

BEYOND AUTONOMY

Autonomy Is Not the Scarce Resource: Why Enterprise Agents Fail at the Decision, Not the Task

Reflections on agentic workflows in the enterprise, and the accountability gap that the race for autonomy is quietly widening

Giovanni Leonardi

November 2024



Autonomy Is Not the Scarce Resource: Why Enterprise Agents Fail at the Decision, Not the Task

Transformation · No. 397

Originally written November 2024 | Re-edited and re-rendered for the WhereBeyond Edition, 2025

“The dangerous metric is not how often the agent is right; it is how long a wrong action survives unseen.”

1. When the demo grew up

Almost every organisation I have watched wrestle with this over the past year has had the same meeting. Someone wires up an agent — a language model handed a set of tools and a loop in which to use them — and demonstrates it live. It reads the ticket, queries the order system, checks the refund policy, drafts the reply, and resolves the case end to end, unattended, in eleven seconds. The room goes quiet for a moment and then slightly giddy. A year ago this was a research curiosity, something hobbyists chained together for fun; now it is running on a laptop in a conference room, and it works.

Then the pilot goes into production, and three months later someone in finance is trying to work out why a run of account credits went out that should never have been approved.

This is the shape of 2024 as I have observed it: the demo is genuinely astonishing, the deployment is genuinely messy, and the gap between the two is not a gap in capability. The models are good enough. The tools connect. The loop runs. What breaks is the thing the demo never tested, because a demo is a single happy path watched by its proudest engineer, and production is ten thousand paths watched by no one.

The lesson of the year, stated plainly, is this: autonomy is not the scarce resource. Accountability is. And almost everything being written about how to build enterprise agents has the priority backwards.

2. What the textbooks prescribe

The prevailing recipe is now familiar enough to recite. Give the agent a goal. Give it a set of tools — search, a database, an API, the ability to call other agents. Let it plan: break the goal into steps, take a step, observe the result, decide the next step, and continue until it judges the goal met. What sophistication the discussion has tends to concern how to make this loop more capable — longer horizons, more tools, more autonomy, fewer human interruptions. The implied trajectory is a ladder, and every rung climbs towards

the agent needing us less.

For a research demonstration this is exactly the right frame. The interesting question there is how much can it do on its own, and the answer keeps getting more impressive. But an enterprise is not trying to answer that question. An enterprise already has enormous capability sitting idle — people who could do more if they were freed from toil. What it chronically lacks is a clean answer to a different question: when this goes wrong, who is answerable, and how fast will we know?

The textbook optimises the thing enterprises are not short of and ignores the thing they are. That is why so many agentic pilots demo brilliantly and deploy badly. They were engineered to maximise autonomy in an environment whose binding constraint was accountability.

The demo asks how much can it do without us? The enterprise should be asking how much of what it does can we not take back? Those are different questions, and only the second one has money attached to it.

3. The unit of design is the decision, not the agent

Here is the reframing I have watched separate the deployments that held from the ones that quietly unravelled. The unit you are designing is not the agent. It is the decision.

Every action an agent takes sits somewhere on two axes that have nothing to do with how clever the model is. The first is reversibility: if this action is wrong, how hard is it to undo? Redrafting an email is free to reverse; issuing a refund is expensive; deleting a customer's records or sending a message to ten thousand people cannot be reversed at all. The second is blast radius: if this action is wrong, how far does the damage travel — one ticket, one account, or the whole ledger?

Autonomy is safe, and enormously valuable, in the corner where actions are reversible and contained. It is dangerous exactly where the textbook is most eager to deploy it: on consequential, hard-to-reverse decisions, taken quickly, at scale, with no one watching. Speed, which is the entire selling point, is precisely what makes an irreversible mistake worse, because the cost of an error is not fixed. It grows with how long the error runs before anyone notices — and autonomy shortens the time to act while lengthening the time to detect. An agent that resolves a case silently has removed the very human who would have raised an eyebrow.

Consider a composite that will be recognisable to anyone who ran one of these pilots. A customer-operations function deploys an agent to triage and resolve billing queries. On triage and drafting it is a triumph: it handles the reading, the lookup, and the first draft for the large majority of tickets, tirelessly and around the clock. Encouraged, the team lets it also action adjustments below a modest threshold without review — a small credit here, a fee waiver there — reasoning that the amounts are individually trivial. Within a quarter the agent is resolving perhaps seventy per cent of eligible tickets end to end. It is also wrong on something like four per cent of the adjustments it makes — not wildly wrong, but confidently and plausibly wrong, in ways that survive a casual glance. Because each

error is small and no human ever sees it, the mistakes are not caught at the point of action. They surface weeks later, in aggregate, in a reconciliation that now costs more to untangle than the whole exercise saved. The seventy per cent was real. So was the four per cent. The failure was not the model's accuracy; it was letting an irreversible, unwitnessed action ride on it.

