WHITE PAPER · PORTFOLIO & AI GOVERNANCE

Every Agent Needs a Human Operating Model

Why an AI agent belongs in production only after a named human can answer for what it decides, escalates, and leaves behind

Marco Policani

Enterprise Portfolio · PMO · AI Operating Governance

policani.net · linkedin.com/in/marcpolicani

July 2026

Executive premise

An AI agent is not a tool you deploy, and it is not a hire you onboard. It is something the organization has little practice managing: a system that acts like labor but is funded, procured, and monitored like technology. That mismatch is where governance breaks. A business can approve an agent, assign it a license, point it at a workflow, and still have no answer for who owns its work when it makes a judgment, escalates poorly, or produces an output that looks finished and is wrong.

The claim of this paper is narrow and testable. An agent belongs in production only after it has an operating model: a role and scope, decision rights, an escalation and stop mechanism, a challenge protocol, acceptance criteria, an evidence and logging duty, and one accountable human owner. Without those seven elements, the organization has not deployed capability. It has deployed unmanaged work.

The market is not short on agent enthusiasm. McKinsey reported in 2025 that 88 percent of respondents were using AI in at least one business function, while only 39 percent reported enterprise-level EBIT impact.[1] The gap is widest around agents. Gartner projects that more than 40 percent of agentic AI projects will be canceled by the end of 2027, citing escalating costs, unclear business value, and inadequate risk controls.[2] Those are not model-quality problems. They are operating-model problems: agents introduced into work that was never redesigned to hold them accountable.

This is a governance argument, not a caution against agents. The distinction matters because the two get confused. Slowing an agent program down to satisfy a committee is not governance; it is friction with no owner. Governance is the opposite: it is what lets an organization move fast and still answer for the result. The organizations that will capture the value are not the ones with the most agents running. They are the ones whose humans can still answer for what the agents did. The unit that makes that possible is the operating model, and it can be defined precisely enough to put on one page.

Why agents break the governance model you already have

Traditional operating models were engineered for control. They assume stable processes, clear handoffs, and predictable outcomes, with a human placed at each decision juncture to make the consequential call. Agentic systems do not behave that way. They make autonomous decisions, adapt as they run, and produce results that cannot always be anticipated in advance. Bolting an autonomous agent onto a model designed for human workers, as Deloitte puts it, is like fitting a jet engine to a bicycle.[3]

The deeper problem is categorical. Agents are neither capital nor labor. They act like workers but are funded like technology, and that creates governance gaps precisely where accountability is supposed

to live: decision rights, risk and liability, quality assurance, and performance accountability.[3] An organization knows how to govern a software purchase and how to govern an employee. It has almost no settled practice for a thing that is procured like the first and behaves like the second.

The evidence that this gap is unmanaged is direct. Deloitte's State of AI in the Enterprise 2026 survey found that 84 percent of companies had not redesigned jobs to fit AI, even though automation expectations were high.[3] Redesign is exactly the work that produces decision rights, escalation paths, and ownership. Skipping it does not make the questions disappear; it defers them to the moment an agent does something no one authorized and no one can explain.

Layering agents onto broken processes does not fix those processes. It amplifies them. An agent given an ambiguous workflow will resolve the ambiguity faster and more confidently than a person would, and it will do so at a scale that makes the error harder to catch. The instinct to move quickly is not wrong. The failure is treating deployment as the finish line when it is the point at which the governance question becomes real.

The vendor is not the owner

Because agents are bought like technology, there is a standing temptation to treat accountability as something that came with the purchase. It did not. A vendor can warrant that a model performs to a specification; it cannot own the consequences of a decision the agent makes inside your workflow, against your data, on behalf of your customer. Accountability is not a feature of the product. It is a property of the operating model you build around the product, and it cannot be outsourced to the party that sold you the capability.

This is why the accountable owner has to be an internal, named person rather than a contract clause. When an agent misclassifies a case, releases a wrong figure, or escalates something it should have handled, the question a regulator, a customer, or an executive asks is not which vendor supplied the model. It is who in this organization was answerable for letting the agent act. If the honest answer is a procurement line item, the agent did not have an operating model. It had a license.

The practical test is simple and worth applying before any agent goes live: name the person who would stand up in a review and account for the agent's decisions. If that name does not exist, or if it is really a group that collectively means no one, the deployment is premature regardless of how capable the model is or how confident the vendor is.

