

## Know it. Prove it. Improve it.

Sample insights for engineering leaders to expect from Hivel to measure the ROI of AI, built on Hivel's SURGE framework.

Figures are grounded in real, anonymized Hivel client data. Some values are illustrative, adjusted to present a clean sample of the reporting format.

## Know it

See what you actually have

S - Spend

U - Utilization

## Prove it

Show the impact in outcomes

G - Gains

## Improve it

Act on what you found

R - Recovery

E - Efficiency

## CONTENTS

KNOW IT - Spend + Utilization

PROVE IT - Gains

IMPROVE IT - Recovery + Efficiency

- 01 Cost per merged PR
- 02 Start with identity resolution
- 03 Adoption cohorts vs outcomes
- 04 Contribution is concentrated
- 05 AI output is compounding

- 06 AI code survival funnel
- 07 Team productivity trends
- 08 Impact on Speed, Quality & Throughput
- 09 Impact of AI on Quality
- 10 The real ROI

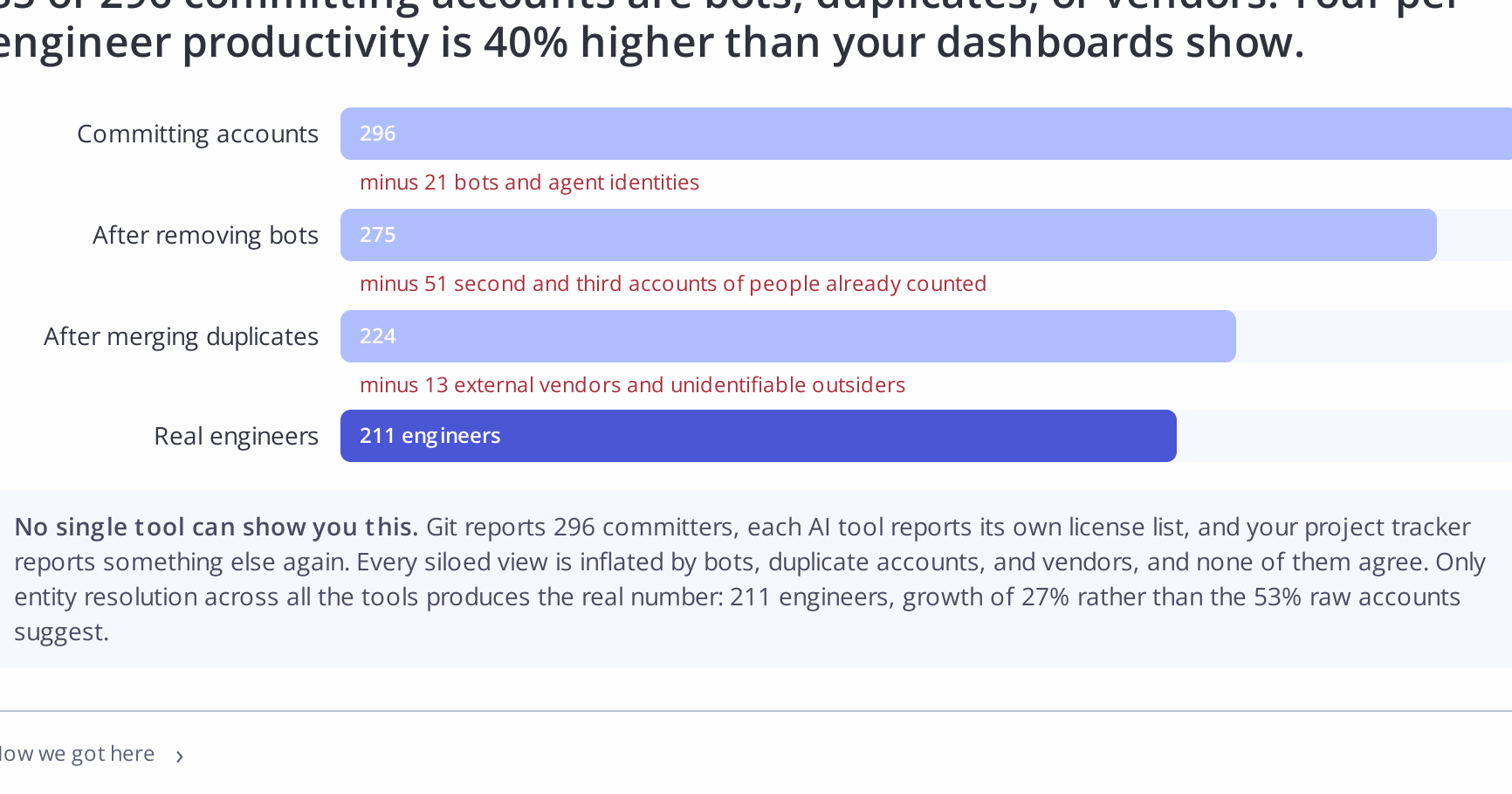
- 11 Redeploy the idle licenses
- 12 Where the bottleneck moved
- 13 Not reviewed by anyone
- 14 Reality of code review
- 15 Code review after the merge
- 16 The review capacity gap
- 17 Adoption vs impact quadrant

