navigara · RESEARCH

The Open Source Engineering Performance Report

Navigara's Commit-Level Analysis Across Cloudflare, Vercel, OpenAI, Google, Meta, and Microsoft · Q1 2025 – Q1 2026

Published 2026-04-30 · Data sourced from Navigara Knowledge Graph metrics

Peter Malina, Jirka Bachel

Corresponding author: research@navigara.com

1. Abstract

This report describes changes in output across six organizations' **SWE** (Software Engineer) contributors over five consecutive quarters (Q1 2025–Q1 2026). A contributor is classified as SWE when they have recorded commit activity in at least **10 weeks** of the measurement window. All findings are **descriptive and correlational**; the report makes no causal claim about why any observed pattern occurred. Measurements cover public-repository commits attributed to those SWEs (see Section 6 for the full definition).

The six organizations were drawn as a purposive sample of AI-forward engineering teams — three companies that make frequent public claims about AI productivity gains (Cloudflare, Vercel, and OpenAI) and three large-scale incumbents with strong engineering reputations (Google, Meta, and Microsoft).

Performance, reported throughout this paper as *Engineering Throughput Value* (**ETV**), is a commit-level measure of the complexity of the changes developers ship. Every merged commit is scored by the structural complexity and intent of the change, the context in which it landed, and whether it produced lasting value in the codebase. Each commit is classified as *Growth* (new feature development and net-new capability), *Maintenance* (refactoring and incremental improvements to existing code), or *Fixes* (repairs to defective or broken behavior). ETV is the scalar sum of those per-commit scores, aggregated per developer.

Between Q1 2025 and Q1 2026, the developer-weighted mean ETV per qualifying SWE moved by **+116%** (endpoint 95% CI [+84%, +148%]) — **+98%** on a fixed panel of 418 SWEs active throughout the full window (see Section 2). The headline averages across qualifying SWEs in each quarter, with each SWE contributing one observation.

Because the open cohort grew from **565** to **676** qualifying SWEs over the window, the headline above reflects both within-SWE change and changes in cohort composition. To isolate within-SWE movement, Section 2 reports a parallel *fixed-panel* view tracking the 418 SWEs active in every qualifying quarter. In that panel the change is **+98%** (endpoint 95% CI [+63%, +140%]). The difference between the open-cohort and fixed-panel figures is the share of the headline attributable to cohort growth rather than individual-level change.

The work classification mix (Section 3) shifts toward Growth and Fixes over the window, with a corresponding decline in Maintenance share. These are observations about output composition, not statements about strategic intent.

Per-organization changes vary in direction and magnitude (Section 4). The trajectory chart rebases each organization to its first qualifying quarter and traces quarterly progress on a shared axis; the work-classification small multiples decompose that trajectory into Growth / Maintenance / Fixes composition per quarter; and the appendix table lists per-quarter values alongside per-cell SWE counts (N).

OpenAI enters the cross-org aggregate beginning Q2 2025. Its Q1 2025 qualifying cohort had 6 SWEs, below the sample floor used in this edition; the subsequent quarters reflect a rapid ramp in OpenAI's public-repository engineering activity and should be read as a growing cohort rather than a steady-state baseline. Whether OpenAI's figure stabilizes in later editions is an empirical question for future reports.

