SmartCo

The Delivery Intelligence Dividend

A five-pillar framework and 90-day plan for turning isolated AI activity into governed, repeatable delivery outcomes, for CIOs and Programme Directors.

Second edition paper following the AI in Delivery Roundtable for financial services leaders, London, August 2026.

By Sahar Khan, Transformation and AI expert in financial services and **Callum Campbell**, Head of Clients & Growth at SmartCo.

October 2026



Table of contents

Executive summary **4**

1. What changed since the Gap **7**

1a. The value question has closed 7

1b. The constraint has moved to data, controls and operating model 7

1c. Delivery models are visibly lagging the technology 8

1d. Adoption is a confidence problem, not a training problem 11

2. From debt to dividend **12**

2a. The compounding test 13

3. The environment is now the constraint **15**

3a. The DevOps moment, again 15

3b. Architecture, not bolt-ons 16

3c. The approval clock is the real constraint 18

3d. Memory, loops and graphs 20

4. The Delivery Intelligence framework **22**

Pillar 1 — Intelligence Foundation 23

Pillar 2 — Continuous Assurance 24

Pillar 3 — Governed Autonomy 24

Pillar 4 — Delivery Memory 27

Pillar 5 — Capability and Confidence 28

5. The CIO and Programme Director compact **29**

6. Measuring the dividend **31**

6a. The first 90 days 32

6b. What comes next 33

7. The open question: evolution or replacement **35**

References 37

About this paper 38

SmartCo 39

About the authors 40

Executive summary

> AI has already proven it creates value in delivery. What remains unproven is whether the delivery model you inherited, and the environment you have bolted it onto, can hold that value.

The first paper in this series, The Delivery Intelligence Gap, argued that organisations were accumulating Delivery Intelligence Debt: the widening distance between how much AI activity is happening inside programmes and how little of it is converted into structured, governed, reusable insight. That paper diagnosed the debt. This one is about collecting the return.

Since publication we have tested the argument with senior technology leaders across the sector, including at SmartCo's AI in Delivery Roundtable, which we co-hosted and which brought together leaders from institutions including MUFG, NatWest, HSBC, LSEG, Irish Life Group and ClearToken. The evidence from that room has changed this paper materially, and we have rebuilt it accordingly.

Three things came through with unusual consistency. First, the value debate is over: leaders have stopped asking whether AI can create value and started asking how to embed it at scale. Second, the constraint has moved. It is no longer model capability but the surrounding architecture: fragmented data, governance designed for a slower world, and operating models that lag the technology by several years. Third, and most uncomfortably, the delivery models most organisations run were not built for a world in which AI participates in the work rather than merely assisting it.

Those three findings point at two things that now have to be designed together. The first is the delivery function: how programmes are governed, evidenced, remembered and staffed once AI participates in the work. The second is the technical environment: the architecture, data foundation, controls and memory within which that AI is built, approved and run. What struck most in writing this edition is how

similar the two challenges are. An environment good enough for AI to work in and an environment good enough for people to work in are made of the same materials: architecture before capability, foundations before bolt-ons, controls that permit change rather than prevent it, and memory that compounds rather than resets. Where this paper prescribes, it does so in two streams, marked Delivery function and Technical environment, side by side.

What this paper sets out

- **A definition of the dividend** that separates the three returns AI delivery pays — velocity, assurance and memory — and explains why most organisations are collecting only the first.
- **An argument that the environment is now the constraint** — why AI has handed programme teams the power to build and run their own technology, why that power sprawls into an unusable estate unless it is architected, and why the approval clock, not the model, is the real limit in a regulated institution.
- **A five-pillar view of the Delivery Intelligence framework** — Intelligence Foundation, Continuous Assurance, Governed Autonomy, Delivery Memory, and Capability and Confidence — each set out in two streams, delivery function and technical environment.
- **An explicit compact** between the CIO and the Programme Director, setting out who owns what when AI is inside the delivery loop — including who owns the environment.
- **A measurement model** that replaces activity metrics with six intelligence metrics, and a 90-day path to standing the framework up.
- **A statement of the open question** the roundtable could not settle: whether this is an evolution of existing delivery methodology or its replacement. Section 7 takes a position on it, and paper three will set the answer out as a method.

What this paper does not argue

- That delivery should pause while a platform is built. The environment is what makes the dividend compound. It is not a prerequisite for starting, and the 90 days in section 6a begins on two programmes with what you already own.
- That AI in delivery needs a separate control regime. The consistent regulatory signal is that AI will be supervised through the frameworks institutions already run: the FCA's long-term review of AI in retail financial services, opened in January 2026, restated that it does not plan AI-specific rules while acknowledging that supervisory frameworks may need to evolve as systems become more autonomous, and that position stands as at September 2026. The autonomy ladder maps onto those frameworks rather than sitting beside them.
- That the published productivity gains will transfer. They were largely measured on greenfield work. On a regulated legacy estate they will be smaller, and a business case built on the headline figures will not survive its first benefits review.
- That any particular tool, platform or vendor is required. Every recommendation here can be met with systems most institutions already hold. What is missing is the architecture connecting them and the decision about who owns it.

> Every organisation in the room could point to AI improving programme assurance, accelerating compliance work and sharpening decisions. Almost none could show that improvement compounding from one programme to the next. That difference is the dividend.

1. What changed since the Gap

The Gap was written from practitioner experience and desk research. This edition has a second evidence base behind it: a structured, invitation-only discussion among senior technology leaders in financial services, facilitated under conditions that encouraged candour about what is not working rather than showcase accounts of what is.

Four shifts are worth recording, because each one changes what a delivery framework now has to do.

1a. The value question has closed

A year ago, the dominant executive question was whether AI could produce defensible value inside a regulated delivery environment. That question has been settled in practice, not in theory. Leaders described AI already enhancing programme assurance, accelerating compliance activity, improving the quality of decision-making, and supporting innovation work. The proof points exist.

What has replaced it is a harder question about scale: how do you move from a portfolio of promising experiments to repeatable, measurable outcomes? Experimentation is cheap and locally rewarded. Repeatability requires foundations that nobody is individually incentivised to build.

