



Ferrari in the Garage

Field notes from an engineering-led AI transformation practice — on why most programmes stall, and what actually works when they don't.

By Walter González, Founder & CEO — The AI Studio

Over the past eighteen months I have sat in a lot of rooms having the same conversation about AI. The rooms differ: some are founder's kitchens at early-stage startups with a dozen people on the payroll, some are Series A and Series B scaleups trying to figure out how to grow into an AI-native operating model, some are boardrooms at large corporations with tens of thousands of employees. Some of that work has been a direct mandate from the leadership of the company in question. A meaningful slice of it has been sponsored by private-equity and venture-capital backers looking for an honest external read on whether their portfolio company is actually doing AI, or merely talking about it. The engagements span retail groups, industrial manufacturers, consumer brands, financial services, B2B SaaS, and a handful of companies whose category did not really exist two years ago.

The engagements differ. The conversations, with dispiriting consistency, do not.

What follows is a consolidated view of what I have learned from that work. The vignettes are drawn from real engagements. They are deliberately anonymised and, in places, combined, because the patterns matter more than the logos — and because the story of AI transformation across the corporate landscape is not a hundred different stories. It is roughly three stories told a hundred different ways, whether the company has ten thousand employees or eighty.

Where we come from, and why it matters for what follows

Before I say anything else, a disclosure about the lens. The AI Studio is not a traditional consultancy with a hired engineering arm bolted on. We are the inverse. Our consultants — including the ones running strategy sessions with executive committees — are hands-on AI engineers. We built the last agent we shipped. We will build the next one. The strategy work we do is inseparable from the engineering work we do, and vice versa, because we do not believe the two are separable in a field moving this fast.

This is relevant to the rest of this piece for two reasons. First, it shapes what we see: the patterns I describe below are visible from where engineering meets boardroom, and they tend to be invisible from either end on its own. Second, it shapes what we do: every framework I describe is something we run against ourselves before we run it against a client. We use AI-native workflows to do our own research, write our own code, produce our own deliverables, and operate our own business. If we are going to tell a client to rewire the way their knowledge workers operate, we had better be doing it ourselves — and we are.

Call this consulting depth with an engineering pulse if you want a label. I find myself using that phrase often, because it is the thing that most consistently distinguishes the work we do from the adjacent categories — strategy firms that cannot build, and dev shops that cannot strategise.

The conversation every leader is having with themselves

The meeting runs smoothly. The AI slide is confident. The CEO — or, just as often, the PE-backed managing director, or the founder presenting to their board — says, with feeling, that AI is the single most important thing happening to the business. Heads nod. The slide moves on.

Twenty minutes later, over coffee, the same person pulls me aside and asks the question they could not ask the room: *so where are we, really?*

The honest answer, in most cases, is somewhere between "further behind than you think" and "we don't actually know." This is not a judgement on the leader — it is a statement about how AI programmes actually progress, or fail to, inside real organisations. The gap between stated ambition and operational reality is, in my experience, the most consistent diagnostic in the entire field. Everything else flows from it.

I have taken to calling this the execution deficit — the distance between what leadership has publicly committed to and what the organisation is actually capable of shipping. It is almost always larger than leadership believes, and it is almost always the first thing we have to confront honestly before any real progress is possible.

The published research is uncomfortable. Depending on whose numbers you trust, somewhere around one in a hundred companies has reached genuine maturity in AI adoption; average workforce proficiency across most organisations hovers around a tenth; and only a minority of them — roughly a quarter, by most credible surveys — will tell you candidly that they are getting real, measurable value out of their AI investments. Even those numbers are generous, because most self-reporting against AI maturity benchmarks is optimistic to the point of fiction.

The question that matters is not *are you doing AI?* Everyone is doing AI. The question is: *is your execution deficit shrinking or widening, quarter on quarter?*

What we usually find in the first two weeks

Every engagement starts with a diagnostic. By the end of week two we are usually able to hand the leadership team — or the sponsoring investor, depending on who commissioned the work — a scoreboard that nobody was expecting and nobody particularly enjoys.

The specifics vary but the shape is almost always the same.

