WNS
Part of Capgemini

AGENTOPS

The Missing Operating Layer for Enterprise AI Agents

Archit Agrawal,
Senior Technical Architect

Riya Patel
Senior Business Analyst

The enterprise AI problem is not a model problem.
Nearly two-thirds have not yet begun scaling AI across the enterprise—not because the models are wrong, but because the operating model is absent. This paper argues that Agent Operations (AgentOps) is the missing discipline: the difference between AI agents that impress in demos and agents that enterprises can trust, audit, govern, and continuously improve in production.

1. The Problem Is Not What You Think

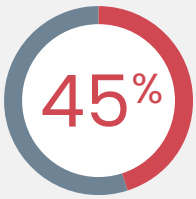
Ask any enterprise AI team where they spend most of their time, and the answer is almost always the same: model selection, prompt design, and benchmark comparisons. These are important. They are not, however, where production deployments fail.

What actually breaks production AI agents is operational: invisible reasoning chains, governance structures that don't survive contact with real data, prompt changes that silently degrade quality in adjacent scenarios, and cost curves that no one saw coming because there was no instrumentation to see them. The model was fine. The operating model was missing.

According to McKinsey, 88% of organizations now use AI in at least one function, yet nearly two-thirds have not begun scaling it across the enterprise.[1] The last five years have produced genuinely capable AI agents. They have produced almost nothing in the way of the discipline needed to run those agents reliably in production. That discipline is AgentOps—and the enterprises that build it first will hold an operational advantage that compounds over time.

of business processes are expected to reach semi- to fully autonomous levels in 12 months.[2]

of respondents report scaling AI agents in any individual function.[3]



of enterprises with high AI maturity keep AI projects operational for at least 3 years.[4]

Three forces are converging to make the operational gap acute right now:

Autonomy has crossed a threshold.
Copilot-style AI that recommends actions is inherently forgiving, with humans catching errors before they propagate. Agents that execute actions are not. A single misconfigured agent operating at scale can corrupt process outputs, violate data policies, or make thousands of wrong decisions before any alert fires. The blast radius of an unobserved agent is qualitatively different from the blast radius of an unobserved recommendation.

Multi-agent systems fail differently.
Enterprise deployments increasingly rely on coordinated agent clusters—planners, retrievers, validators, executors—operating in non-linear pipelines. When one agent in a chain produces a subtly wrong output, the error doesn't surface at the point of generation; it manifests several hops downstream, distorted and compounded. Traditional Application Performance Management (APM) was designed for service-to-service calls. It has no model for agent-to-agent reasoning chains and no mechanism for assigning causality across them.

Regulation has arrived, and it asks operational questions.
The EU AI Act, DORA, FCA guidance, and RBI sector mandates are not asking enterprises to explain their model architecture. They are asking: who made this decision, under which policy version, with what data, and how do you know it was right? These are infrastructure questions. Answering them requires operational systems that most enterprises have not yet built.

[1]The State of AI in 2025 | McKinsey & Company
[2]Rise of Agentic AI: How Trust is the Key to Human-AI Collaboration | Capgemini Research Institute
[3]The State of AI in 2025 | McKinsey & Company
[4]45% of Organizations with High AI Maturity Keep AI Projects Operational for at Least Three Years | Gartner

2. Why Current Tooling Doesn't Solve This

Enterprises are not ignoring the problem. They are assembling point solutions—and discovering that the sum of those solutions leaves the most critical gaps unaddressed.

Tool Category	What It Does Well	What It Cannot Do
APM Platforms (Datadog, Dynatrace, Splunk)	Infrastructure telemetry: latency, error rates, uptime	Semantic understanding of reasoning chains, prompt behavior, or output quality. Surfaces symptoms, not causes.
LLM Monitors (LangSmith, Arize Phoenix)	Prompt/completion capture, basic LangChain tracing	Scoped to single frameworks. No governance, policy enforcement, cost attribution, or cross-agent causality.
Evaluation Frameworks (Ragas, TruLens, DeepEval)	Offline quality benchmarking for RAG pipelines	Developer tools, not production systems. No alerting, no CI/CD gate, no real-time correlation with cost or behavior.
Prompt Tools (PromptLayer, Langfuse)	Prompt versioning, basic A/B testing	Versioning without evaluation gates is configuration management, not operational control. No quality-gated promotion.