## Know it

SPEND + UTILIZATION - SEE WHAT YOU ACTUALLY HAVE

## INSIGHT 01

S - SPEND

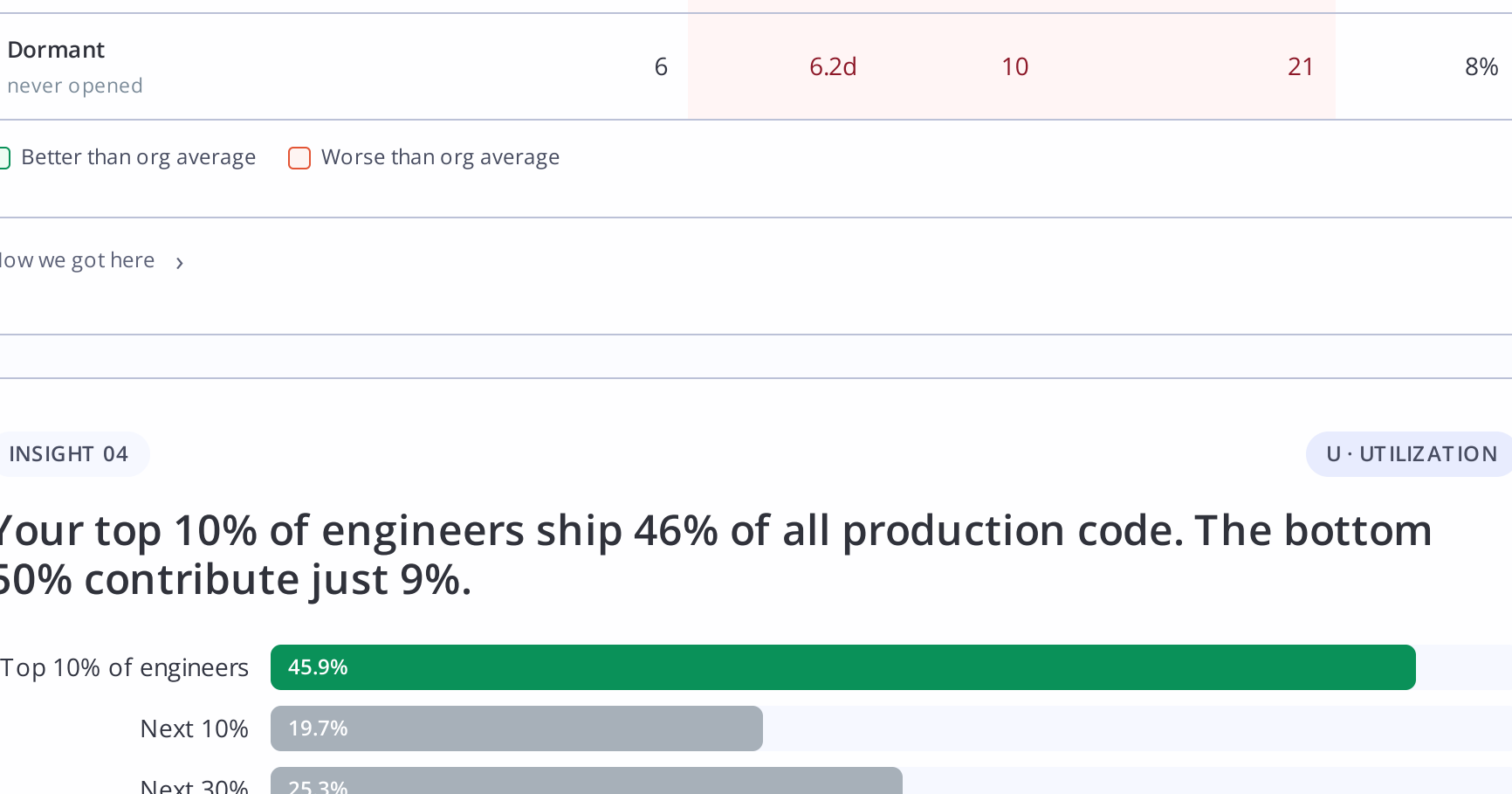
As adoption climbed from 22% to 71%, every new engineer made AI cheaper for the whole org. You're paying **42% less** per PR today than in January.

