Output doubled.
Quality improved.

epilot has **no platform team** and no ops, SRE, DevOps or backend-only roles. Every one of our **43 product engineers** works across the whole stack, alongside one security engineer, one QA and their coding agents. The results here belong to the whole epilot team, not to a platform or AI team.

Our engineers were already working with AI assistants. In February 2026 we brought autonomous coding agents into Slack and set up the tools and environments for them to work across our whole codebase. **We did not grow the engineering team**: between 40 and 44 engineers have committed code in every month since January 2025. Eight months later the same team ships more than twice as much, adds about three times as much new product capability each quarter, and production is steadier than before. Every number here comes from our own GitLab, Datadog, Jira and Zendesk data (how we measured).

2.4×

PRs* merged every week

495 in Q3 2026, up from 204 in Q1 2025, with the same team size

244

production deploys in an average week

up from 137; best week 359

2

customer-facing SEV-1 incidents in 2026

4 in 2025, 14 in 2023

22

people outside engineering now contribute code

up from 8 in all of 2025

* PRs: pull requests, our term for both GitLab merge requests and GitHub pull requests. PR counts come from GitLab.

OUTPUT DOUBLED

244 production deploys in an average week

Every deploy goes through the same pipeline and test gates, whether a person or an agent wrote the change.

Production deploys rose from 137 to 244 a week



Successful production deployments per week across 180 services, January 2025 to October 2026 (GitLab).

244 deploys a week, without surprising your users

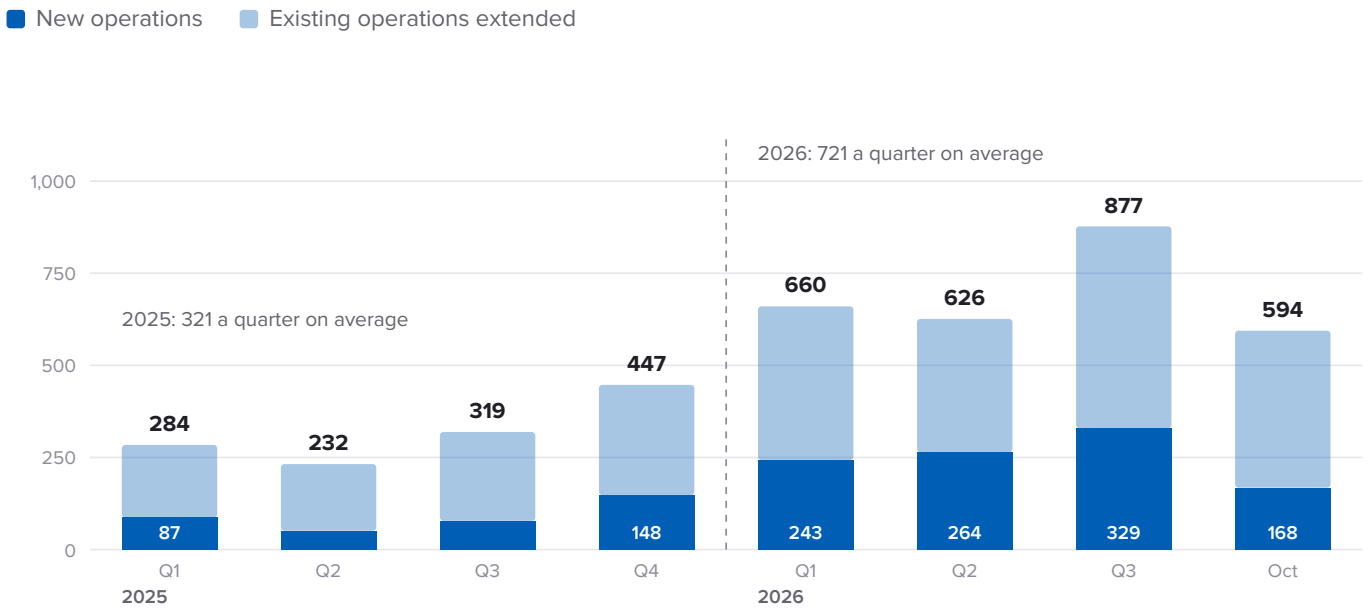
Backend services deploy continuously, behind the same test gates, and an agent writes every API change into our public API changelog, with breaking changes and deprecations marked and an RSS feed to subscribe to. The apps your teams work in move on a schedule: new frontend code goes first to an opt-in Canary channel, while 77% of customer organisations stay on Stable. Stable takes a major release about every two months, after a company-wide Breakathon and acceptance testing, with fixes in between. There is no fixed release date: we release when we're confident.

