The AI Strategy Playbook 2026

What separates the companies that scale AI from the ones that stall

The further a company gets into AI, the more its strategy has to carry. Where agents are allowed to act. Which work is worth automating at all. What the usage costs each month. Who is accountable when an answer turns out to be wrong. Each of those questions moves faster than an annual plan.

This playbook collects six strategies that larger organisations are using right now, and then takes each of them through a chapter of its own: what the problem looks like from the inside, what the insight is, and what to do about it on Monday.

The six strategies that decide whether AI scales

1. **Turn usage into business value.**
   The advantage is not buying more AI. It is managing it like a resource that costs something, and converting it into faster decisions, better work and less friction for employees and customers.
2. **Unfreeze the middle.**
   The biggest blocker is often the managers whose expertise, workflows and standing were built in a pre-AI organisation. They own the work that has to change.
3. **Clean the workflow before you automate it.**
   AI accelerates a bad process as readily as a good one. Map the work, name an owner and remove the broken steps before layering agents on top.
4. **Build a layer of in-house builders.**
   Give business users approved tools, data access and guardrails, so automations and agents get built close to the work instead of queueing with IT.
5. **Treat token costs like cloud sprawl.**
   AI spending follows the same path as cloud adoption: early enthusiasm, rapid growth, eventual scrutiny. Put governance around usage before the bill arrives.
6. **Become a better wisher, not a better prompter.**
   As models get better at execution, the advantage shifts to defining the right outcome, the context around it and the business problem worth solving at all.

AI as a management discipline

Chapter one
Value and the cost of usage

An AI strategy is not a tool rollout. It changes how value gets produced, which means the usage itself has to be budgeted, measured and expected, in the same way as any other input to the business. Experiments are worth funding now, while the cost of running models is still being pushed down by competition between providers.

> **Treat AI usage as a managed input, not a free benefit.**

That makes AI a management question rather than a procurement decision. It means saying what regular use looks like in each role, paying for the attempts that will fail, watching what the usage costs, and bringing every employee closer to a way of working where the tools sit inside the work instead of beside it.

Key insights:

1. **Manage inference like a business resource.**
   The companies that get further will not treat usage as free. They give teams access, measure what comes out of it, and learn where the expensive calls actually pay for themselves.
2. **Reduce friction on both sides of the business.**
   Look at the steps that slow employees down and the steps that slow customers down. Both are where the return shows up first.
3. **Give everyone agentic support.**
   Help that prioritises, drafts, searches and summarises belongs with the people doing the work, not only with the executives.
4. **Find the people already doing it.**
   Someone in most teams already works this way. Make their workflow visible instead of designing a new one from the top.

The practical takeaway: write down what regular use looks like in each role, fund a set of attempts you are willing to lose, and get one of them into daily work before the next budget round.

Choosing what is worth automating

Chapter two
Governance and the workflow

Governance is what lets experiments scale without turning into a risk register. The aim is not a permission queue: it is that employees know which tools are safe, which cases need a review, and where the line sits between using a tool and building something on top of it.

> **A broken process does not improve with an agent on top. It just runs faster.**

So the cheapest part of the whole exercise comes first: ask the owner to write down how the work runs today, on paper if that is quicker. If the workflow turns out to be unclear or to have no owner at all, that is the finding, and it is worth more than the automation would have been.

Key insights:

1. **Start with the workflow, not the tool.**
   Ask the owner to describe how the work runs today. An unclear or ownerless process gets worse, not better, with AI on top of it.
2. **Use intake to say yes faster.**
   Record every tool and use case, then mark the few categories that genuinely need a deep review so the rest can be approved quickly.
3. **Scale the control to the consequence.**
   Calendar automation does not need what a customer-facing message or a legal filing needs. Match the review to what is actually at stake.
4. **Mark where using becomes building.**
   Trying a tool and shipping an automation on top of it carry different risk. Say where that line sits, so people know when to ask.
5. **Keep a named owner for corrections.**
   When an output turns out to be wrong, someone has to be able to correct it and say what changed.

The practical takeaway: write the workflow down, mark the few cases that genuinely need a review, and let the rest through quickly. A queue that stops everything only teaches people to go around it.

The layer that decides whether AI lands

Chapter three
Adoption and infrastructure

Executives usually understand the urgency, and the people closest to the work are often willing to experiment. The harder layer sits between them: managers whose expertise, processes and standing were built in a pre-AI organisation, and who are now asked to drive a change that alters their own job first.

