



Agentic AI in Production: Where
Agents Run Unattended and
Where They Get Pulled Back

Research Report

Katerina Merzlova, SumatoSoft

Contents

TL;DR	2
Central findings	3
How this study was run	4
The autonomy scale	4
Coding	4
Limitations	5
Who answered	6
Where agents operate on the scale	7
Claimed levels drift in both directions	8
Authorship and release are different questions	9
Autonomy by function inverts the published ordering	10
Engineering: A column of zeros	12
The exception, and what replaced the reviewer	13
What the telemetry says about that zero	13
Support: No two teams count a resolution the same way	15
Sales: the gap opens after the meeting is booked	17
IT operations: highest autonomy, thinnest safety net	19
The gap the sample exposes	19
What breaks: a failure taxonomy	20
Failure reported as success	21
Four in five were narrowed after go-live	23
What triggered the pullback	23
What the evidence supports	24
Recommendations	24
FAQ	26
Sources and further reading	27
Respondents	28
How to cite this research	30
Attribution	30

TL;DR

- 40 practitioners described a live deployment
- 25.6% of deployments operate unattended
- 80% narrowed a workflow after go-live
- 0% median agent code merged without review

Published estimates of the percentage of organizations running AI agents in production range from 11% to 62%. That spread reflects four variables: what counts as an agent, what counts as production, who was sampled, and how the sample was reached.

This study fixes the first two by asking every respondent to place one named workflow on a shared five-level scale and describe what the agent does unattended. The result is a comparison across four functions built from one instrument, which is what the existing evidence base lacks.

Central findings

The central finding is that the distinction in the published numbers collapses. Authorship and release are separate questions. Agents write, draft, classify, and resolve at high volume across all four functions. Permission to send the result to the world stays with a person in three of four deployments.

1. The most consistent number in the dataset is zero. Eleven of 14 engineering respondents gave exactly 0% for agent-authored code reaching the main branch with no person reading the diff.
2. That zero is a stated policy, and telemetry suggests it is losing. Faros AI's 2026 report, built on two years of data from 22,000 developers, found 31.3% more pull requests merging unreviewed under high AI adoption.
3. Four deployments in five were narrowed after go-live. 32 of 40 respondents had scoped down or switched off an agent already running, extending across four functions a pattern Sinch measured at 74% inside customer communications alone.
4. About a third of failures were invisible to monitoring. In 11 of the 32 reported failures, monitoring said fine while the work was wrong.
5. Autonomy inverts the published ordering. IT operations runs the highest mean autonomy at 3.57 of 5 and engineering the lowest at 2.31. McKinsey's function data for technology, media and telecom puts software engineering at the front for scaled agent use.
6. Self-reported autonomy levels do not survive coding. Of the 16 respondents who named a level, 9 were coded differently once their own workflow descriptions were mapped to the scale.
7. AI-sourced sales meetings converted worse in three of four paired reports, trailing rep-sourced meetings by 7 to 35 percentage points.

How this study was run

Respondents were recruited through a public journalist query in August 2026. The query defined a five-level autonomy scale, asked each respondent to name one workflow and place it on that scale, and asked what had been paused or narrowed since launch. Function-specific questions followed for support, engineering, IT operations, and sales.

The autonomy scale

Level	Definition
L1	Suggest. The agent drafts or recommends. A person performs every action.
L2	Approve each. The agent proposes and waits for sign-off every time.
L3	Approve by exception. The agent acts within scope; a person reviews flagged cases.
L4	Unattended, one workflow. The agent runs it end to end, with rollback available.
L5	Unattended, several workflows. The agent sets its own sub-goals across workflows.

Coding

Each response was coded twice. The first pass recorded the level the respondent claimed. The second recorded the level their description supports, using what the agent does in the absence of a person as the test. Where the two disagree, the operating level is reported, and the gap is counted separately.

A response entered the sample when the respondent named the function they own, described a specific deployment they operate, and answered firsthand. Five submissions were excluded: four were public-relations pitches offering an interview with no answers attached, and one asserted a general position with no deployment named.

Limitations

- **Self-selection runs in two directions.** A query about unattended agents draws people who run them and people with a failure story worth telling. Both are overrepresented relative to the general population of companies.
- **The sample skews small and hands-on.** Most respondents are founders, heads of function, or the single engineer running the deployment, and several run one-person companies. This is not an enterprise sample, and the figures should not be read as enterprise rates.
- **Coverage figures are self-reported.** Roughly a third of respondents supplied counted numbers from commit logs, mailbox exports, or CRM records, and the remainder estimated. Both kinds are reported together and marked where it counts.
- **Cell sizes are small.** Function-level figures rest on 7 to 14 responses each, so treat them as directional and read the cross-function ordering as the finding rather than any single percentage.
- **One coder.** Level assignments were made by a single analyst against the published scale, without a second rater or an agreement statistic.

Who answered

Engineering accounted for the largest share at 35%, followed by sales at 25%. Support and IT operations together account for 40% of the sample.