How we got here &gt;

## INSIGHT 02

U - UTILIZATION

85 of 296 committing accounts are bots, duplicates, or vendors. Your per-engineer productivity is 40% higher than your dashboards show.



No single tool can show you this. Git reports 296 committers, each AI tool reports its own license list, and your project tracker reports something else again. Every siloed view is inflated by bots, duplicate accounts, and vendors, and none of them agree. Only entity resolution across all the tools produces the real number: 211 engineers, growth of 27% rather than the 53% raw accounts suggest.

How we got here &gt;

## INSIGHT 03

U - UTILIZATION

Daily habit engineers ship in 3.4 days with 31 PRs. Dormant engineers take 6.2 days and ship half the work items.

| COHORT - Q2 '26, 63 WORKING DAYS        | ENGINEERS | CYCLE TIME | PRS / ENG | WORK ITEMS / ENG | REWORK % |
|-----------------------------------------|-----------|------------|-----------|------------------|----------|
| Daily habit<br>AI-active 41+ of 63 days | 112       | 3.4d       | 31        | 42               | 7%       |
| Occasional<br>11 to 40 days             | 56        | 4.1d       | 20        | 32               | 9%       |
| Rarely used<br>1 to 10 days             | 37        | 5.6d       | 13        | 21               | 8%       |
| Dormant<br>never opened                 | 6         | 6.2d       | 10        | 21               | 8%       |