IMPACT

About 3× more new product capability per quarter

PRs and deploys count changes, not what they add up to. So we measured it two ways: every operation in every API behind our apps and integrations, and the roadmap items our product managers rate high impact in Jira.

API operations added or extended per quarter: 321 on average in 2025, 721 in 2026



OpenAPI specifications in all active epilot GitLab repositories. Extended: contract differs from every earlier version. Q4 2026: October only.

3.1×

new API operations a quarter

91 in 2025, 279 in 2026

1.9×

existing API operations extended a quarter

230 in 2025, 442 in 2026

2.1×

high-impact roadmap items per major release, rated by our PMs in Jira

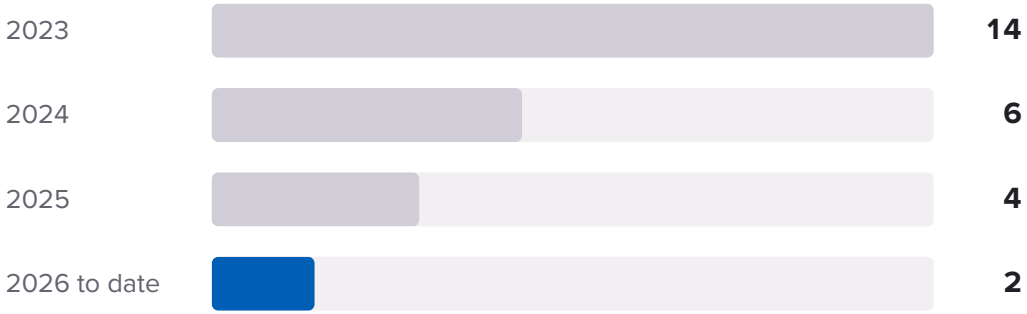
about 8 in 2025, 17 in 2026

QUALITY IMPROVED

More output, fewer severe incidents

Severe incidents and reverts fell. Customer bug reports grew with our customer base, not with our output: per customer they're about flat, while we ship 2.4× as much.

Customer-facing SEV-1 incidents per year



Datadog Incident Management; drills and false alarms excluded. Live status: status.epilot.cloud.

0.5%
of merged PRs reverted
down from 1.7% in Q1 2025

Flat
bug reports per customer organisation: 2.0 to 2.2 a quarter, while we ship 2.4× as much

≤45 min
customer impact of each 2026 SEV-1

~96%
of bugs caught before a release reaches customers

69%
of bugs found by us, not reported by customers

~40%
of bug fixes opened by agents since June, merged by engineers

And the app got faster

−76%
longest main-thread block in our browser tests since April
1,020 ms to 243 ms

−38%
JavaScript loaded at start
14.9 MB to 9.2 MB

99.98%
of requests through our public API gateway answered without a server error

Speed gets deliberate work: we merged 181 performance PRs so far in 2026, against 49 in the same months of 2025.