Enterprise LLM licences have been bought. Actual daily usage sits somewhere in the low teens as a percentage of the intended user base. Weekly usage is roughly double that. A meaningful majority of the employees who were supposed to be using the tool, are not using the tool. The company is, in cashflow terms, paying for a gym membership it isn't using.

Two or three "lighthouse" pilots have been announced with some fanfare. None of them are in genuine production. One or two are technically live but running in shadow mode next to the pre-existing process, because nobody fully trusts them and no executive has been willing to stake a quarter on the switch-over.

There is no consistent definition of what a pilot is supposed to produce. Each one has a different sponsor, a different vendor, a different metric of success, and — if you ask the right question in the right way — usually a different understanding of what would count as failure.

The measurement discipline is focused on inputs, not outcomes. The AI team reports on token consumption, licence activation, and webinar attendance. Finance has no line of sight to business value. The CFO is polite in meetings and visibly sceptical in private.

And underneath all of it, there is already a layer of unsanctioned AI use that nobody is quite willing to name. In one engagement we found an unapproved tool that had been processing customer email for half a year. In another, an operations team had wired up a home-grown retrieval pipeline against a consumer API with no data-loss prevention whatsoever. Nobody was acting in bad faith. Governance had simply failed to keep pace with enthusiasm, and the most capable employees had stopped waiting for it.

I call this picture the deployment plateau: money spent, tools in place, vision articulated, and nothing meaningfully changing about how work gets done. It is by far the most common starting position I see, and it is the profile for which the standard consulting response — *let's do another pilot* — is exactly wrong. The pilots are not the problem. The operating conditions under which pilots run are the problem.

The Ferrari in the garage

When I have to get this across to a room of decision-makers in a single image, I reach for the same one every time. *You have bought a Ferrari. It is parked in the garage. The reason it is parked in the garage is that nobody in the organisation has been taught to drive it, the insurance has not been sorted, the roads to the places you actually want to go are unmapped, and the board keeps asking why the Ferrari has not yet replaced the fleet of vans.*

It is an annoying metaphor. Rooms rarely love it. It is correct.

The point of the metaphor is not to be rude about the technology investment. The technology investment was, in most cases, the right call. The point is that the technology is the *smallest* component of the system you actually have to build. The organisational, procedural and human components are two-thirds of the work and will absorb the majority of the budget in year two regardless of whether you planned for it — so you may as well plan for it.

Most clients I work with — and this holds across both the enterprise engagements and the PE/VC-backed scaleup diagnostics — are what I'd call asymmetrically ready. They are strong on one dimension, usually whichever one the budget or the founder's personal interest has been flowing toward, and weak on the others. The weak dimensions cap the return on the strong ones. In one industrial client the technology stack was genuinely world-class and the workforce capability was, to put it diplomatically, nowhere near it. Most of their AI investment was being quietly absorbed by an organisation that had no idea what to do with it. At the opposite end of the spectrum, I have seen scaleups where the founding team were brilliant builders whose AI literacy was the envy of their sector, and whose commercial, operational and governance posture was several years behind their technology. Same diagnosis, different symptom. In both cases the answer was not more technology. The answer was to stop buying technology for a while and do the unglamorous work underneath.

How we measure it, concretely

The diagnostic is not a vibe check. At The AI Studio we run two instruments that produce the shared evidence base from which everything else flows.

The first is a people-side AI maturity survey run across the target workforce. It is short — twenty-five scored questions, five to seven minutes to complete — and built around four dimensions: awareness and mindset, current usage and adoption, proficiency and confidence, and organisational enablement. Each dimension is weighted by how much signal it actually carries about real maturity: usage and adoption get the most weight, because behavioural reality beats stated intent every time; awareness gets the least, because knowing AI exists is a necessary but insufficient condition for doing anything useful with it.

The instrument produces a per-respondent classification on a four-level scale, from L1 Aware (knows AI exists, isn't using it) through L2 Experimenter (dabbles but hasn't found real value), L3 Practitioner (integrates AI into recurring workflows and gets results) to L4 Builder (creates AI systems that multiply team impact). We apply explicit override rules — you cannot be classified as an Experimenter if you've answered that you never use AI; you cannot be classified as a Builder if you do not share what you learn. The composite score is a default; the override rules stop it lying.