1b. The constraint has moved to data, controls and operating model

Three constraints were named repeatedly, and always together: fragmented data foundations, the absence of appropriate governance for AI-assisted work, and operating models that have not adapted. They are usually treated as separate workstreams owned by separate functions. They are in fact a single problem with three faces, and attacking any one in isolation produces the pattern I described in the first paper: high activity, low retained intelligence.

The public evidence points the same way. A survey of 142 enterprise technical leaders published in February 2026 and updated this month found data privacy, governance and regulatory compliance to be the most-cited blocker to AI adoption at 61%, ahead of both accuracy risk and legacy integration complexity; the authors conclude that the binding constraint is organisational and regulatory readiness rather than model capability. McKinsey's 2026 AI Trust Maturity Survey, published in March across roughly 500 organisations with direct responsibility for AI governance or investment, describes the same transition from experimentation towards scaled and increasingly agentic deployment.

61%

of enterprise technical leaders name data privacy, governance and regulatory compliance as their leading blocker to AI adoption — ahead of accuracy risk and legacy integration.

> **"AI is the future, but it is only built on foundations of core data governance and management."**
>
> Tony Gill, Director of Data & Analytics, Irish Life Group

1c. Delivery models are visibly lagging the technology

The most direct challenge from the room was that many of the delivery models in use today simply were not built for an AI-powered world. Technology has moved quickly; operating models, processes and ways of working have not. Stage-gate governance was designed around one assumption: that producing evidence is expensive, so you sample it periodically at gates. That assumption has quietly stopped being true now that evidence can be generated continuously, at near-zero marginal cost, and the design has not caught up.

The most rigorous public evidence for that lag now sits in the delivery data itself. Google Cloud's DORA research on the return on AI-assisted software development describes a J-curve: an initial dip in productivity while teams absorb what it calls the verification tax — the additional effort of checking whether AI-generated output is reliable, secure and architecturally sound. Its central argument is that AI amplifies existing organisational strength rather than substituting for it, and that rising individual effectiveness is accompanied by rising delivery instability wherever pipelines and review gates cannot absorb the extra volume. Atlassian's Teamwork Lab reached the same conclusion independently in its 2026 workforce research, naming it an AI efficiency paradox: faster individual output simply queues at reviews and approvals.

Stanford research cited in the DORA report sharpens the point for institutions like ours. Productivity gains run at 35–40% on simple greenfield work, but often 10% or less on complex legacy code — which is precisely the estate most financial institutions are changing. Anyone building a business case on headline productivity figures drawn from greenfield studies is budgeting for a benefit their estate will not produce.

> **"The delivery models we use today simply weren't built for an AI-powered world. Giving people better tools is only part of the answer. We also need to rethink how work gets done, how decisions are made and how we give teams the confidence to use AI without losing accountability. Otherwise, we risk producing more, faster, only to leave it waiting for the same reviews and approvals. The real opportunity is to build a delivery model where better evidence supports better decisions, and what we learn on one programme makes the next stronger."**
>
> Oli Kane, Managing Director, SmartCo

EXHIBIT 1

AI-assisted delivery follows a J-curve: performance dips before it returns

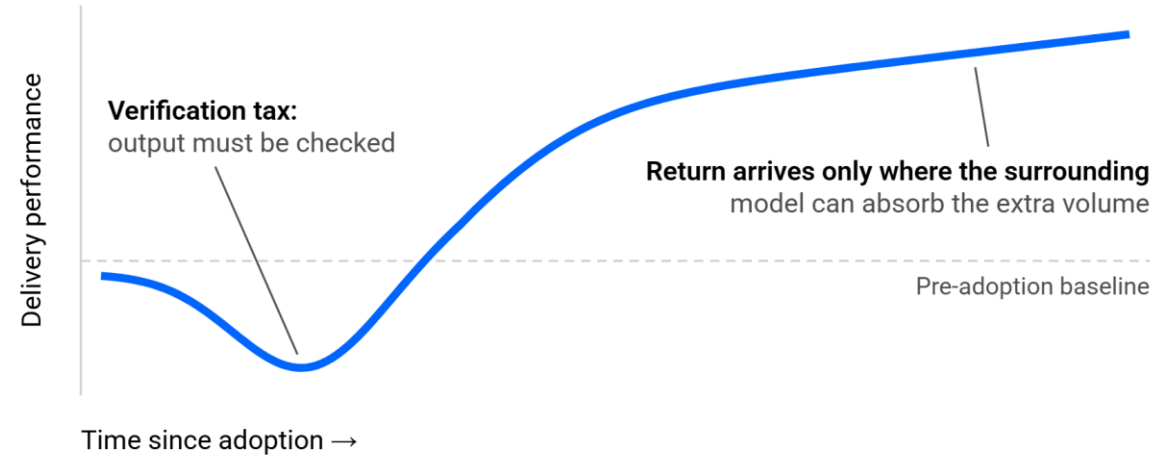


Figure 1. Illustrative, after Google Cloud DORA, “The ROI of AI-Assisted Software Development” (2026).

EXHIBIT 2

The gains concentrate where the work is simplest — not where financial institutions work

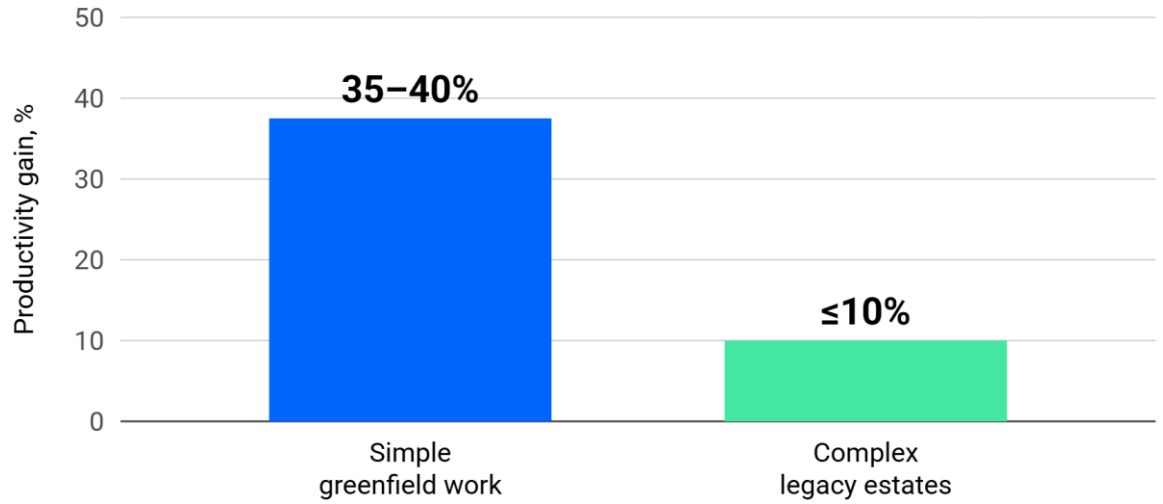


Figure 2. Source: Stanford software engineering productivity research, cited in DORA (2026).