An operating model, defined

An agent has an operating model when a named human can answer six questions before it goes live: what it may decide on its own, what it must escalate, who owns its output, what evidence it must leave

behind, how it is stopped, and what a good result looks like. If any answer is missing, the agent is not ready for production, however capable the underlying model is.

This is deliberately a work-design test, not a technical one. Model accuracy, latency, and cost matter, but they are not what fails in the field. What fails is the absence of an owner when a judgment goes wrong, the absence of an escalation path when the agent hits a case it should not decide, and the absence of an evidence trail when someone asks why a decision was made. Those are governance objects, and they can be specified on a single canvas.

The Agent Operating-Model Canvas

The canvas has seven elements. None of them are novel to anyone who has run a program or a portfolio; the discipline is applying them to a non-human actor before it starts working, rather than after it has caused a problem.

1. Role and scope

State what the agent is for and, just as important, what it is not for. Scope is the first control because it bounds every other one. An agent scoped to draft a document and flag exceptions is a different governance object from one scoped to send the document to a customer. Scope creep is the most common way an agent ends up making decisions no one designed it to make; naming the boundary in writing is what makes creep visible before it becomes an incident.

2. Decision rights

Decide, in advance, what the agent may decide on its own, what requires human sign-off, and what must be escalated. Deloitte frames the same three-way split and adds the crucial rider: set expectations for the evidence the agent must produce to support each decision.[3] Decision rights are where most agent programs are silent, and silence defaults to the widest possible authority. A written rule that says an agent may reclassify but not approve, or recommend but not release, converts an implicit risk into an explicit control.

3. Escalation and stop mechanism

An agent needs a path for the cases it should not handle and a way to be halted when something is wrong. Deloitte's guidance is concrete: build stop mechanisms, clarify who resolves exceptions, and decide where friction is valuable to keep errors from propagating before a human notices.[3] Speed is the point of an agent, but unbounded speed is also how a single bad assumption becomes a thousand bad outputs. The stop mechanism is the difference between an incident and a catastrophe, and it has to exist before it is needed, not be improvised during the failure.

4. Challenge protocol

An agent that only confirms is more dangerous than one that occasionally objects, because confident wrong answers are the expensive ones. In mature agentic systems this becomes structural: McKinsey describes critic agents that challenge outputs, guardrail agents that enforce policy, and compliance agents that monitor regulation, with every action logged and explainable in real time.[4] Most organizations are not there yet, and do not need to be. What they need first is a rule that some outputs get challenged, whether by a second agent, a checklist, or a human reviewer, rather than accepted because they arrived quickly and looked complete. The protocol is the standing answer to a simple question: before this output is trusted, what had to disagree with it and be overruled?

5. Acceptance criteria

Define what a good result looks like before the agent produces one. Acceptance criteria are the difference between reviewing an agent's work and merely receiving it. Without them, review collapses into a plausibility check, which is exactly the check a fluent model is best at defeating. Criteria should be specific to the workflow: the fields that must be correct, the sources that must be cited, the thresholds that must not be crossed, and the conditions under which the output is rejected rather than edited. Written criteria also make review delegable, because a second person can apply them without having to reconstruct the reviewer's private judgment.

6. Evidence and logging duty

An agent should leave a trail that lets a human reconstruct what it did and why. Deloitte's list is a workable minimum: log and review agent activity, audit behaviors, document rationales and human interactions, and track disagreements.[3] This is also where public standards help. The NIST AI Risk Management Framework organizes AI oversight around four functions, govern, map, measure, and manage, that together create a cycle for understanding context, assessing performance and risk, and deciding how a system should be used over time.[5] Its Generative AI Profile goes further, treating risk as a lifecycle concern across design, deployment, operation, and decommissioning, and calling for continuous monitoring, structured feedback, and incident response.[6] Evidence duty is what turns those frameworks from documents into daily practice.

7. Accountable human owner

Every agent needs one named person who owns its outcomes, not a committee and not the vendor. McKinsey's framing is that humans move above the loop, defining policies, monitoring outliers, and adjusting the level of involvement rather than reviewing every action line by line.[4] Above the loop is not absent from the loop. The owner is the person who answers for the agent's decisions, who can invoke the stop mechanism, and who is accountable for the evidence trail being real. If no name can be attached to an agent, the agent does not have an operating model, and it should not be in production.

Reading the gap

Most agent failures announce themselves as a symptom in the work long before anyone calls them a governance problem. The discipline is to read the symptom back to the missing element of the operating model, because each gap routes to a different fix. The table below pairs the missing element with how the gap tends to show up and what it costs when it is left open.