The Compounding Problem with Point Solutions

- Enterprises typically operate 4–7 disconnected tools to approximate what integrated AgentOps provides, each with its own data model, access controls, and alert surface. The integration overhead alone consumes engineering capacity that should be spent improving agents.
- The insights that matter most—cost spikes that correlate with specific prompt versions, quality regressions that trace to retrieval pattern changes, compliance risk that maps to model routing decisions—only emerge from correlations across these systems. Siloed tools cannot surface them.
- The result: most enterprises know something went wrong. They lack the integrated context to understand why, prove it is fixed, or prevent the next occurrence.

3. The AgentOps Operating Model

AgentOps is not a product category. It is a discipline—with a specific scope, a specific set of capabilities, and a specific relationship to the agents it governs. The analogy to DevOps is deliberate and precise: just as DevOps did not replace software development but enabled software to be shipped reliably at scale, AgentOps does not replace agent development—it enables agents to run reliably in production.

A mature AgentOps practice unifies six capabilities that current tooling treats as separate concerns:

Capability	What It Actually Means	Capability	What It Actually Means
Trace-level Observability	Not logs—causal execution graphs. Every reasoning step, tool call, retrieval, model input, and output is linked end-to-end, so failures are diagnosable, not just detectable.	**Prompt Engineering & CI/CD**	Prompt registry, version control, A/B testing, and evaluation-gated promotion. No prompt reaches production without passing a quality gate against ground-truth data.
Continuous Evaluation	Scheduled, dataset-driven quality measurement—not one-off benchmarks. Eight dimensions: relevance, accuracy, completeness, reasoning, safety, grounding, context handling, and policy alignment.	**FinOps & Cost Intelligence**	Real-time token, retrieval, and infrastructure costs by agent and workflow. Threshold alerts, anomaly detection, cost-quality correlation, and optimization recommendations.
Governance & Guardrails	Runtime policy enforcement: escalation thresholds, role-based tool permissions, and responsible-AI checks. Every decision produces an auditable artifact before anything executes.	**Continuous Improvement Loops**	A/B testing, drift detection, and refinement recommendations. The platform tells you what to change, not just that something is wrong.

The architecture that delivers these capabilities has a clear structure: AgentOps operates as a control plane above the agent execution layer. Agents execute; AgentOps observes, evaluates, governs, and feeds insights back into how agents are improved. It instruments without constraining—meaning teams building on LangChain, LlamaIndex, Semantic Kernel, AutoGen, or custom orchestrators do not need to rebuild their execution logic.

One design principle matters above all others: the feedback loop must be closed. Observability that doesn't feed evaluation is just logging. Evaluation that doesn't feed prompt improvement is just scoring. Prompt improvement that doesn't go through a CI/CD gate is just guessing. AgentOps value is cumulative—each capability multiplies the others.

> ***The enterprises that win with Agentic AI are not the ones with the best models. They are the ones with the shortest cycle time between 'something degraded' and 'we fixed it and proved it.'***

4. The Hyperautomation Blind Spot

Here is something the analysts largely miss: the largest concentration of enterprise AI agent deployments is not happening on greenfield LLM platforms. It is happening inside UiPath, ServiceNow, and Appian—the hyperautomation platforms that enterprises have spent years and hundreds of millions of dollars standardizing on.

UiPath's Agentic Automation framework, ServiceNow's Now Assist, and Appian's Agent Studio are shipping AI agents into production workflows that organizations did not build as AI workflows. The automation pipelines already exist. The AI reasoning layer is being inserted into them. And the operational infrastructure—the observability, governance, and cost management—is almost entirely absent, because these platforms were not designed with AI agent operations in mind.