1d. Adoption is a confidence problem, not a training problem

The final shift is the one most often under-resourced. Success will not come from technology alone; organisations must bring people with them and build genuine confidence in how AI is used. In regulated delivery, a practitioner who does not trust an AI-produced artefact will quietly reproduce it by hand. The cost of that shadow work is invisible in every dashboard we have seen, and it is the single largest silent drag on the dividend.

Source: SmartCo AI in Delivery Roundtable, London, August 2026; co-hosted by the authors and facilitated by enterprise CIO Andy Richards. Discussion themes summarised with participants' agreement; no individual attribution.

2. From debt to dividend

Delivery Intelligence Debt accrues whenever a programme generates insight it cannot retain in governed, reusable form. The dividend is what an organisation earns once it stops accruing that debt and starts compounding the intelligence instead.

It pays in three distinct currencies and conflating them is why so many AI delivery business cases disappoint.

EXHIBIT 3

AI delivery pays three returns — most organisations collect only the first

Return	What it is	Why most stall here
Velocity	Faster production of delivery artefacts: plans, papers, test evidence, impact assessments, regulatory mappings.	It is the easiest to capture and the easiest to measure, so organisations bank it and stop. Velocity alone leaves the debt untouched.
Assurance	Continuous, population-level evidence in place of periodic sampling — assurance that runs alongside delivery rather than interrupting it.	Requires the control framework to accept machine-generated evidence, which is a governance decision, not a tooling one.
Memory	Delivery experience captured as structured, queryable institutional knowledge that improves the next programme's starting position.	Nobody owns it. It sits between the PMO, the CIO function and the risk function, and falls between all three.

Figure 3. Source: authors' analysis.

Velocity is a productivity gain. Assurance is a risk-adjusted gain. Memory is the only one that compounds. An organisation collecting velocity alone is running faster

through the same terrain each time; an organisation collecting memory arrives at the next programme already knowing where the ground is soft.

EXHIBIT 4

Velocity plateaus. Only delivery memory compounds across programmes

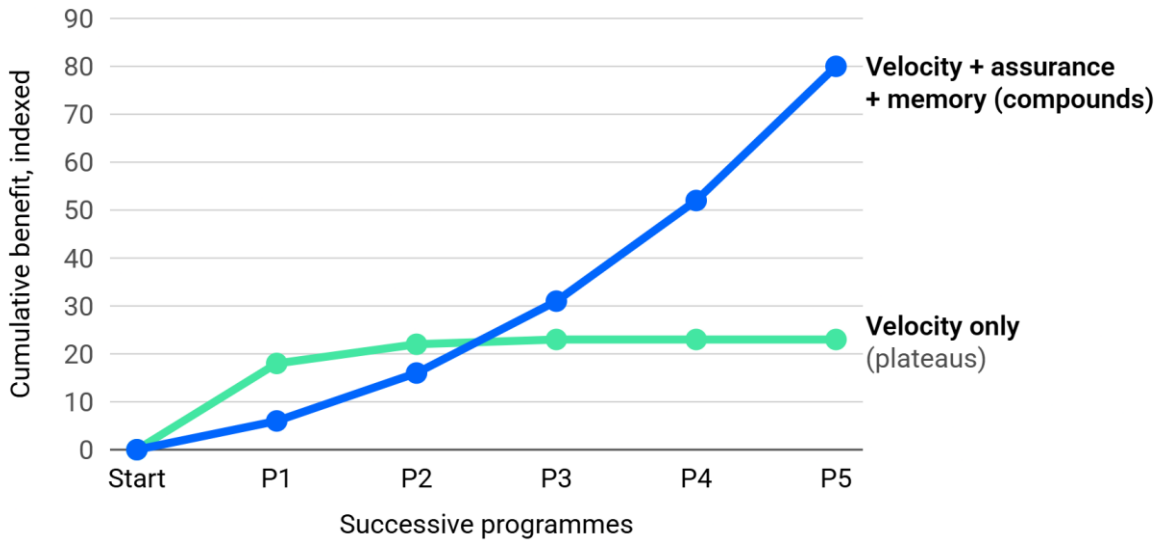


Figure 4. Illustrative. Indexed cumulative benefit across five successive programmes.

> AI hands you productivity for free. Intelligence — the kind that compounds — you still must build yourself.

2a. The compounding test

There is a single diagnostic I now recommend to every CIO before any framework discussion begins. Take two comparable programmes, one that closed 12 months ago and one starting next month. Ask what the new programme starts with that the old one did not. Not in tooling, but in structured knowledge: known risk patterns, reusable control mappings, calibrated estimates, evidenced assumptions.

If the honest answer is nothing beyond individual memory and a document repository, the organisation is at Stage One of the maturity model regardless of how

much AI it has deployed. The gap is architectural, not a failure of ambition, and the rest of this paper is about closing it.

3. The environment is now the constraint

Sahar's argument so far has been about the delivery model. Mine is about what it runs on. I come at this as someone who builds: over the past 18 months I have built production-grade systems for delivery leaders using AI coding tools, inside and alongside Tier-1 institutions. That experience has convinced me of one thing the roundtable circled but did not quite land on. The question has stopped being what AI can do. It is what environment you are asking it to do it in.

3a. The DevOps moment, again

Fifteen years ago a similar shift ran through technology operations. DevOps and agile did not primarily give BAU teams new tools; they gave them permission, and the means, to build and run their own future. Teams that had waited months for a central function to change a system began to change it themselves, continuously, inside guardrails they had helped design. The organisations that did well did not simply hand out the tools. They built the platform, the pipelines and the controls that made that autonomy safe, and they redrew the operating model around them.