Aggregated across the organisation, this gives us a distribution: what percentage of your people sit at each level, how that differs across departments, and — crucially — whether the bottleneck is on the individual side (people lack skills) or on the organisational side (people are ready and the company is the brake). If respondents score high on awareness, usage and proficiency but low on enablement, the company is the bottleneck, not the people. That single diagnostic re-routes the entire programme.

The second instrument is a workflow audit run with each priority team. It starts at the top — pick the team's most important goal, the one where a 10–20% improvement would materially move the business — and walks the team through four steps: identify the workflows that contribute to that goal, prioritise them on Frequency × Effort × Importance, break the priority workflow into tasks, and map each task to what we call generative AI superpowers — generate ideas, generate content, analyse content, give feedback, research — deciding which tasks stay human and which are handed off. The output is not a report. It is a prompt pack: concrete, documented, reusable instructions the team can start running tomorrow.

The two instruments together produce a shared language across the leadership team, so that "we need to improve our AI maturity" — which means nothing and obligates nobody — becomes "we are a company of mostly L2 Experimenters, the bottleneck is enablement not

skill, and our top three workflows for AI augmentation are these, in this order." That is what a real starting point looks like.

The Five Pillars: how we actually run an engagement

Transformation programmes fail when they try to fix everything simultaneously. They also fail when they fix the wrong thing first. The trick is to work sequentially through a small set of well-defined stages, resource each one properly, and have the discipline to ignore the political pressure to skip stages or run them in the wrong order.

At The AI Studio we structure engagements around five pillars that form both a sequence and a self-reinforcing loop.

The first pillar is clarity of goals and identification of champions. This is not a workshop line item — it is an insistence that every AI initiative be explicitly anchored to an outcome the business already values, expressed in the company's own language of quarterly and annual objectives. Not to a technology, not to a tool, not to a hype cycle. Goals first. And alongside the goal, a named human — the champion — who will advocate for the transformation inside the organisation when we are not in the room. Without a champion with political skin in the game, nothing ships.

The second pillar is the initial study. A proper diagnostic, running our two instruments above alongside targeted interviews with key players. The output is a written diagnosis of where AI maturity actually sits, a map of opportunities ordered by value and viability, and a candid read on the people and technology landscape. This phase typically runs thirty to ninety days depending on scope, and it ends with a deliverable the client owns outright. If they wanted to walk away at that point and execute themselves, they could. That is by design.

The third pillar is the right starting points. Not the most exciting projects. Not the ones that would make the best LinkedIn post. The ones that combine three properties: ripe for impact in the near term, visible enough to create organisational belief, and supported by data that is actually accessible. The pilot graveyard in most organisations is populated almost entirely by projects that failed on the third criterion — the data looked fine on the surface and was unusable underneath. We refuse to start a pilot until we have validated that the data feeding it exists, is reachable, and is clean enough to work with.

The fourth pillar is quick wins, shipped and socialised. The goal here is not the win itself — it is the demonstration effect. Every organisation contains two populations: the people who believe this is going to change things, and the people who are quietly hoping it will not. Quick, visible, unambiguous wins move people from the second camp into the first camp faster than any amount of executive rhetoric. Ship something. Make it real. Show it to the rest of the business. Then do it again.

The fifth pillar is repeat and scale. Take the learnings from the first win and apply them to adjacent workflows, adjacent teams, adjacent problems. Build the playbook the rest of the organisation can reuse. Continuously adapt it. This is the pillar that converts a successful pilot into a transformed business, and it is the pillar most clients are least prepared for, because it requires sustained operational discipline at a time when the political pressure is to declare victory and move on.

This looks linear on paper. In practice it is a loop: every scaled win surfaces new goals, which triggers another micro-cycle of diagnostic, selection, delivery and socialisation. The organisations that succeed are not the ones that run the loop once particularly well. They are the ones that run it continuously, getting faster each time.

Try, Buy, Build — in that order

The most consequential architectural decision on any AI programme is also the one most consistently made wrong: how to source the capability. Build it yourself, buy it from a vendor, or something in between?

The industry's default framing — *build versus buy* — is a false binary. It skips the option that is right the majority of the time, and it flattens an important sequencing question into a one-off bet. At The AI Studio we work a three-tier hierarchy, deliberately in this order.