Missing element	How the gap shows up in the work	What it costs
Role and scope	The agent is quietly doing tasks no one assigned it	Unowned decisions accumulate outside any review
Decision rights	No one can say whether the agent was allowed to make the call	Authority defaults to the widest reading; risk is invisible
Escalation and stop	A bad output is discovered only after it has propagated	A single wrong assumption scales into many wrong actions
Challenge protocol	Confident answers are accepted because they look complete	Fluent errors pass review; quality erodes silently
Acceptance criteria	Review becomes a plausibility check, not a real one	The organization cannot tell good output from convincing output
Evidence and logging	No one can reconstruct why a decision was made	Failures cannot be diagnosed, audited, or learned from
Accountable owner	When something breaks, the answer is that the system did it	Accountability evaporates; the same failure repeats

The right column is the argument in miniature. Every entry is a governance cost, not a technology cost. None of them is fixed by a better model. All of them are fixed by deciding, in advance, the seven things a human owner must be able to answer.

Oversight capacity is the real ceiling

The canvas raises an operating question that leaders underestimate: how many agents can one human actually own? The honest answer is that oversight is finite, and it is becoming the binding constraint. McKinsey's own observation from early adopters is that a human team of two to five people can

already supervise an agent factory of 50 to 100 specialized agents running an end-to-end process.[4] That is a striking ratio, and it is also a warning: the same report concludes that the scale of agentic adoption will be capped by how much oversight capacity humans can provide, making governance itself a potential bottleneck to productivity.[4]

This reframes the scaling problem. The instinct is to treat model capability as the limit and to add agents until the models stop improving. The operating reality is that human oversight capacity runs out first. An organization that adds agents without adding, or redesigning, the oversight to hold them will not scale its output; it will scale its exposure. This is an inference from the evidence rather than a measured law, and it should be treated as one, but the direction is consistent across the consulting and standards literature: capacity to supervise, not capability to generate, is what caps safe scale.

There is a partial escape, and it is already visible. Oversight can itself be scaled with technology, what Deloitte describes as agents guarding other agents that are then guarded by humans.[3] Critic, guardrail, and compliance agents extend the reach of a small human team, and the NIST Generative AI Profile supplies the operating actions that make such oversight credible: continuous monitoring, red-teaming, independent evaluation, and incident and feedback loops used in deployment and decommissioning decisions.[6] But layered oversight does not remove the human ceiling; it raises it. The final accountability still rests with a named person, and the practical planning number is the span of supervision, how many agents a human owner can genuinely answer for, which belongs in the portfolio conversation next to cost and value.

What it looks like in practice

The safest illustrations are operating patterns, not claims of broad enterprise AI ownership. In portfolio and PMO work, the recurring shape is the same: an agent is most useful when it is attached to a defined recurring workflow with a named human owner and an evidence trail, and most dangerous when it is handed a workflow no one has redesigned.

In one pattern drawn from portfolio governance, an agent prepares decision evidence, decomposing plan data, surfacing inconsistencies, and assembling the status that a leadership cadence would otherwise spend its time reconstructing. The operating model matters more than the capability. The agent's scope is preparation, not decision; its decision right is to recommend, not to approve; its acceptance criteria are the fields and dependencies a reviewer must be able to trust; and a named owner signs off before anything reaches the meeting. The value is real, but it is created by the surrounding governance, not by the model alone.

In a second pattern, an agent proposes reclassifications across a large initiative set. Left unsupervised, it would impose a taxonomy no one agreed to. Inside an operating model, it recommends, a product owner confirms or amends, every change is logged, and the disagreements are tracked as signal. The agent widens coverage; the human keeps the judgment. That division of labor, coverage from the agent, judgment from the human, is the pattern that survives contact with real work.

A third pattern is intake triage, where an agent reads incoming requests and routes them to the right queue. It looks low-risk, which is exactly why it is worth governing: a mis-route is quiet, and a systematic mis-route is invisible until a backlog appears somewhere no one is watching. The operating model that makes it safe is unglamorous. The agent routes; a threshold of low-confidence cases is escalated rather than guessed; the routing decisions are logged so a weekly review can catch drift; and one owner is accountable for the routing quality. The pattern is a reminder that the agents most likely to escape governance are the ones that seem too minor to need it.

What this changes for the portfolio

Read together, the canvas and the oversight ceiling change how an agent program should be funded. An agent is not a one-time purchase whose cost ends at the license. It carries a standing oversight cost, the owner's attention, the review time, the evidence handling, that persists for as long as the agent runs. A portfolio that funds agents without funding their oversight is understating the true cost and overstating the available capacity, and it will discover the gap at the worst time, when too many agents are live for too few owners to answer for.