> ***The hyperautomation platforms are becoming AI agent platforms, whether they planned to or not. Their observability models were built for deterministic robots, not probabilistic reasoning agents.***

This creates a specific, underappreciated risk profile. A UiPath robot that fails does so visibly—it throws an exception, it stops. A UiPath AI agent that reasons incorrectly does not stop. It produces a plausible-looking output and continues. The same dynamic applies in ServiceNow ticket resolution and Appian document extraction. The failure mode has changed; the detection infrastructure has not.

Platform	AI Agent Capability	Native AI Observability Gap	What AgentOps Adds
UiPath	Autonomous AI Agents that can orchestrate human and digital workers with enterprise-grade governance.	Visibility on LLM usage, audit logging, and full traceability for model calls, and policy-aware and sensitive data observability from the orchestrator.	FinOps alerts on token cost per automated workflow type.
ServiceNow	Now Assist—AI agents for ticket resolution, knowledge generation, and change drafting across ITSM, HR, and CSM.	Platform metrics cover resolution rate and CSAT but not LLM prompt-level traceability, cross-domain quality benchmarking, or cost-per-resolution attribution.	Per-domain Eval-8 scoring; threshold alerts on cost-per-query; governance layer that enforces policy consistency across ITSM, HR, and Finance AI simultaneously.
Appian	Appian's Agent Studio enables goal-driven AI agents that seamlessly integrate into workflows, interacting with humans, systems, and data across diverse use cases. It also provides basic monitoring through process-level metrics and logs.	It has limited cost transparency and weak AI-specific governance, making explainability and control challenging. It lacks deep AI observability, with no visibility into agent reasoning, prompt governance, or continuous quality tracking.	AgentOps provides end-to-end visibility into AI agent execution, along with continuous quality evaluation and structured prompt lifecycle management. It also enables real-time cost optimization and robust governance through audit trails and explainability.

WNS-Vuram's position in this landscape is specific: as a hyperautomation delivery partner with deep implementation history across all three platforms, we have direct visibility into how AI agents are actually deployed within these systems—and where operational gaps are creating risk that platform-native tooling cannot close. AgentOps is not an alternative to UiPath, ServiceNow, or Appian. It is the operational layer that makes enterprise-scale AI deployment on these platforms safe.

5. From the Field: What We Found in Production

The following two deployments are not case studies of what AgentOps promises. They are accounts of what production AI operations actually required—told through the specific signals that only became visible once we built the operational infrastructure around these agents.

CDMS—Card & Dispute Management Banking & Financial Services | 2M+ transactions/month	TrustHall—Contract Lifecycle Management Legal & Procurement | Multi-Jurisdiction Contract Review
The Deployment	
A coordinated agent cluster handling end-to-end dispute resolution: transaction retrieval, fraud-pattern analysis, policy interpretation, chargeback recommendation, and regulator explanation—at 2M+ transactions per month across multiple retail banking clients.	AI agents processing clause extraction, obligation mapping, risk scoring, and counterparty comparison across vendor agreements, MSAs, SOWs, and NDAs—spanning multiple jurisdictions, replacing paralegal review at scale.
Observability: What the Traces Revealed	
Before instrumentation: Agents returned outcomes, but the reasoning chain—which fraud signal fired, which policy clause was applied, which model version generated the recommendation—was a black box. Regulators were already asking questions no one could answer. **After instrumentation:** Step-wise causal traces across all five agent hops revealed that 62% of all latency overruns originated at a single step—policy interpretation—where context windows were being flooded with irrelevant transaction history. A failure that looked systemic was a one-step retrieval problem. Additionally, 18% of dispute decisions carried no recoverable rationale in any log, meaning nearly 1 in 5 decisions would have been indefensible to an auditor. This was invisible until traces made it countable.	**Before instrumentation:** Legal teams were shadow-reviewing most AI outputs—a signal that trust in the system was low, but no one knew why. Error reports were vague: 'the extraction was wrong.' The trace layer made errors diagnosable. **After instrumentation:** Execution traces exposed that a prompt change intended to improve NDA extraction had silently degraded MSA performance—a regression that would have been caught in production review only when a client raised it. Evaluation-gated CI/CD caught it before promotion. Obligation coverage for multi-schedule agreements was running at ~74%. Traces showed this was a chunking issue—long documents were being processed as monoliths, causing the model to lose context across schedule boundaries. The failure mode was architectural, not a model limitation.