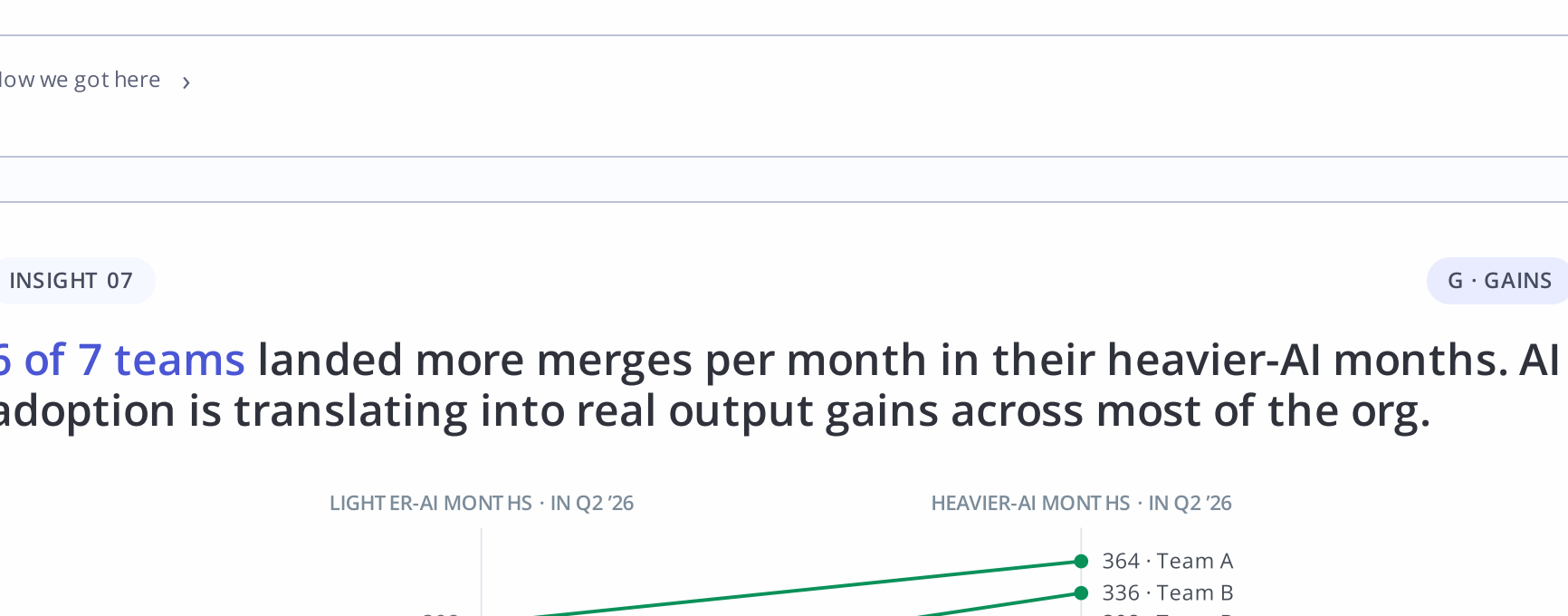
☒ Better than org average ☐ Worse than org average

How we got here &gt;

## INSIGHT 04

U - UTILIZATION

Your top 10% of engineers ship 46% of all production code. The bottom 50% contribute just 9%.

