The Open Source Engineering Performance Report

Navigara's Commit-Level Analysis Across Cloudflare, Vercel, OpenAI, Google, Meta, and Microsoft · Q1 2025 – Q2 2026

Published 2026-09-07 · Data sourced from Navigara Knowledge Graph metrics

Peter Malina, Jirka Bachel

Corresponding author: research@navigara.com

> **Why this report exists.** Scored output per engineer rose +150% across this sample in six quarters. So any sense of what good looks like that formed before 2025 is calibrated to a level that no longer holds. No organization can see that from inside its own history. Size does not settle it either. Similar-sized organizations here differ severalfold in output per engineer, and very differently sized ones match. What follows is one reference point, measured the same way for every organization in it.

705	+1,031	1,736
real engineers	**capacity, not headcount**	**needed at the opening level**
qualifying SWEs in Q2 2026, counted per organization	engineers of output gained without hiring them; +766 excluding OpenAI	opening-era engineers required to ship Q2 2026 output

Engineering capacity added since each organization's own opening reading, as the public index reports it, from shipped output rather than AI usage. OpenAI supplies 26% of the total on an opening window of 6 engineers, the widest multiple in the study, which is why the figure without it is given alongside. Section 5.2 carries the detail.

1. Abstract

One finding leads this edition: **the quarter-over-quarter performance change is not statistically distinguishable from zero.** Mean ETV per qualifying SWE moved +17.5% between Q1 2026 and Q2 2026 on the open cohort, with a 95% interval of [-0.7%, +38.8%] that contains zero, and +3.5% on a fixed panel of the same 388 SWEs. The year-over-year reading is not ambiguous in the same way: +130% open and +82% fixed, both intervals clear of zero. That distinction is load-bearing for how the rest of this report should be read: the aggregate supports a year-over-year statement about scored output per engineer and does not support a quarter-over-quarter one.

The two comparisons. Year over year places Q2 2026 against Q2 2025, removing quarter-specific seasonality; quarter over quarter places it against Q1 2026, the shortest interval at which this dataset can detect a change in direction. Each runs on two cohorts: the open cohort, every SWE qualifying in the quarter, and a fixed panel of 388 SWEs present in all six, which removes composition as a source of change at the cost of survivorship bias. Across the full window the open-cohort change is +150% ([+111%, +195%]).

What is measured. Performance is Engineering Throughput Value (ETV), the scalar sum of per-commit scores for the complexity, intent, context, and lasting value of each merged commit; the scoring factors are documented in full at research.navigara.com/methodology. A contributor is a SWE when Navigara's role classifier assigns that role and they have commit activity in at least 10 weeks of the 78-week window, tested once over the whole window; they join a quarter's cohort by authoring at least one scored commit in it. Findings are descriptive and correlational: the report makes no causal claim about why any pattern occurred.

The work mix. Work is reported on the five categories the scorer emits. Q2 2026 output splits 34.9% Features, 27.5% Tests, 17.5% Maintenance, 14.5% Fixes, and 5.6% Docs. Classification is file-scoped, so a test or doc file scores into Tests or Docs even when it lands in the same commit as the feature it accompanies. Tests is the quarter's largest mover at +3.7pp and now the second-largest category of scored output. Maintenance's share falls in every quarter of the window, from 28.4% to 17.5%, while absolute maintenance output per SWE per month rose +54% over the same span: a smaller share of a larger total, not less maintenance. Section 4 carries the quarterly shares and both endpoint changes for all five categories.

Composition, frequency, and intensity. The qualifying cohort contracted from 793 SWEs to 699, the first quarter-over-quarter contraction in the window, the case in which open-cohort and fixed-panel readings separate. Inside the aggregate the components moved opposite ways: commits per qualifying SWE fell -11.2% while ETV per commit rose +32.4%; year over year both rose, by +14.9% and +100.4%. They multiply to the headline exactly, all three being pooled quarterly ratios, as Section 6 sets out.

Snapshot. Re-scoring, role reclassification, and late-arriving commit attribution move historical quarters between snapshots, so every quarter is recomputed here rather than carried forward: the Q1 2025 to Q1 2026 open-cohort change on this snapshot is +113%. Comparisons should be drawn within one snapshot.

Interpretation. The window coincides with broad production adoption of AI coding assistants across the organizations in this sample. The year-over-year picture remains one of substantial growth in scored output per engineer that survives holding the SWE population constant. The quarter-over-quarter picture is flatter than the Q1 2026 trajectory would have projected, and a single quarter of deceleration in a noisy series is not yet evidence of a plateau. The study does not quantify what share of the level shift, if any, is attributable to AI adoption.

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 388 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)

