Automation Is the Backbone. AI Handles Ambiguity.

Where agentic AI creates real value on the Atlassian platform, what it costs when that boundary is drawn in the wrong place, and how to measure whether an agent is actually working before Rovonomics tells you the hard way.

Artem Fomin · Founder & CEO, Veraxent

Veraxent

SEPTEMBER 2026

ATLASSIAN | Gold Solution Partner

EXECUTIVE SUMMARY

Three questions, one architecture problem

Most organizations evaluating agentic AI are asking the wrong question. "How do we make this process agentic?" assumes more autonomy is the goal. It isn't. The goal is better business outcomes, which means deciding, process by process, where deterministic automation should carry the work and where AI should take over because the step genuinely requires judgment.

That principle holds regardless of platform. This paper starts with the tool-agnostic architecture question, then applies it to Atlassian Rovo, where new usage-based pricing makes the consequences of that decision unusually visible.

The cost of getting that boundary wrong used to be easier to hide. Starting December 3, 2026, when Atlassian's usage-based billing for Rovo credits and Automation steps goes live, that mistake can consume metered resources on one or both sides of the workflow, even before either one produces an overage bill. An agent invoked against high volume and low signal, or a rule that fires more often than the job requires, burns through finite included allowances on outcomes that never materialize, and no governance conversation catches that unless someone is actually measuring it.

This paper lays out the architecture principle in Part One, works through the concrete cost of ignoring it in Part Two, and gives engineering leaders a practical framework in Part Three for measuring whether an agentic workflow on Jira, Confluence, or Rovo is creating value, or just producing activity that looks like progress.

IN THIS PAPER

Part One · The architecture principle: where automation should give way to AI, and why most processes are a mix of both.

Part Two · The economics: what Atlassian's December 3 usage-based billing makes visible about agent placement, in Rovo credits and Automation steps.

Part Three · The measurement framework: ten working principles for telling whether an agent is creating value, plus three metrics to start with.

PART ONE · THE ARCHITECTURE QUESTION

Automation answers "what should happen next." AI answers "what does this mean."

Walk into most organizations evaluating agentic AI right now and you'll hear a version of the same question: how do we make this process agentic? It's an understandable instinct. Agents promise more autonomy, less manual work, and less effort spent encoding every business rule up front.

But it's the wrong question, and answering it well doesn't fix that. The goal was never to make every workflow agentic. The goal is better business outcomes, and those are not the same project.

> **Automation answers "what should happen next." AI answers "what does this mean."**

That's a real difference, not a rhetorical one. When the next step in a process is knowable in advance, meaning it can be expressed as state plus rules plus permissions, deterministic automation is the better answer. It's simpler to build, cheaper to run, easier to test, and easier to govern, because you can prove what it will do before it does it. When a step requires interpreting something genuinely ambiguous, unstructured text, conflicting signals, a judgment call that depends on context a rule can't fully capture, that's where AI creates real value, and where it should complement the workflow rather than replace it.

Most real processes aren't purely one or the other. A support ticket usually needs interpretation at the front (what is this actually about, where does it belong) and determinism everywhere else (routing, approvals, execution, notification). The interpretation step is where AI earns its place. The rest stays deterministic, not because AI can't do it, but because deterministic automation does it more cheaply, more predictably, and with a smaller blast radius when something goes wrong.

That's the actual architectural challenge: not maximizing AI, but placing it precisely where ambiguity or judgment creates enough value to justify the variance an agent introduces. Get the placement right and automation stays the backbone of the system while AI handles the parts that genuinely need it. Get it wrong, and you've traded a system you could reason about for one you can't, without actually buying better outcomes in return.

None of this is static. The boundary between what's deterministic and what needs judgment shifts as a process matures. A pattern that starts as an ambiguous call an agent has to interpret can, over time, become well enough understood to encode as an explicit rule. Some organizations eventually retire the agent from that step entirely; others leave it handling a narrowing tail of genuine exceptions. Either way, the direction of travel matters more than the starting point. Architecture that keeps deterministic logic doing everything it can do, and reserves agentic reasoning for what's left, gets cheaper and more reliable as it matures. Architecture that defaults to agent-first usually moves the other way.

This is where the general architecture principle becomes operational on Atlassian. Cheaper is not the same as free: Rovo credits and Automation steps now make the consequences of agent placement measurable in finite usage and, eventually, spend, on both sides of that architectural boundary, not just the AI side. Part Two works through that economics.

PART TWO · THE ECONOMICS

The architecture question just acquired a price tag

On September 1, 2026, Atlassian made official what had been an open question since Rovo's rollout: usage-based billing for Rovo credits and Automation steps goes live December 3. Organizations now have a firm deadline to work out what they had been able to treat as a governance question rather than a financial one.

Two mechanics matter here, and neither was fully confirmed until the announcement.

Two separate meters, not one

The automation flow wrapped around an AI action, the trigger, the conditions, the branches, the actions that fire before and after, draws from the `Automation steps` meter. The AI action itself draws from `Rovo credits`. A noisy agentic workflow can draw down both finite allowances at once, not just one.