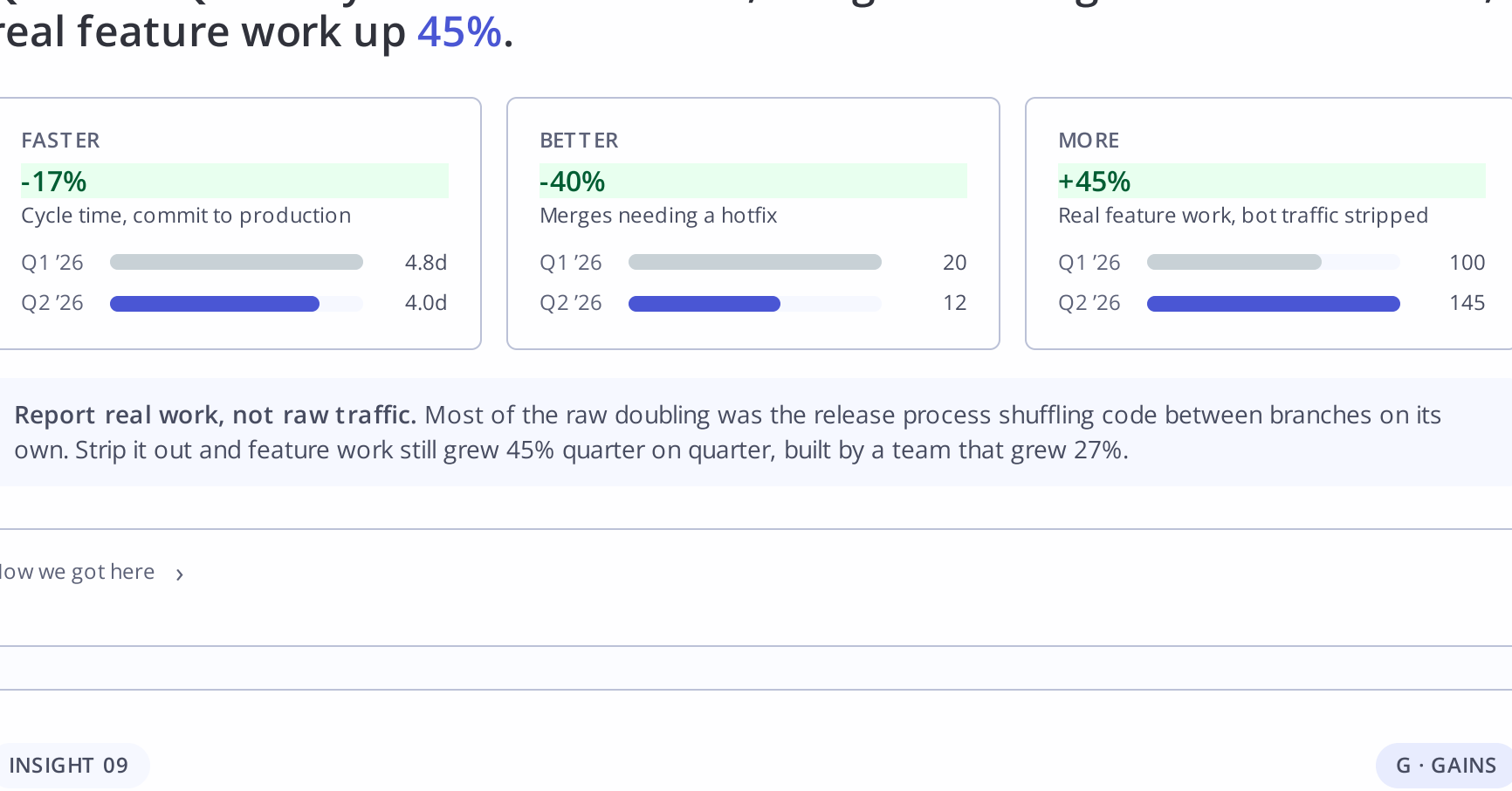


How we got here &gt;

## INSIGHT 05

U - UTILIZATION

AI code in production grew 2.7x in 6 months while PRs merged grew 1.7x. AI adoption is picking up pace and driving more output every month.



Participation widened too. Engineers active in AI in a given month rose from 137 in Jan '26 to about 168 in Jun '26, spread across teams. Q2 '26 ends at its peak, so Q3 '26 starts from a higher base.

How we got here &gt;

