

THE FUTURE OF WORK. THE POWER OF PEOPLE.

The AI Jobs

MAGAZINE

STOP HIRING AI ENGINEERS. TRAIN YOUR BAs AND PMs INSTEAD.

THE BANKS WINNING WITH AI ARE
UPSKILLING THEIR BUSINESS AND
DELIVERY TEAMS.



THE REAL AI ADVANTAGE
It's not your tech stack.
It's your people.



BEYOND AUTOMATION
How PMs and BAs drive
AI strategy, adoption and
measurable value.



UPSKILL. DON'T REPLACE.
Why the future belongs to
augmented teams, not
automated ones.

**LEAD THE SHIFT.
YOUR BUSINESS
EXPERTISE IS THE
COMPETITIVE EDGE
IN THE AGE OF AI.**

Stop Hiring AI Engineers. Train Your BAs and PMs Instead.

The model isn't the product. The wrapper is. And building wrappers is a management discipline.

OpenAI's Codex team shipped roughly a million lines of production code last quarter. **Not one of them was written by hand.**

Anthropic's own coding tool, Claude Code, is more than 512,000 lines wrapped around their model. Five compaction strategies. A three-layer pointer-based memory system. An always-on background agent that monitors your repository while you sleep. The model is the small part.

If the company that built the model still needs half a million lines of *wrapper* code to make it reliable in a single coding tool, it should change how you think about every "AI wrapper" your vendors are pitching as if it were a toy.

The wrapper is the product. And the work that builds a good wrapper is not coding.

It is business analysis.

Three misconceptions costing corporations money

Before another budget cycle disappears into AI experiments, kill these three.

1. "The model is the product."

It is not. ThoughtWorks now puts it as flatly as you can: **Agent = Model + Harness**. The model is raw intelligence. The harness is everything that turns intelligence into a reliable system, the constraints, the feedback loops, the documentation, the decision log, the controls.

Strip away the harness and you have a language model guessing its way through your codebase. Add the right harness and you have a system that ships production code.

OpenAI calls this *Harness Engineering*. Anthropic published three papers on it inside a month. Red Hat wrote implementation guides. ThoughtWorks formalised a framework. A new engineering discipline emerged in ninety days.

Note who is missing from that list: **nobody is calling it programming.**

2. "AI projects need more engineers."

They need fewer engineers and more *constraint designers*.

Vercel removed 80% of the tools their agent had access to. Performance went up, not down. OpenAI's Sora Android team, four engineers, twenty-eight days, shipped an app that hit number one on the Play Store with a 99.9% crash-free rate. The engineers spent their time on architecture, scope, and verification. The agent did the typing.

The bottleneck moved from "can we write the code" to "can we describe the system clearly enough that the agent writes the right code". More of a business analyst's or project manager's job?

3. "Once we deploy the AI, we are done."

You are not done. Anthropic's own data shows that every model upgrade made part of their own wrapper unnecessary. Components that were essential in March were dead weight by April. They call it *harness decay*.

A wrapper is not a one-time build. It is a control framework, a living set of decisions, constraints, and reviews that has to be maintained. Treat it like a product, not a project.

Bottom line: if your AI strategy is "buy the model and plug it in", you are buying the smallest, cheapest part of the system and ignoring the part that decides whether it works.

What an AI wrapper is

Strip the jargon and a wrapper has four parts. Every serious deployment has all four. None of them require code to design.

1. **An operating manual.** A short document the agent reads at the start of every session, project goal, source of truth, what is forbidden, how *done* is defined. OpenAI calls it AGENT.md, Anthropic calls it CLAUDE.md, the principle is identical: onboarding documentation for a new starter who gets wiped clean every morning.
2. **A decision log.** A folder of short, numbered records that capture every non-obvious choice, vendor, data residency, escalation path, cost cap, threat model. Each one written once. Each one read by the agent on demand. This is the Architectural Decision Record, and it is the most underrated tool in corporate AI right now.

3. **A feature ledger.** A structured list of what is to be built, what is built, what is broken. The agent reads it, picks the next priority, ships it, marks it done, commits. A project board the machine can parse.
4. **A feedback loop.** Tests, reviews, audits, automated where possible, human where it matters. The trail your auditors and regulators will eventually ask to see.

Read that list with the eyes of a programme manager. Operating manual, decision log, project board, audit trail. None of this is new. It is what you already build for any serious change programme. **It just has not been called a *wrapper* before.**

The ADR: the corporate AI tool nobody is talking about

An **Architectural Decision Record** is a one-page document that captures a single decision, its context, its consequences, and the alternatives considered. Append-only. Numbered. Versioned alongside the code.