navigara

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

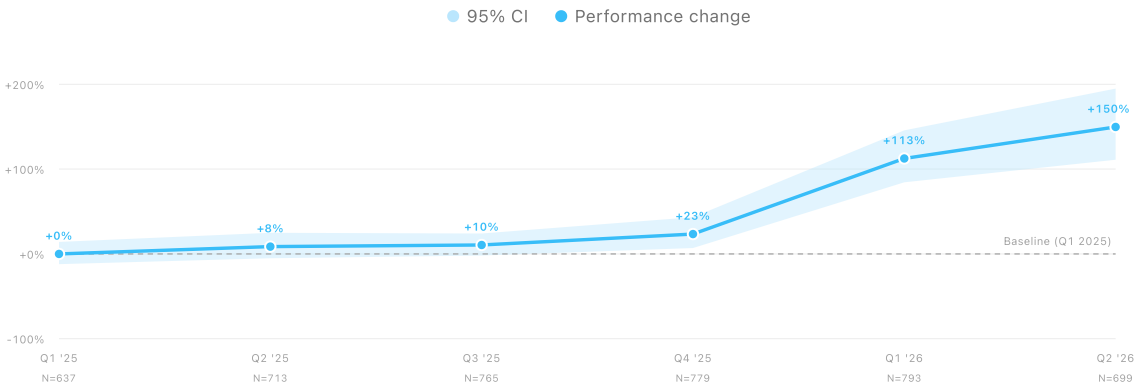
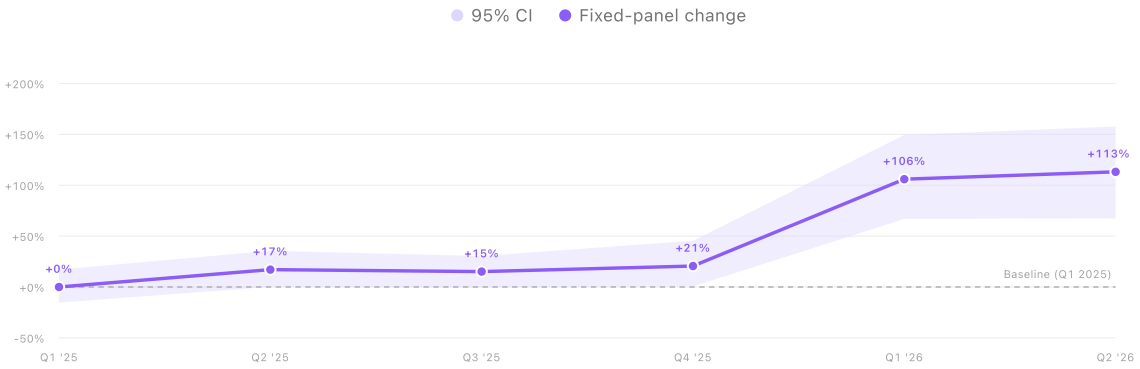


Figure 1b: Performance change for the fixed-panel cohort (388 SWEs)

navigara

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 7.

Reading Figures 1a and 1b together. Across the first four quarters the two series track each other closely and both are close to flat. The level shift arrives in Q1 2026 and is present in both, which is the pattern that rules out pure composition: it survives holding the SWE population constant. Q2 2026 is where the two series separate. The open cohort adds +17.5% while the fixed panel adds +3.5%, so roughly 20% of the quarterly move is preserved once the population is held fixed. Over the year-over-year interval the correspondence is much stronger, +130% open against +82% fixed.

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 work 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 and the other stabilizing it, and the single number will not separate them. This motivates the decomposition in Sections 4 and 5.

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. It matters particularly here, because the headline quarter-over-quarter interval [-0.7%, +38.8%] contains zero. Directional interpretation is best anchored to the year-over-year endpoints, and the shaded CI bands should be consulted before treating any single inter-quarter movement as meaningful.

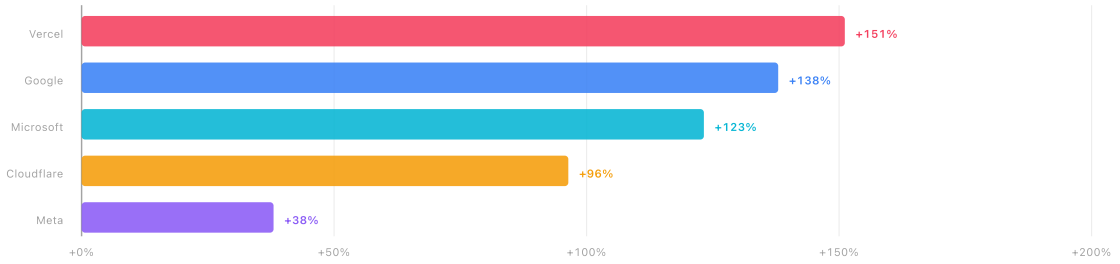
3. The Two Comparisons

Sections 1 and 2 established that the two readings disagree in magnitude. This section reports both at the organization level, and which of the two orderings survives a re-scoring of the same quarters. One does and one does not.

Figure 2a: Year over year, Q2 2025 to Q2 2026

navigara

Percentage change in mean ETV per qualifying SWE, by organization



OpenAI, held out of the scale above

Its Q2 2025 cohort held 18 qualifying SWEs, below the 20-SWE floor, so the year-over-year change against that quarter, **+757%**, is measured from a baseline this report does not consider reportable on its own. Rebased to Q3 2025, its first at-floor quarter (25 SWEs), the change to Q2 2026 is **+230%**. The second figure is the one to read; the first is shown so the omission from the chart is explicit.

Bars cover the 5 organizations whose Q2 2025 baseline cohort is at or above the 20-SWE sample floor. OpenAI is excluded from the bars for that reason and reported beneath them on both baselines.

Two of the three orderings reproduce. The quarterly one does not. Section 7.1 records that this dataset re-scores closed quarters between snapshots, which is testable on the ordering itself: rank these organizations on two panels built thirteen days apart. Each test covers the organizations that comparison can rank, so the denominators differ: 6 for the cumulative and quarterly tests, where every organization qualifies, and 5 year over year, because OpenAI's Q2 2025 baseline is below the floor and Figure 2a holds it out. All three are Kendall's tau, and each is quoted with the exact one-sided probability of matching it by reshuffling, enumerated over every permutation.

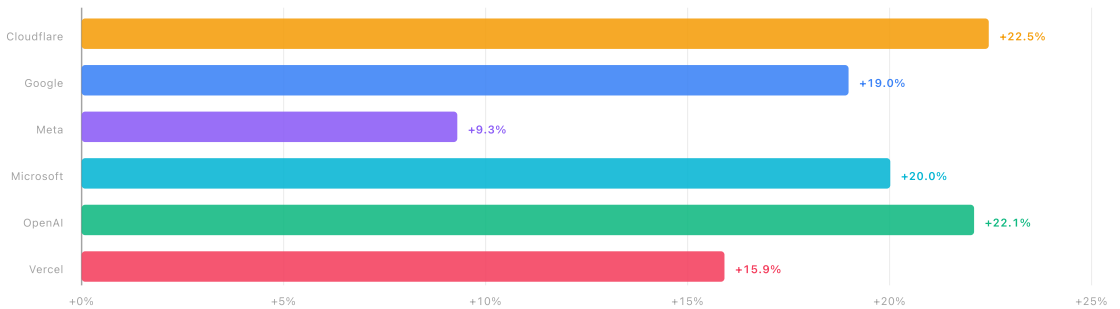
The cumulative endpoint Section 5.1 reports is the strongest: 13 of 15 pairwise orderings survive, +0.73, $p = 0.028$, failing only on Cloudflare against Microsoft, and Vercel against Google. Year over year 8 of 10 survive, +0.60, which at 5 organizations is $p = 0.117$ and does not clear conventional significance on its own. The sharper statement is that both failures are the same organization: set Microsoft aside and all 6 orderings among the remaining 4 reproduce exactly, $p = 0.042$. The annual comparison is therefore stable except in Microsoft's position, rather than uniformly noisy. Quarter over quarter 8 of 15 survive, +0.07, $p = 0.500$: chance exactly, not as a figure of speech. Cloudflare moves from last (+3.6%) to first (+22.5%) on the same closed quarter.