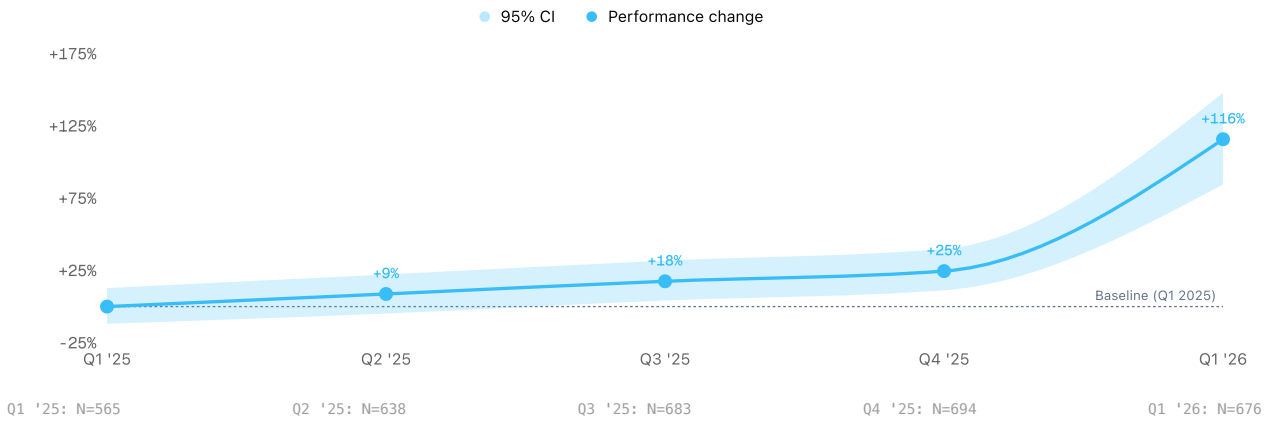
Interpretation. The measurement window coincides with broad production adoption of AI coding assistants across the organizations in this sample — code-generation tools, inline completion, and agentic workflows. The coincident timing, the concentration of gains in Growth-classified output, and the persistence of the pattern in the fixed-panel cohort (which holds the SWE population constant) point most naturally to AI adoption as the leading explanation for the observed rise. The study does not quantify what share is attributable to it; Section 2 discusses which trajectories the ETV scalar can and cannot distinguish.

2. Average SWE Performance

Figure 1a plots the percentage change in developer-weighted mean performance per qualifying SWE per quarter, relative to Q1 2025 = 0%. The shaded band is the 95% bootstrap confidence interval obtained by resampling SWEs within each quarter. Figure 1b plots the same metric restricted to the fixed-panel cohort — the same SWEs active in every qualifying quarter — so its trend is not confounded by changes in who is included each quarter.

Figure 1a: Performance change per qualifying SWE (open cohort)

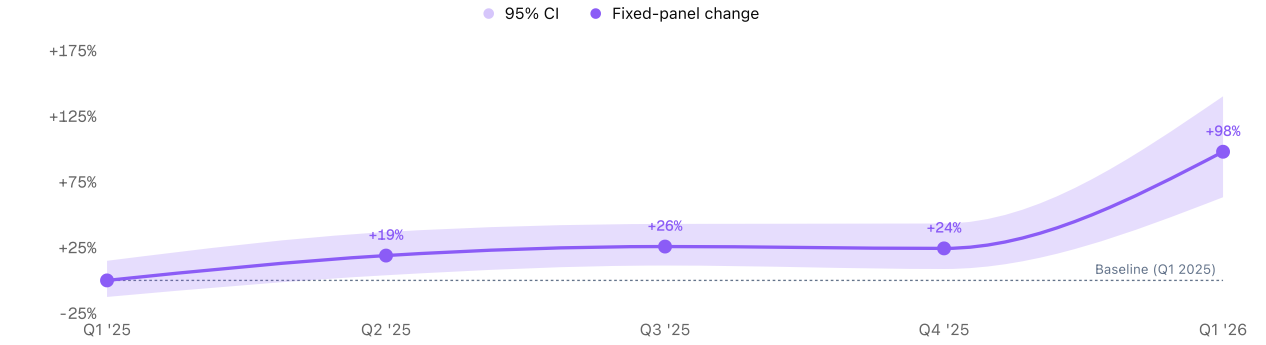
Percentage change vs Q1 2025 baseline | developer-weighted across included organizations | shaded band = 95% bootstrap CI



Q1 '25: N=565 | Q2 '25: N=638 | Q3 '25: N=683 | Q4 '25: N=694 | Q1 '26: N=676

Figure 1b: Performance change for the fixed-panel cohort (418 SWEs)

Same developers tracked across every quarter | percentage change vs Q1 2025 baseline | 95% bootstrap CI



The fixed panel removes cohort composition as a source of change, at the cost of survivorship bias toward continuously-active SWEs. Definition and window detail in Section 6.

Reading Figures 1a and 1b together. The open cohort blends two distinct forces — per-SWE change and quarter-over-quarter turnover in who qualifies — while the fixed panel holds the sample constant and isolates the within-SWE component. In this edition the two series agree in direction across every quarter: the open cohort ends Q1 2026 at **+116%** vs Q1 2025, and the fixed panel ends at **+98%**. That convergence is informative. When the within-SWE series tracks the open-cohort series, the measured rise is not a pure composition artifact — it survives holding the population constant. About 85% of the open-cohort delta is preserved in the fixed panel; the residual is attributable to cohort expansion (new qualifying SWEs entering the sample) and, to a lesser extent, quarter-over-quarter churn in who clears the benchmark.