## Prove it

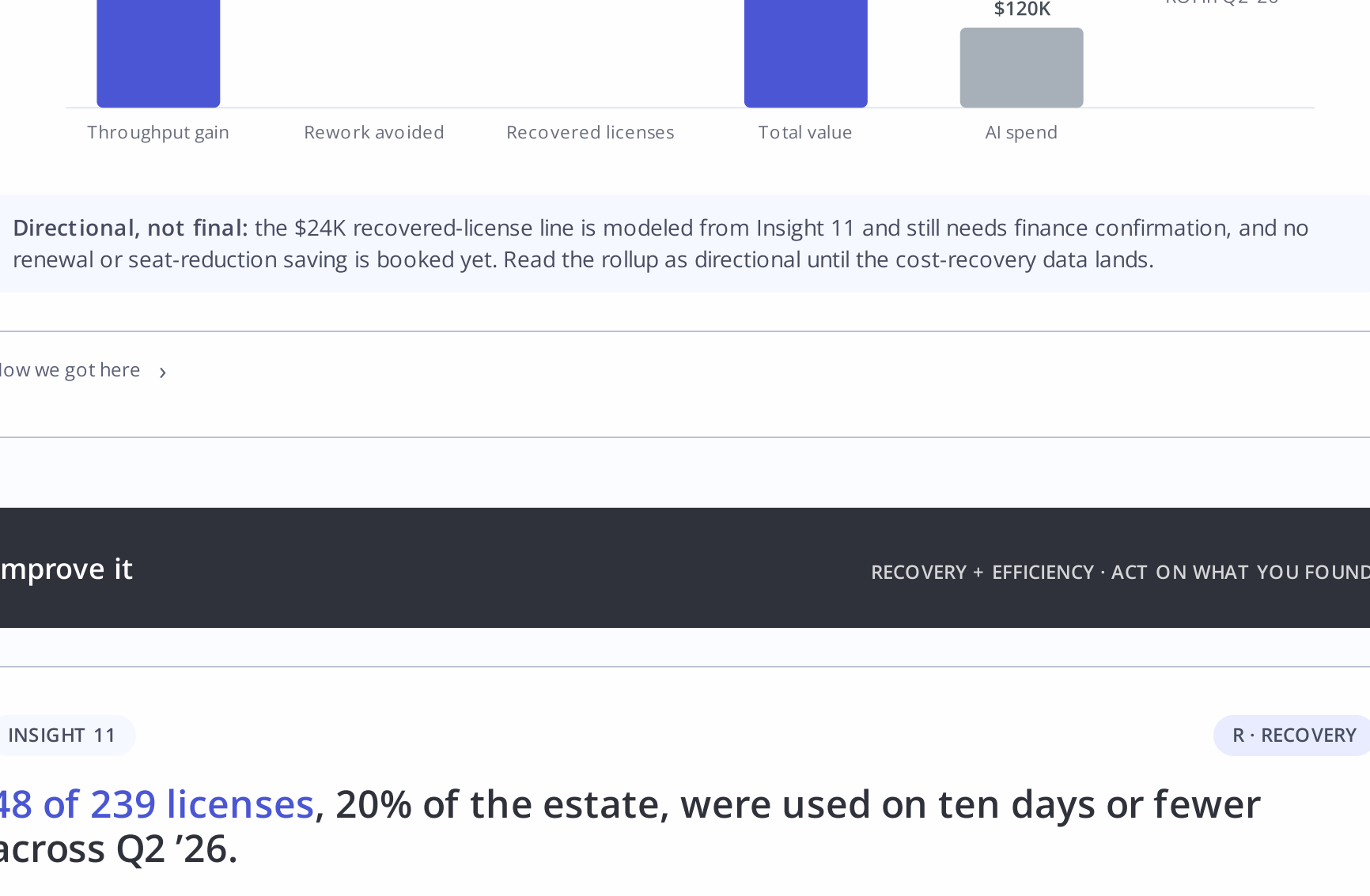
GAINS - SHOW THE IMPACT IN OUTCOMES, NOT ACTIVITY

## INSIGHT 06

G - GAINS

THE JOURNEY OF AI CODE FROM GENERATION TO PRODUCTION

For every 100 lines of AI code generated, only 39 make it to production. 61% is filtered out across acceptance, review, and post-merge reverts.