The pattern is not arbitrary. A one-quarter delta divides two adjacent quarters that are both still being re-scored, so error in either end moves it; a cumulative change and a year-over-year change divide by a baseline quarter that has had a year or more to settle, and only the numerator moves. The orderings that survive the test are the ones in Figure 2a, Figure 4a and the endpoint column of Section 5.1. Figure 4b is not among them: it reports composition rather than an ordering, its shares divide by total ETV in the same quarter so re-scoring moves numerator and denominator together, and that different failure mode was not tested here. Figure 2b is printed for completeness, is ordered alphabetically rather than by value, and **should not be read as a ranking**: no organization's position in it is reproducible. That is the organization-level counterpart of the abstract's finding on the aggregate.

Figure 2b: Quarter over quarter, Q1 2026 to Q2 2026

navigara

Percentage change in mean ETV per qualifying SWE, by organization



Every organization in the sample is positive on both horizons. On comparable baselines the annual changes run from +38% to +151%, but the top of Figure 2a is not a winner: Microsoft placed 1st of 5 on the earlier panel and 3rd here, and both failed pairs involve it. The bottom of that range is steadier: Meta is last on both panels. Every organization clears the sample floor in both Q1 2026 and Q2 2026, so Figure 2b covers all six, which is a statement about coverage and not about precision.

Mean ETV per qualifying SWE per month	Q2 '25	Q1 '26	Q2 '26	YoY	QoQ
Aggregate, open cohort	**0.99**	**1.94**	**2.27**	**+130.2%**	**+17.5%**
Aggregate, fixed panel (388)	**1.33**	**2.34**	**2.42**	**+82.2%**	**+3.5%**
Cloudflare	0.53	0.85	1.04	**+96.4%**	**+22.5%**
Vercel	1.27	2.76	3.20	**+151.1%**	**+15.9%**
OpenAI*	0.96	6.73	8.21	**+756.6%**	**+22.1%**
Google	0.89	1.78	2.11	**+138.0%**	**+19.0%**
Meta	0.88	1.11	1.21	**+38.0%**	**+9.3%**
Microsoft	1.43	2.67	3.20	**+123.2%**	**+20.0%**

** Q2 2025 cohort below the 20-SWE sample floor, so that row's YoY cell is measured from a below-floor baseline; the rebased figure is given under Figure 2a. Aggregate rows are developer-weighted means over every qualifying SWE, so they are not the average of the rows below them.*

4. Work Classification Distribution

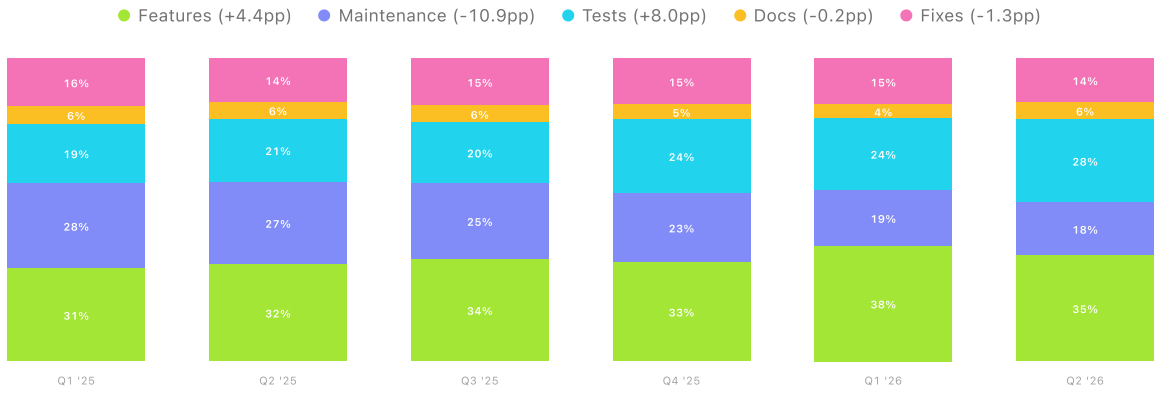
Figure 3 decomposes performance into its five constituent sub-scores as shares of total output per quarter. Classification is file-scoped, so a commit that adds a feature, extends its test suite, and updates a README splits across Features, Tests, and Docs rather than booking whole to one category.

Over the full window. The mix rotates out of Maintenance and into Tests: Maintenance falls from 28.4% of scored output to 17.5% and Tests rises from 19.5% to 27.5%, while Features, Docs, and Fixes each move by at most 4.4pp. All five endpoints are in the table below.

Figure 3: 100% Stacked Work Classification (Features / Maintenance / Tests / Docs / Fixes)

navigara

Proportional allocation of SWE output by work classification per quarter



Docs runs between 4.3% and 5.8% of output across the window, so its slice is labelled only where the band is tall enough; the table below carries all five values.

Q1 2026 to Q2 2026. The largest single movement of the quarter is Tests at +3.7 percentage points, against Features at -3.2. The remaining three categories each move by at most 1.4pp; the QoQ and YoY columns of the table below carry all five on both horizons.

