A POINT OF VIEW

Jobs Don’t Get Automated. Tasks Do.

Why role-level thinking is producing role-level disappointment in enterprise AI — and the task-level operating model that actually compounds value.

Smit Jani, CAITL · AI Governance & Data Strategy Consultant · June 2026

The argument in five lines

- **Most enterprise AI programs use the wrong unit of analysis.** Jobs are bundles of tasks. The tasks inside a single role differ wildly in complexity, judgment, and error cost. Treating the role as the unit produces over-investment in the wrong work and under-investment in the highest-leverage work.
- **Most programs measure the wrong thing.** “Tasks per hour” is the wrong scoreboard for knowledge work. The scoreboard that matters is time-to-competency, decision quality, capacity freed for irreducible work, and error rate on what already exists.
- **One-size-fits-all AI is malpractice.** Plot tasks on a 2×2 of complexity vs. operator experience. Each quadrant demands a different intervention — service-layer assistant, deterministic guided workflow, strategic co-pilot, or plain process automation. Putting AI where it does not belong is more expensive than putting nothing there.
- **Autonomy must be calibrated to error cost.** Automate → Augment → Anchor. As error cost rises and knowledge becomes more tacit, AI moves from independent agent to assistant to retrieval tool. This decision belongs at the task level, not the program level.
- **Build foundations, not point solutions.** A first deployment treated as reusable infrastructure compounds across every subsequent team. A first deployment treated as a project does not. The firms pulling ahead are choosing infrastructure.

1. The wrong question, and the costly answer

Most executive conversations about AI in operations start with the wrong question: which roles can we automate? The framing is a category error. Roles are bundles of tasks, and the tasks inside a single role differ wildly in complexity, judgment, regulatory weight, and tacit knowledge.

When the question is framed as “what roles can AI replace,” the answer is necessarily binary and the resulting investment is necessarily wrong. Programs over-invest in trying to automate functions that contain irreducible human judgment, and under-invest in augmenting the very work where AI would deliver the largest gains: the high-stakes, high-judgment tasks where senior practitioners spend their best hours.

I have watched this pattern play out across several operations functions in regulated industries. Programs are framed at the role level, ROI is modeled at the role level, and the result is a portfolio of point solutions that touch the wrong work, deliver the wrong metrics, and frustrate the practitioners they were meant to serve. The narrative inside the firm becomes "AI did not work for us," when what actually did not work was the unit of analysis.

There is a better unit of analysis. There is a better scoreboard. There is a better operating model. This piece sets out what I believe each of them looks like, based on direct implementation work — not on a press release.

2. Decompose to the task

The first move is the boring one, and it is the one almost every program skips: break every role into its constituent tasks.

A senior practitioner in a complex operations function typically performs somewhere between twenty and forty discrete tasks in a normal week. Some are routine, rule-based, and high-volume — running the same lookup, applying the same validation, generating the same report. Some are deeply tacit — interpreting an ambiguous policy in a novel situation, judging whether an exception is a one-off or a pattern, deciding when to escalate to leadership. Some demand cross-functional coordination, regulatory awareness, or pattern recognition that no procedure document fully captures.

Treat each of these as a separate object of analysis. For each task, ask three questions:

1. What is the cost of an error — financial, regulatory, reputational?
2. What kind of knowledge does the task require — explicit and documented, or tacit and lived?
3. Does the task require human accountability for audit, regulatory, or relationship reasons — independent of who or what could technically perform it?

The answers cluster tasks into three intervention buckets:

- **Automate** — routine, rule-based, low-judgment, low-error-cost work that AI can perform end-to-end with periodic audit.
- **Augment** — work where humans must remain in control of the outcome, but where AI can draft, validate, surface precedent, or accelerate the underlying analysis.
- **Preserve** — high-judgment, relationship-driven, audit-critical work where AI may assist with retrieval but the human owns the decision end-to-end.

This taxonomy is older than the current AI cycle. What is new is that the boundary between Automate and Augment has moved significantly in the last eighteen months, and the boundary between Augment and Preserve is moving every quarter. If a firm has not redrawn these lines in the last six months, the investment plan is already stale.

3. A better scoreboard

"Productivity" has been the wrong scoreboard from the start. The word implies a denominator — hours worked, tasks completed, tickets closed — and a numerator that grows by squeezing more output from the same time. Applied naively to knowledge work, this framing optimizes for the wrong thing and quietly degrades the work that matters most.

The right scoreboard for AI in complex operations measures four things:

- **Time-to-competency for new hires**. How long until a new joiner is productive at the median level for their role. This is the most under-discussed source of value: shortening onboarding from many months to a few weeks by giving new hires a guided workflow rather than a binder full of procedure documents.
- **Decision quality on the highest-stakes work**. Stronger gap analyses, fewer overlooked precedents, cleaner audit trails. Faster is a byproduct; better is the point. The hard part is establishing a defensible baseline for "better," and most programs skip this.
- **Senior capacity freed for irreducible work**. Measured not as abstract hours saved but as hours redeployed to strategy, governance, tacit knowledge transfer, and the work only a senior practitioner can do.
- **Error rate on existing work**, particularly on exception handling — where the cost of a missed pattern is high and the cost of an unnecessary escalation is non-trivial.