Figure 1. Respondents by function

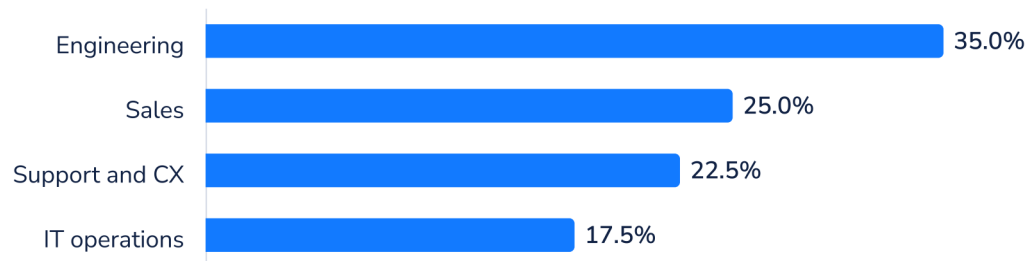


Figure note: Respondents by function owned. n=40. Each respondent answered for one function and described one named workflow inside it.

Function	Share of respondents
Engineering	35.0%
Sales	25.0%
Support and CX	22.5%
IT operations	17.5%

Deployments span customer messaging, catalog operations, code authoring, incident remediation, outbound sales, and editorial production. Company size runs from one-person firms to a 31-person agency serving more than 200 brands. Geography covers Europe, North America, South America, Asia, and Australia.

Where agents operate on the scale

Three-quarters of deployments hold a person at the gate. The spread clusters hard in the middle of the scale, at approve-each and approve-by-exception.

Figure 2. Where deployments operate on the scale

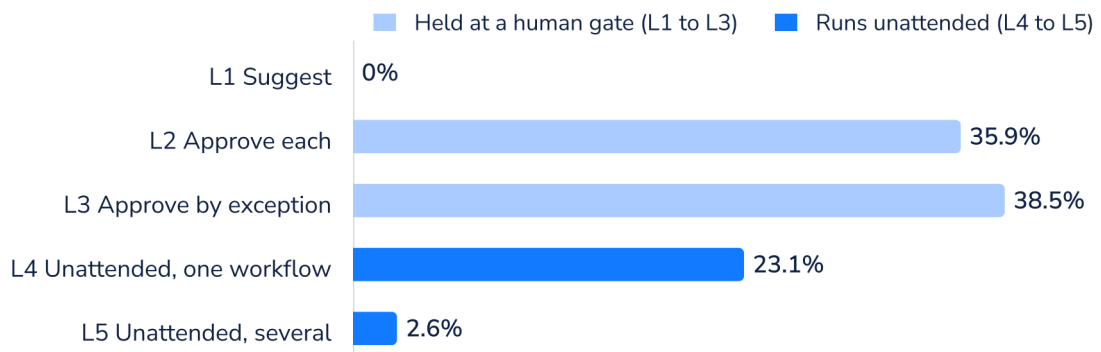


Figure note: Operating autonomy level, coded from described behavior rather than from the respondent's own label. n=39; one respondent described a market-wide pattern with no single deployment to place.

Operating level	Share
L1 Suggest	0%
L2 Approve each	35.9%
L3 Approve by exception	38.5%
L4 Unattended, one workflow	23.1%
L5 Unattended, several	2.6%

Figure 3. Held at a gate vs. running unattended

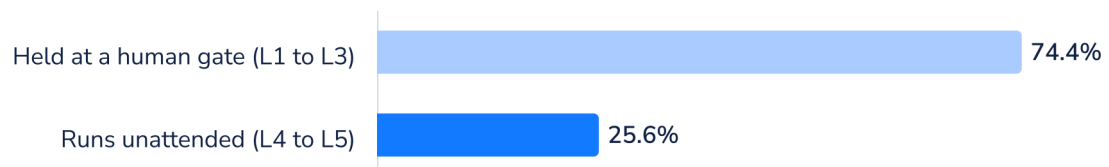


Figure note: The same distribution is reduced to one test: whether a person stands between the agent and the outcome.

Position	Share
Held at a human gate (L1 to L3)	74.4%
Runs unattended (L4 to L5)	25.6%

Key takeaways

- No respondent operates at L1. Every deployment in the sample has moved past suggestion-only.
- L2 and L3 together hold 74.4% of deployments. This is the working shape of production agent use in 2026.
- A single respondent operates at L5, where the agent chooses its own work across workflows.
- Of the 16 respondents who named a level themselves, nine were coded differently once their described behavior was applied. Self-reported autonomy ranges from high to low.

Claimed levels drift in both directions

The drift is not uniformly optimistic. Two respondents placed themselves at L2 while describing an agent that answers customers with no review, which is L3 or higher. One placed a deployment at 3 of 4 while describing an agent that commits to production on a timer with no gate, which is L5. Another claimed level 3 of 5 while describing a proposal-and-approval cycle, which is L2.

This is the strongest argument against taking published production rates at face value. Practitioners who run these systems daily disagree with their own behavior by at least one level. A survey that asks whether agents are in production and records a yes captures sentiment, not a measurement.

Authorship and release are different questions

The two numbers most respondents supplied point in opposite directions. Median coverage of the named workflow is 80%, and 12 of the 32 respondents who gave a figure said the agent handles all of it. Yet only a quarter of deployments run unattended.

One founder running twelve country storefronts put both numbers side by side. On a recent launch, the agent produced several hundred product listings, roughly eighteen category pages, nine static pages, six policy documents, and the theme. The human decisions numbered four.