Programme and change teams never had that moment. Their structure — the PMO, the stage gate, the steering committee, the plan in one tool and the risks in another — has been essentially unchanged for 30 years while the technology beneath it has changed several times over. AI is now doing to programme delivery what DevOps did to operations: it hands the delivery lead the ability to specify, build and run technology directly. A programme director can describe a dashboard, an assurance check or a data reconciliation and have a working version by the afternoon.

That is the power. The similarity of the challenge is what should give us pause. DevOps succeeded where it was treated as an architecture and operating-model problem, and produced sprawl where it was treated as a tooling one. Programme delivery is about to make the same choice, at higher speed and inside a regulated perimeter.

EXHIBIT 5

The DevOps shift and the AI shift make the same demand of the organisation

	Then: BAU teams and DevOps	Now: programme teams and AI
Who gained the power to build	Operations and application teams	Programme, PMO and change teams
What made it safe	Platform, pipelines, guardrails, automated tests	The AI environment: data foundation, harness, controls, memory
What happened without it	Snowflake servers, shadow tooling, estates nobody could patch	Bot graveyards, agent sprawl, unmaintained “helpers” with their own copies of the data
The operating-model change	Product teams; you build it, you run it	The compact in section 5, with a named owner for the environment
What everyone underestimated	Culture and confidence	Approval time and architecture

Figure 5. Source: authors’ analysis.

3b. Architecture, not bolt-ons

Every institution in the room in August had AI in live use, and in most of the firms we work with the strategic model decisions are already made. What has not been built is the environment those models work in: the tools they can call, the delivery data they can see, the memory they carry, and the way they recover when they get it wrong. There is suggestive evidence from outside financial services that this is now where performance comes from. NVIDIA’s August 2026 comparison of two agent systems running on the same underlying model found that one needed roughly 12% more actions to complete the same benchmark, and attributes the difference to harness design — persistent memory, tools, feedback and recovery. NVIDIA is explicit that this is a cross-system comparison rather than a controlled ablation, and a public

games benchmark is not a delivery estate. Read it as a pointer, not a proof; the proof that matters is in the two sub-sections that follow, and it comes from the approval process of a real institution. But note what that environment needs: connected data, clear controls, a memory, and a way to recover from mistakes. It is the same list your people need.

The same conclusion is reached from the engineering side. DORA's finding on returns is that they come from the underlying organisational system — the quality of the internal platform, the clarity of workflows, the alignment of teams — rather than from the tools, and that without that foundation AI produces local pockets of productivity that are lost to downstream chaos. Section 1 used that to explain the J-curve. It is worth reading a second time as a statement about the environment: the platform is the variable, and it is the one most institutions have not touched.

The instinct when a tool can be built in an afternoon is to build one for each problem. A tracker here, a summariser there, a reconciliation script for the thing that hurt last week. Each is locally sensible. Together they are a new estate: dozens of small systems with their own logic, their own copies of the data and no shared identity for the things they describe. Estates like that demolish adoption, because nobody trusts a number that lives in five places. Every programme manager has watched this happen with spreadsheets. It happened again with low-code and robotic process automation, which left many institutions with hundreds of orphaned bots. AI compresses the time to reach that point from years to months.

The alternative is to treat the environment as architecture from the start. In practice that means a foundational data layer before any capability: the delivery record — plans, risks, decisions, dependencies, controls, resourcing, spend — drawn from the systems that already hold it (planning and PPM tools, ticketing, collaboration platforms, finance and HR systems, the architecture repository) into one governed layer with one identity for each thing. Capabilities are then built on that layer, not beside it. This is the environment-side expression of Sahar's Intelligence Foundation: the pillar and the platform are the same decision viewed from two chairs.

This is also the honest answer to the claim that systems built with AI assistance are inherently poor. Some are — but the failure is never in who did the typing. It is in whether anyone held the architecture: whether the system was structured so that it could be changed without breaking, scaled without rewriting, and worked on by more than one person. A programme director who wants to build does not need to learn to code. They need to approach the environment with architecture in mind, and to have someone accountable for it who does.

- **A foundational data layer** with a shared identity for programme, decision, risk, control and dependency, fed from the systems of record rather than re-keyed.
- **A harness, not a set of tools:** a shared layer for permissions, model access, memory, logging and recovery that every capability inherits rather than re-implements.
- **Configuration over code:** an environment designed so that most change is a permitted configuration inside an approved envelope, not a new system requiring a new approval.
- **Model agnosticism:** the models will be swapped several times over the life of the environment, and nothing in the architecture should depend on which one is behind it.

3c. The approval clock is the real constraint

Here is the part the speed narrative leaves out. A programme leader can make a decision tomorrow, and it is now widely assumed that AI can act on it at the same speed. In a regulated institution that is not true, and pretending otherwise is what produces shadow AI.

Consider the route to production at one Tier-1 UK market infrastructure group, which is typical rather than exceptional. A new system must first be registered in the enterprise architecture repository and pass a quality approval. It is then registered in the service-management platform and the configuration database, which requires the architecture record to have been approved at least a day earlier. Only then can the team onboard to the engineering platform — source control, secrets management,

pipeline templates, identity and access — before building under mandatory guardrails: compliance checks, static analysis, rotation of secrets. Go-live then requires solution design documents, security, cloud and network reviews and a governance-board decision, with any integration to the engineering platform first passing an architecture review forum. None of that is unreasonable. All of it takes weeks at best, and it takes weeks per system.

EXHIBIT 6

Every bolt-on pays the approval cost again. An architected environment pays it once

Step	What it requires	Bolt-on tools	Architected environment
1. Architecture registration and quality approval	Factsheet, ownership, data classification	Once per tool	Once, for the environment
2. Service-management and CMDB registration	Support group, application services, an approved factsheet	Once per tool	Once
3. Engineering platform onboarding	Source control, secrets, pipeline templates, identity and access	Once per tool	Once
4. Build under guardrails	Compliance checks, static analysis, secrets rotation	Re-implemented each time	Inherited by every capability
5. Go-live gates	Design documents; security, cloud and network reviews; governance board	Once per tool, weeks each	Once for the envelope; changes inside it follow a pre-agreed lighter path

Figure 6. Source: authors' analysis, based on the documented route to production at a Tier-1 UK market infrastructure group (anonymised).