If a program cannot show movement on at least two of these against a documented baseline, it is not working, regardless of what the throughput dashboard says. A throughput dashboard that has nothing to say about decision quality or onboarding speed is a vanity instrument.

> *Throughput is the easiest thing to measure and the least important thing to optimize. Capability growth is the hardest thing to measure and the most important thing to compound.*

4. The matrix that tells you what to build

Once tasks are decomposed, score each on two axes.

The first axis is task complexity — from rule-based and single-step, through structured with documented exceptions, to multi-source analytical judgment requiring domain expertise. A simple five-point rubric, applied honestly, is sufficient.

The second axis is operator experience — from a new joiner with limited product knowledge, through a competent mid-tenure practitioner, to a veteran with deep tacit pattern recognition and audit sign-off authority. Three levels are usually enough.

Plot every task on the 2×2 of complexity against experience. Four quadrants emerge. Each demands a different intervention, and the difference between them is the difference between an AI program that lands and an AI program that gets quietly shelved.

Quadrant A — Low complexity, low experience (Accelerated Ramp-Up)

A new hire doing routine work. The bottleneck is not the work; it is the time it takes to learn what to do, who to ask, and where to look. The right intervention is a service-layer assistant that answers procedural questions, looks up data definitions, routes requests, validates required fields before submission, and replaces the binder.

The win in this quadrant is collapsing time-to-competency from months to weeks. Practitioners should experience AI as a colleague who knows the procedure manual cold and never sighs when asked the same question for the third time. The risk is low; the gain is one of the largest the program will deliver.

Quadrant B — High complexity, low experience (the Equalizer)

A junior practitioner handling work that would normally require a senior. This is the highest-leverage quadrant in most operations functions. It is where attrition hits hardest, where errors cluster, where overnight pressure on a small senior bench is most damaging, and where the productivity gap between top and median performers is widest.

The right intervention here is a deterministic, guided workflow — pre-checks, validation rules, step-by-step decision support, contextual procedure lookups — with mandatory human review on the output and a maker-checker audit trail captured by design. The win is enabling a junior practitioner to safely handle complex work that previously required escalation. This is where AI becomes a knowledge equalizer rather than a productivity gimmick, and it is where its credibility with the operations function is earned.

Quadrant C — High complexity, high experience (the Amplifier)

The most senior practitioners doing the work only they can do: designing controls, leading audits, navigating regulators, building cross-functional consensus, drafting governance positions. Do not put AI between this person and the work. Put AI behind them.

The right intervention is a strategic co-pilot that drafts first-pass gap analyses, runs pre-mortems on proposed approaches, synthesizes cross-product data into a coherent narrative, and prepares executive materials. The expert stays in control; the AI accelerates their thinking and frees them from the parts of the work that taxed without teaching them anything new. The audit trail still belongs to the human.

Quadrant D — Low complexity, high experience (the Caution Zone)

A senior practitioner spending their week on rule-based work is a system failure that AI will quietly mask if allowed to. The mistake here is to deploy a generative AI co-pilot on top of work that should not require a senior in the first place.

The right intervention is process automation: APIs, deterministic workflows, plain RPA, and the redeployment of freed senior capacity into Quadrant C. The objective is not to make the senior faster at routine work; the objective is to stop spending the senior on routine work at all.

> *Putting AI where it does not belong is more expensive than putting nothing there. One-size-fits-all deployment is the most common, and most damaging, mistake.*

5. Calibrating autonomy to error cost

The intervention type tells you what to build. The autonomy ladder tells you how much of the decision the AI is permitted to own.

The ladder has three rungs:

- **Automate** — the AI acts independently. Appropriate only where the cost of an error is low, the knowledge is explicit and rule-based, and detection of errors is cheap. Periodic audit sampling is usually sufficient as the control.
- **Augment** — the AI drafts, the human approves. Appropriate where error cost is moderate, knowledge is a mix of explicit and contextual, and the human can meaningfully validate the output. This is the right default for most complex operations work today.
- **Anchor** — the human decides, the AI retrieves. Appropriate where error cost is high, knowledge is tacit, and there is regulatory or audit expectation of human ownership. The AI is permitted to surface historical patterns, flag anomalies, and supply context, and nothing more.

Move down the ladder as error cost rises and knowledge becomes more tacit. The architectural mistake is to choose an autonomy level once, at the program level, and apply it across every task. The right approach is to choose per task and to write the autonomy decision into a control layer that the firm can audit. Three short diagnostic questions are usually sufficient: What happens if the system is wrong? Can the answer be found in approved sources, or does it require judgment? Does this work need maker-checker evidence or audit sign-off?

6. A first deployment is a foundation, not a project

The economics of enterprise AI are dominated by reuse. Every program I have seen succeed has treated its first deployment as an infrastructure investment, not as a point solution.