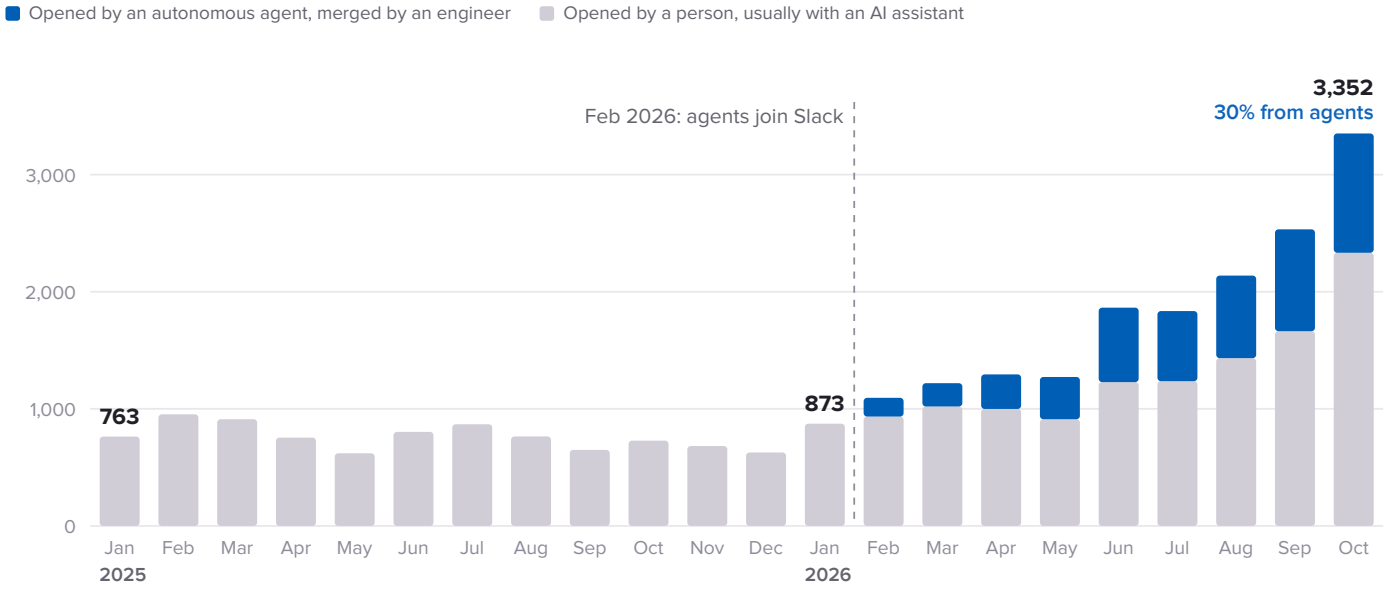
HOW WE DID IT

Agents do the work. Engineers stay in charge.

Anyone at epilot can tag a coding agent in Slack. It spins up a cloud sandbox with our whole codebase and comes back with a PR.
An engineer checks what it does and merges it, and only CI deploys it.

Autonomous agents now open about 30% of the PRs we merge

Merged PRs per month, January 2025 to October 2026



GitLab, all epilot groups. October 2026 counted to the 27th.

4,844

agent PRs merged by engineers, out of more than 8,000 opened since February

873 → 1,663

PRs opened by people a month, January to September: agents add to what engineers ship

Context, not prompts

Every repository and a compressed index of our internal docs sit in every agent's context, so the code follows our conventions instead of guessing.

Review the behaviour, not every line

Engineers check the spec or agent session behind a PR and click through the running preview. The code they skim for shape; tests and CI gates catch the rest. Only CI deploys.

On-call through agents

Monitoring opens a Slack thread for every alert, and the on-call engineer has an agent investigate it right there. One engineer merged about 100 fixes in a single on-call week.

Background agents that don't wait to be asked

More than a dozen agent workflows run on their own, on a schedule or on a pipeline event. Each one reports to Slack or opens a PR, learns from every run, and leaves the decision to an engineer.

Production triage

Every 30 to 60 minutes: groups errors and alerts into patterns, investigates across services and fixes the trivial ones.

Bug and translation tickets

Several times a day: picks up new bug and translation tickets and opens PRs for the fixes it is confident about.

Tech debt

Daily: takes one item from the engineers' backlog, such as an upgrade, a missing test or a migration.

PR review

On every PR: looks for bugs, security issues and risky changes, and leaves review threads for the author.

Risk check

Before a production deploy: reviews the changeset and rates its risk.

API changelog

On every API release: writes the public changelog from the spec diff.

Self-healing tests

When a browser test fails: reproduces it and fixes it in the right place, whether test, product, infrastructure or CI.

Health-check coverage

On a new bug or support thread: checks that our customer configuration health check covers the case, and adds a check if not.

AGENT-READY CI

CI built for agent scale

Thousands of agent PRs only work if each one gets a fast pipeline and a running preview. Pipeline volume grew 3 to 4× this year while each pipeline got faster and cost roughly a quarter of what it did.

Before and after, in minutes



Pipeline time: GitLab medians, Q1 2025 vs Q3 2026. Staging e2e: Datadog CI Visibility, January 2026 vs the last 3 days. Build and test jobs: CI runner migration, January vs September 2026.

- **AWS runners with cached ARM images**, rolled out to about 80 production repositories.
- **Browser tests on AWS Lambda runners.** The full staging suite of about 210 browser tests runs in just over 3 minutes. AI test selection runs only the tests a change can affect, so the gate averages about 2.5 minutes.
- **A review app on every PR.** Reviewers, PMs, designers and agents click through the change running before it merges, and agents test their own changes with Playwright before a human looks.

