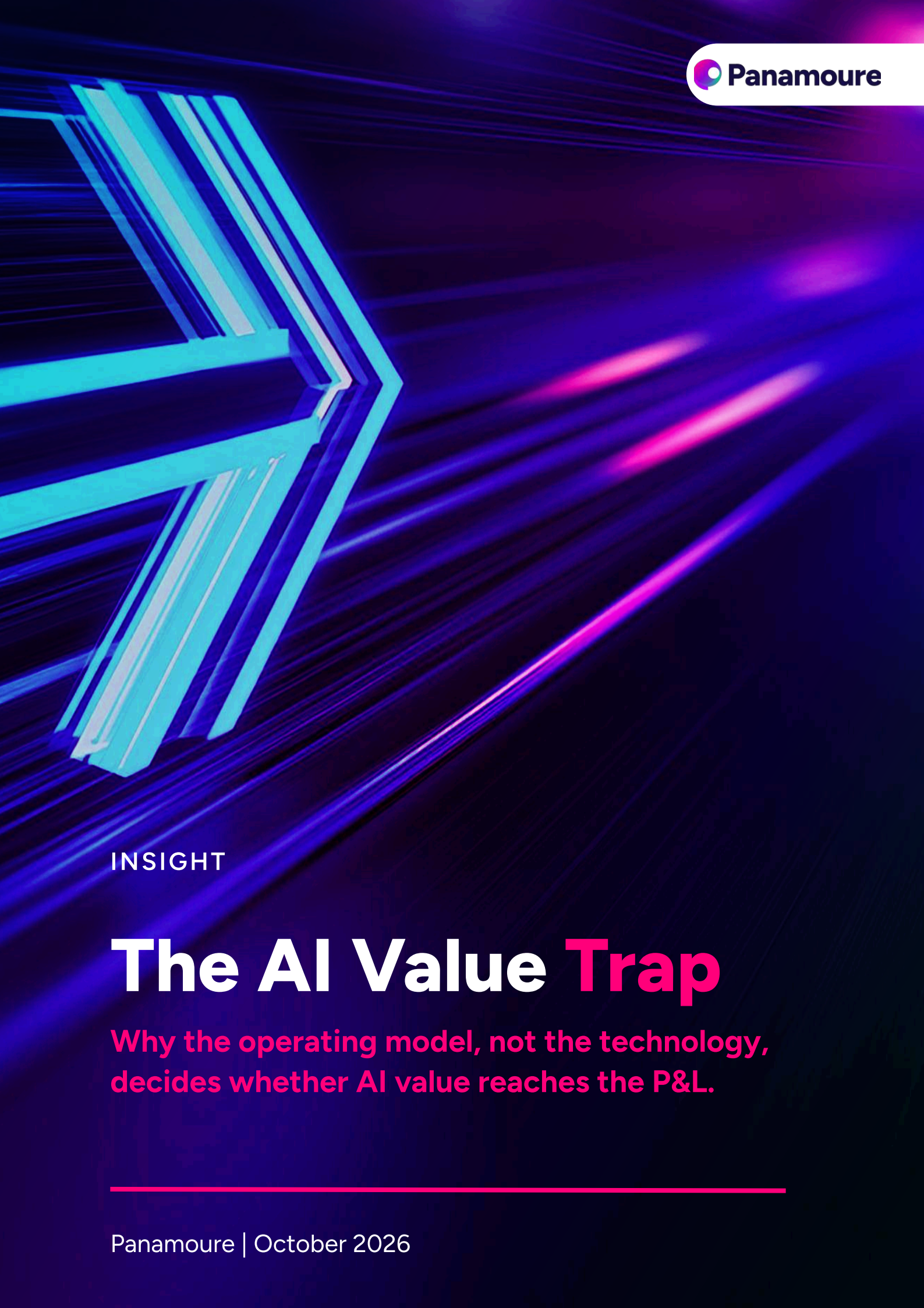


Panamoure

INSIGHT

The AI Value Trap

Why the operating model, not the technology, decides whether AI value reaches the P&L.

Panamoure | October 2026

Introduction

AI has democratised access to the tools needed to deliver change. Anyone in a company can now describe what they want in plain language and have a model write the software (a practice usually called vibe coding). We see this play out all the time: an operator can automate a process over a weekend, or an analyst can build a working tool without needing to join the IT queue.