Try comes first. Before any engineering, before any procurement, the question is: *can we solve this with the AI tools we already have access to?* A frontier general-purpose assistant, a Custom GPT or Claude Project, a Gem. Foundation stack, configured thoughtfully against a specific workflow. In a surprising number of cases the answer is yes — and the work ships in days rather than quarters. The teams that skip this step end up building things they could have configured.

Buy comes second. If configuration of the foundation stack is not enough, the next question is: *does a mature SaaS solution exist that solves this problem?* AI-enabled CRM, agentic payments, customer-success agents, marketing automation suites, knowledge-management systems. The market for AI-enabled SaaS has matured faster than most enterprise buyers realise, and in a growing number of categories a buy decision is the right one — faster, cheaper, and better-supported than any internal build.

Build comes last, and only when Try and Buy have both been honestly ruled out. Custom automation pipelines using orchestration tools — Zapier, n8n, Make, Relay, or bespoke engineering. Custom applications. Custom fine-tuned or deployed models. This is where the engineering work gets genuinely deep, where the competitive moat is genuinely built, and where, in our experience, most organisations overspend because they started here instead of ending here.

The sequence matters because each tier tests the assumptions of the next. A team that has Tried against a workflow and found the ceiling of Try will know exactly what they need from a Buy. A team that has Bought and hit the limits of off-the-shelf will know precisely what they need to Build. Skipping stages produces either over-engineered solutions to problems that did not need engineering, or under-powered purchases that could never have met the need.

What we actually build: the engineering spine

This is the section most consulting write-ups skip, because most consulting firms do not build anything. It is also the section our engineering readers care about most, so I will spend some time on it, and I will describe two real delivery patterns — one in depth, one in brief — that represent the kind of work we ship most frequently.

Pattern one: the engineering knowledge agent

We have now delivered versions of this system across multiple clients in regulated and engineering-heavy environments — financial services institutions, an industrial engineering group, and a large regional bank. The pattern generalises.

The problem. Critical operational and engineering knowledge sits scattered across wikis, ticketing systems, PDFs, runbooks, and proprietary codebases. When a senior engineer leaves, the tacit knowledge walks with them. When a new engineer joins, onboarding takes months because the context they need is nowhere they can find it. When an incident hits, someone is doing digital archaeology across five systems to reconstruct a decision that was made and documented three years ago. The cost is not a single visible line on a P&L; it is a pervasive drag on velocity, a persistent source of operational risk, and a slow leak of institutional memory.

The system we ship. A context-aware support agent deployed securely against the organisation's proprietary knowledge and codebase. Architecturally it rests on four pillars.

The first is a semantic retrieval layer built on a vector store — we have used FAISS in several deployments and other backends where the client's infrastructure constraints required it. The retrieval layer ingests the distinct knowledge sources as separate indexed corpora rather than as one undifferentiated blob: the wiki is a different data dictionary from the ticket history, which is a different dictionary from the runbooks, which is a different dictionary from the code. Keeping them separate, and teaching the system to reason over which to consult for what, is the difference between a useful agent and a plausible-sounding liability.

The second is a set of heuristic guardrails that encode the organisation's internal engineering standards, architectural patterns, and compliance constraints. These are not bolted-on filters — they are part of the prompting and orchestration logic, so that the agent's outputs are not just technically correct but structurally consistent with how the organisation already builds. A suggestion that is right but architecturally unlike anything the team has ever shipped is a suggestion that will be rejected, correctly, by the team.

The third is secure orchestration built on the Model Context Protocol. MCP manages tool access and permissions, so that agents can navigate across connectors — wikis, code

repositories, ticketing systems, runbook stores — without data leakage and with a full audit trail. This is the architectural decision that lets legal and compliance sign off. Without it, the conversation with the CISO ends the project before it starts.

The fourth is a set of long-running agents for recurring work — migration playbooks, dependency analysis, documentation gap detection — that run in the background between human-driven sessions. This is the element most clients do not ask for up front and almost all of them value most once it is in place.