Share of scored output	Q1 '25	Q2 '25	Q3 '25	Q4 '25	Q1 '26	Q2 '26	YoY	QoQ
Features	30.5%	31.9%	33.7%	32.6%	38.0%	34.9%	**+3.0pp**	**−3.2pp**
Maintenance	28.4%	27.3%	25.2%	22.9%	18.5%	17.5%	**−9.8pp**	**−1.0pp**
Tests	19.5%	20.7%	19.9%	24.3%	23.8%	27.5%	**+6.8pp**	**+3.7pp**
Docs	5.8%	5.6%	5.7%	4.9%	4.3%	5.6%	**−0.0pp**	**+1.3pp**
Fixes	15.8%	14.4%	15.5%	15.2%	15.3%	14.5%	**+0.1pp**	**−0.9pp**

Shares are computed over commits, as sum of the classification sub-score divided by sum of total ETV within the quarter. YoY and QoQ columns are changes in percentage points, against Q2 2025 and Q1 2026 respectively. Quarterly percentages are rounded independently and may not sum to 100.

These are shifts in composition, not statements about strategic intent, and a falling share is not falling output. Maintenance runs the clearest case: its share fell -10.9pp across the window while total output per SWE rose +150%, so absolute maintenance output per SWE per month rose +54%, from 0.26 to 0.40 ETV (+47% year over year). Less of a larger total, not less work.

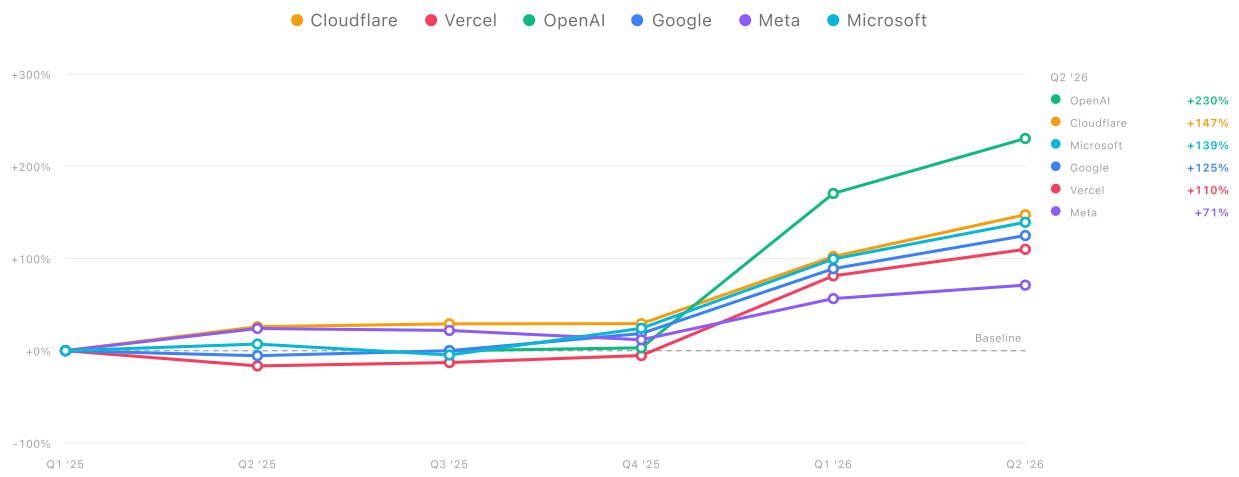
5. Per-Organization Comparison

Figure 4a 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 4b, on the following page, decomposes the same trajectory into its work classification, so two organizations with similar headline deltas can be distinguished by how the underlying mix shifted.

Figure 4a: Per-Org Performance Trajectory per SWE

navigara

% 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 at or above the 20-SWE sample floor. Pre-baseline quarters are suppressed, so line length reflects how long the org has been at floor. OpenAI's line covers four quarters against the other organizations' six, which amplifies its visual slope on the shared axis. Endpoints match the first to last deltas in the Section 5.1 detail table.

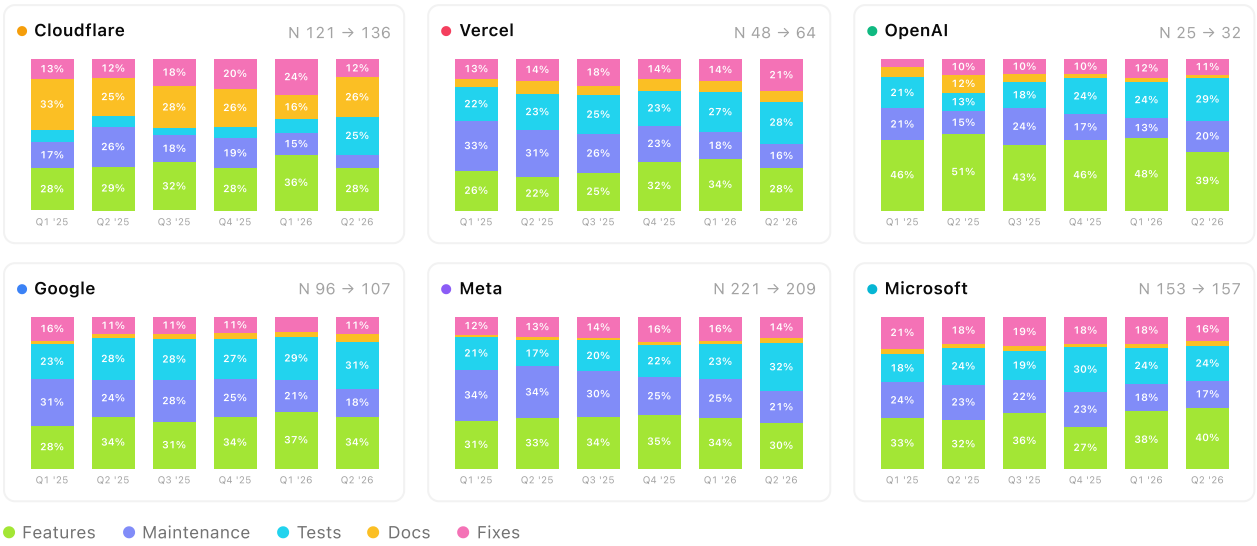
OpenAI enters the per-organization view at Q3 2025. Its Q1 2025 cohort held 6 qualifying SWEs and its Q2 2025 cohort 18, both below the 20-SWE floor. The subsequent quarters reflect a continuing ramp in OpenAI's public-repository engineering activity and should be read as a growing cohort rather than a steady-state baseline; its cohort reached 38 SWEs in Q1 2026 before easing to 32 in Q2 2026.

