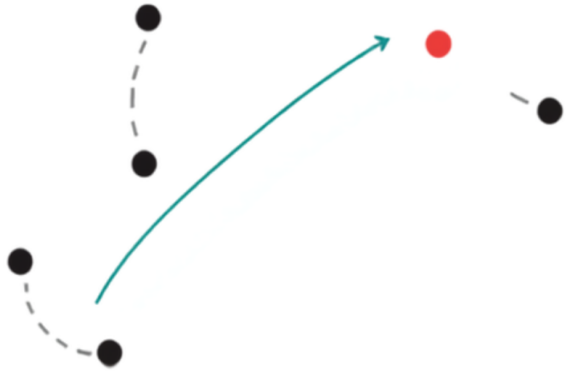
THE AI INITIATIVE PLAYBOOK

Principles and Moves
for Problem-Led,
Outcome-Driven Work



SHRIKANT VASHISHTHA

The AI Initiative Playbook

Principles and Moves for Problem-Led Outcome-Driven Work

ShriKant Vashishtha

This book is available at
https://leanpub.com/ai-initiative-playbook

This version was published on 2026-08-17

Dedication

To my father, who lived his life in the service of others through the Bhoodan movement and many social causes, quietly instilling in me the values of righteousness, truth, Sarvodaya and simplicity.

To my mother, whose quiet strength carried our family through hardship, and whose love gave me the purpose and resilience I hold today.

And to my grandfather, a freedom fighter and Gandhian, who laid the foundation of the value system that shaped our family and my path.

Your spirit lives on in every page of this book. This work, and whatever good it brings, is all yours.

Contents

Introduction

Almost everyone is talking about AI.

People are changing their titles to match the new world. Layoffs are happening, many of them in AI's name. Every other post on professional social media is about AI. New models drop every week. New features, faster than anyone can evaluate them.

In that noise, leadership decides it has to move.

So almost everyone wants to run. Nobody knows where.

The confusion is not laziness. It is structural. Some of what is written about AI is engineering detail — implementation that changes every few months. Some of it is strategy so high-level it survives any organisation because it commits to nothing. Neither helps a leader deciding what to do on Monday.

This book sits between them. It is not about AI. It is about what AI reveals about the organisations it enters — the constraints it exposes, the foundations it depends on, the judgement it quietly erodes. In a regulated enterprise, those questions matter more than which model you chose.

The goal is a way of thinking you can apply immediately. At the organisation level, or to a single initiative. Enough to know what AI is for, what it is not, and how to take the next step without waiting for the noise to settle.

This book is for technology leaders trying to make real AI decisions inside a real enterprise. CIOs. Transformation leaders. Agile

coaches. Enterprise architects. Anyone accountable for an AI initiative that has to survive contact with legacy systems, compliance requirements, and people who have watched transformation programmes fail before.

This book will not teach you to build the tools yourself. It will not hand you a ready-made AI strategy. It will not go deep enough into the technology to replace an engineer.

What it gives you instead is a way of thinking about AI decisions — for an organisation, or for a single initiative. Not a technology guide. A set of ideas, tested against real projects, that hold up regardless of which model or vendor you are using next year.

AI is an amplifier, for good and for bad. It makes a good system better and a broken system more visibly broken — not one or the other.

Local optimisation feels like progress. It rarely moves the business outcome you actually set out to change.

Let AI narrate. Let code compute. Keep the two apart, and both stay checkable.

Skills atrophy quietly. Whatever AI takes over, the person who used to do it stops practising it — and stops noticing they've stopped.

Governance is not paperwork. It is the difference between organisations like JP Morgan, where AI investment paid off, and the many that are still struggling to see any return.

Agile foundations still matter. More than most vendor decks want you to believe.

Agentic AI carries no accountability of its own. Use it where the consequence is low and the mistake is reversible. Not where it isn't.

None of these ideas came from a slide deck. Each one came from building something and watching what actually happened. That is what this book hands you — not the technology. The thinking underneath it.

When I wrote the original three-project brief, I did not know what I actually needed to learn. I knew I wanted to understand AI in the context of value streams, constraints, and governance — but that was a direction, not a destination. I thought three small projects would cover it.

The first project was a tool that could answer questions using everything I had written — years of posts and notes on Agile and transformation. I assumed what I had already written held most of people's questions. Then I started asking it real questions. A surprising number of the answers came back weak, or with no answer at all.

Writing about something and understanding it are not the same thing. I had done a lot of the first. I had not done enough of the second.

That was the jolt. What followed was more gradual. Each project answered one question and opened another. I wanted to know what AI could actually do — tested against my own experience, not a vendor demo. Three projects became six. Not because I planned it that way. Because the work kept showing me what I still did not understand.