The delivery timeline we commit to. A focused proof of concept at T+10 days, validating the retrieval layer against a defined set of critical sources. A live MVP with expanded coverage at T+25 days. Then a handover to the client's internal engineering team for full-scale deployment and ongoing optimisation. We deliberately aim for a short window before handover — the goal is to transfer capability, not to create a dependency.

The impact pattern. Time-to-answer for context-heavy engineering queries compressed by roughly five-to-one against the baseline. Onboarding times for new engineers compressed by roughly three-to-one. Dramatic reductions in context-switching fatigue for senior engineers, who had been losing hours a day to the archaeology problem. Incident response accelerated. The engineering team's own sentiment toward the tool, measured weekly, typically dips in week one, recovers by week three, and exceeds baseline by week six. The dip is real. Budget for it.

Pattern two: the voice-enabled agentic customer experience

A shorter vignette from a different domain. We delivered this for a multinational beverages business operating across several countries and languages, serving a customer base of close to a million. The existing support infrastructure was a static, text-only bot with frozen context — acceptable as a cost-centre in 2021, dangerous as a brand-experience touchpoint in 2026.

We rebuilt it as an agentic voice experience. The architecture resembles the engineering agent but with a customer-facing surface and a different integration profile: a RAG repository constructed from legacy support records and product data; tight integration into three critical systems (CRM, billing, logistics) so the agent could execute tasks rather than merely answer questions; MCP-based orchestration between sub-agents; and a voice layer built on a best-in-class speech stack delivering native-quality output across multiple languages.

Delivery rhythm: T+30 days end-to-end POC validating RAG coverage, tool use, and voice latency. T+90 days live MVP with AB testing and global deployment. The results are not mine to quote in full, but the headline pattern — sub-second response times, round-the-clock availability in multiple languages, and a measurable lift in resolution rate against the baseline bot — has held in every deployment of the pattern we have run.

What we insist on every time

Across every agentic system we have shipped, five things do not move.

Humans at decisions, not at keystrokes. The human approves or refines commercial and customer-facing decisions. The system does the retrieval, the drafting, the compliance-checking. The moment you try to let the agent *make* the commercial decision autonomously — issue a refund, approve a claim, commit to an SLA — legal, compliance and the frontline team will combine to kill the project. Keep humans where their judgement adds value and they become the system's biggest advocates. Put them in the wrong place and they become the biggest obstacles.

Structured evaluation before any live deployment. Not vibes-based "*looks fine to me*" evaluation. A proper rubric, scored by multiple senior operators, with inter-rater reliability measured. I have had more than one agent design fail this step and require rebuilding before it was allowed anywhere near live traffic. This feels slow. It is dramatically faster than deploying a bad agent, eroding operator trust, and having to re-earn it over the following six months.

Full observability from day one. Every agent call logged with its prompt, retrieved context, tool calls, and output. When a system produces a wrong answer in production — and eventually it will — you need to trace precisely which component failed. With proper logging, that diagnosis takes hours. Without it, it takes weeks, and usually kills confidence in the system long before the actual bug is fixed.

Match the model to the component, not to the project. Uniform model choice across a multi-component system is uniform money wasted — either on frontier capacity for trivial tasks, or on inadequate capacity for the task that actually needs reasoning. We run different models at different components and will defend that choice against any procurement team that wants a single-vendor answer.

Retrieval quality is the ceiling on output quality. You can have the best model on the planet and it will produce confident, fluent nonsense if the retrieval layer is pulling the wrong passages. A disproportionate share of the engineering effort on these systems goes — correctly — into the retrieval layer: indexing strategy, chunking approach, re-ranking, and the editorial discipline of curating the knowledge base rather than dumping every document the company has ever produced into a vector store and hoping for the best. I have seen more agentic systems fail on retrieval than on any other single component.

What twelve months of this typically produces

I am cautious about presenting standardised "before and after" dashboards, because every client starts from a different place and every engagement is scoped differently. But the directional shape across successful programmes is consistent enough to describe.

Daily active usage of the enterprise or team-wide LLM moves from somewhere in the low double digits to a clear majority of the targeted workforce, given that the foundation pillars are resourced properly. The distribution on the maturity survey shifts tangibly — L1s shrink, L2s grow and then themselves start progressing to L3, and a real L4 Builder population starts to emerge, usually clustered around specific functions rather than evenly across the organisation. The number of internal builder-level automations produced by employees — not commissioned by central, built by employees — grows from scattered dozens into the low thousands in a mid-sized organisation. Most are low-value individually; the valuable minority, perhaps five to ten per cent of them, becomes the discovery engine for the programme's next wave of more ambitious investments.