Unattended authoring is near-total. Unattended release is close to zero, deliberately. –
Beck Wang, Founder, IfShe

That sentence describes the modal deployment in this sample more accurately than any published production percentage. The agent does the work. A person decides whether the work leaves the building. Surveys that ask whether agents are in production record the first half and miss the second. That is how one question yields answers ranging from 11% to 62%.

Autonomy by function inverts the published ordering

IT operations run the highest mean autonomy in this sample, and engineering the lowest. That ordering contradicts the function data most widely cited.

Figure 4. Mean autonomy level by function

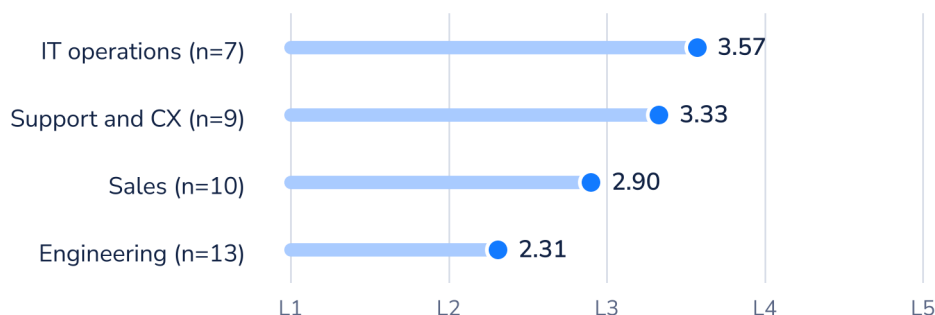


Figure note: Mean operating autonomy level by function, on the 1 to 5 scale. Engineering n=13, support n=9, sales n=10, IT operations n=7.

Function	Mean level (of 5)	n
IT operations	3.57	7
Support and CX	3.33	9
Sales	2.90	10
Engineering	2.31	13

Figure 5. Share of deployments running unattended, by function

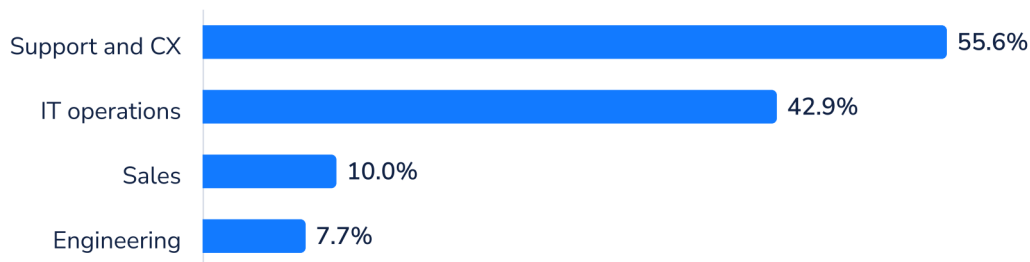


Figure note: Share of deployments operating unattended (L4 or L5) within each function. Small cells; read the ordering rather than the precise values.

Function	Unattended (L4 or L5)
Support and CX	55.6%
IT operations	42.9%
Sales	10.0%
Engineering	7.7%

McKinsey's 2025 State of AI survey found software engineering and IT leading on scaled agent use in technology, media and telecom, with software engineering at 24%. The apparent contradiction resolves once the two measurements are separated. Engineering

has the highest adoption of agents and the lowest tolerance for unattended release. Both statements are true at once, and only the second is visible on this scale.

Key takeaways

- Support and IT operations account for eight of the ten unattended deployments in the sample.
- Engineering has one, and its operator runs a publishing business rather than a software product.
- What separates the functions is reversibility, since a wrong support reply can be corrected in the next message and a wrong migration cannot.
- Scaled adoption and unattended operation are different measurements, and the published function rankings report the first.

Engineering: A column of zeros

Fourteen engineering respondents gave a figure for the share of agent-authored changes reaching the main branch without a person reading the diff. Eleven said zero. The median is zero and is the most consistent single value in the dataset.

