WHITE PAPER

You bought the outcome. You paid for the effort.

Why outcome-based AI fails when only the invoice changes — the delivery model that makes the risk carryable, and the commercial model that follows from it.

XAMUN TECHNOLOGIES

Position paper · v1.0 · September 2026 · xamun.ai

Companion to *You have AI. It isn't in your P&L.* (August 2026) · All sources cited in §9

SUMMARY OF THE ARGUMENT

Enterprises have bought large custom systems the same way for thirty years: specify, pay for effort, run over, accept less. AI has not changed the pattern; it has made it measurable. Outcome-as-a-Service — pay for the result, not the work — is the right correction and it usually fails, because most vendors who offer it change only the invoice. The delivery model underneath is still the one that produced the failure rate, so the vendor is carrying a risk they cannot control, and they hedge: projected returns instead of a baseline, milestones instead of a live operation, change requests instead of a fix.

This paper separates the two halves. The first is a delivery model that makes an operational outcome predictable enough to contract — a baseline measured before anything is built, a governed architecture in which a deterministic layer has the final say, one specification that survives from diagnosis to running software, and a thirty-day gate at which the operation either runs or does not. The second is a commercial model that only becomes credible once the first exists — payment triggered by go-live and nothing before it, a price shape chosen by the buyer's circumstances, and an obligation on the vendor to keep changing the system until it fits, unbilled. Monetisation is not what makes OaaS work. It is what becomes possible once the delivery model has removed enough uncertainty for someone to carry the risk.

IN THIS PAPER

SECTION 1

The scoreboard, thirty years deep

The AI-specific evidence is now familiar and is set out in full in the companion paper [7]. MIT's Project NANDA found 95% of generative-AI pilots produced no measurable P&L impact across more than 300 deployments [1]. McKinsey's State of AI survey, roughly 1,993 organisations, found about 6% of firms attribute more than 5% of EBIT to AI [2]. Gartner forecasts that over 40% of agentic-AI projects will be cancelled by end-2027, citing cost, unclear value and inadequate controls [3].

95%
of pilots: no P&L impact
MIT NANDA, 300+ deployments [1]

~6%
attribute real EBIT to AI
McKinsey, ~1,993 organisations [2]

40%+
agentic projects to be cancelled
Gartner, by end-2027 [3]

The figure that matters for this paper is older. McKinsey and the University of Oxford studied some 5,400 large IT projects and found that, on average, they ran 45% over budget and delivered 56% less value than predicted — with one project in six a "black swan" overrunning by more than 200% [4]. That study predates the current AI cycle by more than a decade. Its subject is large custom software generally, and its findings are almost indistinguishable from the AI-specific rows above it.

45%
average budget overrun
McKinsey–Oxford, ~5,400 projects [4]

56%
less value than predicted
same study [4]

1 in 6
overrun by more than 200%
same study [4]

The failure is not new and it is not about models. Three things happen in nearly every failed engagement, and all three are structural rather than accidental: the buyer was asked what to build, so every gap in the specification became the buyer's fault; it ran over or never ended, because every discovery became a change request and the model rewarded duration; and the buyer accepted less, a compromise on the original problem signed off because the money was already spent.

> *What the buyer wanted to know was simple: will it fix this, will it pay back, will it be done in time. The market has only ever answered: probably, we hope, eventually.*

SECTION 2

Three suppliers, three partial answers

The three suppliers an enterprise can engage today each sell a fragment of the answer, and each is paid in a way that makes the whole answer impossible.

Supplier	Sells	Paid for	What the buyer gets
Consultants	A strategy	Days	A deck, then they leave
Development shops	Deployed FTEs	Headcount-months	A spec the buyer wrote, and the overrun
AI vendors	A wrapped model	Seats and credits	A demo the buyer had to make work

Figure 1 — Who an enterprise can engage today. Under every model the risk of the outcome stays with the buyer; the supplier is paid regardless.

None of this is a criticism of the people involved. It is a description of incentives. A consultant paid by the day has no stake in whether the strategy is executed. A development shop paid by the headcount-month is rewarded when the project is long and penalised when it is short. An AI vendor paid by the seat is rewarded when the tool is licensed, whether or not it is used and whether or not the use moves anything. The field data on seat-based tools is unambiguous: 2.8% of work hours saved, and no significant effect on earnings or hours in any occupation studied [5]; experienced developers 19% slower on real tasks while believing themselves 20% faster [6]. Minutes saved by individuals are not enterprise value, and nobody in the chain is paid on whether they become it.