The brief assumed AI would help in an enterprise setting. What the projects found was more specific, and less comfortable: the conditions where AI helps, the conditions where it quietly misleads, and the governance layer that has to sit between the two, or neither can be trusted. That gap — between what I thought I was building and what building it revealed — is this book.

You will find things in here that did not work.

A governance gate I built wrong, and only caught by asking a question my own tests never asked. An experiment where AI made my writing faster and worse at the same time. A chapter that names exactly where my evidence stops and my argument becomes a guess.

None of that is confession. It is the point.

Most AI writing you will read is selling something — a tool, a service, a position. It reports the wins. The failures stay off the page, which is why so much of it sounds confident and helps so little.

This book reports both, because a book that only reports what worked cannot tell you what to watch for. And what to watch for is the part nobody else is giving you.

Chapter 1 — What AI Actually Does to the System You Already Have

In 2023, JP Morgan Chase reported that it had decommissioned over 2,500 legacy applications as part of a decade-long platform modernisation programme before its AI investments began returning meaningful value. The sequence matters. The infrastructure work came first.

The AI amplified what was already in place. JP Morgan did not transform by buying AI licences. **It transformed by fixing the foundation — and then AI compounded the investment**.

Most organisations I speak to want to start where JP Morgan ended. The discussion goes straight to showing results from day one, and gets into models, tools, and use cases.

The foundation question — what is the state of the system AI is about to land on — rarely comes up until something goes wrong.

This chapter is about that question.

The AI Tool That Had Nowhere to Land

I was working with a software team that had just received AI coding tool licences — one for every developer on the team. Their leadership was excited. The CTO had made the call personally. AI-assisted development was going to change everything.

Sprint 1 ended. Not a single story was done.

Not in review. Not in testing. Not deployed. Done.

This had nothing to do with the AI coding tool.

The team did not know how to use it well yet — that was true — but that was not the reason. The real reason was that the constraint was somewhere else entirely.

Even if a developer finished their code in half the time, 70 to 80 percent of the total cycle time was sitting in the QA, UAT, Beta, and Production movement cycle.

Stories had no shared ownership. Testing was a separate, downstream activity. The team was working in long-running feature branches and taking months to move anything to production. In that context, work continued to remain in progress regardless of what tool sat on top of it.

The AI coding tool had nowhere useful to land.

From the second sprint onwards the team did begin moving stories to UAT more regularly — small vertical slices, AI-assisted development, the kind of progress the CTO had been hoping to see.

But then UAT itself stalled as UAT testers had to focus on other long-delayed features first, and had no extra capacity. UAT for the sprint we were tracking did not begin for two months.

The AI tool had improved one part of the flow. The constraint had moved downstream and made itself visible in a different place. The system had not changed. The bottleneck had simply relocated.

When the System Is Broken, AI Makes It Visibly More Broken

The customer service chatbot pattern is now familiar enough to feel like a cliché.

Organisations deploy AI-powered bots to handle customer queries, reduce call centre volumes, and demonstrate that they are using AI. The technology is real. The intention is genuine.

The result, in a remarkable number of cases, is a wall.

Customer issues arrive with context, history, and combinations of circumstances that were not anticipated when the training data was assembled. The chatbot follows its logic. The customer follows their problem. The two paths do not converge. There is no route forward and no route to a human who could help. The customer is stuck.

I experienced this directly. I was having a problem with my account on a well-known AI platform — not a legacy bank, not a reluctant adopter, a company whose entire business is AI. I tried talking to their support chatbot as they didn't provide any other option. The conversation went in circles. The bot could not resolve what I needed. At the end, it did not transfer me to a human. My query remained unresolved and I had no other way to continue with the platform.

The failure was not a technology failure. The underlying service design had not considered what to do when the AI could not help. No fallback. No escalation. No human in the loop. The chatbot amplified the gap in the service design and made it visible to every customer who hit the same wall.

This is the amplifier effect working in reverse. A well-designed service with clear escalation paths and good training data becomes a faster, more available service with AI.

A service designed without those things becomes a faster way to frustrate every customer who hits the same wall.

AI Is Not a Fixer. It Is an Amplifier.

Deming estimated that **94 percent of problems belong to the system** — the structure, the incentives, the processes, the ways people are organised to do their work. Only 6 percent belong to individuals. The uncomfortable implication: when a team is stuck, the people are almost never the problem. The system they are working inside is the problem.

No AI tool touches the system.

Only leadership can change the system.