Cohort size fell between Q1 2026 and Q2 2026 at all six organizations, from 802 qualifying SWEs to 705 in total, and this is the composition effect that separates Figures 1a and 1b. Where a cohort contracts and mean output per remaining SWE rises, the aggregate can move without any individual engineer changing output. The fixed-panel series in Figure 1b is the control for exactly this, and it is the reason the quarter-over-quarter headline in this edition is reported with a caution the year-over-year headline does not require. Note that a single engineer committing to more than one organization's repositories is counted in each, so these per-org totals exceed the 699 distinct SWEs in the Q2 2026 aggregate.

How settled is that contraction? It is measured at the least settled point in the window, so it is worth asking directly. Comparing the two panels thirteen days apart described in Section 7.1, every quarter's qualifying cohort gained members except the most recent: Q1 2025 +1, Q2 2025 +5, Q3 2025 +11, Q4 2025 +11, Q1 2026 +12, Q2 2026 +0. Quarters keep accruing members for a year or more after they close, and Q2 2026 has not begun to. The contraction reported here should therefore be read as an **upper bound**, with one condition attached. Both quarters it spans are still moving, and they move it in opposite directions: members added to Q1 2026 widen the contraction, members added to Q2 2026 narrow it. The bound holds only if Q2 2026 accrues more from here than Q1 2026 has left. That does not follow from decay: across the three most recent closed quarters the accrual is flat, Q3 2025 +11, Q4 2025 +11 and Q1 2026 +12, and only the two oldest quarters have slowed. It rests on the weaker point that Q2 2026 has accrued nothing at all yet and is the younger quarter, so it has further to go. This snapshot cannot confirm it. The report states the direction rather than a correction, and a later edition can settle it.

Figure 4b: Per-Org Work Classification (Features / Maintenance / Tests / Docs / Fixes)

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



Slices below roughly 10% of a column carry no inline label; the per-quarter values are tabulated on the following page. Docs is the smallest of the five categories at every organization except Cloudflare, and that is sample construction rather than practice: cloudflare/cloudflare-docs is the only repository in this report that is itself a documentation corpus, and file-scoped classification books its output as Docs. The Cloudflare panel therefore runs 15.9% to 33.3% Docs in every quarter of the window, not only in Q2 2026.

Reading the panels against Figure 4a separates organizations whose headline gain came from feature work from those whose gain came from test and documentation output. Cloudflare is the clearest case in this window: its Docs share is the largest in the sample at 25.9% of Q2 2026 output, against 5.6% for the aggregate, and its Tests share moves from 9.0% to 25.2% across the final quarter while its Maintenance share falls to 8.5%.

Why Cloudflare's panel looks different. The Docs figure is a sample effect and should not be read as a documentation culture. Cloudflare's repository set includes cloudflare/cloudflare-docs (Appendix A), the only repository in this report that is itself a documentation corpus rather than a codebase with documentation in it; classification is file-scoped, so that repository's output books almost entirely as Docs. The elevated share is present in every quarter of the window, from 15.9% to 33.3%, which is the signature of sample composition rather than of a change in behaviour, and it is 4.6 times the aggregate in Q2 2026. Its Tests figure has the mirror property: the +653% reported in Section 5.1 is measured from the smallest base in the sample, 0.03 ETV per SWE per month in Q1 2025 against 0.26 in Q2 2026, so the multiple is large while the absolute move is not. Neither figure is comparable with organizations whose documentation and tests live inside their product repositories.

5.1 Per-Organization Detail

Each organization contributes seven rows: total performance, the five work categories, and the qualifying SWE count. Values are percentage change against that organization's first qualifying quarter.

Organization	Metric	Q1 '25	Q2 '25	Q3 '25	Q4 '25	Q1 '26	Q2 '26
Cloudflare	Performance	baseline	+26%	+29%	+29%	+102%	+147%
	Features	baseline	+31%	+47%	+31%	+165%	+151%
	Maintenance	baseline	+88%	+36%	+44%	+72%	+21%
	Tests	baseline	+13%	−33%	+19%	+121%	+653%
	Docs	baseline	−4%	+9%	−1%	−3%	+92%
	Fixes	baseline	+17%	+73%	+88%	+259%	+127%
	N (SWEs)	121	134	145	140	148	136
Vercel	Performance	baseline	−16%	−13%	−5%	+81%	+110%
	Features	baseline	−29%	−17%	+18%	+136%	+126%
	Maintenance	baseline	−22%	−32%	−33%	−3%	+3%
	Tests	baseline	−11%	+1%	+1%	+119%	+164%
	Docs	baseline	+37%	−1%	+26%	+135%	+157%
	Fixes	baseline	−10%	+17%	−3%	+99%	+239%
	N (SWEs)	48	55	62	67	70	64
OpenAI	Performance	–	–	baseline	+3%	+170%	+230%
	Features	–	–	baseline	+11%	+199%	+197%
	Maintenance	–	–	baseline	−28%	+49%	+174%
	Tests	–	–	baseline	+40%	+263%	+442%
	Docs	–	–	baseline	−47%	+62%	+9%
	Fixes	–	–	baseline	+3%	+233%	+249%
	N (SWEs)	6	18	25	32	38	32
Google	Performance	baseline	−6%	0%	+19%	+89%	+125%
	Features	baseline	+15%	+9%	+44%	+151%	+175%
	Maintenance	baseline	−26%	−11%	−5%	+27%	+32%
	Tests	baseline	+14%	+20%	+38%	+139%	+201%
	Docs	baseline	+2%	+32%	+91%	+165%	+436%
	Fixes	baseline	−32%	−29%	−20%	+16%	+62%
	N (SWEs)	96	119	126	136	130	107
Meta	Performance	baseline	+24%	+22%	+12%	+57%	+71%
	Features	baseline	+33%	+33%	+27%	+69%	+65%
	Maintenance	baseline	+22%	+7%	−19%	+15%	+4%
	Tests	baseline	+2%	+18%	+13%	+73%	+158%
	Docs	baseline	+70%	+28%	+31%	+79%	+244%
	Fixes	baseline	+39%	+42%	+57%	+112%	+101%
	N (SWEs)	221	238	245	244	243	209
Microsoft	Performance	baseline	+7%	−5%	+24%	+99%	+139%
	Features	baseline	+4%	+5%	+3%	+128%	+192%
	Maintenance	baseline	+4%	−13%	+20%	+48%	+72%
	Tests	baseline	+42%	+1%	+101%	+162%	+206%
	Docs	baseline	−28%	−29%	−38%	+34%	+74%
	Fixes	baseline	−8%	−11%	+6%	+69%	+85%
	N (SWEs)	153	162	170	166	173	157