Most of the market treats this as good news, and in many ways it is. For decades, the limiting factor in digital transformation was your engineering capacity, but that limit has largely gone. However, across the PE-backed and mid-market companies we work with, we keep seeing the same pattern emerge: engineering constraints are lifted, but new challenges emerge elsewhere and value fails to appear on the bottom line.

The tools work, the models do what the vendors said they would, legacy data issues are starting to get resolved and people across the business are building. Yet when we look for the effect on the P&L, it is often hard to find. Democratisation has widened who can create change, but how organisations define ownership, governance, maintenance and adoption of what is built have not always kept up.

On the surface this often manifests as two separate problems. In some companies, a business case was approved, a tool was delivered, but it's unclear who owns it, few people are using it, capacity hasn't been released and efficiency remains stubbornly plateaued. In others, staff have built a lot very quickly, speeding up high-friction cumbersome tasks, but nobody can say what exists, or whether critical AI safeguards have been followed. We think these are two symptoms of one gap: there is no operating model for owning what AI makes easy to build. Tools that nobody owns do not get integrated, trusted or used. Tools approved on a business case do not get absorbed into the process around them. Tools developed outside of policy guardrails introduce new risks to your business. In each instance, the value never reaches the P&L.

TWO CONVERSATIONS MADE THIS CONCRETE FOR US

A Chief Technology Officer at a mid-market engineering firm described a Monday morning meeting. Over the weekend, an engineer had vibe coded a working replacement for part of a core business process, using coding assistants and a few hours of prompting. He told his team he was putting it into production. He had skipped the IT backlog because he was tired of waiting and with it the data sovereignty checks, security review, access permissions, integration standards and testing.

The following day, a private equity partner then asked the question that frames this article. In a traditional transformation, engineering effort was scarce, so gateways, business cases and steering committees decided what got built.

THE CRITICAL QUESTION:

When anyone can build a working capability in an afternoon, what controls the change mechanism?

That question is really about the operating model, not the technology. Three things decide whether democratised building compounds into value or into risk:

1. **Who owns what gets built**
2. **How the end-to-end flow around it is actually redesigned, rather than automated piecemeal**
3. **Whether the people expected to rely it trust it enough to.**

What has changed?

Shadow IT is not new. Excel macros, Access databases and low-code platforms all produced unowned tools that quietly ran parts of the business. Four things are different now: building takes hours, not weeks; the volume is far higher; agents being build can act on live systems; and citizen developers lack the technical skills to understand the code being written and are unable to review, explain or change it.

This presents two fundamental risks which need to be proactively managed:

RISK 01 · AI-ASSISTED DEVELOPMENT RISK

AI-assisted development, including vibe coding, produces ordinary code with risks in security, integration, testing and ownership, plus one more: the accountable person may not be able to read it.

RISK 02 · AI AT RUN TIME

AI at runtime puts a model inside the process, with risks of drift, hallucination, bias and data leaving the perimeter.

The scarce resource has moved from build capacity to ownership capacity.

The impact of these is that the cost of building may have dropped, but cost of running, integrating, securing and owning what was built was not. The scarce resource has moved from build capacity to ownership capacity: knowing who is responsible for each tool, what it touches and what happens when it fails.

Where the value goes

From an investor's perspective, the thesis begins to fail and value erodes in three key areas:

01

Headcount is removed from the plan to match and AI business case, but the operating model around the work stays the same. The savings are absorbed by workarounds, overtime and re-work.

02

Integrations are built outside any standard and break without notice, sometimes damaging the data in core systems such as the ERP. Vibe coded integrations are especially prone to this, because they are written to make the demo work and rarely handle the edge cases a production system meets in the first month.

03

Operational data is sent to external model endpoints with no check on residency or sovereignty, and no one has defined what happens when an agent gets a calculation wrong.

Islands and orphans

Two patterns show up when building runs ahead of ownership.

PATTERN 01 · AGENTIC ISLANDS

Each department builds its own automation. Customer service has an agent, finance has a script, operations has a workflow. None of them connect and the end-to-end process still breaks at the same manual handoffs as before.

PATTERN 02 · AGENTIC ORPHANS

A tool built by an enthusiastic individual who has since changed role. It still runs in the background of a core process, with no named owner, no performance tracking and no one accountable for it. Vibe-coded orphans are harder to deal with than the older kind. The code may be unreadable to the people who inherit it, and the only record of why it works the way it does is a chat history on someone's personal account.

Buyers' diligence teams look for orphans. They test data provenance and ask who owns each risk. When the answer is nobody, the finding is more likely to appear as a price adjustment, an indemnity request or a remediation condition than as a headline change in the multiple. In sectors where a credible AI story supports valuation, an unowned estate also undermines the story, because it shows the capability is not under management.