This is why bolt-ons fail twice: once at the estate and once at the gate. Every new tool restarts the clock. An environment approved once, with a defined envelope of what

may be changed within it and by whom, is the only way to give a programme anything like the responsiveness the speed narrative promises. It is also why the environment question and Sahar's autonomy ladder are one question. What an agent may do inside the environment, and what the environment may change without a fresh approval, are the same control expressed in two vocabularies.

A dedicated environment that permits rapid change is not something a regulated institution will stumble into, and it will never arrive from a portfolio of experiments. It comes from planning: understanding your own approval processes well enough to design an environment that satisfies them once and understanding your own delivery processes well enough to know which changes need to be fast. Done properly, AI then gives you a system that can be reconfigured in days. Done as bolt-ons, it gives you a queue.

3d. Memory, loops and graphs

Sahar's fourth pillar is delivery memory: the institution remembering what its programmes learned. From the environment side that has to be engineered, and two disciplines from systems engineering apply directly.

The first is loop engineering. Memory is not a repository; it is a loop. Every AI-produced artefact should write back what it drew on and what happened next. Every assurance finding, every reversed decision and every estimate variance should return to the environment as structured feedback rather than sit in a close-down report. The NVIDIA result is, at bottom, a result about loops: feedback and recovery are what let the system keep going over a long horizon. The same is true of a programme team.

The second is graph engineering. Delivery knowledge is relational. A risk belongs to a workstream, threatens a milestone, is owned by a person, was raised in a decision and is mitigated by a control. Held as rows in separate tools, those relationships are lost. Held as a graph — entities and the connections between them — they become the thing an AI can reason over and a new programme can interrogate at mobilisation. The delivery ontology Sahar describes is, in engineering terms, a graph schema. Build it as one.

And gather the team's memory deliberately. The tacit knowledge of a delivery team — why the vendor was chosen, what the regulator actually objected to, which dependency always slips — leaves with the people unless the environment gives them somewhere to put it and a reason to do so. Instrument the capture at source, in the tools the team already uses, rather than asking for it at close-down when everyone has moved on.

> The model is a commodity. The harness, the data beneath it and the discipline to change it safely are the asset — and they are the same asset your people need.

4. The Delivery Intelligence framework

The framework has five pillars. They are deliberately ordered: each one is difficult to stand up without the one before it, and organisations that attempt pillar three before pillar one reliably produce controls nobody trusts.

In this edition each pillar is set out in two streams. The delivery function stream is what governance, roles and practice must do. The technical environment stream is what the platform beneath must provide. The order matters in both: attempt Governed Autonomy before the environment can log and reverse what an agent did, and you have controls nobody can evidence.

EXHIBIT 7

Five pillars, in the order they have to be built

Pillar	The question the delivery function must answer	What the technical environment must provide
1. Intelligence Foundation	Is delivery data structured, connected and trustworthy enough for AI to reason over?	A foundational data layer with one identity per entity, fed from systems of record and held as a graph
2. Continuous Assurance	Can we evidence control effectiveness continuously rather than at gates?	Controls as executable checks in the harness; lineage on every artefact by default
3. Governed Autonomy	Where AI acts rather than assists, who is accountable and how is that provable?	Permissions, audit and reversal built into the harness; an approved change envelope
4. Delivery Memory	Does what we learn on one programme measurably improve the next?	Write-back loops from every artefact and finding; capture at source

Pillar	The question the delivery function must answer	What the technical environment must provide
5. Capability and Confidence	Do practitioners trust the output enough to stop duplicating it?	One environment, one answer: consolidated, consistent, with visible lineage

Figure 7. Source: authors' analysis.

Pillar 1 — Intelligence Foundation

Fragmented data was the most frequently named blocker in the room, and delivery data is usually the worst-governed data an organisation holds. Plans, risks, decisions, dependencies and assurance evidence live in different tools with different owners, different taxonomies and no shared identifier for the thing they all describe.

Delivery function

- A single delivery ontology: one definition each of a programme, a decision, a risk, a control and a dependency, applied across the portfolio.
- Decision records as first-class artefacts — what was decided, by whom, on what evidence, and what was expected to follow.
- Lineage on every AI-generated artefact: which inputs, which model, which version, which human review.
- A minimum-viable scope. Do not boil the portfolio. Two programmes fully instrumented beat 40 partially.

Technical environment

- Build the ontology as a graph schema, not a set of tables. The relationships are the intelligence.
- Feed the layer from systems of record — PPM and planning tools, ticketing, collaboration, finance, HR, the architecture repository — never by re-keying.
- Take the layer through the route to production once, as the environment, so that capabilities built on it do not each restart the approval clock.

Pillar 2 — Continuous Assurance

Stage-gate governance samples evidence because evidence used to be expensive. It no longer is. Continuous assurance moves from testing a sample of artefacts at a point in time to testing the whole population continuously, with the gate reduced to a decision forum rather than an evidence-production exercise.

Delivery function

- Controls expressed as testable assertions against the delivery record, not as narrative sections in a pack.
- Exception-based escalation: the gate reviews what failed, not what was produced.
- Assurance findings written back into the delivery record, so they inform the next programme (this is where pillars 2 and 4 join).

Technical environment

- Express controls as executable checks the harness runs against the delivery record, with results logged automatically.
- Give every AI-generated artefact its lineage — inputs, model, version, reviewer — as a property of the environment rather than a task for the author.

This is the pillar with the highest regulatory sensitivity and the highest return. It also requires second-line and internal audit engagement from the first week, not at the point of rollout. Every organisation we have seen skip that step has had to rebuild.

Pillar 3 — Governed Autonomy

The distinction that matters is between AI as a tool the delivery process uses and AI as a participant in the delivery process. The first requires better tooling controls. The second breaks the accountability chain, because accountability frameworks assume a named human at every decision point.

The gap here is measurable and widening. Deloitte's "State of AI in the Enterprise 2026" found that close to three-quarters of organisations plan to deploy agentic AI within two years, while only around one in five reports a mature governance model

for autonomous agents. Roughly a quarter already report at least moderate agentic use today. Deployment is running several years ahead of the control frameworks meant to hold it.