Values are percentage change against each organization's first qualifying quarter, which is shown as baseline. Quarters below the 20-SWE sample floor are suppressed. A category starting from a small base produces a very large percentage change on a modest absolute move; the clearest cases here are Tests at Cloudflare (+653%), Tests at OpenAI (+442%), Docs at Google (+436%). Read those against the share panels in Figure 4b rather than as standalone growth rates.

5.2 Engineering Capacity

Sections 5 and 5.1 report change against each organization's own first qualifying quarter. This section reads that same distance as capacity: the number of opening-era engineers it would take to ship what each organization ships now. Every subject divides by its own opening reading, the 90-day window ending 2025-04-01, the earliest window the index publishes and one that closes before the adoption period this report covers; the commit history reaches 90 days further back, so it is a complete window. Each organization starts at 1.00x on the same window as the others, and the figure below is what it has added since.

These are the published index's figures. This section reports what 500.navigara.com serves for each of these organizations rather than recomputing capacity on this report's own basis, so that the two cannot print different numbers for the same organization. The cohorts are not a difference between them: the index applies the same two tests Section 7 defines, so every headcount below is the same qualifying cohort as the matching Q2 2026 column in Section 5.1, and the render fails if they diverge. Both sections are built from one snapshot, for the same reason.

What differs is the divisor, and only the divisor. This section divides by a rolling 90-day window; Sections 5 and 5.1 divide by a quarter. So a multiple here is not the percentage change there: holding the sample fixed at the same six organizations, this report's pooled Q1 2025 quarter of 0.911 gives +1,056 engineers against the +1,031 above. Neither is a correction of the other. Read the table below against the org pages, and Sections 5 and 5.1 against each other.

705
real engineers
qualifying SWEs in Q2 2026, counted per organization

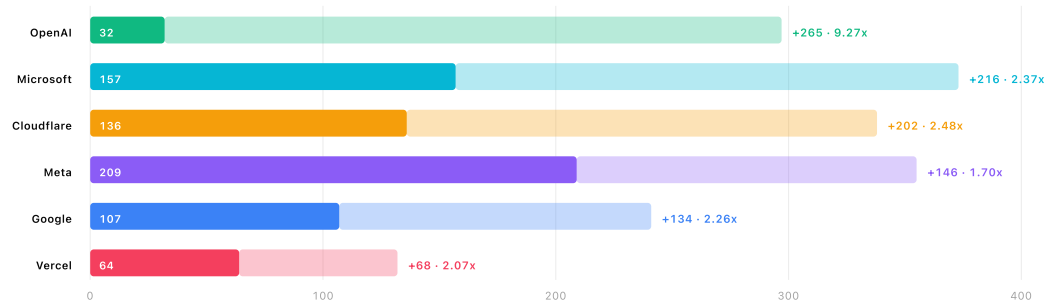
+1,031
capacity, not headcount
engineers of output gained without hiring them; +766 excluding OpenAI

1,736
needed at the opening level
opening-era engineers required to ship Q2 2026 output

Figure 4c: Engineering capacity added per organization

navigara

Q2 2026 qualifying SWEs (solid) and the capacity added since each organization's own opening reading (light) | engineers



Capacity added is the quarter's cohort multiplied by (multiple − 1), where the multiple is that organization's Q2 2026 ETV per SWE per month over its own opening reading. Measured from shipped output, not from AI usage.

Organization	SWEs	ETV / SWE / mo at opening	ETV / SWE / mo, Q2'26	vs own opening	Capacity added	Ships like
OpenAI*	32	0.89	8.21	9.27x	+265	297
Microsoft	157	1.35	3.20	2.37x	+216	373
Cloudflare	136	0.42	1.04	2.48x	+202	338
Meta	209	0.71	1.21	1.70x	+146	355
Google	107	0.94	2.11	2.26x	+134	241
Vercel	64	1.54	3.20	2.07x	+68	132
Six organizations	**705**				**+1,031**	**1,736**

Ships like is the row's cohort plus its capacity added: opening-era engineers needed to produce that organization's Q2 2026 output. The total is every row added up, marked rows included. OpenAI is the one row measured from an opening window below this report's 20-SWE reporting floor, so it carries the widest multiple here by construction and supplies +265 of the +1,031, or 26%. Without it the other five come to +766, which is the figure to quote if that divisor is not one you want to rest on. This total is not the index total, which pools seven organizations including a private team this report excludes throughout; per-organization capacity does not sum to the index in any case, since each row divides by its own opening and the index by its own. Every row is a point estimate: the bootstrap intervals quoted elsewhere are computed on the panel, and the index publishes none on a rolling window.