FinOps: The Cost Picture

Before instrumentation:
Token consumption spiked 3–4× during the festive season and year-end reconciliation. Finance teams discovered this in billing statements. There was no mechanism to distinguish expected volume peaks from operational waste.

After FinOps instrumentation:
Cost attribution by agent step revealed that redundant evidence-summarization calls—triggered by a retry logic flaw—accounted for 31% of total token spend. Threshold-based alerts flagged abnormal cost-per-dispute within hours, not weeks.

Cost-quality correlation showed that 60% of disputes could be resolved with a lighter, cheaper model configuration at equivalent accuracy. Dynamic model routing was introduced: the most complex disputes routed to the full model, the rest to the lighter variant. Net result: 27% cost reduction with no accuracy regression.

Before instrumentation:
Infrastructure costs were climbing faster than contract volume. No one could explain why. Token usage per contract had no baseline, so there was no way to identify waste or justify model spend to leadership.

After FinOps instrumentation:
Cost-per-contract analysis revealed that long-form documents (40–80 pages with multiple schedules) consumed 8–10× the tokens of a standard NDA, not because they were more complex, but because they were processed monolithically regardless of structure.

Contract-type classification was introduced upstream to route document types to structure-aware chunking strategies. Real-time cost alerts now fire if any contract type exceeds its cost-per-unit threshold—catching runaway processing before it compounds. Token spend on long-form contracts dropped 36%.

Governance & Outcome

Mandatory explanation generation was embedded as a guardrail—no chargeback recommendation executes without a structured rationale. 100% of decisions now produce a queryable audit artifact. Decision consistency improved 32%. Resolution turnaround 41% faster.

Prompt CI/CD with evaluation gates blocked the MSA regression before it reached production. Obligation coverage lifted to >94%. Every extraction now includes a prompt-version hash and context audit trail. Legal teams stopped shadow-reviewing—the most meaningful adoption signal.

Metric	CDMS (Dispute Management)	TrustHall (Contract CLM)
Observability	Full causal trace across 5 agent hops; latency root cause identified at the step level.	Chunking failure & silent prompt regression both caught via trace; paralegal shadow-review eliminated.
FinOps	31% of spend from retry flaw discovered; dynamic model routing cut cost 27%.	8–10× cost overrun on long-form docs identified; contract-type routing cut spend 36%.
Governance	Unexplained decisions: 18% → near-zero; 100% auditable artifacts.	Evaluation-gated CI/CD blocked MSA regression; full prompt-version audit trail per contract.
Quality/Speed	+32% decision consistency; 41% faster resolution.	+38% extraction accuracy; obligation coverage 74% → >94%.

In both deployments, the first 60 days of AgentOps instrumentation surfaced cost anomalies, quality regressions, and governance gaps that had been accumulating undetected for months. This is not unusual; it is the baseline state of most production AI agent deployments that have not been operationally instrumented.

6. The WNS-Vuram AgentOps Platform

The platform is the operational infrastructure we built to deliver the AgentOps discipline described above—refined through real deployments rather than designed in the abstract. It is framework-agnostic, integrates natively with UiPath, ServiceNow, and Appian, and operates as a control plane that instruments agents without constraining how they are built.

