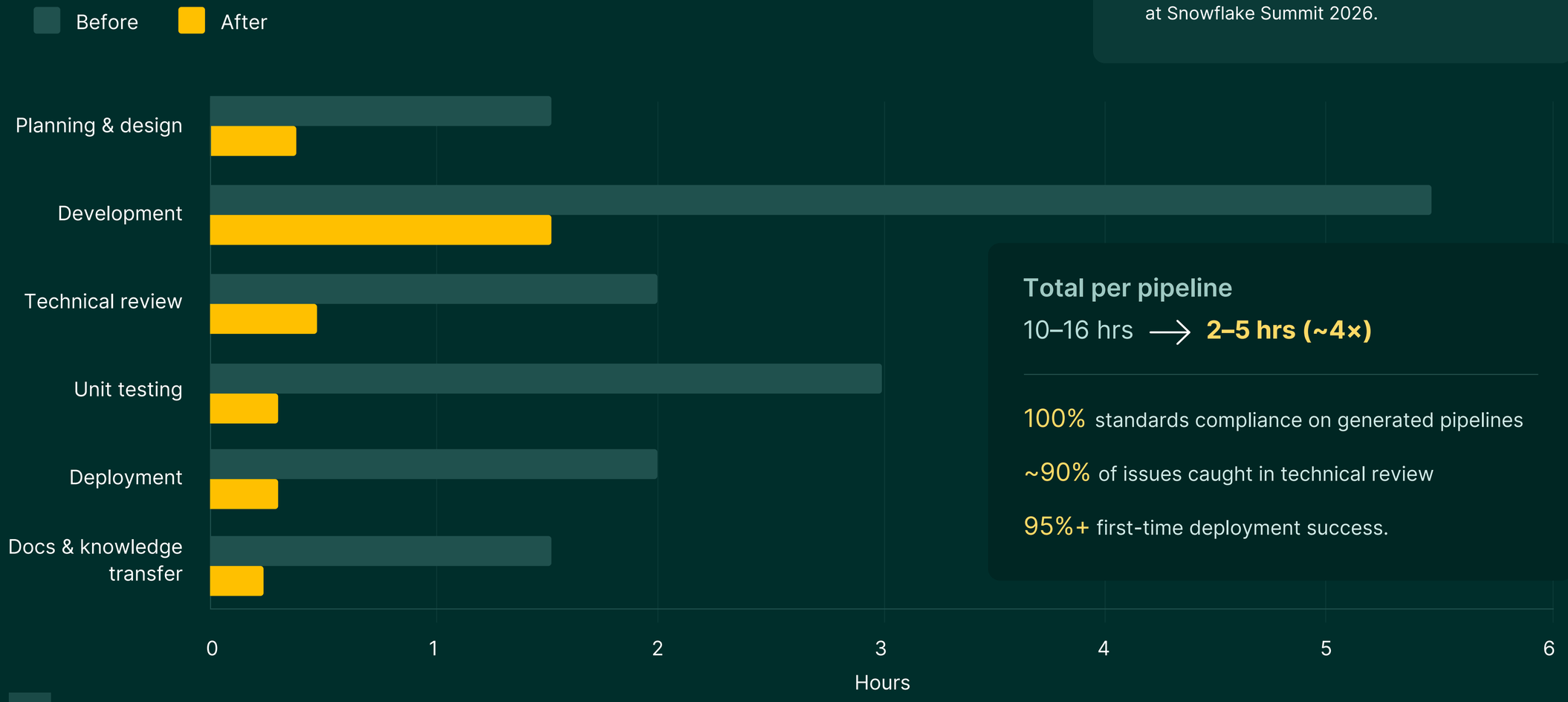
A fast build lost to hours of senior validation is not a saving, and the gains sit in design, documentation, and testing as much as code generation. As Jason Mulvin, Director of Enterprise Data at Sophos, put it: **“It’s really the full lifecycle. The design, the development, the documentation, and the testing. That’s where the gains are.”** Measure only code generation and you understate your own case.

Present the conservative end

A case that survives at its low end is the one that holds.

Time saved across the pipeline lifecycle

Faster, with the quality bar moving up.



Source:
Sophos, presented these findings at Snowflake Summit 2026.

Total per pipeline

10–16 hrs → **2–5 hrs (~4×)**

100% standards compliance on generated pipelines

~90% of issues caught in technical review

95%+ first-time deployment success.

"Why this, and not the AI we already have?"

This one usually comes from a technical architecture review rather than the board.

"We've seen quite an advancement in the warehouse vendors recently. Their assistant is really a language model on top of their whole ecosystem. Where's the differentiating value over a tool we already have?"

Director of Software Engineering
Global Industrial Manufacturer



Match the tool to the job rather than forcing an either/or. Sometimes the two run together, sometimes you'll choose one over the other. One logistics team in our research runs both: the warehouse agent for its broad understanding of their data and ecosystem, and the automation platform to build production-grade pipelines, then document and deploy them. Leading with where the two tools fit together is what survives an architecture review, because it doesn't force anyone to give up something they already value.

Every one of those four questions came out of a real evaluation. Answer them in advance, and none of them will stop you.

Two modern agents. *Different jobs.*

	A coding assistant	Autonomous end-to-end automation
What it does	Generates code faster	Designs, builds, tests, deploys, monitors, governs
Who gets faster	One engineer	The delivery model
What's left to you	Wiring, documentation, deployment, monitoring	Review and direction
Which bottleneck it clears	Consumption	Supply

Consumption was never the bottleneck.