Automation has economics too

Deterministic automation is still cheaper than agentic reasoning, but it is no longer economically invisible. Every trigger, condition, action, branch, and loop that executes counts as one Automation step, including a trigger that fires and finds nothing to do. Starting December 3, usage beyond the included allowance is billed at **$0.50 per 1,000 additional steps**. A rule that runs 100,000 times a month doesn't become efficient just because it never calls an agent; if each run evaluates a trigger and a couple of conditions, that's several hundred thousand steps drawn from the same allowance your higher-value automations depend on.

The gap between the two meters is real. Automation steps are priced well below Rovo credits at the margin, but a step and a credit measure different work, so the honest comparison stops at "automation is cheaper," not a specific multiple. The architectural goal isn't automation instead of AI at any cost. It's using the least expensive, most deterministic mechanism that can actually do the job, at every point in the workflow, not just the point where an agent gets invoked. That cuts both ways: a filter that adds ten automation steps to avoid one agent call isn't automatically a win either. What matters is the total cost of the path, not which meter it managed to avoid.

Noise costs you before it ever becomes a bill

The allowance mechanic has an answer worth understanding before December 3 rather than after. Noise doesn't turn into a bill the moment it happens. It draws down your included Rovo allowance first. Only once that pool is exhausted does the excess become incremental spend, billed in arrears at **$10 per 1,000 credits**, with extra usage enabled by default. So even an organization that never crosses into paid overage is still spending a finite monthly resource on invocations that produce nothing. The bill is the visible failure mode. The invisible one, using up capacity your higher-value use cases need, happens first, and happens regardless of whether you ever pay for it.

SAME CAPABILITY, DIFFERENT ECONOMICS

High volume, low signal: run an agent against every one of 10,000 incoming comments looking for the rare ones that need escalation. All 10,000 fire the trigger and its conditions, drawing on the Automation steps allowance whether or not anything happens next. If only 50 produce a meaningful outcome, the other 9,950 also invoke the agent, drawing on Rovo credits for outcomes that never materialize. The agent isn't broken. It may be working exactly as designed. A trigger that fires on volume instead of on signal produces poor economics on both meters, no matter how well the model performs once invoked.

Constrained trigger, defined job: invoke an agent when an issue is created, where every invocation has a clear task, classify it, enrich it, route it, flag what's missing, start a workflow. The automation flow still costs something, every trigger and condition is a step, but it's a smaller, more predictable draw, paired with a Rovo invocation that has real work to do. Tighter trigger, higher odds each invocation produces something useful, meaningfully better economics on both meters for the same underlying capability.

None of this makes Rovo a bad bet. It's genuinely capable, and Atlassian is shipping real usage visibility and admin controls ahead of the change. The point isn't caution for its own sake. It's that agent audit, going forward, is an economics exercise as much as a governance one. For every agent running in production, you need an answer to four questions: what triggers it and how often; what percentage of those invocations produce a meaningful outcome; could deterministic logic filter the noise before probabilistic reasoning gets involved; and what the useful outcome actually costs once everything the agent touched is counted.

Those aren't new questions. They're the placement question from Part One, asked with a dollar sign attached. That is Rovonomics: the economics of what a workflow consumes relative to the useful outcomes it produces. Rovonomics didn't create this problem. It made the cost of getting the placement wrong visible on a monthly statement.

PART THREE · THE MEASUREMENT FRAMEWORK

You're not measuring a system. You're measuring a decision process under uncertainty.

Knowing where AI belongs, and what it costs when it's misplaced, still leaves the harder question: how do you tell whether a well-placed agent is actually working? Most teams don't have a good answer, because most teams are measuring the wrong layer.

What follows is a practical framework for closing that gap, built from years of watching agentic AI initiatives get evaluated well and evaluated poorly. It's meant to be applied directly, one process at a time, not completed once as an audit checklist.

1. Start with the right question

Most teams track activity. Few measure impact. Before choosing any metric, force the actual question: what are we trying to improve? Not what the agent does, but what changes in the business if it works.

2. Think in layers

Metrics cluster in three layers, and most measurement programs stop at the first one.

LAYER 1 · WEAK SIGNAL

Activity

Usage, volume, latency. Tells you the system is running, not that it's helping.

LAYER 2 · BETTER

Quality

Accuracy, error rate, how often a human overrides the output. Tells you it's roughly right, not that being right matters.

LAYER 3 · REAL VALUE

Outcome

Time saved, incidents avoided, measurable business or customer impact. Most teams never get here.

3 **Define the counterfactual**

Value only exists relative to an alternative. Before crediting an agent with an outcome, ask what would have happened without it. Are you accelerating something that would have happened anyway, or actually replacing work, risk, or delay that would otherwise have occurred? A security system that flags a leak is only valuable if that leak would have caused real damage without it. No counterfactual, no real ROI, just activity dressed up as an outcome.

4 **Not all value is direct**

Some of the most important value an agent creates is indirect: risk avoided, trust preserved, a problem that never became visible because it never happened. That's harder to quantify but not impossible. Use severity tiers, low, medium, high, instead of forcing false precision, and estimate value as frequency times impact rather than pretending you can price every prevented incident to the dollar.