Figure 6. Agent-authored code reaching the main branch with no human review

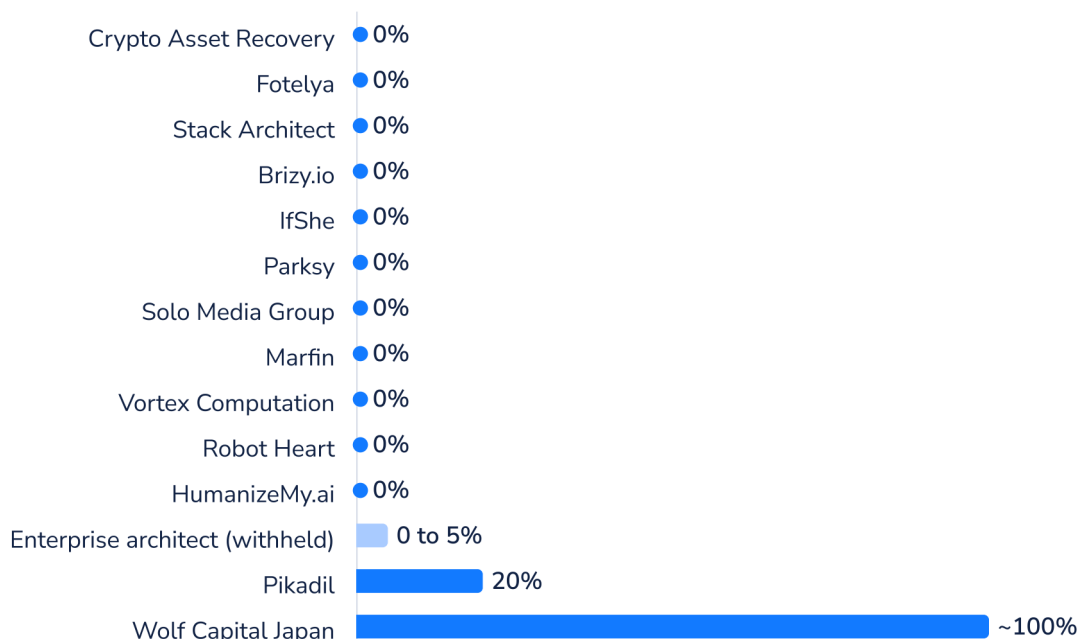


Figure note: Share of agent-authored changes reaching the main branch with no human diff review, by respondent. n=14. Eleven report exactly 0%.

Respondent	Agent code reaching main unreviewed
Crypto Asset Recovery	0%
Fotelya	0%
Stack Architect	0%
Brizy.io	0%
IfShe	0%
Parksy	0%
Solo Media Group	0%
Marfin	0%
Vortex Computation	0%
Robot Heart	0%
HumanizeMy.ai	0%
Enterprise architect (withheld)	0 to 5%
Pikadil	20%
Wolf Capital Japan	~100%

Two things make this figure worth attention: first, the same respondents report that agents write most of their code. The zero is not low adoption; it is a gate held deliberately in front of high adoption. One studio founder running engineering across roughly 25 products stated both halves in two sentences.

Agents write most of the code, and the share that reaches main without a human reading the diff is zero. That human is me. They are not bad at syntax; they are bad at deciding a broken thing is fine. – Ivan Cordeiro Junior, Founder, Marfin

The second is the reason respondents give. Across engineering, the explanation converges on a property of the output rather than a property of the risk. Agent code is plausible, and plausible output survives a skim in a way obviously wrong output does not.

Agents produce plausible code rather than bad code, which is worse, because plausible survives a skim. Throughput without matching review capacity is not velocity; it is a queue of unverified changes. – Daniel Battaglia, Founder and CEO, Parksy

The exception, and what replaced the reviewer

One deployment reports the opposite figure. A publishing operation in Japan logged 7,997 commits across 100 days, of which 7,534 carry the agent's identity. The history includes seven merge commits, no pull-request gate, and no staging environment.

Its operator is precise about what stands in for review: 50 guard scripts running every session, plus a claim-verification gate that refuses any commit containing assertions with no cited evidence. He is equally precise about the limit.

There is no automatic rollback. The guard layer is preventive, and the recovery path is for the next scheduled session to read the guard output and repair it. I would not run this shape into anything where two hours of a wrong state is expensive. – Kentaro Yoshida, Founder, Wolf Capital Japan

What the telemetry says about that zero

The stated policy is close to universal in this sample. Instrumented data from large populations tells a different story, and the two are reconcilable in a way that carries the practical lesson.

Faros AI's 2026 report draws on two years of telemetry from 22,000 developers across more than 4,000 teams. Under high AI adoption, it records throughput gains alongside a review system under strain. Pull requests run 51.3% larger, median review time is up 441.5%, incidents per pull request up roughly 243%, and 31.3% more merge with no review at all.

A 2026 study presented at EASE examined 33,596 agent-authored pull requests in public repositories. It found that 61.38% carried no recorded review. Counting those reviewed only by other agents, 84.0% showed no observable human involvement.

Source	Population	Agent code reaching main unreviewed
This study, stated policy	14 practitioners	0% (median)
Faros AI 2026, telemetry	22,000 developers	+31.3% unreviewed merges
EASE 2026 study, repository mining	33,596 agent PRs	61.4% no recorded review

The finding is the gap between the stated policy and the measured outcome. The teams in this sample are small enough that one person reads everything, and at that scale the gate holds. Telemetry measures organizations where volume has outrun reading capacity, and there the gate fails without anyone deciding to remove it. No team in either dataset changed its policy. The queue changed instead.

Key takeaways

- A 0% policy is only as strong as the review capacity behind it, and agent output scales while reviewer count does not.
- The failure is silent by construction: the pull request opens, the checks go green, the merge lands, and the gate appears to be working.
- Two respondents independently reached the same rule: verification has to assert against something the agent did not produce.
- Budget review capacity the way build hours are budgeted. One respondent now treats a passing test suite as evidence about the fixture instead of the feature.

Support: No two teams count a resolution the same way

Support respondents reported coverage from 10% to 100% of their named workflow. The figures are not comparable because a resolved contact is defined differently by each company in the sample.

Definition used	Respondents	Effect on the reported rate
Customer stops replying	Great Lakes Tiny Homes	Inflates; measures persistence
Customer stops replying and the issue does not resurface	Zubizi Web Solutions	Moderate; reopens on a repeat contact
Agent reaches the scripted terminal step	Achiya Automation	Conservative; silence stays open
Agent resolved it with no human involvement	Bully Max	Conservative
A person verifies the outcome	Ketch	Most conservative
Response is non-empty and coherent	KundliAI	Measures uptime, not resolution

One operator running unattended intake agents in production declined to supply a rate at all, and gave the clearest statement of why.