What the scalar can and cannot tell us. A rising Engineering Throughput Value is consistent with several underlying trajectories that carry very different strategic implications: expansion into new product surfaces, heavier performance on existing systems, or increased operational load from defects and incidents. The aggregate conflates these regimes. Two organizations with identical quarter-over-quarter deltas can be on substantially different paths — one building capability, the other stabilizing it — and the single number will not separate them. This is the motivation for the decomposition that follows: projecting performance onto its *Growth* / *Maintenance* / *Fixes* components reveals where the gains are being booked, not only that they exist.

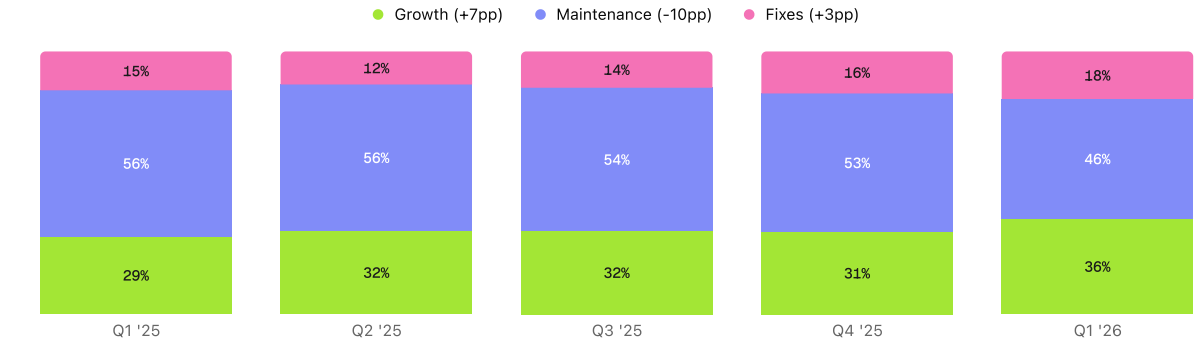
Interpretive guardrail. The bootstrap intervals in Figures 1a and 1b are wide enough that single-quarter jumps should be read cautiously; the multi-quarter slope is the signal of interest. Per-organization views (Figure 3b) have wider bands still, proportional to the square root of cohort size. Directional interpretation is best anchored to the endpoints of the reporting window rather than to any single inter-quarter delta, and the shaded CI bands should be consulted before treating a quarter-over-quarter movement as meaningful.

3. Work Classification Distribution

Figure 2 decomposes performance into its three constituent sub-scores as shares of total output per quarter. In Q1 2025, Maintenance accounted for 56% of classified output; in Q1 2026, 46%. Growth share moves from 30% to 36%; Fixes from 15% to 18%.

These are shifts in the composition of scored output. Directional interpretation should be supported by organization-specific context not captured here.

Figure 2: 100% Stacked Work Classification (Growth / Maintenance / Fixes)
Proportional allocation of SWE output by work classification per quarter