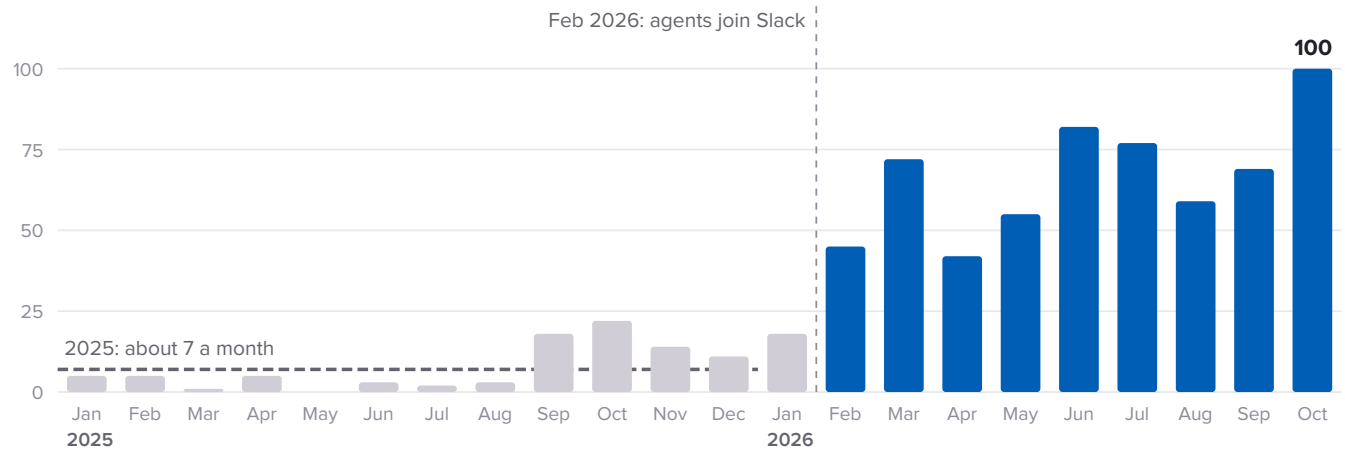
THE DEVELOPMENT WE'RE PROUDEST OF

22 people outside engineering now contribute code

Product managers, designers, customer success and QA now fix what slows their customers down.

Contributor days from people outside engineering: from about 7 a month in 2025 to 100 in October

Days on which a colleague outside engineering committed code, counted once per person per day, GitLab and GitHub, January 2025 to October 2026



Git history across epilot's active GitLab and GitHub repositories, merge commits excluded. Changes they request from agents in Slack are committed by the agent and not counted here. October 2026 counted to the 27th.

9×

more contributor days a month since February than the 2025 average

807

PRs they drove between February and September

543

merged: two in three, more than the agents' 59%

Product managers ship features

One has opened 141 PRs since February, including a feature spanning frontend, API and a shared library.

Customer success fixes what customers hit

A customer complaint becomes a PR with a before-and-after recording, not a ticket.

Designers send working software

Proposals arrive as preview links, not mockups, and designers run the design system themselves.

Teams build their own tools

Finance, projects, product and QA have built internal tools with agents, such as the control center our QA built in a few weeks to run major releases.

WHAT KEEPS IT STABLE

Guardrails built into how we ship

The most important guardrail is people: engineers still own merging and shipping code.

Engineers own merge and deploy

Agents and colleagues outside engineering open PRs; an engineer merges them and CI ships them. The engineers who build a service also run it on call, with agents triaging alerts alongside them.

Browser tests, twice

About 460 Playwright tests gate every release: around 210 on staging and 200 more on production. AI picks the tests each change can affect, and self-healing tests repair themselves when the UI changes.

Performance budgets

Every pipeline checks budgets for JavaScript, CSS and long tasks and fails when something creeps back.

Serverless, certified, locked down

Every backend service is a short-lived AWS Lambda function with least-privilege rights, so a vulnerable dependency reaches one function, not the whole backend. Customer data stays in Frankfurt, we're ISO 27001:2022 certified and independently pen-tested, and no laptop or agent reaches production without an approved, expiring session. In October, agents merged fixes for a Node.js security release across about 15 services within 24 hours.

> **“Our guardrail is simple: agents create MRs, engineers review & merge. main triggers deployments, so we need engineers in that loop.”**
>
> Viljami Kuosmanen, Head of Engineering, 10 March 2026

AI AND YOUR DATA

Coding agents never touch customer data

They work on code, in isolated, epilot-owned sandboxes running in the EU. The AI features inside epilot follow strict rules of their own.

No customer data, no deploy rights

Agent sandboxes run in the EU and have no access to customer data. Agents open PRs; engineers merge them, and only CI deploys.