The evidence published since has moved the same way. Deloitte's follow-up study of 501 senior leaders whose organisations are at least piloting agentic AI, published on 12 August 2026, found that only 5% consider their business processes highly prepared for AI agents and only 15% have scaled orchestrated, cross-functional multi-agent adoption — against 74% who expect roughly half their business processes to be redesigned or rebuilt around agents within four years. Even among organisations that have scaled, fewer than half believe their processes are ready. Forrester reached the same conclusion from a different direction in June 2026: around three-quarters of enterprise leaders report adopting agentic AI, while only a small minority have anything running in meaningful production. The ambition is not the constraint. The ground it has to run on is.

EXHIBIT 8

Intent to deploy agents runs far ahead of the governance to control them

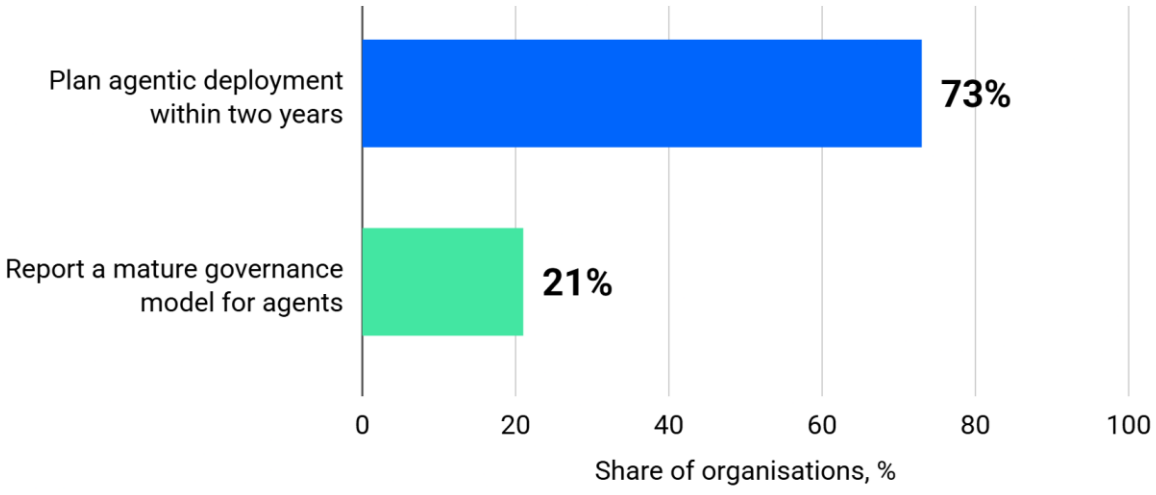


Figure 8. Source: Deloitte, "State of AI in the Enterprise 2026".

> **“We’re using AI to design processes and build agents, but we’re not thinking enough about what happens when those agents go wrong: the governance, the controls and, crucially, the auditability.”**
>
> Eleanor Hruzik, Head of Ops and IT, ClearToken

The regulatory direction argues for acting now rather than waiting for rules to arrive. Speaking at techUK in June 2026, FCA chief executive Nikhil Rathi acknowledged that technology is moving faster than many regulatory paradigms and that legislation will never keep pace, with the regulator itself exploring agentic AI to speed up wholesale market monitoring. The Bank of England and PRA confirmed in spring 2026 that AI adoption is now a core supervisory priority and that a fourth biennial Bank–FCA survey of sector AI usage would run this year, while the Treasury Committee has recommended that major AI and cloud providers be designated critical third parties by the end of 2026. The consistent signal is that AI will be supervised through frameworks institutions already run. That is precisely why the autonomy ladder must be mapped onto existing controls rather than built as a parallel AI regime.

Gartner has put a date on the cost of leaving this implicit. Its May 2026 forecast is that by 2027, 40% of enterprises will demote or decommission autonomous AI agents because of governance gaps identified only after production incidents occur. The load-bearing half of that sentence is the second. The gap is found after the incident, not before it, and the safest available response at that point is almost always to switch the thing off — which is how an institution loses two years of adoption to one afternoon. An autonomy ladder agreed in advance is what turns that discovery into a decision taken in daylight.

Delivery function

- A defined autonomy ladder: which classes of task an agent may draft, which it may execute, and which remain human-decided.

- Named accountable owners for each rung, with the boundary conditions written down before deployment rather than discovered after.
- Reversibility as a design requirement: any autonomous action in a regulated delivery path must be auditable and undoable.
- Explicit alignment to the operational resilience and third-party risk frameworks the institution already runs under, rather than a parallel AI control set.

Technical environment

- Enforce the autonomy ladder in the harness: permissions, scopes and tool access per rung, not per individual's judgement.
- Log every autonomous action with enough state to reverse it.
- Define the change envelope: which configuration changes to the environment are pre-approved and which require the full gate.

Pillar 4 — Delivery Memory

This is the pillar that produces the compounding return, and the one most likely to be dropped when budgets tighten, because its benefit lands in the next programme's numbers rather than this one's.

Delivery function

- Close-down converted from a report into a structured contribution: risks realised versus predicted, estimate variance, control failures, decisions later reversed.
- A queryable delivery corpus that a new programme can interrogate at mobilisation: what has this institution learned about changes of this shape?
- Portfolio-level pattern detection, so recurring failure modes are visible as portfolio facts rather than individual programme misfortunes.
- Explicit ownership. Our recommendation is that the CIO owns delivery memory as an asset, because no programme will ever fund it out of its own budget.

Technical environment

- Engineer the loop: findings, reversed decisions and estimate variances write back to the graph as structured feedback, not prose.
- Capture at source, in the tools the team already uses, so that close-down becomes a review of what the environment already knows.

Pillar 5 — Capability and Confidence

Confidence is a control, not a soft factor. Where practitioners do not trust AI-produced artefacts they reproduce them manually, and the organisation pays twice while reporting a productivity gain.