The number of genuinely productionised agentic workflows — not pilots, not proofs of concept, but systems embedded in how a business unit actually operates — typically reaches a small handful per selected business unit by month twelve, with the next wave already in design. Unsanctioned AI tool usage moves from uncounted to counted and from ungoverned to governed. And executive accountability shifts from zero leaders with AI in their annual objectives to most of the senior leadership team having it there explicitly. That last change, in my experience, is the single most predictive indicator of whether year two of the programme will succeed or stall.

What does not generally happen in twelve months is a wholesale transition to an AI-native operating model. That is a three-to-five year horizon for any organisation I have ever worked with. The twelve-month goal is to produce enough proof, enough capability and enough governance to make year two both fundable and credible. If you reach that point, the compounding begins. If you do not, year two becomes an argument about why the budget should continue at all — an argument that is very hard to win once scepticism has set in.

Four things I tell every leader who asks

If you skip everything else I have written here, read these.

Your AI strategy is not a strategy if it cannot survive contact with your operating model. Most of the AI strategies I read are wish lists. A real strategy names the two or three business units or functions that will change, the specific workflows within them, the accountable executive for each, and what those executives will stop doing to make room. If your strategy does not contain all four of those elements, you do not have a strategy. You have a press release.

Good governance is an accelerator, not a brake. The companies moving fastest on AI are the ones with the clearest guardrails. They have pre-approved low-risk use cases so teams can ship without a committee. They have risk-tiered their policies so that heavy review only happens where it genuinely matters. High-risk cases get full scrutiny. Low-risk ones get flight. Every week you delay building this, you are funding the shadow IT problem you will eventually have to clean up at much greater cost.

Concentrate, don't spread. Pick two business units (in a large enterprise) or two functions (in a scaleup). Not seven. The temptation to spread investment evenly across the organisation is enormous and almost always political. Resist it. Evenly-spread AI investment produces evenly-spread mediocrity. The only pattern I have reliably seen produce scaled results inside a year is picking two, going deep, and producing a playbook the rest can eventually copy. Concentration first. Replication second.

Workforce transformation is the actual transformation. Technology is the easy part. It is genuinely, measurably, the easy part. The hard part is changing the default way your knowledge workers approach their work — and that requires visible leadership modelling, role-specific activation, a living network of internal builders, and an operating cadence that does not stop after the launch event. If you have not budgeted for that, you have not budgeted for the programme. You have budgeted for a Ferrari, and you are about to park it in the garage.

Where this goes next

We are eighteen months into the era in which "*we use AI*" was itself a differentiator. That era is ending. By the close of 2026 it will be the baseline, and the companies that did the unglamorous foundation work in 2025 will start compounding away from those that did not.

The next unlock is not model quality. The frontier models are already good enough for almost any business use case that matters, at almost any scale of company. The next unlock is organisational and architectural: how you structure agentic systems, how you govern them, how you measure them, and how you redeploy the human capacity they free up into the work that actually creates value.

The clients I am most optimistic about have stopped asking "*can we do this?*" and started asking "*how fast can we do the next one?*". Their knowledge workers have stopped producing documents they never needed to produce, and started doing the work the documents were a proxy for. Their AI programmes have a P&L line. Their heads of AI have political capital where they used to have excuses.

That transition — from ambition to execution, from announcement to outcome, from garage to road — is the only AI story in any organisation that is genuinely worth telling. Everything else is noise.

Walter González is the founder of The AI Studio, a boutique AI transformation and engineering practice based in Amsterdam. The AI Studio works with European clients ranging from scaleups to large enterprises — often sponsored by private-equity and venture-capital backers, and often directly mandated by leadership — on AI maturity diagnostics, workforce activation programmes, and the design and delivery of agentic systems. Our consultants are hands-on AI engineers; our engineers think strategically. We run our own practice on AI-native workflows, which is the only way we know to credibly recommend them to anyone else. theai.studio