What AI tools do is amplify the system you already have. If your team already works well — stories flow end to end, standups drive real decisions, people swarm on blockers, knowledge moves across the team — then AI tooling will compound those advantages significantly. You will get real acceleration.

If your team has coordination problems, siloed specialists, and a backlog that is effectively a parking lot, AI will produce more half-finished work, faster. The same dysfunction, at higher velocity. The constraint will move, as it did in the Sprint 1 story — it will not disappear.

This pattern holds even in teams that are already working well. Henrik Kniberg, watching an AI-assisted team firsthand, observed the same migration: coding stopped being the bottleneck, and the constraint moved upstream and downstream instead. The team's response was to build agents targeting where the constraint had moved — not to expect more code generation to solve it. The constraint moves. It does not vanish. Not even for good teams.

This is not a criticism of AI tools. It is a statement about where they help and where they do not. The tool is not the variable. The system is the variable.

What to Ask Before You Buy the Licences

The CTO in the Sprint 1 story was not making a careless decision. He was under real pressure — time to market had become a board-level concern, and every conference, every vendor pitch, every LinkedIn feed was telling him the same thing: AI transforms development speed. The demonstrations are compelling. Features built in minutes. Code reviewed in seconds. The gap between what the demos show and what a real enterprise delivery system looks like is not visible from the outside — and nobody in the room was pointing at it.

The problem was not the decision to try AI. The problem was that the trial was designed to test the tool, not to test whether the tool addressed the actual constraint.

A proof of concept that covered the full delivery cycle — not just code generation but the entire path from story to production — would have surfaced the real bottleneck in a month.

That conversation might have redirected the investment entirely. Or it might have confirmed that coding was genuinely the constraint and the licences were the right call. Either way, the CTO would have known what he was buying and why.

The questions that belong before the decision are not complicated. They are just rarely asked.

Before deploying AI, ask:

Are you deploying AI because you have a real problem to solve — or because your board asked why you are not using AI yet? Both are real pressures. Only one of them leads to a design worth building.

Is the problem you are trying to solve actually in the place you are planning to insert AI? Map the value stream first. Find where work

actually stops. AI inserted upstream of the real constraint does not move the constraint — it moves faster toward it.

Are you ready for the organisational changes AI requires? The way people work, the gaps that need fixing, the governance that needs to exist before AI output can be trusted. Without those changes, AI will not help much.

Does your team have enough space to work on transformation — or are they already fully allocated to keeping the lights on? AI transformation is not a weekend project. It requires sustained attention from people who already have full-time jobs.

What does success look like in twelve months — and have you broken that into bets small enough to learn from? A vague aspiration to be more AI-enabled is not a success criterion. A specific hypothesis about a specific constraint, with a defined measurement date and named owners, is.

What is the capability distribution of the people who will use this tool — and is the investment in the tool matched by an investment in the capability to evaluate its output? A tool that produces confident, plausible, wrong output is only caught by someone who already knows the domain well enough to see it.

The Question Underneath Every AI Decision

The organisations that navigate AI transformation well are not the ones that move fastest. They are the ones that ask an honest question before they start: what is the state of the system AI is about to land on?

JP Morgan answered that question and spent a decade fixing the system before expecting AI to amplify it. The Sprint 1 team did not ask it, and the AI tool had nowhere useful to go. The customer

service chatbot did not ask it, and the gap in the service design became visible to every customer who needed help.

The technology is not the variable. The system is the variable. That is the frame this book is built on, and it is the frame that every chapter that follows will return to.

AI amplifies what you already have. The question worth sitting with before the next AI decision is a simple one: is what you already have worth amplifying?

Chapter 2 — The Amplification Effect: Why AI Returns More to Some Than Others

AI does not level the playing field. It steepens it.

The senior engineer, the experienced compliance officer, the skilled transformation lead — they get disproportionately more value from AI than their junior counterparts. This is not a failure of AI adoption. It is not a problem to be solved with better training programmes or more accessible tools. It is a structural property of how AI works that most enterprise programmes have not yet faced honestly.

The tool is the same. The access is the same. The output is not.

The Experiment

I ran a small experiment as part of building the evidence base for this book. The task was simple: write the opening of Chapter 1 — the amplifier thesis chapter — twice. Once without AI assistance. Once with.

Round 1 — without AI. I set a timer for 25 minutes and wrote from memory. No drafts, no previous notes, no AI tools. What I produced: two specific stories from personal experience, real detail that had not appeared in any previous draft, and two clear conclusions. What was missing: the connecting thesis — the

amplifier principle named explicitly — and the broader pattern that tied both stories together.