AI in the product stays in the EU

All AI inference for our product features runs in the EU, including failover. No personal data leaves the EU.

Never used to train models

Prompts, emails and records are never used to train foundation models, and our provider contracts guarantee it. We don't fine-tune models on customer data.

AI suggests, people decide

Every AI feature is decision support with a non-AI fallback, and admins can switch AI off for their organisation within a minute. Grid connection, §14a EnWG and feed-in decisions never run on AI.

HOW WE MEASURED

- From epilot's own GitLab, git, Datadog, Jira and Zendesk data through 27 October 2026. Comparisons are Q1 2025 against Q3 2026 unless stated.
- PRs: merged GitLab merge requests across all epilot groups. Agent PRs: opened by our autonomous agent accounts and merged by an engineer.
- Team size: between 40 and 44 engineers committed code in every month of 2025 and 2026.
- Production deploys: successful production deploy jobs per week across 180 services.
- SEV-1: customer-facing incidents in Datadog Incident Management; drills and false alarms excluded.
- Reverted PRs: merged PRs that revert an earlier change, as a share of all merged PRs.
- Bug reports per customer organisation: Zendesk tickets classified by support as bugs, divided by the customer organisations that filed any ticket that quarter; 2.0 in Q1 2025, 2.2 in Q3 2026. In absolute terms, bug tickets rose from about 56 a month in 2025 to about 97 a month in Q3 2026, as the organisations filing tickets grew from 85 to 133. Support also classifies more of its tickets than before (81% in Q1 2025, 93% in Q3 2026).
- Customer-reported bugs in Jira (bugs escalated from customer support) show the same trend: about 50 a month in 2025, about 100 a month from April 2026, while merged PRs grew 2.4×. Per 100 merged PRs they fell from 6.1 to 4.5. Part of that step is our release process: since spring 2026 we release when we're confident rather than on a fixed date, and customer-reported bugs flagged as release blockers, fixed before they reach Stable, rose from about 4 to about 18 a month. Earlier years aren't comparable, because customer bugs only moved into this Jira project in 2025.
- Bugs caught before release: bugs found during each 2026 release cycle against customer-reported bugs on Stable in the five days after each release. Found by us: bugs not escalated from customer support, 2026.
- SEV-1 customer impact: start to resolution of customer impact, per incident.
- Agent bug fixes: Jira bugs fixed through a PR that one of our agent accounts opened, as a share of all bug fixes linked to a PR; 35 to 43% each month since June 2026, 30% since February.
- Performance: Datadog browser-test monitors, first week of April against the last seven days. API: requests through our public API gateway in October 2026; internal service traffic not included.
- People outside engineering: anyone outside an engineering role who authored commits on the main branch of our active GitLab or GitHub repositories since February 2026 (22 people); 807 PRs and 543 merges cover February to September 2026. Chart: contributor days per month, a day on which one of them authored at least one commit in our GitLab or GitHub repositories, counted once per person per day.
- API operations: every OpenAPI specification in our active GitLab repositories (archive excluded), read at the start of each quarter and on 30 October 2026. An operation is one method and path in one API, counted once even when a spec is copied into several repositories; a renamed API counts as removed and added. Net growth (added minus removed) rose 2.9×, from 82 to 234 a quarter. Product repositories alone show the same 3.1×. Existing operations extended: an operation that existed at the start of the quarter whose parameters, request body, responses or the schemas they reference in the same file differ from every earlier version of it; descriptions, examples and other wording are ignored, and an operation counts once a quarter however often it changed. 230 a quarter in 2025, 442 in 2026 (1.9×); new and extended together 321 and 721 (2.2×). Changes to schemas kept in separate files are not counted. Changes that only touch the UI are not counted.
- High-impact roadmap items: items on our product roadmap in Jira that product managers rated high impact, per major release they targeted; 7.9 on average across 2025's seven releases, 16.8 across 2026's first five. All roadmap items per release rose 1.7×, from 22 to 39. The roadmap in Jira tracks product-manager-led initiatives; much engineer-led work, such as integrations, AI features and Config Hub, is not on it.
- CI: GitLab pipeline medians and Datadog CI Visibility. Staging browser tests: the 62 successful full runs (207 tests) in the last 7 days took 3.2 minutes at the median, half of them between 3.0 and 3.6 minutes; across the last 3 days' 556 runs of the staging gate, runs narrowed by AI test selection (25 tests at the median) took 2.3 minutes. Distinct browser tests run in the last 14 days: 459.