05

The Compounding Cost

Why every day of hesitation costs more than the last



The gap doesn’t *grow* in a straight line

Defending the move to your board is one half of the decision. The other half is the cost of not moving, and unlike the risk of acting, it doesn’t hold still.

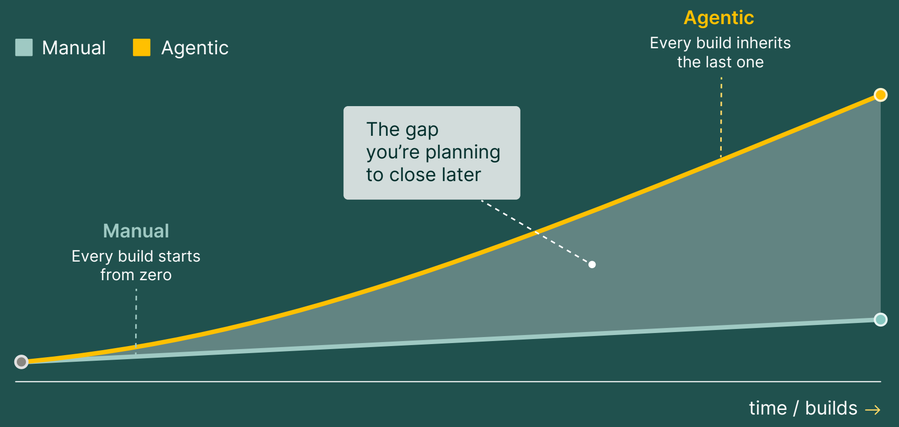
Most cost-of-delay arguments treat the gap between movers and waiters as a fixed distance, as though you could close it later by working harder. You can’t, because the two kinds of work accumulate differently.

Manual data work does not compound. Every pipeline your team hand-builds starts from roughly zero. The tenth build is barely faster than the first, because the knowledge lives in individual heads and has to be reassembled each time an engineer leaves, joins, or forgets.

Agentic work compounds. Every standard you document, every naming convention you encode, every architectural pattern you capture makes the next build faster than the last. The context layer is an asset, and it appreciates. One team we studied spent about three of every ten project days building and the other seven understanding the business. It sounds inefficient right up until you realize the understanding is permanent and the build is not.

The compounding gap

Two diverging lines over time



Organizations with successful AI initiatives invest up to 4× more in foundations. That investment is what bends the curve.

You aren’t standing still while others move. You’re building nothing while they build an asset.

Maia by matillion

The gap doesn't *grow* in a straight line

The field is still open, and that won't last

The statistic we started with: fewer than 10% of enterprises have scaled agents to tangible value, while roughly two-thirds have only experimented. Most of your competitors are exactly where you are: convinced, nervous, and stalled. That won't hold for long, and which side of the split you end up on is still yours to decide.

You don't have to rip anything out

The most common objection is that the legacy estate works and replacing it is a multi-year program nobody has budget for. It doesn't have to be. At one FTSE-listed infrastructure firm, a migration scoped as a multi-year systems-integrator program became ongoing work the existing team absorbed: legacy logic that had taken a senior engineer a week to reverse-engineer was analyzed in six minutes.

An imperfect start is not a failed one

The leaders who moved early did not get clean results immediately. Their agents didn't follow every standard, usually because they hadn't been given full context and the gaps weren't yet obvious. Their teams used a product that was still shipping features. Their first pilots surfaced data quality problems they hadn't known they had.

None of that is failure. It's the organizational learning curve, and it's the same curve you'll climb whenever you start. The only variable is whether you climb it before or after the people you compete with.



Maia by matillion

The teams pulling ahead didn't move because they were certain. They moved because they understood that certainty was never coming, and the *cost of waiting for it compounds daily*.

06

Your move

The guide ends and your advantage begins.



The mandate inverted and the accountability landed on *you*

Teams already ahead run lean, put context before build, and measure net of review. 90 days of parallel work gets you a proven pilot, a funded path, and a plan to scale. Four questions de-risk it with your board. And every quarter you wait, the teams ahead compound an advantage you can't buy back.

Five factors decided every adoption we studied, regardless of industry, size, or how they bought.

01

Executive engagement unlocks the hard deals

The biggest commitments closed on a senior-leader-to-senior-leader conversation, often the one you're best placed to open. Build that path early, before you need it.

02

Governance is the dominant timeline driver

Not the technology. Start it on Day 0, run it in parallel, and get the data-training answer in writing.

03

Funding the AI layer is its own hurdle

A strong pilot doesn't guarantee it gets funded. Secure the path before any trial clock expires.

04

Context determines whether the tooling delivers

Every other factor on this list assumes you've already got it right.

05

You and your data engineers will read it differently

Your experience rarely matches the daily user's, and theirs is what decides whether adoption holds.

Underneath all five sits the idea this guide opened with. AI does not create advantage by being deployed. It creates advantage by being supplied, and the supply side is the part of your AI roadmap nobody else is resourcing. The organizations that succeed here aren't luckier or better funded. They follow a different discipline, and they start sooner.

Both are choices you can make this quarter.



About *Maia*

Maia is an AI Data Automation platform powered by autonomous AI agents that build, maintain, and evolve data products, thus eliminating manual data work.

Maia empowers CDAOs and enterprise data teams to deliver data products at machine scale while maintaining governance. Its integrated platform combines specialized agents in Maia Team, grounded in the organizational data intelligence of the Maia Context Engine and executed through the governed data tools of Maia Foundation.

Organizations like Sophos, EDF, and St. James' Place use Maia to automate data work at scale, modernize platforms, and accelerate AI roadmaps without expanding headcount.

Explore Maia.ai



Maia by matillion