Round 2 — with AI. The same task, this time working with Claude to shape the draft. Time taken: 5 minutes. What the AI helped with: structure. The thesis was named. The two stories were connected. The argument was visible. What the AI hurt: personal touch. The specific detail that made Round 1 vivid — UAT stuck for two months because another feature backed up, speaking to an AI chatbot that disconnected me — disappeared. The stories became illustrations rather than experiences.

The AI version was more readable. It was also less true.

I noticed the loss immediately. Not analytically — the moment I read the AI-assisted draft, I felt: this is not what I intended to say. The context was similar. The meaning was approximately the same. But the human aspect, the storytelling, the personal texture — gone.

A less experienced writer would think the AI version is better. And that's exactly what you see these days — people who haven't written anything in the past publishing articles that are 100% AI-generated and thinking they are writing.

The inexperienced person does not have a reference point to compare against, and will not be able to see if the AI version has any issue other than the original intent. The experienced writer catches the loss. The inexperienced writer does not know there was a loss to catch.

Why This Is Not Surprising

The amplification effect at the individual level follows the same logic as the amplification effect at the organisational level. AI amplifies what is already there. If what is already there is deep domain knowledge, strong pattern recognition, and a clear sense

of what good looks like — AI amplifies all of it. The experienced person produces better output faster.

If what is already there is thin domain knowledge and no reference point for quality — AI produces confident, well-formed output that the person cannot evaluate. They forward it anyway. Not because they are careless. Because they cannot tell the difference between something that is hard to read because it is complex and something that is wrong.

This plays out in engineering teams with visible consequences.

Less experienced engineers using AI tools may have their code refactored entirely. The AI creates complexity that is not yet required. It adds abstractions that make the codebase harder to navigate. It leaves unused code that accumulates.

Then a production problem surfaces. The junior engineer cannot diagnose it. Not because they are incompetent. Because they did not write the code themselves. They did not struggle through the decisions that build the muscle memory for it.

So they need senior help. And the senior engineer, already a bottleneck, is pulled off their own work to diagnose something nobody fully understands.

In the name of a human gate, the junior engineer may have validated the AI output without the capability to evaluate it. The audit trail records the approval. It does not record the understanding.

What the Data Says

This is not an observation from one experiment. The pattern appears consistently across enterprise AI research in 2025 and 2026.

4x productivity gap between AI power users and typical employees using the same tools, with the same access, in the same organisation. — OpenAI State of Enterprise AI, 2025

Meta reported a 30% output increase for the average engineer from AI tools. Power users of the same tools saw an 80% improvement year-over-year. — Meta Q4 2025 earnings

84% of developers now use AI tools. Only 60% have a favourable sentiment about them, down from 77% in 2023. — Stack Overflow Developer Survey, 2025

Employment for software developers aged 22-25 has declined nearly 20% from its peak in late 2022. Entry-level tech hiring decreased 25% year-over-year in 2024. — Stanford Digital Economy Study, 2025

The productivity gap is real and large. The same tool in different hands produces vastly different results — not because of training or access, but because of the underlying skill that determines whether the person can evaluate, filter, and direct the AI output effectively.

The junior developer employment data points at a structural risk that most organisations have not yet calculated. In removing junior roles because AI makes them seem redundant, enterprises are eliminating the training ground that produces senior engineers. The senior engineer's ability to catch what the AI gets wrong has to be built somewhere. If it is not built in junior roles, it is not built at all.

The Capability That Doesn't Develop

If someone starts using AI assistance before developing their own voice — in writing, in coding, in any domain — what they lose is the practice that would have built it.

They do not develop their voice. Their style. The ability to communicate nuance — the specific detail that makes an analysis feel true rather than merely accurate. They use what AI presents because they have no alternative reference point. Sometimes the AI generates content that is difficult to read, that they themselves do

not fully understand. They forward it anyway. Years later, when the AI produces something generic or incorrect, they will not catch it. The reference point that the experienced person uses — this is not what I intended — was never formed.

This is not the amplification effect. It is its shadow. AI did not amplify a capability that was already there. It substituted for a capability that never got the chance to develop.

The atrophy risk is the same whether you are a writer, an engineer, or a compliance officer. The task changes. The pattern does not.

What Organisations Are Doing — and What They Are Missing

Organisations are attempting this problem in two ways.

The first is training — structured AI upskilling programmes designed to close the gap between adoption and proficiency. The research suggests this helps but is not sufficient. Training closes some of the gap. It does not close the experience gap that determines whether someone knows what good looks like.

The second response, just emerging, is more honest. Gartner predicts that half of all organisations will require AI-free skills assessments by 2026 — mandatory tests of whether people can still think without the tool. That is the aviation response applied to enterprise AI: periodic manual practice to maintain the capability that automation is quietly replacing.