** Opening window below this report's 20-SWE sample floor: OpenAI's opening window holds 6 SWEs. The index's own floor on a usable opening is 3 contributors, which that window clears, so the index publishes the figure and this report prints it and holds it out. A divisor drawn from a small window moves further between syncs than the rest, making that row the least stable figure here. Holding it out rather than moving it to a later window keeps every organization on one opening; printing it rather than dropping it keeps the figure visible. Match this table to the Q2 2026 row of the quarterly table on an org page, not to the capacity tile above it: the tile is a trailing-90-day reading ending at the build date, so it belongs to no quarter and moves with every sync.*

6. Commit Volume and Per-Commit Performance

Beyond organizational variance, the aggregate decomposes into frequency and intensity: commits per SWE and per-commit performance. This decomposition is where the two comparison horizons diverge most clearly. Year over year both components rise, commits per qualifying SWE by +14.9% and performance per commit by +100.4%. Quarter over quarter they move in opposite directions, commits per SWE falling -11.2% while performance per commit rises +32.4%.

The Q2 2026 pattern is therefore entirely an intensity story, and the two series account for the headline between them: falling frequency times rising intensity is exactly the +17.5% of Section 2, as the note below sets out. Engineers in the qualifying cohort landed fewer merged commits than in Q1 2026 but each commit carried more scored complexity. That is consistent with larger or more structurally involved changes per commit, and equally consistent with a shift in commit-granularity practice such as heavier squashing before merge, which would raise ETV per commit and lower commits per SWE without any change in work done. That alternative is **not testable on this dataset**, and the report states so rather than leaving the hedge open: the commit records carry author, timestamp, repository, the five sub-scores and the merge and exclusion flags, and no diff size or pull-request linkage, so median files or lines per commit and commits per pull request cannot be computed at any layer. Separating the two readings requires those fields at ingestion. Until then the decomposition says which component moved and cannot say why.

Figure 5a: Commits per qualifying SWE, change vs Q1 2025

navigara

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

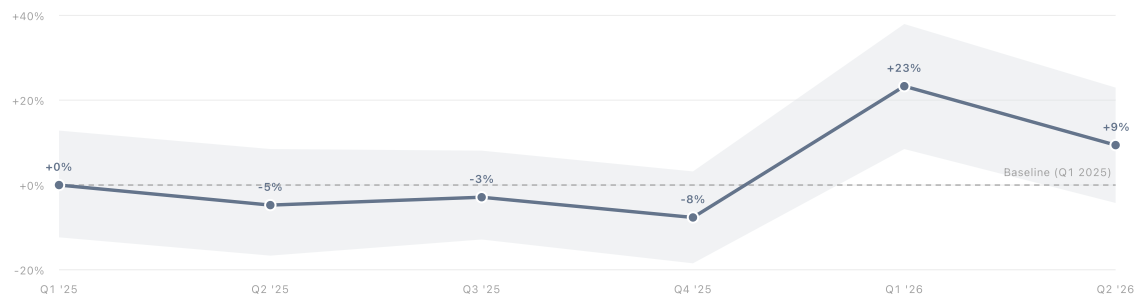
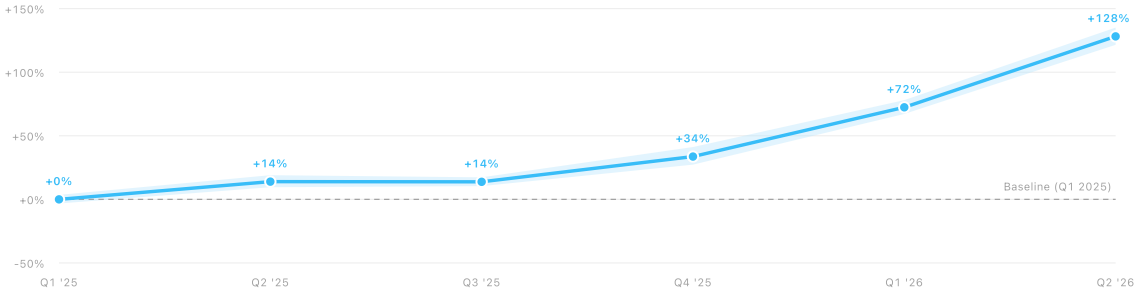


Figure 5b: Performance per commit, change vs Q1 2025

navigara

Per-commit performance averaged over commits | percentage change | 95% bootstrap CI



Note: the two components multiply to the headline exactly, because the decomposition is an identity rather than an approximation. All three statistics are pooled quarterly ratios over the same commit set: mean ETV per SWE per month is total ETV divided by the qualifying SWE count and by three, commits per SWE is total commits over that same count, and ETV per commit is total ETV over total commits. The commit count cancels. So (1 + commits/SWE change) × (1 + ETV/commit change) returns the Section 2 headline: 0.888 × 1.324 = 1.175 quarter over quarter, against a headline of 1.175, and 1.149 × 2.004 = 2.302 year over year, against 2.302. What the identity cannot do is attribute the movement. Both factors are cohort-level ratios, so either can move because the cohort changed rather than because any individual engineer did, which is the caution Section 3 and Figure 1b address.

7. Methodology

Per-commit scoring engine. For each merged commit, the engine produces five sub-scores, one per work category. Each sub-score is assembled per file and summed across the commit, so classification is file-scoped: a test file contributes to Tests and a documentation file to Docs regardless of what else the commit touches. 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, so 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 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 five per-commit sub-scores into a single scalar metric per commit, the Engineering Throughput Value (ETV). The sum is unchanged by the move from three categories to five. Commits are attributed to a contributor via the commit's primary author; contributors are attributed to organizations via the repositories they commit to, with each repository assigned to the organization contributing the most commits to it. Per-SWE, per-quarter ETV is the sum of sub-scores across the SWE's qualifying commits in that quarter. Per-quarter per-org performance is the mean across qualifying SWEs, expressed per calendar month by dividing the quarterly figure by three. The cross-org aggregate is a developer-weighted mean: every qualifying SWE contributes one observation per quarter, weighted equally regardless of organization size. Figures label the scalar as "performance" for readability; the formal definition is ETV.