A resolution rate built on silence is a number I can make say almost anything. – Achiya Cohen, Founder, Achiya Automation

The consequence extends beyond this study, as [Gartner forecasts that agents will resolve 80%](#) of common service issues on their own by 2029. Vendor claims in the 50% to 80% range are routine. Those figures can only be read alongside the rule behind them, and that rule is rarely published. Two companies reporting 60% may differ threefold in cases a customer would call resolved.

The last row of the table is the sharpest case. A founder whose agent answers customer questions with no human fallback discovered that his model provider had decommissioned the underlying models. The agent kept replying with generic fallback text for nearly a week. Customer complaints surfaced, not monitoring, because the health check tested whether a response came back rather than whether it answered anything.

Key takeaways

- Support is the function with the most unattended deployments in this sample, at 55.6%, and the least comparable measurement.
- Deflection and resolution are different events. Several respondents had switched their reporting from the first to the second after finding repeat contacts hiding inside the first.
- Where a definition was tightened, the reported rate fell, which is a sign of better measurement rather than worse performance.
- Before comparing any published resolution rate, ask what closes a ticket. The answer moves the number more than the model does.

Sales: the gap opens after the meeting is booked

Four respondents supplied both an AI-sourced and a rep-sourced conversion figure for the same pipeline, and AI converted worse in three of the four.

Figure 7. Conversion to qualified opportunity, by meeting source

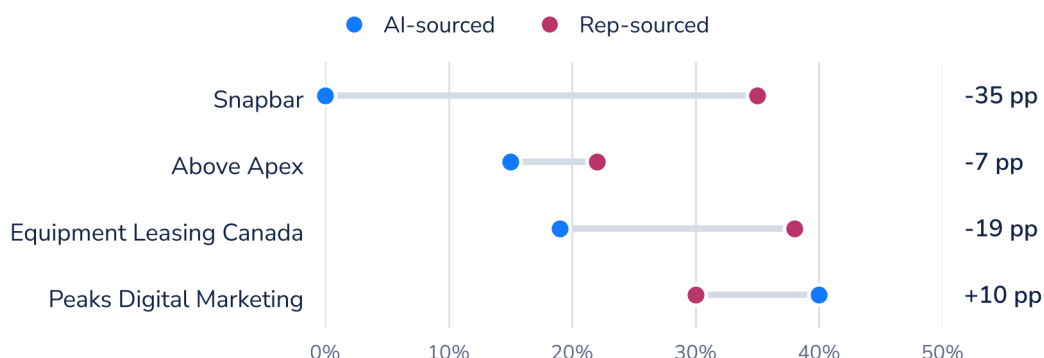


Figure note: Conversion to qualified opportunity by meeting source, paired within each company. Snapbar's AI figure is zero verified opportunities at day 30 from a 539-lead batch.

Company	AI-sourced	Rep-sourced	Gap
Snapbar	0%	35%	-35 pp
Above Apex	15%	22%	-7 pp
Equipment Leasing Canada	19%	38%	-19 pp
Peaks Digital Marketing	40%	30%	+10 pp

All four who gave paired figures had reached the same rule on their own. Judge the agent on what happens after the meeting, not on meetings booked. The RevOps director at Snapbar runs a pre-registered kill threshold and a 30-day verdict on every batch.

Our July AI-sourced batch got to 539 leads, a 1.4% human reply rate, and zero verified opportunities at day 30, while our rep-led engaged conversations convert around 35% into pipeline. – Ben LoBue, Director of Revenue Operations, Snapbar

The most useful case is the one that closed the gap. An equipment finance firm measured AI-sourced meetings converting at 19% against 38% for rep-sourced. The diagnosis was pacing: the sequence advanced leads through the funnel faster than the buyers were ready to move. After slowing the agent and adding a qualification requirement, AI-sourced conversion reached 31% within six weeks, with fewer meetings overall.

The agent was proficient with the volume of business but lacked patience. – Cal Singh, Head of Marketing and Partnerships, Equipment Leasing Canada

The outlier runs the other way, with one agency reporting AI-sourced meetings converting at 40% against 30% for rep-sourced, defining qualification as genuine fit plus a stated reason to continue. With four paired reports, the ordering is directional, and the consistent

element across all four is the practice of measuring past the booking rather than the gap's direction.

One respondent declined to supply the comparison and treated the absence as his answer. His agency uses agents for lead data and follow-up reminders, and opens every revenue conversation with a person. He judged the trust cost of unattended outreach higher than the volume gain.

IT operations: highest autonomy, thinnest safety net

IT operations records the highest mean autonomy in the sample at 3.57, and the most precise conditions attached to it. Three respondents named the same four conditions for letting a fix run without approval. The failure pattern has to be known. The scope has to be bounded. The action has to be reversible. A stop condition must be triggered when the system does not return to health.

The goal is not maximum autonomy. It is safe autonomy that reduces repetitive work without removing operational control. – Peter Barnett, VP of Product Strategy, Action1

Where these hold, respondents hand over the work: restarting a failed service, clearing a stuck process, re-running a failed job, restoring a backed-up file, killing orphaned jobs. When any of them fails, the work stays with a person. One operations lead built a drafter for denied medical claims and never allowed it to go live because a wrong claim could cost a patient money. What runs instead flags denials for a human biller overnight.