SECTION 3

Outcome pricing as a hedge

Outcome-based pricing proposes to move the risk. The proposal is correct. But moving a risk to a party that cannot control it does not make the risk smaller; it makes the party hedge. That is why so many "outcome-based" contracts contain projected returns produced before any baseline exists, milestone payments tied to documents rather than results, warranty periods measured in weeks, and change-request clauses that quietly restore effort billing through the side door.

The vendor is not being dishonest. They are being rational. If their delivery model gives them no way to make the outcome predictable — if the burden still sits on the user, if a language model still sits in the transaction path, if four handoffs still separate the problem from the code — they cannot afford to be paid only on the outcome. The pricing is not the problem to solve first.

> *A vendor pricing on outcomes with an effort-era delivery model is carrying a risk they have no mechanism to control. The hedge is not a flaw in the contract. It is the contract telling the truth.*

SECTION 4

What an outcome has to be

An outcome that can be paid for has to be a number, and the number has to exist before the work starts. The usual substitute is a projection: a business case, an ROI model, a forecast of savings. Projections are produced before a baseline exists, by the party that wants the deal approved, and they are what the buyer is later asked to accept less than. Quoting a number before a baseline exists is the behaviour that produced the 95% [1][7].

The discipline that replaces it is short: baseline first, then build. In week one, before anything is specified, the current operation is measured on the terms that will later decide whether the work is done. Three measures cover nearly every operational process, and a board already tracks all three.

Measure	How it is counted
Cycle time	Per case, end to end — not per task. From a case entering the operation to leaving it, regardless of how many steps or people it passes through.
Error rate	Against the governing rules — not against opinion. A case is wrong if it breaches a rule, an entitlement, a calculation or a compliance requirement that can be written down.
Throughput	Same team, no added headcount. Cases completed per period by the people already doing the work.

Figure 2 — The three contractable measures. Each is observable by both parties, fixed before the build, and operational rather than financial.

Each has three properties that make it contractable. It is observable by both parties. It is defined before the build, so neither side can move it afterwards. And it is operational rather than financial, so it moves in the quarter after go-live rather than in a year-end reconciliation both sides can argue about. Done is then judged against the baseline, not against a slide.

SECTION 5

The delivery model that carries the risk

This is the half of OaaS that is usually missing. A vendor can commit to an operational number only if their way of building gives them control over three things: where the value sits, whether the system can be trusted in the transaction path, and whether what was specified is what gets built. Each is a design choice with a wrong answer that is the industry default.

Where the value sits

The companion paper sets out the maturity ladder in full [7]. Levels 1–3 — a language model, retrieval, or a graph, each behind a chat window — change how knowledge is stored and leave the burden of the

work on the user: learn to prompt, judge the answer, verify the source, re-key the result, absorb the blame. Levels 4 and 5 move the burden onto the system, first inside the existing process and then inside a process redesigned around what governed intelligence makes possible. Workflow redesign is the strongest EBIT correlate McKinsey found, ahead of talent, tooling and budget, yet only 21% of adopters have done it [2].

Level	Pattern	Burden sits with	What changes
1	LLM + chat	The user	Faster drafting
2	RAG + chat	The user	Better-grounded answers
3	Graph + chat	The user	Answers with structure
4	Two Minds, existing process	The system	The process runs itself
5	Two Minds, re-engineered process	The system	The process is redesigned

Figure 3 — The Xamun maturity ladder, abbreviated from [7]. The divide sits between levels 3 and 4. An outcome contract written against a level-2 deployment is a contract against a productivity tool.

Only above the divide can cycle time, error rate and throughput move without adding people, because only above it does the burden leave the user. A vendor selling outcomes on a chat interface is selling a number they cannot move.

Whether the system can be trusted in the transaction path

Enterprises stay below the divide for a well-founded reason: language models are probabilistic. The mechanism that drafts a good clause is the mechanism that invents a citation, and retrieval reduces the problem without removing it — commercial legal-research tools marketed as hallucination-free still erred on 17–33% of queries in a preregistered evaluation [7]. The mistake is to treat this as a limitation to patch rather than a division of labour to design.