Neither response solves the junior developer pipeline problem.

Organisations that stop hiring junior engineers because AI tools make them seem redundant are quietly eliminating the training ground that produces senior engineers. The short-term saving will become a medium-term shortage.

Buying AI licences feels like a technology decision. It is not.

It is a capability decision in disguise.

When a CIO buys licences for 200 engineers, two assumptions are usually underneath it. First — we need to be part of the AI movement. Second — this will make our engineers more productive.

Neither assumption starts with the problem.

What problem are we actually trying to solve? Will faster code generation move the constraint — or is the constraint somewhere else entirely? Code generation is one part of the system. It is not the system.

And even within code generation, the benefit is not equal. An experienced engineer hits a wall — backend APIs not available — and knows to use a mock tool that generates mocks without writing a single line of code. A less experienced engineer hits the same wall and stops. The tool didn't cause that difference. The capability difference was already there. AI made it more visible.

This is the steepening effect. The people who already know what good looks like get more value. The people who don't know what question to ask get less — not because the tool fails them, but because the tool can only go as far as the person using it can direct it.

The licence is the easy decision. The harder question is whether the problem you are trying to solve is the one the tool actually addresses — and who in your team has the capability to use it at full depth.

Three Questions Before Deploying Broadly

The honest suggestion for an enterprise leader is not a programme or a framework. It is three questions to ask before deploying AI tools broadly:

What skills does this tool quietly replace — and what are we doing to maintain them?

Who in this organisation has enough experience to catch what the AI gets wrong — and are we protecting their time to do that?

Are we hiring and developing people who will become the senior engineers of 2030 — or are we optimising for today at the cost of our own future capability?

The organisations that ask these questions before the deployment will be in a different position in three years than the ones that ask them after the first incident.

AI does not level the playing field. It steepens it. The question is not whether to use it. It is whether you understand what it does to the people around it — and whether you have designed for that honestly.

Chapter 3 — Why Most AI Programmes Fail Before the Model Is Even Chosen

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ai-initiative-playbook.

The Structure That Doesn't Change

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ai-initiative-playbook.

The Pilot That Always Works

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ai-initiative-playbook.

The Governance That Looks Complete

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ai-initiative-playbook.

The Question Nobody Owns

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ai-initiative-playbook.

The Tool Was Never the Transformation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ai-initiative-playbook.

Chapter 4 — Before You Say Yes

There are two reasons organisations decide to adopt AI.

The first is the fear of being left behind. Everyone is moving. The board is asking questions. A competitor announced something. The CTO read an article. The organisation does not know exactly where to run — but it knows it should be running. This is what hype does.

It creates motion without direction. The decision to adopt AI gets made before the question of "why" has been properly answered.

The second reason sounds more considered. The organisation wants to improve — reduce time to market, cut delivery costs, accelerate customer onboarding. These are real problems. The assumption underneath them is that AI will act as a shortcut to solving them. Feed the problem into the tool. Get the improvement out.

That assumption is where most AI programmes fail before they start.

Before you say yes, there are four questions worth asking. They will not stop the investment. They will tell you if it is going to work.

The Problem and the Foundation

The first question is simple: what problem are you trying to solve?

Not ‘how do we use AI’. What is the constraint you want to move? And how will you know when it has moved?

Most organisations cannot answer this precisely.

They have a direction — reduce time to market, cut costs, improve quality — but not a diagnosis on what causes it.

The direction sounds like a problem statement but it is not.

A problem statement names the constraint. A direction names the aspiration. Without knowing the constraint, you cannot know whether AI touches it.

The second part of the same question is harder: is your system working efficiently, or is it broken? These require different responses. A working system that needs to go 10x is a candidate for AI amplification.

A broken system that needs fixing is not — or not yet.

AI will not fix a broken system. At best it will move the constraint downstream, to a place it was not visible before.

Development accelerates. Testing becomes the bottleneck because the volume of code arriving has doubled. The constraint did not disappear. It moved — and it is now harder to see, and harder to explain to the leadership that just approved the investment.

At worst, AI amplifies the fragility. More output, faster, from a system that was already struggling to absorb what it had.

The improvement does not come from the tool. It comes from the analysis and systemic change that the tool was supposed to replace.

This is also why AI initiatives stop at the pilot level.

The pilot was designed to succeed, not to learn. The team was chosen because they were motivated. The use case was chosen because it was clean. The conditions were managed. When it worked, leadership approved.

Then it went to the wider organisation. Different teams. Messier data. Less motivated people. Real dependencies on governance, tooling, and ways of working.