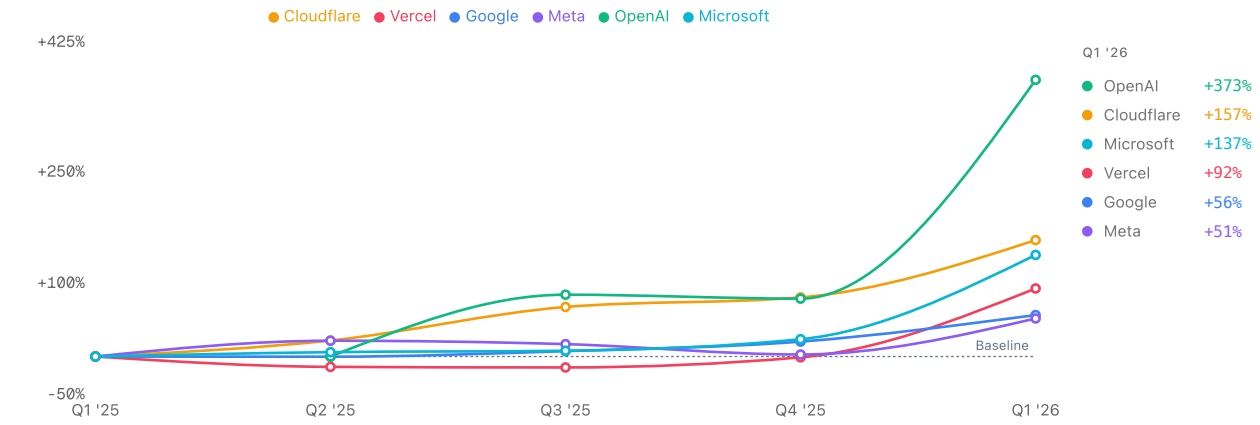


4. Per-Organization Comparison

Figure 3a traces each organization's quarter-by-quarter change in mean performance per qualifying SWE, rebased to 0% at its first qualifying quarter. The shared axis makes magnitudes comparable across organizations; the shape of each line separates steady drift from late-window inflections.

Figure 3b decomposes that same trajectory into its work classification. Each panel is a 100% stacked view of Growth / Maintenance / Fixes per quarter for one organization, so two orgs with similar headline deltas can be distinguished by how the underlying mix shifted.

Figure 3a: Per-Org Performance Trajectory per SWE
% change in mean performance per qualifying SWE, vs each org's first qualifying quarter

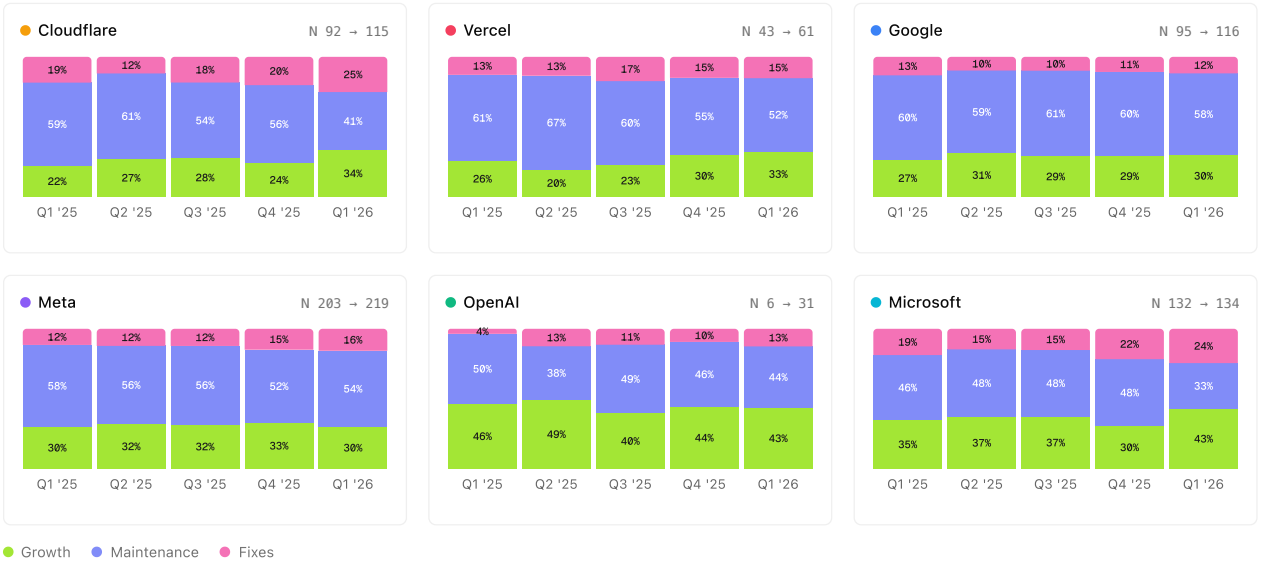


Each line is rebased to 0% at that organization's first qualifying quarter (the first quarter above the sample floor). Pre-baseline quarters are suppressed, so line length reflects how long the org has been in the sample — OpenAI's line covers four quarters of data against the other organizations' five, which amplifies its visual slope on the shared axis. Endpoints match the first → last deltas in the table below.