Contributor inclusion. Two tests apply at two different scopes. The *role* test is evaluated once over the whole 78-week window, never per quarter: a contributor is a SWE if Navigara's contributor-role classifier assigns them the Software Engineer role (distinct from Data Scientist, SRE, Documentation, or other roles) and has recorded commit activity in at least 10 of those 78 weeks. The *cohort* test is evaluated per quarter: a SWE joins that quarter's qualifying cohort by authoring at least one scored commit to an in-scope repository in it. The cohort movement reported in Section 3 is therefore movement in the second test alone. Bots are excluded by a pattern match on email and display name (dependabot, renovate, github-actions, service accounts, and similar). 137,592 qualifying commits across 6 organizations enter the window.

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. Where a below-floor quarter is nonetheless used as a comparison baseline, as with OpenAI in Figure 2a, it is marked in place.

Uncertainty. 95% confidence intervals are computed by bootstrap (B = 1,000 iterations, seeded for reproducibility) over the unit of analysis for each metric: commits for the classification shares and per-commit performance, SWEs for per-developer metrics. Within each quarter the sample is resampled with replacement and the statistic recomputed; the 2.5th and 97.5th percentiles of the resulting distribution are reported. Percentage-change intervals quoted in the text are *endpoint* intervals: the endpoint quarter's bootstrap bounds expressed against the point estimate of the baseline quarter. They therefore reflect sampling error in the endpoint only and understate total uncertainty in the change, since the baseline quarter carries its own sampling error. Figures 4a and 4b display point estimates only. Cells with 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 to Q2 2026) and contains 388 SWEs. OpenAI contributes few SWEs to it because its early cohorts fell below the sample floor. The panel removes cohort composition as a source of trend, at the cost of introducing survivorship bias toward continuously-active SWEs.

7.1 Study Design, Alignment, and Limitations

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. One further organization present in the underlying dataset is a private team and is excluded from every figure in this report.

Subject-matter confound. 21 of the 65 repositories in the sample (32%) are AI or agent SDKs, and they are not spread evenly: OpenAI 8 of 8, Microsoft 5 of 12, Vercel 3 of 9, Google 3 of 12, Cloudflare 2 of 14. Demand for that category grew over the same window this report measures, so for those repositories output per engineer and market growth are not separable here. It bears most on OpenAI, whose sample is entirely of that kind and whose cohort went from 6 SWEs to 38 and back to 32 across the window, a shape that is as consistent with staffing a launch as with any change in productivity. The report does not attempt to separate the two.

Temporal alignment. Quarters follow the standard calendar definition (Q1 = Jan to Mar, Q2 = Apr to Jun, Q3 = Jul to Sep, Q4 = Oct to Dec). Commits are attributed to the quarter in which they were authored. Long-lived feature branches therefore distribute across the quarters in which their commits were written rather than concentrating at merge. Q2 2026 closed 2026-07-01, 10 weeks before this snapshot was taken. Late-arriving commit ingestion is small at that remove, but re-scoring is not, and a closed quarter is not a settled quarter: between two snapshots of this dataset taken thirteen days apart, both more than eight weeks after the quarter closed, Q2 2026 ETV per SWE moved +11% pooled and +36% at Google, on identical cohorts and with commit counts differing by 1.3%. The cohorts are identical because Q2 2026 is the one quarter whose membership did not move between those panels, as Section 5 sets out; that is what isolates re-scoring here from the ingestion that is still adding members to every earlier quarter. That is the reason every figure in this report is recomputed on one snapshot and the restatement policy below is not a formality. Q3 2026 is in progress at publication and is excluded rather than reported as a partial quarter.

Restatement policy. Every quarter is recomputed on the snapshot this report was built from rather than carried forward. Commit re-scoring, contributor-role reclassification, and repository-scope changes move historical values between snapshots, so figures here should be read as a single internally consistent series and compared within it rather than line by line against separately published tables.

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 repository. Cross-org comparisons are descriptive; differences in organizational structure, product maturity, monorepo versus polyrepo workflow, squash-merge policy, and public-versus-private code mix are not controlled.

7.2 Sensitivity to the Scoring Weights

ETV sums the five sub-scores with equal weight. That is a modelling choice, and a metric assembled this way invites the question of what happens if it is assembled differently, so the headline is recomputed below under perturbations of those weights. Dropping a category sets its weight to zero; the Fixes row asks what happens if corrective work is counted against output rather than for it, which is the most contestable of the five.

Weighting	QoQ	YoY	Full window
As published, all weights 1.0	**+17.5%**	**+130.2%**	**+149.7%**
Drop Features	+23.5%	+120.2%	+134.1%
Drop Maintenance	+18.9%	+161.3%	+187.5%
Drop Tests	+11.8%	+110.5%	+124.9%
Drop Docs	+15.9%	+130.2%	+150.3%
Drop Fixes	+18.7%	+129.9%	+153.8%
Fixes counted against output (weight −1)	+20.5%	+129.6%	+159.6%
Tests and Docs at half weight	+14.0%	+121.2%	+138.6%

The year-over-year finding does not depend on the weighting: it ranges +111% to +161% across every variant above, and no variant changes its sign or its order of magnitude. The upper end is not an independent reading: it comes from dropping Maintenance, the category whose share falls furthest across the window in Section 4, so removing it mechanically steepens the series it was already diluting. The quarter-over-quarter figure ranges +11.8% to +23.5%, a factor of 2.0 between the extremes, which is a further reason it is not reported as a finding. What this cannot reach is the scorer's internal coefficients, the engagement multiplier and the decay and waste factors described in Section 7: those are applied upstream of anything this report observes, so the analysis perturbs how the five sub-scores are combined and not how each is produced. That is where most of the model risk sits, and closing it is scoped work rather than open-ended: it needs the scorer to emit per-file sub-scores before its multipliers are applied, or to accept a coefficient override and re-score the window, after which this table can be rebuilt against each perturbed run. Point estimates only: the bootstrap is not rerun per variant, so the range above is a spread of point estimates across weightings and carries no sampling uncertainty of its own. The interval on the published weighting is the one in Section 2.

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 7 (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

9 repositories

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

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

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

Microsoft

12 repositories

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