The architecture that works has two minds with different jobs. A deterministic mind holds the rules, entitlements, calculations, compliance logic and audit trail; it runs first and has the final say; it decides and does not generate. A governed model handles language, extraction, drafting and explanation; it imagines where imagination helps and is never the last word. A shared graph memory holds one model of the business — entities, obligations, precedence, lineage — read and written by both, so that what the model extracts and what the rules decide refer to the same things. The user touches an application. No prompt, no chat window.

> *Hallucination is a feature. Just never in your ledger. Generation is not patched. It is placed.*

This is what makes the error-rate commitment possible. If the rules sit in a deterministic layer that runs first, the error rate against those rules is controllable, and a vendor can put their fee behind it.

Whether what was specified is what gets built

Most enterprise software arrives technically correct and operationally wrong, and the reason is translation. A strategy is handed to an analyst, who writes requirements for an architect, who designs for developers, who build software. Four handoffs; meaning degrades at every one. By the time working software exists nobody can trace a line from the problem to the code, and the people who own the problem are told the system does what was asked.

The alternative is one specification that runs from diagnosis to running software, approved once and never re-interpreted. The specification is the artefact that decides whether the system is right, so it gets the time — weeks, with the people who own the operation. The build is generated from it, gated by automated quality checks, reviewed by a person at each acceptance point, deployed on the client's own infrastructure, with the client owning the code. Fewer translations is not a speed claim. It is why the outcome is right.

The gate

The three choices above are what make a hard gate possible: thirty days from approved specification to an operation running end to end, digitally. Not a demo, not a pilot, not a proof of concept. A case flows through the system and comes out the other side, or it does not, and both parties can see which. The gate is where the delivery model hands over to the commercial model. Nothing about money needs to be decided before it.

SECTION 6

The commercial model, kept separate

With a baseline measured, a governed architecture, a single specification and a go-live gate, the vendor now controls enough of the outcome to carry its risk. Only at this point does outcome pricing stop being a hedge and start being a commitment. Four principles follow.

The gate is the only trigger

The meter starts at go-live and nowhere else. This is the direct inversion of milestone billing, where a vendor is paid for producing deliverables whether or not those deliverables produce a live operation.

At the gate	Usage model	Licence model
Live	Meter starts. Licence accepted.	Final tranche paid. Source delivered.
Not live	You owe nothing. The meter never started.	The final tranche is never paid. You keep what was built.

Figure 4 — The gate. No obligation exists past it; a second phase happens only if the buyer chooses it.

The buyer's circumstances select the price shape

Two questions decide which of two doors applies: who holds the platform after go-live, and is there a unit of work worth metering?

- **Usage.** The vendor invests the build, deploys the system and is paid a small fee per unit the operation processes — a case, a transaction, a shipment — from the day it runs and not before. No capital expenditure. Appropriate when there is a countable unit and the buyer prefers the vendor to hold the platform.
- **Licence.** One capital fee, source code included, final tranche due only at go-live. Appropriate when the buyer must own the platform outright, or when per-unit pricing is not an honest measure of the value delivered.

Offering both matters. A single price shape forces the vendor to choose engagements that suit the shape rather than problems that suit the buyer. Two doors let the delivery model stay constant while the commercial terms flex — usage in one jurisdiction, licence in another, without touching how the system is built.

Effort is never billed

Scope changes, change requests, timesheets and rework are not invoiced. There is no rate card. This is the clause that separates an outcome contract from an effort contract wearing an outcome contract's clothes. If any of these appear as billable line items, the risk has moved back to the buyer and the pricing is not outcome-based, whatever it is called.

The vendor keeps changing it until it fits

The commitment that closes the loop is the one most vendors will not make. If the system does not fit the way the work is done, the vendor changes it — without asking why, without distinguishing a bug from a change or a data problem from a code problem — for as long as the operation stays the same. When the operation itself changes, both parties re-baseline and begin again.

This replaces the invoice as the buyer's leverage. Under effort billing the buyer withholds payment to keep the vendor honest. Under a genuine outcome model the buyer does not need to: the vendor has already been paid only for a live operation, and every subsequent change is theirs to make. The vendor's incentive is now to get it right quickly, because every hour of rework is unbilled.

> *You pay for a live operation — never for effort. Every change after that is the vendor's to make, so you never need to hold an invoice to keep them honest.*

SECTION 7

Designed together, sold apart

The separation this paper proposes is not an accounting nicety. It has three consequences.