The governance move is to make the operating model a funding gate, not an afterthought. Before an agent is approved, the canvas is complete: it has an owner, decision rights, a stop mechanism, acceptance criteria, and a budgeted share of someone's oversight capacity. Agents that cannot clear that gate are not rejected on principle; they are sent back to have their operating model designed, which is usually where the real work was hiding. This keeps the portfolio honest about what it can actually supervise, and it keeps the count of live agents tied to the count of humans who can answer for them.

Evidence gaps this paper cannot close

Two things in this argument are not settled by the current evidence, and it is more useful to name them than to paper over them. First, the span of supervision is not a known constant. McKinsey's 50 to 100 agents per small team is an early-adopter observation from selected engagements, not a benchmark that transfers to every workflow, risk level, or regulatory context; the safe number in a high-liability process may be far lower.[4] Treat span of supervision as a variable to be measured in your own setting, not a target to be hit.

Second, the return on oversight investment is not yet well quantified. The literature is clear that oversight capacity caps safe scale, but it does not tell a leader precisely how much oversight a given agent portfolio needs to stay safe, or what the marginal failure rate is when that oversight is thin. Until that evidence matures, the defensible posture is conservative: budget oversight as a real cost, measure the span of supervision you can actually sustain, and let the number of live agents follow the oversight you have, rather than the other way around.

The shift

The temptation with agents is to measure progress in deployment: how many are running, how much they touch, how fast they move. That number is easy to grow and easy to mistake for value. The number that predicts whether the value survives is different. It is whether a named human can still answer, for each agent, what it may decide, what it must escalate, who owns its output, what evidence it leaves, how it is stopped, and what good looks like.

An agent that can act without an operating model is not autonomy. It is unmanaged work, moving quickly. Give every agent a role, an owner, decision rights, a challenge protocol, acceptance criteria, an evidence duty, and a way to be stopped, and the organization has deployed capability it can answer for. Skip that, and it has deployed something it will eventually have to explain. Build the operating model first. Then let the agent work.

Notes and sources

1. Alex Singla, Alexander Sukharevsky, Bryce Hall, Lareina Yee, Michael Chui, and Tara Balakrishnan, "The state of AI in 2025: Agents, innovation, and transformation," McKinsey & Company, November 5, 2025. Verified July 10, 2026. https://www.mckinsey.com/capabilities/quantumblack/our-insights/the-state-of-ai

2. Gartner, "Gartner Predicts Over 40% of Agentic AI Projects Will Be Canceled by End of 2027," press release, June 25, 2025. Verified July 10, 2026. https://www.gartner.com/en/newsroom/press-releases/2025-06-25-gartner-predicts-over-40-percent-of-agentic-ai-projects-will-be-canceled-by-end-of-2027

3. David Mallon, Brad Kreit, and Natasha Buckley, "Rethinking operating models for humans with agents," Deloitte Insights, April 2, 2026. Verified July 10, 2026. https://www.deloitte.com/us/en/insights/topics/talent/operating-models-for-humans-ai-agents.html

4. Alexander Sukharevsky, Alexis Krivkovich, Arne Gast, Arsen Storozhev, Dana Maor, Deepak Mahadevan, Lari Hamalainen, and Sandra Durth, "The agentic organization: Contours of the next paradigm for the AI era," McKinsey & Company, September 26, 2025. Verified July 10, 2026. https://www.mckinsey.com/capabilities/people-and-organizational-performance/our-insights/the-agentic-organization-contours-of-the-next-paradigm-for-the-ai-era

5. National Institute of Standards and Technology, "AI Risk Management Framework Core," excerpt from AI RMF 1.0, 2023. Verified July 10, 2026. https://airc.nist.gov/airmf-resources/airmf/5-sec-core/

6. National Institute of Standards and Technology, "Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile," NIST AI 600-1, July 2024. Verified July 10, 2026. https://doi.org/10.6028/NIST.AI.600-1

About the author

Marco Policani is an enterprise portfolio, PMO, and AI operating-governance leader. He builds portfolio governance systems where AI carries the analytical load and named humans own every decision — an approach documented across case studies, walkthroughs, and working governance guides on his portfolio site.

Portfolio & governance library: policani.net/governance · Full portfolio: policani.net · LinkedIn: linkedin.com/in/marcpolicani