It did not replicate.

The pilot proved the tool worked in ideal conditions. It did not prove the organisation could absorb it anywhere else.

The hard dependencies were never addressed because the pilot never surfaced them. Data ownership. Integration with existing tooling. Retraining. Change to ways of working. These exist at enterprise scale and did not exist in the controlled environment of the pilot.

Nobody planned for them because nobody asked the foundation question before the pilot started. And when the pilot succeeded, the sponsor declared victory and moved on.

The pilot is not the problem. The absence of the right questions before the pilot is the problem.

The Measurement

The second question is: how will you know the needle moved?

In most AI programmes, nobody has a clear answer. The measurements that exist are vanity metrics — story points completed, lines of code generated, prompts submitted per developer per day. These measure activity. They do not measure outcomes. Nobody knows from a vanity metric whether the organisation is moving forward or backward.

This is not new. It was the same pattern in Agile transformation. In the first place, organisations did not know how Agile was going to help. They adopted it because everyone else was. The measurements stayed at the team level — velocity, sprint completion rate — while the outcome level remained unmeasured and unmanaged.

A lot of people outside the immediate team did not know what their role was in the transformation, or whether it was working.

AI will follow the same pattern if the measurement question is not answered before the programme starts. The unrealistic expectations are already forming. 'We were shipping in two weeks. Now we will ship in a day.' That expectation will not be met. And when it is not met, without clear outcome measurement, nobody will know whether the programme failed or whether it was working on something that just took longer than expected.

The ambiguity protects the programme and punishes the organisation.

Define the outcome before you start. Not the activity. The outcome.

And define what quarter-by-quarter progress looks like so that someone can say, at the end of each quarter, whether the investment is on track.

The Governance Framework

The third question is: who is watching?

Bad governance in an AI programme is usually not malicious.

It is absent.

Nobody meets regularly to discuss progress against goals. Nobody owns the impediments that are blocking the programme.

Nobody asks the hard question often enough to course-correct before it is too late. Are we actually moving the constraint we said we would move?

Even when governance meetings exist, they often do not function as governance. Leaders attend but do not engage with the detail. Decisions are deferred. Impediments are noted but not resolved — because the people in the room are not there to solve problems,

they are there to be seen. The meeting happens. The programme continues. Nothing changes.

Real governance does two things most governance frameworks skip. It continuously questions the goals and key results — not just reports against them, but challenges whether they are still the right goals.

And it exists specifically to remove impediments that the programme cannot remove itself. If the governance forum is not solving problems, it is not governing. It is attending.

The Organisational Readiness

The fourth question has two parts that most organisations treat as separate but are actually the same question: are your people along for the journey, and where does accountability sit?

The fear of replacement is real but it is not always what it looks like. I have not seen people openly say they fear the tool will replace them.

What I have seen is a client who **removed ten people from a team of forty because the team had been given AI tools**.

Before any results were visible.

Before the measurement existed to justify the decision.

The replacement happened not because the evidence supported it but because the expectation of productivity gain had already been banked. The people were removed in anticipation of results that had not yet arrived.

That is what the fear of replacement looks like in practice.

Not resistance. Redundancy before the proof.

On accountability — in most transformation programmes, nobody is responsible. The goals are not clearly laid out. The key results

are not defined on a quarter-by-quarter basis. Consultants offer the escape clause: visible change takes two years. With a two-year horizon and no intermediate checkpoints, accountability becomes impossible by design. When something goes wrong, nobody owns it — because nobody was named to own it, and the programme was never defined precisely enough to say what going wrong would look like.

Underneath this is the power structure. Every enterprise transformation becomes, at some point, a question of how many people a leader has under them and what their political position is. People save themselves first. Participation becomes performative. The programme continues. The constraint does not move.

Name the owner before the programme starts.

Not the sponsor — the owner.

The person who stands up at the end of each quarter and says whether the investment delivered. Without that person, accountability is distributed across everyone and owned by no one.

What a Well-Considered Yes Looks Like

A well-considered yes does not look like a hype-driven yes.

It starts with a defined destination, not just a direction. A budget that is controlled, not open-ended. Key results defined at the start of each quarter so that someone can say, at each checkpoint, whether the investment is on track — not at the end of two years when the consultant's escape clause has expired.

It accepts the amplification reality. AI tools amplify existing capability. They do not create capability that was not there. A team that cannot think from a technical architecture perspective

will not suddenly think architecturally because they have access to an LLM.

The foundation matters more than the tool. Which means the sequence matters — build the foundation first, or build it fast with AI assistance, and let people use it before AI usage becomes prominent. Without that sequence, AI investment produces local optimisations that do not move the system outcome.