Figure 3b: Per-Org Work Classification (Growth / Maintenance / Fixes)

100% stacked composition of scored output per quarter, by organization

navigara



Organization	Metric	Q1 '25	Q2 '25	Q3 '25	Q4 '25	Q1 '26
Cloudflare	Performance	baseline	+22%	+67%	+80%	+157%
	Growth	baseline	+49%	+107%	+95%	+287%
	Maintenance	baseline	+25%	+52%	+68%	+79%
	Fixes	baseline	−22%	+66%	+97%	+251%
	N (SWEs)	92	102	113	113	115
Vercel	Performance	baseline	−14%	−15%	−1%	+92%
	Growth	baseline	−34%	−24%	+15%	+142%
	Maintenance	baseline	−6%	−17%	−11%	+63%
	Fixes	baseline	−10%	+15%	+15%	+127%
	N (SWEs)	43	52	61	62	61
Google	Performance	baseline	0%	+7%	+20%	+56%
	Growth	baseline	+17%	+17%	+33%	+76%
	Maintenance	baseline	−2%	+9%	+19%	+51%
	Fixes	baseline	−26%	−20%	−2%	+39%
	N (SWEs)	95	115	120	126	116
Meta	Performance	baseline	+21%	+17%	+3%	+51%
	Growth	baseline	+29%	+23%	+12%	+51%
	Maintenance	baseline	+16%	+12%	−8%	+40%
	Fixes	baseline	+28%	+24%	+33%	+106%
	N (SWEs)	203	220	230	230	219
OpenAI	Performance	–	baseline	+83%	+78%	+373%
	Growth	–	baseline	+47%	+60%	+317%
	Maintenance	–	baseline	+135%	+116%	+444%
	Fixes	–	baseline	+67%	+34%	+377%
	N (SWEs)	6	12	22	29	31
Microsoft	Performance	baseline	+6%	+8%	+23%	+137%
	Growth	baseline	+13%	+16%	+8%	+192%
	Maintenance	baseline	+10%	+10%	+27%	+66%
	Fixes	baseline	−17%	−13%	+43%	+209%
	N (SWEs)	132	137	137	134	134

5. Commit Volume & Per-Commit Performance

Beyond organizational variance (Section 4), the aggregate decomposes into frequency and intensity: commits per SWE and per-commit performance. Over the study period, average commits per qualifying SWE per quarter moved by **+35%** relative to Q1 2025, while average performance per commit moved by **+51%**. Shaded bands are 95% bootstrap CIs over SWEs.

The two series describe different properties. A rise in commits per SWE with stable per-commit performance is consistent with more granular commit practices or with real performance gains. A rise in per-commit performance with stable commits is consistent with larger or more complex changes per commit. The report describes which moved without claiming why.

Note: the two changes do not compound to the headline. The headline is a mean across SWEs of per-SWE performance, while the two series here are aggregate commit-weighted means. Different math, different number. The mean of products and the product of means differ in general, so (1 + commits/SWE Δ) × (1 + performance/commit Δ) is not expected to equal the Section 2 headline.

Figure 4a: Commits per qualifying SWE — change vs Q1 2025

Percentage change in distinct merged commits per SWE per quarter | shaded band = 95% bootstrap CI

navigara

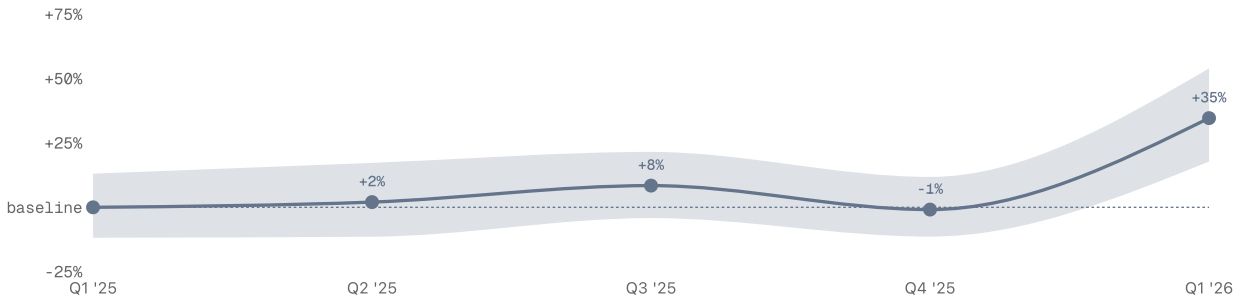
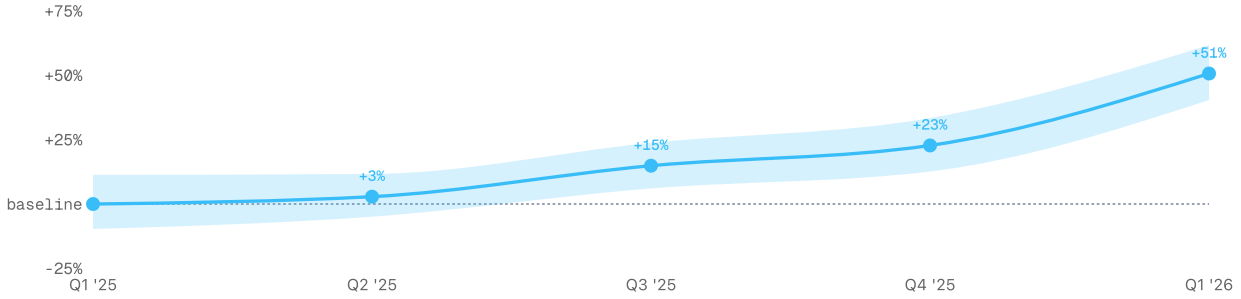