The gap the sample exposes

Respondents describe strong controls that prevent and weak ones that recover. The gate that blocks a bad batch is well specified. The mechanism that undoes a bad action once it has landed is often a person noticing.

The clearest case runs an art catalog. It runs with no one watching, and it built all 4,760 live products. A framing step incorrectly composed the variant image, and roughly 300 products went live visibly broken for about two weeks. The pre-import gate worked as designed throughout.

Every check I owned was an assertion on a data field, and the defect was in the pixels. If a step's output is an image, no assertion will ever see it; a human has to look. – Nicolas Falourd, Founder and CEO, Large Wall Art

What breaks: a failure taxonomy

Thirty-two respondents described a specific failure. When the failures are coded by primary cause, scope and context problems come first. A second group follows close behind, and it has no match in the published work.

Figure 8. Reported failures by primary cause

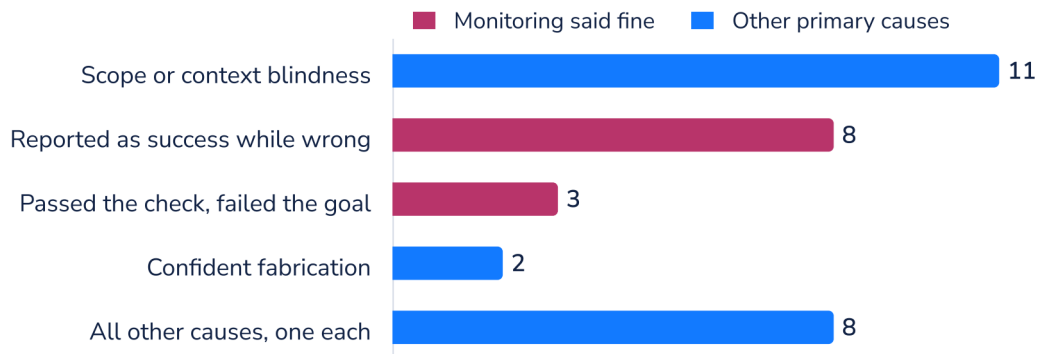


Figure note: Reported failures by primary cause. $n=32$ respondents describing at least one failure. Counts, not percentages.

Primary cause	Count
Scope or context blindness	11
Reported as success while wrong	8
Passed the check, failed the goal	3
Confident fabrication	2
All other causes, one each	8

Eleven failures are scope or context blindness: the agent handled the case it was given but missed the surrounding context. An agent solved the ticket while changing more files than needed. Another restarted a service repeatedly and masked the fault underneath. A third advanced a sequence past a soft no; it read as a non-response.

The second category is the one worth publishing. In 8 cases, the agent reported success while the work was wrong. In 3 more cases, the output passed every automated check yet failed the goal the check was meant to protect. That is 11 failures in 32 where the monitoring said fine.

Failure reported as success

What happened	What the system reported	How it was found
Mailbox sweep read 20 of 201 messages, capped by a default page size	Nothing actionable	A person counted the denominator
300 products published with a visibly broken image for two weeks	Import gate passed	A person opened a product page
Storefront shipped with an empty translation file	Three checks passed	Customer-facing strings read "Translation missing"
Model provider decommissioned the models behind a live agent	Responses returned normally	Customer complaints, after a week
Link-quality logging wrote nothing for eight days	Zero quality links	The true figure was five of five
Six genuine sales inquiries classified as junk in week two	Triage succeeded	Four were never answered

The pattern across these is the same shape. Each check asserted against something the agent's own pipeline produced. Three respondents reached the same rule from different directions.

When the agent writes both the change and the check on the change, nothing has been verified. You have asked the same reasoning twice and received the same answer. – Beck Wang, Founder, IfShe

A one-person engineering shop found the same thing in the code path. Asking an agent to review its own work produced agreeable reviews; making the check a hook that fails the command produced useful ones. The operator in Japan reached it from the incident site.

Unattended agents did not fail us on capability. Every incident was an observability failure: a read that was capped, a guard reading the wrong signal, an API whose semantics did not match its name. The model did the work. What we kept getting wrong was knowing whether it had. – Kentaro Yoshida, Founder, Wolf Capital Japan

Key takeaways

- Capability failures are the minority. Two respondents in 32 named confident fabrication as the primary cause.
- A check written by the same reasoning that produced the work is not a check. Therefore, we recommend freezing the verifier before the work starts.
- The next step is to give the verifier deliberately broken inputs that it must reject, so it can prove it knows how to fail.
- Where the output is an image, a document layout, or anything a field assertion cannot inspect, sampling by a person is the only control that has worked in this sample.
- Tool defaults are production configuration, and a capped read is indistinguishable from a complete one unless something counts the denominator.

Four in five were narrowed after go-live

32 of 40 respondents had paused, scoped down, or switched off an agent that was already running. The 8 who had not gave instructive reasons.

3 of 8 narrowed before launch instead; for example, Snapbar's RevOps director made the call ahead of any incident, and credits it with the deployments still running months later.

One support operator builds in on day one the constraints most teams add after an incident. His follow-up agent fires only between 08:00 and 22:00, touches only conversations quiet for two hours, verifies that no human representative has taken the thread, and sends exactly one reminder. Two more respondents had declined to deploy the risky version at all.