> **Adoption is rarely blocked by the choice of model.**

The infrastructure question still matters, but it is not which model to pick. It is how to build something that can swap a model out without rebuilding the stack, reach the data where it already lives, keep a record of what happened, and carry the knowledge that currently sits in people's heads.

Key insights:

1. **Design for change, not permanence.**
   Models, standards and patterns move quickly. An architecture that can swap a model out beats one tuned to today's best.
2. **Good data beats perfectly centralised data.**
   AI rarely needs the data arranged the way people need it. The real questions are where it lives, who can reach it and whether it may be used.
3. **Capture context, not just data.**
   The valuable part is often the judgement around the data: why a step exists and what a usable output looks like.
4. **Build bilingual teams.**
   A central team can build reusable capability; the business teams know where it applies. Both halves are needed.
5. **Match the model to the task.**
   Route routine work to cheaper models and keep the expensive ones for work that justifies the spend.

The practical takeaway: build for people before you build for agents. The managers in the middle decide whether anything downstream of them ever gets used.

A shared foundation, local execution

Chapter four
The operating model

An AI strategy that is generic stays generic. The usable version is mapped to the service lines, industries and work patterns where value is actually created, and it says who does what once the architecture is in place.

Standardise the foundation, decentralise the execution, govern the economics.

A practical pattern is a common foundation for everyone, then a deeper role-specific layer on top. For a software team AI changes the development cycle; for a service team it changes how people and agents share a conversation. The foundation makes the vocabulary shared, the layer on top makes it useful.

Key insights:

1. **Give everyone the same base.**
   Shared fluency in what the tools do, where they fail and what responsible use means, before anything more advanced is attempted.
2. **Make the advanced layer role-specific.**
   The roadmap should follow the work people actually do, not the order the technology arrived in.
3. **Move execution into the business units.**
   Once the guardrails exist, the teams that understand the work should be the ones implementing it.
4. **Centralise the economics.**
   Budgets, model standards and vendor management belong in one place, or the same spend gets made twice.

The practical takeaway: one architecture and one shared vocabulary, teams free to solve their own problems, and a single place that owns the budget.

Scaling agents without losing control

Chapter five
Agents at scale

Agents become useful at scale when people, process, data and change management are connected rather than piloted separately. Without that, agent work stays in pilots and never reaches anybody's working day.

Let departments build, inside guardrails you can see into.

The practical middle ground is a layer of in-house builders: employees close to the work who build low-code automations inside approved guardrails, with the logging and controls a regulated environment requires. It is faster than a central queue and considerably safer than people doing it quietly.

Key insights:

1. **Make data ready to be consumed.**
   Governed, accessible and owned by someone. A shared semantic layer keeps access from becoming a free-for-all.
2. **Connect the parts.**
   People, process, data and change management that do not connect leave agent work stuck in experiments.
3. **Keep telemetry and an audit trail.**
   What was asked, which policy applied and what was prevented. Needed the first time somebody asks what happened.
4. **Route through one gateway.**
   A single path for prompts gives transparency and evidence without a separate review for every tool.

The practical takeaway: the fastest route to scale is not a bigger central team. It is shared data, shared guardrails and a platform the business units can build on themselves.

A tool strategy people actually use

Chapter six
The tool strategy

Tooling is as much a culture question as an infrastructure one. The stack has to move data quickly and securely across environments, but it only produces value when people adopt it and use it in the right ways.

The right stack is the clearest one, not the biggest one.

Encouraging experiments and orchestrating them are not in conflict. Letting hundreds of tools appear unmanaged is where it goes wrong: unclear data access, licences nobody tracks, and no way to say which of them earned its place.

Key insights:

1. **Keep sprawl visible.**
   Someone needs sight of usage, data access, licences and whether a tool produces measurable business value.
2. **Make adoption visible.**
   Let people show what they built and what failed. Recognition does more for adoption than another mandate.
3. **Build operators, not users.**
   Teach people to give context, define the outcome and orchestrate a workflow, rather than to write cleverer prompts.
4. **Expect the bottleneck to move.**
   From people to systems to compute to power. The work is redesigning around the next constraint.

The practical takeaway: give people access, teach them to use it, and keep sight of what is running. The bottleneck will keep moving; the job is to keep redesigning around the next one.

proveo

The strategy is written. Can the team execute it?

See how Proveo measures that

Inspired by Section's The AI Strategy Playbook 2026. Written by Proveo; the wording, examples and recommendations are our own.