Figure 4b: Performance per commit — change vs Q1 2025

Per-commit performance averaged over commits within each SWE, then across SWEs | percentage change | 95% bootstrap CI

navigara



6. Methodology

Per-commit scoring engine. For each merged commit, the engine produces three sub-scores. Each sub-score is assembled per file and summed across the commit. The per-file score begins with a *context complexity* signal derived from the structural properties of the change, scaled by an *engagement multiplier* that accounts for the ratio of surrounding context complexity to the complexity of the change itself — targeted modifications in complex areas score higher than equivalent changes in trivial code. Several decay and amplification factors are then applied. A *similarity dampener* reduces credit for changes that are structurally similar to existing code, such as mechanical refactors or copy-paste patterns. A *blame decay* factor discounts changes that overwrite very recent work by the same author; this signal fades over a short business-day window so that revisiting older code is scored normally. A *copy decay* factor reduces the score for commits where a high proportion of added lines are duplicated from elsewhere in the codebase. For Fixes specifically, a *waste multiplier* amplifies the score based on how long the original code existed, whether the fix targets another author's code, and how frequently the affected area has been modified recently. Machine-learning components are used to tune thresholds and coefficients within this structure; a large-language-model classifier resolves ambiguous work classifications where pattern-based signals are insufficient. The structure of the score itself is deterministic.

Report-layer aggregation. This report sums the three per-commit sub-scores into a single scalar metric per commit, the *Engineering Throughput Value* (**ETV**). Commits are attributed to a contributor via the commit's primary author; contributors are attributed to organizations via the repositories they commit to. Per-SWE, per-quarter ETV is the sum of sub-scores across the SWE's qualifying commits in that quarter. Per-quarter per-org ETV is the mean across qualifying SWEs. The cross-org aggregate is a *developer-weighted mean*: every qualifying SWE contributes one observation per quarter, with each SWE weighted equally regardless of organization size. Figures in this report label the scalar as "performance" for readability; the formal definition is ETV.

Contributor inclusion. A contributor qualifies as a SWE in a given quarter if Navigara's contributor-role classifier assigns them the Software Engineer role (distinct from Data Scientist, SRE, Documentation, or other roles) and they have recorded commit activity in at least 10 weeks of the measurement window. Bots are excluded by a pattern match on email and display name (dependabot, renovate, github-actions, service accounts, etc.).

Sample floor. Per-organization results are reported only when the qualifying SWE cohort for that organization in a given quarter reaches at least **20** SWEs. Quarters below the floor are suppressed from per-org trajectories and excluded from the cross-org aggregate for that quarter.

Uncertainty. 95% confidence intervals are computed by bootstrap over SWEs (1,000 iterations, seeded for reproducibility): within each quarter, the set of qualifying SWEs is resampled with replacement and the mean is recomputed; the 2.5th and 97.5th percentiles of the resulting distribution are reported as the CI. Figures 3a and 3b display point estimates only; per-quarter CIs are available in the underlying data and, for the aggregate cohort, are rendered as shaded bands in Figures 1a and 1b. Cells with very small cohorts produce wide intervals that correctly reflect low precision.

Fixed-panel cohort. The fixed panel is the intersection of qualifying SWEs across every quarter of the reporting window (Q1 2025–Q1 2026). OpenAI contributes no SWEs to the fixed panel because its Q1 2025 cohort fell below the sample floor. The fixed panel removes cohort composition as a source of trend, at the cost of introducing survivorship bias toward continuously-active SWEs.