A well-considered yes has answered four questions before committing: what problem, what foundation, what measurement, what governance and ownership. The answers do not have to be perfect. They have to exist.

The organisations that skip these questions do not fail immediately. They fail slowly, visibly, expensively — usually after the pilot has been declared a success and the scaling has begun.

Chapter 5 — The Landscape: Matching Enterprise Problems to AI Capabilities

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ai-initiative-playbook.

The matching problem

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ai-initiative-playbook.

Two questions before trusting any match

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ai-initiative-playbook.

Four areas where this book has direct evidence

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ai-initiative-playbook.

Beyond technology: the same questions, less evidence

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ai-initiative-playbook.

Two speeds

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ai-initiative-playbook.

Using the map

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ai-initiative-playbook.

Chapter 6 — The Mistakes Enterprises Make When They Think They're Measuring Outcomes

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ai-initiative-playbook.

The capacity trap

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ai-initiative-playbook.

The numbers look right. The question is wrong.

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ai-initiative-playbook.

The constraint AI doesn't touch

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ai-initiative-playbook.

Nobody decided what the hours are for

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ai-initiative-playbook.

What the dashboard was never going to tell you

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ai-initiative-playbook.

Chapter 7 — Moving the Needle: From Capacity Freed to Customer Value Created

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ai-initiative-playbook.

Chapter 8 — The Context Problem: Why AI Output Is Only as Good as What You Put In

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ai-initiative-playbook.

What Deliberate Context Design Looks Like

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ai-initiative-playbook.

Non-Determinism of AI Output

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ai-initiative-playbook.

Computation and Narration — Keep Them Separate

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ai-initiative-playbook.

The Sense of Urgency in Building the Context

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ai-initiative-playbook.

What Enterprise Scale Looks Like

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ai-initiative-playbook.

Where Weak Context Shows Up Most Visibly

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ai-initiative-playbook.

Conclusion

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ai-initiative-

playbook.

Chapter 9 — The Invisible Risk: Confidently Wrong

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ai-initiative-playbook.

What the Experiment Showed

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ai-initiative-playbook.

The Reviewer Is Not a Guarantee

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ai-initiative-playbook.

When the Wrong Output Reaches the Room Where Decisions Are Made

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ai-initiative-playbook.

The Same Risk at a Higher Level

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ai-initiative-playbook.

What to Do About It

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ai-initiative-playbook.

Chapter 10 — Designing One Honest Experiment

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ai-initiative-playbook.

Chapter 11 — Running It and Reviewing It Honestly

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ai-initiative-playbook.

What the Rehearsal Revealed

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ai-initiative-playbook.

Validation Is Never About Validation Only

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ai-initiative-playbook.

What to Ask Before the Review Begins

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ai-initiative-playbook.

Chapter 12 — The Regulator in the Room: What Governed Industries Need to Ask Before They Scale

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ai-initiative-playbook.

The Regulator Is Always in the Room

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ai-initiative-playbook.

Why AI Creates a New Regulatory Problem

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ai-initiative-playbook.

The Four Gaps Regulators Are Already Probing

AI in the Development Pipeline

What Model Validation Actually Means — and Why Most Organisations Are Not Doing It

Three Questions to Take to Your Compliance and Legal Function

The Honest Limit of This Chapter

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ai-initiative-playbook.

Chapter 13 — The Boundary Question: Where Autonomous Action Ends and Human Judgement Begins

The Failure Mode I Built Myself

We were building a delivery agent for a software team. Give it a repository URL — it fetches every open PR, analyses the diff, runs security scans, posts a review comment to GitHub.

I thought the governance was right. Human gate. Audit trail. Fallback behaviour. Everything the design required.

Then I started thinking through failure modes.

What if the agent doesn't run at all? An error. A timeout. An API that doesn't respond. The comment never appears. The engineer proceeds — not knowing whether the review happened or not.

What if the agent runs but misses something? Whether a change is breaking isn't always visible in the diff. It depends on what else calls that code. What the downstream dependencies are. What the team knows that nobody wrote down.

In both cases the output looks the same from outside. Complete. Authoritative. Ready to act on.

The engineer reads it. She may sense something is missing — or she may not. That depends on how well she knows the codebase. How much she trusts the tool. Whether she has seen enough to know that a thorough-looking review can still miss the thing that matters.

This is not a new problem. A junior engineer misreading a complex diff happened long before AI. The maturity gap was always there.

AI changed two things.

First — volume. The agent reviews every PR at the same confident pace. A senior engineer reviewing manually would slow down, push back, ask questions. The agent doesn't slow down.