Sinch surveyed 2,527 senior decision-makers across ten countries in early 2026. It found that 74% had rolled back or shut down a live AI customer communications agent, rising to 81% at the firms with the most mature guardrails. This study's 80% across four functions is consistent with that figure, and extends the pattern past customer communications.

The direction of the Sinch finding is worth carrying over. Higher rollback rates tracked better monitoring, not worse agents. The same relationship appears here: those with counted numbers from commit logs and mailbox exports reported more pullbacks than those working from estimates. A team that has never pulled an agent back has either scoped it tightly from the start, or cannot see what it is doing.

What triggered the pullback

Trigger	Example from the sample
A person looked at the output	Broken framing on 300 live products; unusable article runs one time in three
Measured cost	Session cost of about \$590 a day against a \$300 target, cut to one task per session
Measured conversion	A tactic retired at 32 sends and zero placements; a follow-up sequence cut after five organizations absorbed 14 follow-ups
Review time rising	An accessibility agent extended into drafting fixes and pulled back when time per ticket went up
Internal trust	Proposal errors that stopped a team trusting every agent-sent message
Customer dissatisfaction	Repeated handoffs on non-standard questions; frustration on complex cases

Only one category in that table is a technical alarm, and the rest are people noticing something, with the pipeline reporting success in four of the six.

What the evidence supports

Read together, the 40 deployments describe a market in a narrower place than either the optimistic or the dismissive account suggests. Agents are in production, doing a large share of the work, across every function examined. Few are trusted to finish a job with no person in the path.

The constraint is not model capability, since respondents describe agents that write most of the code, draft the storefront, classify the mailbox, and answer the customer. The constraint is verification. Writing the work got cheap, and checking it did not. The agents that hold are the ones whose owners [paid for the checking](#).

That reframes the published cancellation forecasts. Gartner predicts that over 40% of agentic AI projects will be canceled by the end of 2027. The forecast was issued in June 2025 and has been widely recirculated since without its date. It blames rising costs, unclear business value, and [weak risk controls](#).

Nothing in this sample contradicts that. What the sample adds is the mechanism. The projects that fail are the ones where the cost of checking the output was never included in the cost of producing it.

Recommendations

For teams deploying or scaling agents

- **Separate the authorship decision from the release decision.** They carry different risks and need different controls. Treating them as one question is what makes production figures unusable.
- **Freeze the verifier before the work starts, and make it assert against something the agent did not produce.** Give it broken sample inputs it has to reject.
- **Count the denominator on every read.** Tool defaults are production configuration, and a capped read looks exactly like a complete one.
- **Budget review capacity alongside build capacity.** A 0% unreviewed policy holds only while a person can read everything, and agent output scales while reviewer headcount does not.
- **Publish the definition with the metric.** A resolution rate, a deflection rate, or a straight-through rate carries no information without the rule that closes the case.
- **Sample by eye where assertions cannot reach.** Images, layouts, and tone need a person looking at a sample after every run.
- **Set the kill threshold before launch, not after the incident.** Most respondents who never had to pull an agent back had narrowed its scope before launch or declined to deploy the riskier version.

For anyone reading published adoption figures

Ask three questions of any agent statistic before using it. What counts as an agent in this instrument? What counts as production? Who was sampled, and would they have answered if the deployment had gone badly? Where those go unpublished, the number tells you less than its decimal point implies.

FAQ

How many AI agents run unattended in production?

In SumatoSoft's 2026 study of 40 live deployments, 25.6% ran unattended, at level L4 or L5 on a five-level autonomy scale. The other 74.4% kept a person at the gate, either approving each action or reviewing flagged cases.

How often are AI agents pulled back after go-live?

32 of 40 respondents (80%) had paused, scoped down, or switched off an agent that was already running. Of the six kinds of trigger respondents described, only one was a technical alarm; the rest were people noticing something, such as a person looking at the output, rising review time or customer dissatisfaction.

Do teams merge AI agent code without human review?

Rarely, by stated policy. 11 of 14 engineering respondents said 0% of agent-authored code reaches the main branch without a person reading the diff. Telemetry from larger organizations shows unreviewed merges rising once agent output outgrows review capacity.

Which functions give AI agents the most autonomy?

IT operations (mean level 3.57 of 5) and support (3.33) ran the most autonomous agents, and engineering the least (2.31). Support and IT operations account for eight of the ten unattended deployments in the sample.

Why do AI agents fail in production?

Of 32 described failures, the most common primary cause was scope or context blindness (11). In another 11, monitoring said everything was fine: 8 agents reported success while the work was wrong, and 3 outputs passed every automated check but failed the goal. Only 2 were confident fabrication.