Capability	What It Delivers in Practice
Trace-level Observability	Causal execution graph across every agent, tool call, retrieval, model input/output, and retry—with latency distribution and error diagnostics. Not logs. Diagnosable execution history.
Eval-8 Framework	Eight-dimension continuous evaluation—relevance, accuracy, completeness, reasoning quality, safety, grounding, context handling, and policy alignment—run on schedules and on every deployment event.
Prompt Registry & CI/CD	Central prompt registry with version history, author attribution, and deployment status. Every change triggers an Eval-8 run. Promotion to production requires passing defined quality thresholds. One-click rollback with automatic incident artifact.
Governance & Guardrails	Runtime policy enforcement, role-based tool permissions, responsible-AI checks, and mandatory audit artifact generation per interaction. Approval workflows for high-risk agent changes. Immutable change audit trail.
FinOps Intelligence	Real-time cost dashboards by agent, workflow, and business unit. Threshold-based alerts are routed to Slack, email, or ITSM. Anomaly detection that distinguishes peak-period spikes from runaway loops. Cost-quality correlation. Optimization recommendations with estimated savings.
Hyperautomation Connectors	Pre-built integrations for UiPath, ServiceNow, and Appian—adding semantic observability and governance to AI agent activity within existing automation workflows without requiring architectural changes.
Prompt Best-Practice Library	Domain-indexed library of tested prompt patterns across financial services, legal, insurance, and operations—with performance benchmarks from production deployments. Reduces time-to-quality for new agent rollouts.

> ***AgentOps is not an end state. It is an operating rhythm—a closed loop of observe, evaluate, govern, optimize, and repeat. The platform automates that rhythm so teams spend their time on decisions, not on keeping the lights on.***

7. What Leaders Should Do Now

The organizations that capture durable value from Agentic AI will not be those that deployed first. They will be those who built the operational infrastructure to keep agents performing, compliant, and cost-efficient as scale and complexity grow.

The window to do this without painful retrofitting is closing. Every agent deployed into production without observability is technical debt accumulating invisibly. Every prompt change made without an evaluation gate is a regression waiting to happen. Every FinOps conversation that happens at the billing cycle, rather than in real time, is money already spent.

Horizon	Timeframe	The Priority Action
Immediate	0–30 days	Audit your three highest-impact production agents for observability coverage, prompt version discipline, and governance documentation. Assume the gaps are larger than you expect. They always are.
Near-term	30–90 days	Instrument those agents with trace-level telemetry and establish Eval-8 baselines. Introduce evaluation-gated promotion for any prompt or model change. Surface cost attribution by agent and workflow—the numbers will surprise you.
Medium-term	90–180 days	Embed governance guardrails and audit artifact generation into production execution paths. If you operate on UiPath, ServiceNow, or Appian, assess AI agent activity within those platforms specifically; it is almost certainly the least observed AI in your environment.
Ongoing	Continuous	Run AgentOps as an operating rhythm, not a project. Scheduled evaluations, FinOps reviews, prompt library maintenance, and CI/CD discipline compound over time. The organizations that do this consistently will not just run better agents; they will build institutional knowledge about how to improve them that becomes a genuine competitive asset.

The AgentOps Maturity Curve

Most enterprises sit at Level 1 or 2. The gap between Level 2 and Level 4 is not a technology gap; it is a discipline gap.

L1 Reactive

Agents in production with no systematic observability. Issues discovered by users or at the end of the billing cycle.

L2 Monitored

Basic telemetry in place. Teams can detect failures but cannot diagnose root causes or predict degradation.

L3 Governed

Evaluation baselines, prompt versioning, and policy guardrails are in place. Changes go through defined approval paths.

L4 Optimized

Closed feedback loop: telemetry feeds evaluation, evaluation feeds prompt CI/CD, and cost intelligence drives model routing. Continuous improvement is automated.

The question is no longer whether to deploy AI agents.

It is whether you can operate them well enough to trust them.

Contact WNS-Vuram to schedule an AgentOps maturity assessment for your highest-impact agent deployments.

About WNS

WNS, part of Capgemini, is an Agentic AI-powered intelligent operations and transformation company. We combine deep domain expertise with talent, technology, and AI to co-create innovative solutions for over 700 clients across various industries. WNS delivers an entire spectrum of solutions, including industry-specific offerings, customer experience services, finance and accounting, human resources, procurement, and research and analytics to re-imagine the digital future of businesses. WNS has 66,000+ professionals across 65 delivery centers worldwide, including facilities in Canada, China, Costa Rica, India, Malaysia, the Philippines, Poland, Romania, South Africa, Sri Lanka, Turkey, the United Kingdom, and the United States.

To know more, write to us at **marketing@wns.com** or visit us at **www.wns.com**

WNS
Part of Capgemini