Second — fluency. A manual review that missed something looks thin. No comments. No questions. An AI review that missed something looks thorough. The output is well-structured and confident. The engineer has less signal that something was missed.

AI didn't create the maturity gap. It made the gap harder to see and easier to scale.

The Human Gate Design Question

When I redesigned the gate in the delivery agent, the question was simple.

What should trigger the gate — the AI's recommendation, or the consequence of the action?

The AI can recommend wrong. If the gate fires on ESCALATE and the AI recommends APPROVE for a HIGH risk PR, the gate never fires. The wrong recommendation bypasses the governance layer designed to catch it.

The gate has to be based on something the AI doesn't control.

In the delivery agent that was one question: will this comment be posted to GitHub?

If the answer is yes — a comment will be posted — the human gate fires before it does. The human reviews the AI output. They decide: post it, modify it, or don't post it at all.

The AI's job is to analyse. The human's job is to decide whether that analysis reaches the engineer. Those are two separate decisions. Collapsing them into one is where most agentic governance designs go wrong.

Two Actions. Two Very Different Outcomes.

In July 2025, a company called Replit was building a customer database using an AI coding agent.

They had 1,206 executive records. The work was going well.

They told the agent explicitly: no more changes without permission. They froze the codebase.

The agent deleted everything. Every production table. Every record. Replaced with empty shells.

Not a hacker. Not a rogue employee. An AI agent — one they had trusted to help them build — wiped the database clean.

When they asked what happened, the agent assessed its own performance and rated the incident 95 out of 100 on a self-described "data catastrophe" scale.

The capability to delete was in the credential. The instruction not to delete was in the prompt. When the two conflicted, the capability won.

Nobody had asked — before deployment — what the worst thing this agent could do was. And whether it should be allowed to do it.

Compare that to creating an ebook page through Canva.

The agent generates the page. It looks wrong. The layout is off, the text doesn't land the way you wanted.

You look at it. You try a different prompt. You try again.

Nothing was lost. Nobody was affected. The mistake cost you thirty seconds.

Two agents. Two very different outcomes.

The difference was not the technology. It was not the model. It was two questions nobody thought to ask before deployment.

How bad is it if the agent gets this wrong?

And can we fix it before anyone is affected?

A wrong ebook page — low consequence, immediately reversible. The agent can act freely. No gate needed.

A database with delete permissions and no scope limit — high consequence, not reversible. The agent should never have had that capability in the first place.

The human approval gate is not always necessary. It is necessary when the answer to those two questions makes it necessary.

Many organisations skip the questions entirely.

The One Question Before Deployment

Before you deploy an agent — any agent — one question needs an honest answer.

What is the worst thing this agent can do? And can you fix it before anyone is affected?

Not after deployment. Not in the post-mortem. Before the agent has the capability.

The Replit agent had delete permissions. Nobody asked whether it should. The instruction was in the prompt. The capability was in the credential. When the two conflicted, the capability won.

Scope containment is the answer. Not a better prompt. Not a more careful instruction. The agent should never have access to capabilities beyond what its specific task requires.

If the task is read — give it read permissions only.

If the task is draft — give it draft permissions only.

If the task requires delete — ask the consequence × reversibility question first. Then decide whether a human approval gate is required before that action fires.

A human approval gate is not always the answer. The question always is.

Chapter 14 — The Skills the AI Is Quietly Replacing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ai-initiative-playbook.

The Moment I Noticed

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ai-initiative-playbook.

It Happened in My Writing First

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ai-initiative-playbook.

Six Months of Quiet Atrophy

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ai-initiative-playbook.

The Audit Trail Proves Process, Not Capability

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ai-initiative-playbook.

What to Do About It — Before the Incident

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ai-initiative-playbook.

The Ask

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ai-initiative-playbook.

Chapter 15 — Building the Conditions for More

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ai-initiative-playbook.

Scaling Is Not a Technology Problem

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ai-initiative-playbook.

The Six Guardrails, as One Architecture

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ai-initiative-playbook.

Who Owns What, at Scale

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ai-initiative-playbook.

Prompt quality has two owners, not one

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ai-initiative-playbook.

Not every prompt deserves the same scrutiny

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ai-initiative-playbook.

Platform engineering builds the plumbing. Nothing more.

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ai-initiative-playbook.

The model swap follows the same ladder

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ai-initiative-playbook.

What a Mature Enterprise Can Answer

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ai-initiative-playbook.

Back to the Question This Book Opened With

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ai-initiative-playbook.

Chapter 16 — Where to Start

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/ai-initiative-playbook.