Organization selection. The six organizations studied were not sampled randomly. Three — *Cloudflare*, *Vercel*, and *OpenAI* — were selected because each makes frequent public claims about AI productivity gains in its engineering organization. The other three — *Google*, *Meta*, and *Microsoft* — were selected as incumbent organizations of substantially larger scale with strong public reputations for engineering talent density.

OpenAI entry quarter. OpenAI's Q1 2025 qualifying SWE cohort had 6 SWEs, below this edition's sample floor. OpenAI is therefore reported beginning Q2 2025 and is excluded from the Q1 2025 cross-org aggregate. Its Q2 2025–Q1 2026 trajectory reflects a rapid ramp in public-repo engineering activity and is labeled accordingly.

Temporal alignment. Quarters follow the standard calendar definition (Q1 = Jan–Mar, Q2 = Apr–Jun, Q3 = Jul–Sep, Q4 = Oct–Dec). Commits are attributed to the quarter in which they were merged to the default branch, not the quarter in which they were authored. Long-lived feature branches land at their merge quarter, which can concentrate multi-sprint work into a single reporting interval.

Window context. The measurement window (Q1 2025–Q1 2026) coincides with broad production adoption of AI coding assistants — code-generation tools, inline completion, and agentic workflows — across the organizations in this sample. The study does not attempt to establish which share of the observed rise, if any, is attributable to that adoption.

Limitations. Findings reflect commits to the specific public repositories listed in Appendix A. The report does not observe private-repository activity, code review depth, incident response, planning, mentorship, or any engineering contribution not captured as a merged commit to an in-scope repo. Cross-org comparisons are descriptive; differences in organizational structure, product maturity, monorepo versus polyrepo workflow, squash-merge policy, and public-vs-private code mix are not controlled.

Appendix A. Repositories Analyzed

The repositories listed below form the codebase sample for this report. Metrics are computed over merged commits to each repository's default branch within the reporting window. Selection rationale and sample-floor treatment are documented in Section 6 (Methodology); this appendix enumerates the repositories.

Cloudflare

14 repositories

cloudflare/agents	cloudflare/api-schemas	cloudflare/cloudflare-docs
cloudflare/cloudflared	cloudflare/containers	cloudflare/foundations
cloudflare/lol-html	cloudflare/moltworker	cloudflare/pingora
cloudflare/sandbox-sdk	cloudflare/telescope	cloudflare/workerd
cloudflare/workers-rs	cloudflare/workers-sdk	

Vercel

10 repositories

vercel/ai	vercel/commerce	vercel/next-forge
vercel/next.js	vercel/sandbox	vercel/sdk
vercel/swr	vercel/turborepo	vercel/vercel
vercel/workflow		

Google

12 repositories

google/adk-go	google/adk-python	google/flatbuffers
google/go-github	google/guava	google/perfetto
google/skia	google/syzkaller	google/zerocopy
googleapis/google-cloud-go	googleapis/google-cloud-python	googleapis/python-genai

Meta

10 repositories

facebook/buck2	facebook/docusaurus	facebook/fboss
facebook/fbthrift	facebook/folly	facebook/fresco
facebook/hermes	facebook/pyrefly	facebook/react
facebook/react-native		

OpenAI

8 repositories

openai/codex	openai/openai-agents-js	openai/openai-agents-python
openai/openai-dotnet	openai/openai-go	openai/openai-node
openai/openai-python	openai/plugins	

Microsoft

12 repositories

microsoft/Agents-for-net	microsoft/autogen	microsoft/DeepSpeed
microsoft/fluentui	microsoft/FluidFramework	microsoft/markitdown
microsoft/playwright	microsoft/PowerToys	microsoft/semantic-kernel
microsoft/terminal	microsoft/TypeScript	microsoft/vscode

About the Authors

Peter Malina

Peter has spent 14+ years scaling engineering platforms, leading large engineering organizations, and optimizing how software gets shipped at scale. Most recently he led the platform organization at Kiwi.com, serving tens of millions of customers monthly. He is building the measurement layer that makes engineering work legible to the rest of the organization.

Jirka Bachel

Jirka has 18+ years of technical leadership across Silicon Valley and Prague. Third time as CTO, having architected critical products for Fortune 500 companies and global scaleups. Previously Lead Developer at Seznam.cz, building a browser for 1.5M monthly users. He started Navigara because engineering managers are the only function in the building without a performance layer.

Acknowledgments. Writing and editing by the Navigara team.