Delivery function

- Role-specific competence rather than generic awareness training: what a Programme Director must be able to challenge is different from what a delivery lead must be able to produce.
- A published quality bar — what good looks like for an AI-assisted artefact, and what review it requires before it is relied upon.
- Measured trust: survey practitioner confidence in AI-produced artefacts quarterly and treat a decline as a control failure.
- Visible senior use. Confidence follows behaviour, and the fastest way to legitimise AI-assisted work is for the executive layer to rely on it openly.

Technical environment

- One environment, one answer: consolidate the sprawl so that practitioners stop meeting the same number in five places.
- Make lineage visible to the reader of every artefact. Trust follows traceability.

5. The CIO and Programme Director compact

Most of the failure modes described here are not capability failures but ownership gaps. The framework only holds if two roles agree explicitly on who carries what.

This edition adds a third element to the compact: the environment. Someone must own it as a platform, and in most institutions today nobody does.

EXHIBIT 9

The framework holds only where accountability is divided explicitly

The CIO owns	The Programme Director owns
The delivery ontology and the data foundation beneath it	Correct and complete contribution to the delivery record
The autonomy ladder and its control alignment	Operating within the ladder and escalating boundary cases
Delivery memory as a funded, persistent asset	Structured close-down contribution as a delivery obligation
The measurement model and portfolio-level patterns	Programme-level intelligence metrics as reported outcomes
Second- and third-line engagement	Practitioner confidence within the programme team
The delivery AI environment as a platform: its architecture, harness and route to production	Building and configuring capabilities inside the environment's approved envelope, not beside it
The change envelope: what may be reconfigured without a fresh approval	Requesting changes to the envelope through the environment's owner rather than around it

Figure 9. Source: authors' analysis. The final two rows are new in this edition.

The compact matters most at its edges. When an agent acts inside a programme and the outcome is wrong, the question of whether that is a programme failure or a platform failure must have been answered in advance. Organisations that leave it to be resolved after the fact discover that the honest answer is neither party accepted it, which is how governance gaps become findings.

We would add one recommendation. Name an environment owner: a delivery platform lead, reporting to the CIO but embedded with the delivery function, whose job is the architecture, the harness and the approval envelope. Without that role the environment column above defaults to whichever programme shouts loudest, and the estate sprawls.

6. Measuring the dividend

Delivery measurement still largely counts activity: milestones met, artefacts produced, capacity consumed. Those measures were adequate when producing an artefact was itself evidence of effort. Once artefacts are cheap, activity metrics measure almost nothing and can be gamed without trying.

The framework replaces them with six intelligence metrics. Five measure the delivery function; the sixth measures the environment.

EXHIBIT 10

Activity metrics measure nothing once artefacts are cheap

Metric	Definition	What it exposes
Decision latency	Elapsed time from a decision becoming necessary to it being made and recorded.	Whether governance is keeping pace with delivery.
Evidence coverage	Proportion of controls evidenced continuously rather than sampled at a gate.	Real progress on assurance, as distinct from tooling adoption.
Rework rate	Proportion of delivery effort spent redoing work, including silent manual duplication of AI output.	Where confidence is missing and the organisation is paying twice.
Reuse rate	Proportion of a new programme's baseline drawn from structured prior delivery knowledge.	Whether memory is compounding or resetting.
Assurance cycle time	Elapsed time from control test to remediated finding.	Whether continuous assurance is producing action or only volume.

Metric	Definition	What it exposes
Environment change lead time	Elapsed time from a permitted change being requested to it running safely in the approved environment.	Whether you have built a platform or a pile of tools, and how far the speed narrative is true in your institution.

Figure 10. Source: authors' analysis.

Reuse rate is the one to instrument first. It is the only metric here that cannot be improved by working harder, and it is the closest available proxy for whether the dividend is being collected at all. Environment change lead time is the second. It is the honest measure of the gap between how fast a leader can decide and how fast the institution can act.

1 in 5

organisations report a mature governance model for autonomous agents, against close to three-quarters planning to deploy them within two years.

6a. The first 90 days

Everything below is meant to be used. Take it, run it, and tell us what breaks.

Days 1–30 · Establish the baseline	Days 31–60 · Build the foundation	From around day 60 · Prove and measure

Delivery function Run the compounding test on two comparable programmes and publish the honest answer to the executive committee. Select two programmes for full instrumentation and agree the delivery ontology for them alone. Bring second line and internal audit into the design, before any control is redesigned. **Technical environment** Start the approval clock on day one: register the foundational data layer as a system and begin the route to production. It is the longest pole. Inventory the bolt-ons already in use across the two programmes: every script, tracker and assistant, and which data each one copies.	**Delivery function** Stand up decision records and artefact lineage on the two programmes. Convert three high-frequency controls from gate-sampled to continuously evidenced. Publish the autonomy ladder in draft and socialise it with risk and compliance. **Technical environment** Stand the foundational data layer up on the two programmes as the single source the AI reads from, and retire the bolt-ons that duplicate it. Define the change envelope in draft alongside the autonomy ladder, so that risk and compliance see them as one control.	**Delivery function** Report the six intelligence metrics for the instrumented programmes alongside conventional reporting, not instead of it. Run the first structured close-down contribution and demonstrate one reuse into a live programme. Take a practitioner confidence baseline and commit to repeating it quarterly. **Technical environment** Report environment change lead time alongside the other five, and publish the first before-and-after of a change made inside the envelope.

Figure 11. The first 90 days in three stages, each set out in two streams.

Ninety days proves the mechanism on two programmes; scaling it takes longer, and a plan that promises both in one quarter is really just describing velocity and calling it transformation.

6b. What comes next

What this paper sets out is the architecture and the reasoning behind it. What it cannot yet give you is calibration: the thresholds that separate a good reuse rate from a poor one, the benchmarks that make decision latency comparable across

institutions, and the assessment that lets a CIO know where they actually sit rather than where they believe they sit.

That work is only possible with data from organisations running the framework in anger. If you take the 90 days above and instrument two programmes, we would like to hear what you find — including where the model is wrong. The calibrated version will be built from those findings and published with them.

7. The open question: evolution or replacement

Let’s finish where the roundtable did, on the question the room could not close.

Everything in this paper can be read as an evolution. The five pillars can be mapped onto existing methodology; continuous assurance can be described as a better gate; governed autonomy can be presented as an extension of the control framework. Read that way, the framework is an upgrade to PRINCE2-lineage delivery rather than a departure from it, and that reading has a real advantage: it can be adopted next quarter without asking a regulated institution to abandon a model its assurance functions understand.