It exposes false OaaS. A vendor offering outcome pricing can be asked which of the four delivery choices they have made — baseline before build, burden on the system, a deterministic layer with the final say, one specification without handoffs. If the answers are vague, the outcome pricing is a hedge, and the contract will contain the mechanisms described in §3.

It keeps the first engagement small. Because the delivery model makes a single operation predictable, the first engagement can be scoped to one problem — the costliest bottleneck the buyer can name — approved without a committee, and judged on one live operation. Nothing else is committed. A vendor who cannot scope this small usually cannot control the outcome, and needs the larger programme to absorb the risk.

It lets the delivery model stay constant while the market varies. Data residency, ownership expectations, procurement rules and capital availability differ by jurisdiction and sector. The commercial door can change without touching the architecture or the specification discipline, which are the same regardless of who ends up holding the platform.

SECTION 8

Where to start

Choose the costliest problem, not the strategic one. A single operational constraint that leadership already knows is expensive, that has a countable unit, that a small enough group can approve, and that can be baselined in a week. Anything where cases flow through people and rules, and where cycle time, error rate and throughput are visible.

Demand the baseline before the proposal. Any proposal containing a projected return before the operation has been measured should be returned. Ask for the week-one measurement plan: what will be counted, how, and by whom. If the vendor cannot describe it, they cannot be paid on it.

Ask the four delivery questions.

1. Where does the burden sit after go-live — with our people, or with the system?
2. What has the final say over a decision that touches money, entitlement or compliance — a rule, or a model?
3. How many handoffs sit between the problem we describe and the code that runs, and what artefact survives all of them?
4. What is the go-live gate, when is it, and what happens to payment if it is not met?

Read the commercial terms for the side door. A rate card, billable change requests, milestone payments tied to documents rather than a live operation, warranty periods in weeks, any distinction between defects and changes. Each is effort billing re-entering. A genuine outcome contract has none of them, and says in plain language that the vendor keeps changing the system until it fits.

Hold everything else back. Whatever the vendor's wider platform offers — intelligence, strategy, governance, a loop of subsequent operations — it is worth nothing until the first system is live. The right vendor will say so themselves. Judge them on one live operation first, and choose what comes next only after the number has moved.

We publish no projected figures in this paper deliberately. What moves, and how far, is a property of the workflow, and it is measured against that workflow's own baseline — not against a vendor's slide.

SECTION 9

References

[1] MIT Project NANDA. The GenAI Divide: State of AI in Business 2025. 300+ deployments, 52 executive interviews, 153 survey responses. 2025.

[2] McKinsey & Company. The State of AI: agents, innovation and transformation. n ≈ 1,993 across ~105 countries; with the March 2025 EBIT correlation analysis. 2025.

[3] Gartner, Inc. Press release: over 40% of agentic AI projects will be cancelled by end-2027. Poll of 3,412 respondents, January 2025. June 2025.

[4] Bloch, M., Blumberg, S., Laartz, J. Delivering large-scale IT projects on time, on budget, and on value. McKinsey & Company with the University of Oxford (BT Centre for Major Programme Management). ~5,400 IT projects over US$15 million. 2012.

[5] Humlum, A., Vestergaard, E. Large Language Models, Small Labor Market Effects. NBER Working Paper 33777, 2025. 25,000 workers, 7,000 workplaces, payroll-linked.

[6] METR. Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity. RCT, 16 developers, 246 tasks; 2025, revised February 2026.

[7] Xamun Technologies. You have AI. It isn't in your P&L. Position paper v1.0, August 2026. Sets out the retrieval evidence, the maturity ladder and the Two Minds architecture in full, with sources.

ABOUT XAMUN TECHNOLOGIES

Xamun builds vertical AI operating systems on the Two Minds architecture — a deterministic rule engine paired with a governed language model over one graph memory — delivered inside purpose-built applications on re-engineered workflows, and sold under Outcome-as-a-Service: one problem, one price, unlimited changes, live in thirty days,

judged on the buyer's own number. Xamun is the AI-native evolution of a delivery engine shipping enterprise software since 2000 across more than ten countries, with a platform that has processed over US$500 million in transactions and systems still in production after twenty years. Xamun Technologies Ltd is a DIFC company operating from the Dubai AI Campus and Manila. Enquiries: xamun.ai.

Figures reflect the studies as published; where findings have been revised or contested, this paper says so in the body rather than in a footnote. Correspondence: xamun.ai. *Version 1.0, September 2026.*