Panamoure

What to do about it

None of this is an argument against democratisation, or against vibe coding. Sending teams back into a months-long ticket queue would stop the creativity that produces value, and the backlog was what drove the citizen developers to bypass IT in the first place. The aim is to separate creation from control.

01 **Redesign the end-to-end flow before automating individual tasks.** A business that still organises itseld around tasks and job titles cannot make proper use of anything described in this paper, no matter how sound its governance or how good its intentions. Automating a single task inside an obsolete, human-centred process leaves every handoff, approval and exception intact, meaning capacity is never freed and cost savings never hit the P&L.

Start by changing how work itself is understood: as one continuous flow of value from start to finish, drawn from a single pool of human and digital capacity, rather than a list of individual jobs someone happens to do. Map the full flow first to decide where human judgement, software tools or automated agents belong. Building around an optimized flow, rather than an isolated list of tasks, is the best way to avoid creating disconnected islands of automation.

---

02 **Make the controls technical.** A registration policy only works if the people who might bypass it (whether deliberately or not) follow it. Give builders a sanctioned environment for creation, with approved models, approved data connections and single sign-on and put an API gateway and egress controls in front of core systems. A tool that is not registered should be unable to reach core data.

---

03 **Put a review gate between the prototype and production.** Vibe coding is well suited to prototypes and personal tools, but more structured development processes must be followed when something touches core systems and data, other people's work or a customer. At that point, someone who can read the code reviews it, it moves into a proper repository and it gets a named owner. The original builder can transition to being a product or business owner for the solution (and get credit for the idea!) and the company gets software it can maintain.

---

04 **Run a small control desk with clear decision rights.** For a mid-market company this can be a handful of people drawn from IT, security, data and finance, with one named executive owner. Think of it as a hub and spoke model: the desk is the hub, holding the standards and policies, the register of what exists and the decision rights, while the business units are the spokes, building and owning the agentic workforce that matters to them. A hub with no engaged spokes produces governance nobody follows. Spokes with no hub produce exactly the sprawl this article describes. Tier its decisions by what a tool touches: a read-only tool on non-sensitive data should be approved the same day; a tool that writes to core systems or uses sensitive data gets a fuller review, with a target decision within five working days. These figures are starting points to adjust, but a target is essential else a new bottleneck forms and the bypassing starts again.

05 **Set clear guardrails around human accountability and judgement.** Every automated process needs an explicit boundary between where the machine executes and where a named person is accountable.

06 **Design for trust, not just for control.** A tool nobody trusts does not get used, however well it is governed, and an unused tool never reaches the P&L. People need to know, in plain terms, what has changed about their role, what has not, and what they remain personally accountable for. Give them a genuine way to flag or overrule the system when it is wrong, and measure adoption through how many people actually choose to keep using a tool, not how many were given access to it. Confidence has to be designed in from the start. It cannot be patched on after launch once usage numbers disappoint.

07 **Fix data foundations where agents will operate.** Fixing all data first is unrealistic on a three- to five-year hold. Prioritise remediation based on the value of initiatives in your backlog, those domains where agents will act in the next twelve months, and settle access rights and quality standards there.

08 **Track benefits against a baseline.** Record the cost, throughput and quality of the process before the tool is built. These should be signed off by finance and tracked against realised savings, so the number in the business case can be checked against the number in the accounts. Include the running costs that sprawl creates, such as duplicate licences and uncontrolled model usage.

09 **Start with an inventory.** Before building any of this, run a short diagnostic to identify every AI-built or AI-enabled tool across your estate. This should include anything vibe coded, with who owns it, what data it touches and what it connects to and will show whether a company has an island and orphan problem or an adoption problem. It's likely that both will exist, and definitely worth finding them before the diligence team finds the gap themselves.

Conclusion

The portfolio companies that do well over the hold period will be the ones whose people keep building and whose operating model can absorb what they build. Giving everyone the ability to create change is the easy part, but to secure value, you need to lock in who owns each results, what is it is allowed to touch and how savings and benefits can be measured and confirmed.

That operating model rests on three things above all: who owns what gets built, how the flow around it has actually been redesigned and whether the people expected to rely on it trust it enough to. Get those three right and democratised building compounds into value. Get them wrong and it compounds into risk, usually quietly, until a diligence team or a disappointing set of numbers finds it for you.

Get in touch

Richard Gott
richard.gott@panamoure.com
07881 612572