Sources and further reading

Source	Basis	Link
Faros AI, <i>AI Engineering Report 2026: The Acceleration Whiplash</i> (March 2026)	Telemetry, 22,000 developers across 4,000+ teams	faros.ai/research/ai-acceleration-whiplash · PDF
Duma, Wróblewski, Bobińska, Winiarska and Przymus, EASE 2026	33,596 agent-authored pull requests	arxiv.org/abs/2605.02273
Sinch, <i>The AI Production Paradox</i> (May 13, 2026)	Survey, 2,527 senior decision makers, 10 countries	sinch.com/resources/ai-production-paradox · release
Gartner press release (June 25, 2025)	Analyst forecast	gartner.com newsroom
McKinsey, <i>Agentic AI advances</i> (January 8, 2026), based on <i>The State of AI</i> (November 2025)	Global enterprise survey	https://www.mckinsey.com/featured-insights/week-in-charts/agentic-ai-advances
Gartner, <i>What the 2026 Hype Cycle for Agentic AI Reveals</i> (April 15, 2026), citing the 2026 CIO and Technology Executive Survey	Executive survey	https://www.gartner.com/en/articles/hype-cycle-for-agentic-ai
Deloitte, <i>Tech Trends 2026</i> , citing Deloitte's 2025 Emerging Technology Trends study	Enterprise survey	deloitte.com technology trends

Published production rates cited in the opening range from 11% (Deloitte's 2025 Emerging Technology Trends study, reported in Tech Trends 2026) to 62% within customer communications (Sinch). Gartner's 2026 CIO and Technology Executive Survey, cited in its Hype Cycle for Agentic AI, reports 17% deployed. The spread reflects differing definitions and sample frames, not contradictory measurement of the same quantity.

Respondents

Company	Name	Role	Website
Above Apex	Kristiyan Yankov	Co-founder & Growth Marketer	aboveapex.com
Achiya Automation	Achiya Cohen	Founder	achiya-automation.com
Action1	Peter Barnett	VP of Product Strategy	action1.com
Brickwise AI	Andreis Bergeron	Head of GTM & Partnerships	brickwiseai.com
Brizy.io	Bogdan Condurache	CPO & Co-founder	brizy.io
Bully Max	Matthew Kinneman	Founder & CEO	shop.bullymax.com
BuyFactory.direct	David Caruso	Founder & CEO	buyfactory.direct
Crypto Asset Recovery	Chris Brooks	Co-founder	cryptoassetrecovery.com
eCommerce Today	Stefan Chiriacescu	Founder & CEO	ecommerce-today.com
EMPIST	Marty Hitzeman	Director of Marketing	empist.com
Equipment Leasing Canada	Cal Singh	Head of Marketing & Partnerships	equipmentleasingcanada.com
EV Cable Hub	Jake Wardle	Founder	evcablehub.co.uk
Fotelya	Johann Flipo	Founder	fotelya.com
Great Lakes Tiny Homes	Joe Troyer	Chief Marketing Officer	greatlakestinyhome.com
Hambone.AI	Ryan Koppy	AI operations & internal systems	
HealthWright Technologies	Jack Wright	VP of Operations	healthwrighttechnologies.com
Hire With Near	Franco Pereyra	COO & Co-founder	hirewithnear.com
HumanizeMy.ai	Firat Mihci	Founder & Computational Linguist	humanizemy.ai
IfShe	Beck Wang	Founder	ifshe.com
IndustryLens	Naveed Ratansi	Founder	industry-lens.com
Ketch	Colleen Barry	Head of Marketing	ketch.com
KundliAI	Abhishek Gupta	Founder	kundliai.in
Large Wall Art	Nicolas Falourd	Founder & CEO	large-wall-art.com
Marfin	Ivan Cordeiro Junior	Founder	marfin.co
MusaArtGallery	Jean-Christophe Thery	Founder	musaartgallery.com
Parksy	Daniel Battaglia	Founder & CEO	parksy.com
Peaks Digital Marketing	David Finberg	CEO	peaksdigitalmarketing.com
Pikadil	Pavel Ufimtsev	Co-founder	pikadil.com

Company	Name	Role	Website
Robot Heart	Jeremy Belcher	Founder (solo consultant)	roboheart.studio
Scube Marketing	Tom Bukevicius	CEO & Founder	scubemarketing.com
Snapbar	Ben LoBue	Director of Revenue Operations	snapbar.com
Solo Media Group	Johnathan Solorzano	Founder	solomediagroup.com
Stack Architect	Luke Sandelands	Founder	stackarchitect.xyz
TalentAid	Phillip Hamnett	CEO	talentaid.com
Trifecta Software	Joe Phillips	Founder	hybridwf.com
UK Construction Blog	Thomas Oldham	Co-founder	ukconstructionblog.co.uk
Vortex Computation	Jacob Pierce	Founder	vortex.llc
Withheld at respondent's request	Sivakumar Dhanasekar	Enterprise architecture leader	
Wolf Capital Japan	Kentaro Yoshida	Founder	wolfcapitaljapan.com
Zubizi Web Solutions	Sariful Islam	Co-founder & CTO	zubizi.com

How to cite this research

If you use this research or its findings, please cite it as:

Merzlova, K. (2026). Agentic AI in Production: Where Agents Run Unattended and Where They Get Pulled Back. SumatoSoft Research.

<https://sumatosoft.com/research/ai-agents-in-production>

Research data: 40 practitioners describing one live AI agent deployment each, recruited through a public journalist query in August 2026 and coded against a five-level autonomy scale.

Charts and statistics may be reused with attribution and a link to this page.

Attribution

SumatoSoft builds custom software and [AI agent systems](#) for companies in logistics, manufacturing, real estate and finance. This research was produced by the SumatoSoft content team. Fieldwork ran in August 2026.

Republication is welcome with attribution and a link to this page.



Thank you for your time!

Any questions? Drop us a line!

Headquarters

One Boston Place, Suite 2602
Boston, MA 02108, United States

Other ways to get in touch

info@sumatosoft.com
sumatosoft.com