Concretely, that means treating the following as reusable assets, not as deliverables of a single team:

- the service-layer assistant pattern for procedural Q&A and routing;
- the deterministic intake pattern that pre-checks and validates every request before it reaches a human;

- the maker-checker evidence-capture pattern that produces an audit trail by design, not as an afterthought;
- the challenger and follow-up pattern that converts ad-hoc reminders into tracked control obligations;
- the task-decomposition framework itself, applied evenly across every team in scope.

A program that builds five foundational components delivers compounding value: every subsequent team adopts the pattern at a fraction of the original cost, with the controls and audit affordances already attached. A program that builds five point solutions delivers diminishing value: every subsequent team requires its own bespoke build, its own governance review, and its own integration roadmap.

Point solutions feel faster in the first quarter. Foundations feel faster in every quarter after that. Choose accordingly.

7. The pre-mortem beats the business case

The standard playbook is to model ROI before launch, scale on the strength of the model, and learn whether the model was right somewhere around month eighteen. The risk profile of generative AI does not tolerate this approach.

A better discipline is the pre-mortem: imagine the program has failed twelve months from now and work backward to identify why. The same six failure modes recur:

- **Adoption stalls** — the tool does not fit the workflow, training was thin, end users were not co-designers. Prevent by embedding the tool in the channels practitioners already use, publishing role-specific prompt packs, and treating the first cohort as co-builders rather than subjects.
- **Trust erodes** — the AI gives a confidently wrong answer to a confidently asked question. Prevent by grounding answers in approved sources, setting confidence thresholds that escalate to a human below threshold, and making source citation a hard requirement of every response.
- **Leadership sees no ROI** — KPIs were never established at baseline. Prevent by measuring the right four metrics from week one against a documented baseline, and reporting against them monthly even when the news is unflattering.
- **Governance blocks the launch** — compliance was engaged at month nine instead of week one. Prevent by treating governance as a design input, not a deployment gate.
- **Scope creep turns the pilot into the platform** — success in one team creates pressure to expand before the pattern is stable. Prevent by defining explicit phase-exit criteria before starting.
- **Data quality undermines accuracy** — the AI is asked to reason over data with known gaps. Prevent by running a data-quality assessment as part of the pilot, not as a downstream finding six months in.

A pre-mortem written before the pilot is one of the few interventions I have seen materially change the trajectory of an enterprise AI program. It costs a day. It saves quarters.

8. A phased roadmap, not a big-bang launch

A defensible deployment unfolds in four phases. Twelve weeks is enough to prove or disprove the pattern. It is not enough to build everything. That is the point.

- **Weeks 1–2 — Assess.** Run task decomposition workshops with each in-scope team. Pull tenure data. Run a two-week time-tracking exercise. The output is a single master worksheet of every task in scope, scored on complexity and operator experience, with the owning team noted.
- **Weeks 3–4 — Map and prioritize.** Plot every task on the matrix. Choose the highest-leverage pilots — typically one Ramp-Up quadrant and one Equalizer quadrant. Validate scoring with team leads and executive sponsors. Define and instrument baseline KPIs before any tool is built.
- **Weeks 5–8 — Build and pilot.** Deploy with a single team. Capture maker-checker evidence. Track KPIs weekly against baseline. The output is not a launched product; it is a measured one.
- **Weeks 9–12 — Scale and govern.** Expand the pattern to adjacent teams. Embed the foundational components into the firm's automation governance. Publish a self-serve prompt pack for each role. Document lessons. The output is a replication playbook, not a one-off success story.

The next twelve weeks are about expanding what worked. The twelve after that are about retiring what did not.

9. The quiet implication

The firms that get this right will not look very different from the outside in the next year. They will publish similar press releases, deploy roughly the same vendor tooling, and claim roughly the same wins.

The difference will become visible inside the next three years, when their senior practitioners have spent two years compounding their judgment with appropriately calibrated AI support, while their competitors' senior practitioners have spent two years fighting with chatbots that were never sized to the work. Their new hires will be productive in weeks rather than months. Their exception handling will be faster, more consistent, and more defensible under audit. Their controls will be evidenced by design rather than reconstructed before the audit.

None of this comes from buying the right product. All of it comes from doing the unglamorous work: decomposing the tasks, scoring the complexity, placing each task in the right quadrant, choosing the right intervention, calibrating the right autonomy, building foundations rather than point solutions, running a pre-mortem before each phase, and gating each phase before moving to the next.

> *The unit of analysis is the task. The scoreboard is capability growth, not throughput. The mistake to avoid is the one almost everyone is making.*

The firms that internalize that will own the next decade of operations excellence. The firms that keep asking "which jobs can we automate" will spend that decade explaining why their AI program did not work.

A note on numbers and sources

The ranges and patterns in this piece reflect direct implementation experience and the broader observed practice of practitioners deploying AI in complex operations functions. They are intended as practitioner intuition, not as benchmarks. Any firm using this framework should establish its own baselines before targeting specific numbers; the value is in the method, not in any single statistic. Where this POV refers to industry positions, it does so generically and intentionally avoids attributing specific claims to specific named sources unless those sources are themselves verified and current.