Three reasons it matters more than it looks for AI work.

1. **It stops the agent re-litigating decisions.** Without an ADR, every new session starts from zero and the agent re-derives, usually badly, choices you settled weeks ago. With one, it reads the record and complies.
2. **It encodes constraints the model cannot infer.** Cost caps. Data-residency rules. Regulatory boundaries. Threat models. Things the model has no way of knowing unless you write them down.
3. **It is the audit trail your regulators will ask for.** When the question lands, "*why did your AI make this choice?*", the ADR is the answer.

A real example, paraphrased from a project I shipped this quarter: *which model should the system call by default?* That looks like an engineering question. It is not. Different user populations have opposite risk profiles. First-time founders need responsiveness; corporate users in regulated industries need a hard local-only guarantee. Two defaults, two threat models, two consent flows, one decision log entry that ties the technical choice to the business choice.

Coding the routing took an afternoon. *Deciding* the routing took the work a senior business analyst is paid to do. The ADR is what made the decision survive the next session, the next reviewer, and the next regulator.

Why this is BA/PM work, not just code engineering work

Three independent teams, OpenAI, Anthropic, ThoughtWorks, converged on the same five principles for making AI systems reliable. Read them as an engineer and they sound technical. Read them as a programme manager and they are exactly what your discipline already does.

- **Context beats instructions:** scoping and discovery.
- **Plan and execute separately:** stage gates.
- **Feedback loops are non-negotiable:** governance and assurance.
- **One thing at a time:** sprint discipline and milestones.
- **The codebase is the documentation:** single source of truth.

The work that distinguishes a reliable AI system from an expensive demo is not code. It is the same craft that distinguishes a delivered programme from a stalled one. Decomposition. Decision capture. Definition of Done. Controls. Reviews.

If you have a BA who can write a clear ADR and a PM who can hold a Definition of Done, you already have most of an AI delivery team.

A playbook for corporate teams

Six steps. None of them require buying a new platform.

1. **Pick one workflow with a measurable goal.** Not "improve efficiency". A specific end-to-end objective with a number on it.
 2. **Write the operating manual first.** Project goal, sources of truth, invariants, do-not list. Six pages, not sixty.
 3. **Stand up an ADR folder.** Numbered. Append-only. One entry for every non-obvious decision.
 4. **Define Done.** Tests pass. Audit clean. Diff reviewed by a human. Branch tagged. ADR added if the decision was non-obvious.
 5. **Run the loop.** One feature per session. Read progress, pick the highest priority, implement, verify, commit, update.
 6. **Plan to delete the wrapper.** Every quarter, switch off each scaffold and check whether quality holds. If it does, remove it. The wrapper exists to compensate for what the model cannot yet do.
-

The real competitive advantage

The advantage in 2026 is not having AI. Every company will have AI inside twenty-four months.

The advantage is the discipline of designing the constraints AI works inside.

That discipline already exists in your organisation. It is sitting in the BA team, the PMO, the compliance function, the architects who already write decision records for non-AI systems. The companies that win will move those people to the front of the AI programme, not the back.

Hiring more engineers will not fix a broken AI strategy. Putting your sharpest analyst in the room when the operating manual gets written, and writing one ADR for every decision worth keeping, will.

Key takeaways

- AI Wrappers, not models, are the actual product.
 - A wrapper is a control framework. Your BA, PMO, and compliance teams already build these.
 - The Architectural Decision Record is the connective artefact: one decision, one document, full auditability.
 - AI projects fail the same way IT projects always failed: poor scope, vague outcomes, unclear governance.
 - *Build to delete.* A good wrapper shrinks every time the model improves.
-

[#AIStrategy](#) [#DigitalTransformation](#) [#BusinessAnalysis](#) [#ProjectManagement](#) [#AIGovernance](#)
[#ArchitectureDecisionRecords](#) [#ClaudeCode](#) [#HarnessEngineering](#)

Disclaimer and Disclosure

Third-party Content and AI Assistance: This article references tools and software publicly available and proprietary to their respective creators. The author does not claim ownership or affiliation with these third-party products. This article was written by the author with assistance from Generative AI Language Models.

Transparency Notice: While every effort has been made to ensure accuracy, readers should verify information independently and consult official sources or documentation for the mentioned tools and software. The use of AI in the writing process is disclosed in the interest of transparency, but all opinions and analyses are the author's own unless otherwise stated.