But there is a harder reading, and I think it is the true one. Existing methodology assumes that delivery is performed by humans, evidenced periodically because evidence is expensive, and governed by a chain in which every decision has a named person at the point of decision. AI-participant delivery breaks all three assumptions simultaneously. Repair three foundational assumptions at once and you have not upgraded the method; you have built a different one, still dressed in the old method’s language because the new language does not exist yet.

> When AI is a participant in delivery rather than a tool that delivery uses, what must change — and is that evolution, or replacement?

My operational recommendation to CIOs is unchanged: adopt as evolution. Run the framework inside your existing governance, because that is what will survive contact with your risk committee this year.

My intellectual position is different, and I will state it plainly. I do not think this is an evolution. I think the inherited model cannot be repaired, only replaced, and that the replacement is already visible in outline to anyone running delivery at scale with AI inside the loop.

A note on the environment

I disagree with little in section 3 and its environment argument sharpens the choice. If AI participation only required better tooling, evolution would suffice. It does not. It requires an environment that is approved once and changes continuously inside an envelope, that remembers as a loop and reasons over a graph, and that is owned by someone as a platform. No existing delivery methodology describes such an environment, because none was written for one. Whatever paper three names, it will have two halves — a way of delivering and the environment it runs in — and they will have to be designed together, because they are built from the same materials.

Paper three will name it and set it out as a method: its principles, its roles, its control model, and the assessment against which an organisation can be measured. It is being written from what a small cohort of institutions find as they run the architecture described here through 2027, which is why the series has been built on practitioner evidence rather than assertion. Saying so now is a commitment rather than a hedge. The room in August could not close the question. We intend to.

References

1. Google Cloud DORA, “The ROI of AI-Assisted Software Development” (2026). J-curve, verification tax, and the amplifier thesis; includes Stanford software engineering productivity findings on greenfield versus legacy gains.
2. McKinsey, “State of AI Trust in 2026: Shifting to the Agentic Era”, March 2026. AI Trust Maturity Survey of approximately 500 organisations.
3. Deloitte, “State of AI in the Enterprise 2026”. Agentic adoption intent versus maturity of agent governance models.
4. Atlassian Teamwork Lab, 2026 workforce research. The AI efficiency paradox and downstream queueing at review and approval points.
5. Nikhil Rathi, FCA, “Rethinking regulation for the age of AI”, speech at techUK, 24 June 2026.
6. Bank of England and PRA, response to HM Treasury on AI in financial services, spring 2026. AI adoption as a core supervisory priority; fourth biennial Bank–FCA AI survey.
7. House of Commons Treasury Committee, “AI in financial services”, and the Government and regulator responses. Recommendation on designating major AI and cloud providers as critical third parties.
8. Code Ninety, “Enterprise AI Adoption Survey 2026”, February 2026, updated August 2026. Sample of 142 technical leaders; governance and data privacy as the leading adoption blocker.
9. SmartCo, AI in Delivery Roundtable coverage and filmed participant contributions, August 2026.
10. Chen, T., Zhu, Y. (Eva), Ye, Z., Puget, J.-F. and Shi, H., “NVIDIA AVO Reaches 100% on ARC-AGI-3, Demonstrating a Frontier-Level General-Purpose Architecture for Long-Horizon Autonomous Agents”, NVIDIA Technical Blog, 21 August 2026. Cross-system comparison of AVO and VISTA on the same underlying model; harness design (persistent memory, tools, feedback, recovery) as the enabler of sustained autonomous performance.
11. Deloitte, “AI agents are only the beginning: The path to agentic transformation”, 12 August 2026. Survey of 501 senior leaders across five industries, fielded April to June 2026; 5% report business processes highly prepared for AI agents and 15% have scaled orchestrated, cross-functional multi-agent adoption.
12. Forrester, “The State of Agentic AI, 2026”, 3 June 2026. Roughly three-quarters of enterprise leaders report adopting agentic AI against a small minority running it in meaningful production.
13. Gartner, agentic AI governance forecast, 26 May 2026. By 2027, 40% of enterprises expected to demote or decommission autonomous AI agents due to governance gaps identified only after production incidents occur.
14. FCA, long-term review of AI in retail financial services, opened 27 January 2026. No AI-specific rules currently planned; existing supervisory frameworks acknowledged as needing to evolve as AI systems become more autonomous.

About this paper

The Delivery Intelligence Dividend is the second paper in a three-part series on AI-enabled delivery. The first, 'The Delivery Intelligence Gap' was published in June 2026. The third will address the question set out in section 7.

This second edition paper has been written following the AI in Delivery Roundtable for financial services leaders, which Sahar and Callum co-hosted in London in August 2026 with participants from institutions including MUFG, NatWest, HSBC, LSEG, Irish Life Group and ClearToken. Discussion was held under conditions of non-attribution; themes are reported in aggregate.

SmartCo

SmartCo is a digital transformation partner, enabling global organisations to scale AI and data, de-risk M&A, optimise cloud and ERP estates and deliver complex technology programmes with confidence.

smart-co.co.uk

Contact the authors

If you would like to discuss this white paper, please contact the authors.



Callum Campbell
SmartCo, Head of Clients and Growth
callum.campbell@smart-co.co.uk *linkedin.com/in/callum-campbell-consulting*



Sahar Khan
Transformation and AI expert in financial services
sahar_khan@live.co.uk *linkedin.com/in/sahar-k-88a0091*

About the authors

Callum Campbell is Head of Clients & Growth at SmartCo Consulting, where he leads the firm's financial-services client relationships and its AI proposition. The perspective in section 3 draws on his experience building production-grade delivery systems with AI coding tools, and on the architecture and approval realities of doing so inside Tier-1 institutions.

Sahar Khan is a Transformation and AI expert in financial services, with nearly 20 years across tier-1 institutions. The practitioner perspective in this paper draws on her large-scale regulatory, portfolio, AI and delivery work across global financial-services programmes.

The Delivery Intelligence Dividend is a collaboration between Sahar Khan and Callum Campbell, written with the support of SmartCo.