5 **Tradeoffs are unavoidable**

Every agentic system balances competing forces, and you are always optimizing a tradeoff whether you name it or not. Catch everything and you drown the system in noise. Reduce noise and you accept some risk of missing something real. Medical alert systems tuned to catch everything overwhelm clinicians into ignoring alerts altogether; spam filters tuned to eliminate every false positive let more junk through. No configuration avoids this tradeoff, only some make it consciously instead of by accident.

6 **Watch for the wrong metric**

A bad metric doesn't just fail to help, it actively misleads. It can create false confidence in a system that isn't working, degrade trust in one that is, or shift a problem downstream instead of solving it. Before adopting any metric, ask what it would look like for that metric to improve while the real outcome gets worse. If that scenario is easy to construct, the metric isn't ready to be a target.

7 **Adoption is part of the metric**

A recommendation nobody acts on has zero value no matter how accurate it is. Track whether users accept the agent's output and, just as importantly, how often they override or ignore it. Low adoption is low impact, regardless of what the accuracy numbers say in isolation.

8 **False negatives are the blind spot**

False positives are visible: the system flagged something wrong, someone noticed, the cost is plain. False negatives are usually invisible: the system missed something, and unless it surfaces some other way, nobody finds out. That asymmetry pushes teams toward eliminating misses at any cost, which drives sensitivity up until alert fatigue and workaround behavior erase whatever value the system had left. The goal isn't zero misses. The goal is the right balance between risk reduction and a system people will actually keep using.

9 **A practical starting point**

If you're building this today and don't know where to start, three metrics get you most of the way there. None of them, on their own, catches what the agent missed, so validation also means periodically sampling what it didn't act on and checking for false negatives, not just auditing what it did. Pair every metric with the guardrails that make the risk of being wrong tolerable: scoped access, audit logs, defined escalation paths, and human review on anything high-risk.

PRECISION

How often it's correct on the cases it acts on.

OVERRIDE RATE

Whether users actually trust its output.

TIME TO RESOLUTION

Whether the work is genuinely getting done faster.

10 **You're measuring a decision, not a system**

None of this is really about the tool. You're measuring a decision process operating under uncertainty, and every principle above is a way of asking the same question from a different angle: is this decision process actually better than the one it replaced?

Applied to an agent running on Jira, Confluence, or Rovo, this isn't abstract. Precision is whether the routing, classification, or escalation call the agent made turned out to be right. Override rate is how often a person undoes or ignores what it recommended. Time to resolution is whether the ticket, request, or workflow actually closed faster because the agent touched it. Tracked against the trigger volume from Part Two, those three numbers tell you directly what percentage of invocations produced a meaningful outcome, which is the exact question Rovonomics is now asking you to answer whether you're ready or not.

CONCLUSION

The order of operations

Put the three parts together and the order of operations is simple, even if the execution isn't. Architecture decides where AI belongs in a workflow: automation carries what's deterministic, AI takes the parts that genuinely require judgment. Economics, now that Rovo credits and Automation steps are metered separately starting December 3, makes the resource cost of bad placement visible on both meters, not just the one running the model: first in consumed allowance, then, if it goes far enough, in paid overage. Measurement tells you, process by process, whether an agent that's correctly placed is actually creating outcomes or just generating activity that resembles progress.

Skip the first step and you'll eventually see the consequence, in depleted allowance, an overage bill, or both. Skip the third and you'll never know whether the money was well spent either way.

The objective was never to minimize AI usage in isolation. It's to minimize wasted execution across the whole path: unnecessary Automation steps before and around the decision, and unnecessary Rovo credits where deterministic logic could have resolved it first. Before scaling agentic automation on the Atlassian platform, or auditing what's already running, map what triggers each agent, how often, and what happens on both meters after. That map, more than any tool selection or model choice, is what determines whether the investment pays off.

ABOUT

Veraxent

Veraxent, founded in 2017 as DevOps Action, is an Atlassian Gold Solution Partner and enterprise technology consultancy. The firm works across three connected practices: Atlassian Platform Engineering, AI & Agentic Systems, and Engineering Transformation, built around the same workflow-first discipline reflected in this paper: automation as the backbone, AI applied where judgment genuinely earns its complexity.

WEB

veraxent.com

EMAIL

info@veraxent.com

PHONE

844-VERAXENT

SOURCES

Pricing and billing mechanics for Rovo credits and Automation steps: Atlassian, "Expanded usage-based pricing," Inside Atlassian blog, atlassian.com/blog/company-news/expanded-usage-based-pricing; Atlassian Support, "How Rovo credits work," support.atlassian.com/rovo/docs/rovo-usage-limits; Atlassian Support, "How is your automation usage calculated," support.atlassian.com/cloud-automation/docs/how-is-my-usage-calculated. Verified current as of September 2026; Atlassian may revise terms before the December 3, 2026 effective date.

Veraxent

ATLASSIAN | Gold Solution Partner