Where the action sits	What autonomy delivers	What it costs when wrong
Reversible, contained (draft a reply, summarise a case, propose an action)	Pure leverage — speed and scale with almost no downside	Trivial; the human simply discards it
Irreversible or wide (issue funds, change a contract, delete data, message at scale)	The same speed — now applied to a mistake you cannot take back	Grows with time-to-detection; often exceeds the savings that justified it

Read that table and the design rule falls out of it on its own: ration autonomy against reversibility, not against the model's confidence. The agent's own certainty is the worst possible gate, because a fluent model is most persuasive exactly when it is wrong.

4. The objection worth taking seriously

The strongest version of the counter-argument is not naïve, and it deserves to be met head on rather than caricatured. It runs like this: the entire point of an agent is to scale past the human bottleneck. Every checkpoint you insert — every route this back to a person before acting — reintroduces the constraint you built the agent to remove. Do it enough and you have an expensive language model filling in a form for a human to rubber-stamp, which is worse than either extreme. And besides, capability is improving so fast that today's careful guardrails will look like a handbrake within a year; the right posture is to lean into autonomy and let the models grow into it.

There is real truth in the first half and a real error in the second.

The truth is that indiscriminate checkpoints are a genuine failure mode. A human asked to approve every one of a thousand trivial, reversible actions will not scrutinise them; they will click through, and you will have paid for oversight and received rubber-stamping. But this is an argument for the reversibility frame, not against it. You do not put a checkpoint on every step. You put it where reversibility changes — at the small number of moments where a contained, undoable process is about to take a wide or irreversible action. Those moments are rare by design, which is exactly why a human can afford to look at them properly. The bottleneck worth removing is toil. Judgement on irreversible decisions was never toil; it was the job.

The error is the belief that rising capability dissolves the problem. It does the opposite. A more capable agent is more autonomous, faster, and more convincing — so when it is wrong, it is wrong more fluently, acts more quickly, and is harder to catch.

“Capability improves the average outcome and worsens the tail, and it is the tail that ends up in the reconciliation, the regulator’s letter, or the newspaper.”

Accountability is not a temporary scaffold to be removed once the models are good enough. It is the load-bearing wall.

5. Designing the boundary

If the decision is the unit, then the real craft of an enterprise agent is not the planning loop. It is the boundary — the deliberate seam where the agent hands control back. In the deployments that held up this year, that boundary was engineered rather than assumed, and a few disciplines recurred.

- Draw the line at reversibility, not at task type. Let the agent run freely wherever its actions can be discarded or undone, and require a human hand precisely where an action crosses into the irreversible or the wide. The map of those crossings is worth more than any prompt.
- Make uncertainty legible, and make it act. An agent that surfaces why it did something, and how sure it is, lets a reviewer spend attention where it is warranted. An agent that only ever surfaces a confident answer trains its reviewers to stop looking.
- Give every autonomous action a named owner and a route back. Not “the system” — a person or role who is answerable when it is wrong and can reverse it. If no one can be named, the action should not be autonomous. This is also, not by coincidence, the question the year’s new regulation on automated decisions has begun to ask out loud.
- Instrument time-to-detection, not only accuracy. The dangerous metric is not how often the agent is right; it is how long a wrong action survives unseen. A pilot that measures only its success rate is measuring the seventy per cent and ignoring the four.

None of this throttles the leverage. It concentrates the leverage where it is free and fences it where it is not. The organisations getting durable value from agents this year are not the ones that made their agents the most autonomous. They are the ones that were clearest about where autonomy stops.

6. The question worth putting to the board

The board conversation I have watched repeat itself this year eventually reaches the same question: how autonomous can we make it? It is the wrong question, and it is the one that produces the pilots that demo well and deploy badly. The better question — the one that actually predicts whether an agentic programme creates value or quietly destroys it — is narrower and less thrilling: for each thing this agent can do on its own, who is answerable

when it is wrong, and how quickly would we know?

An organisation that can answer that, action by action, can safely give its agents a great deal of freedom, because it has put the freedom where freedom is cheap. An organisation that cannot answer it is not being bold by pressing ahead. It is mistaking the absence of a visible failure for the absence of risk — and in a system built to act quietly, quickly, and at scale, that is the most expensive mistake of all.

The agents are real, and the leverage is real. But the ambition was never meant to be autonomy for its own sake. It was meant to be judgement, multiplied — and judgement, multiplied, still has to belong to someone.