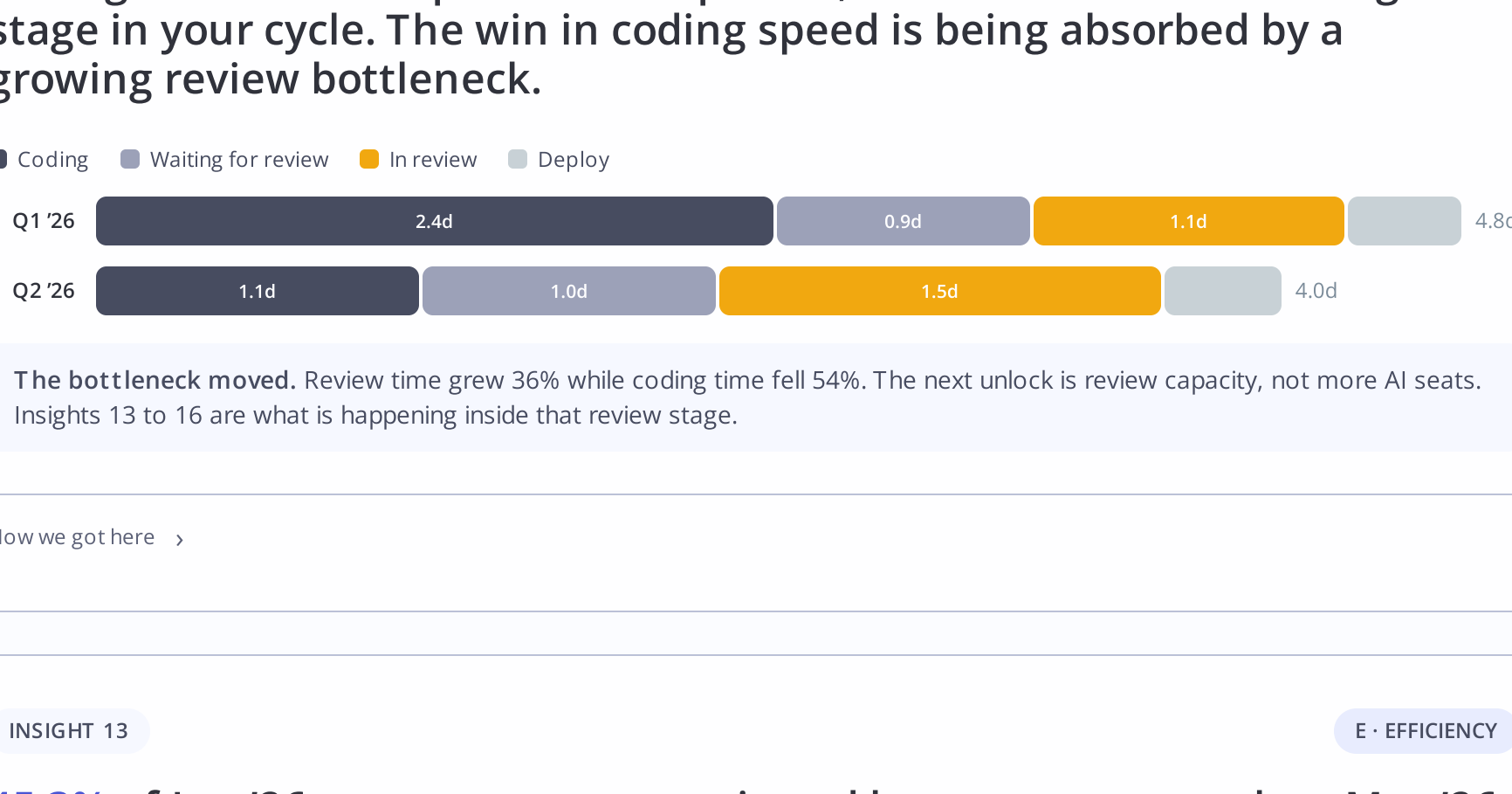
The funnel widens as teams learn. The acceptance step shown above rose from 51% in Q1 '26 to 62% in Q2 '26, and survival to 30 days in production rose from 31% to 39% over the same two quarters.

How we got here &gt;

## INSIGHT 07

G - GAINS

6 of 7 teams landed more merges per month in their heavier-AI months. AI adoption is translating into real output gains across most of the org.



Merges landed per month - each line is one team

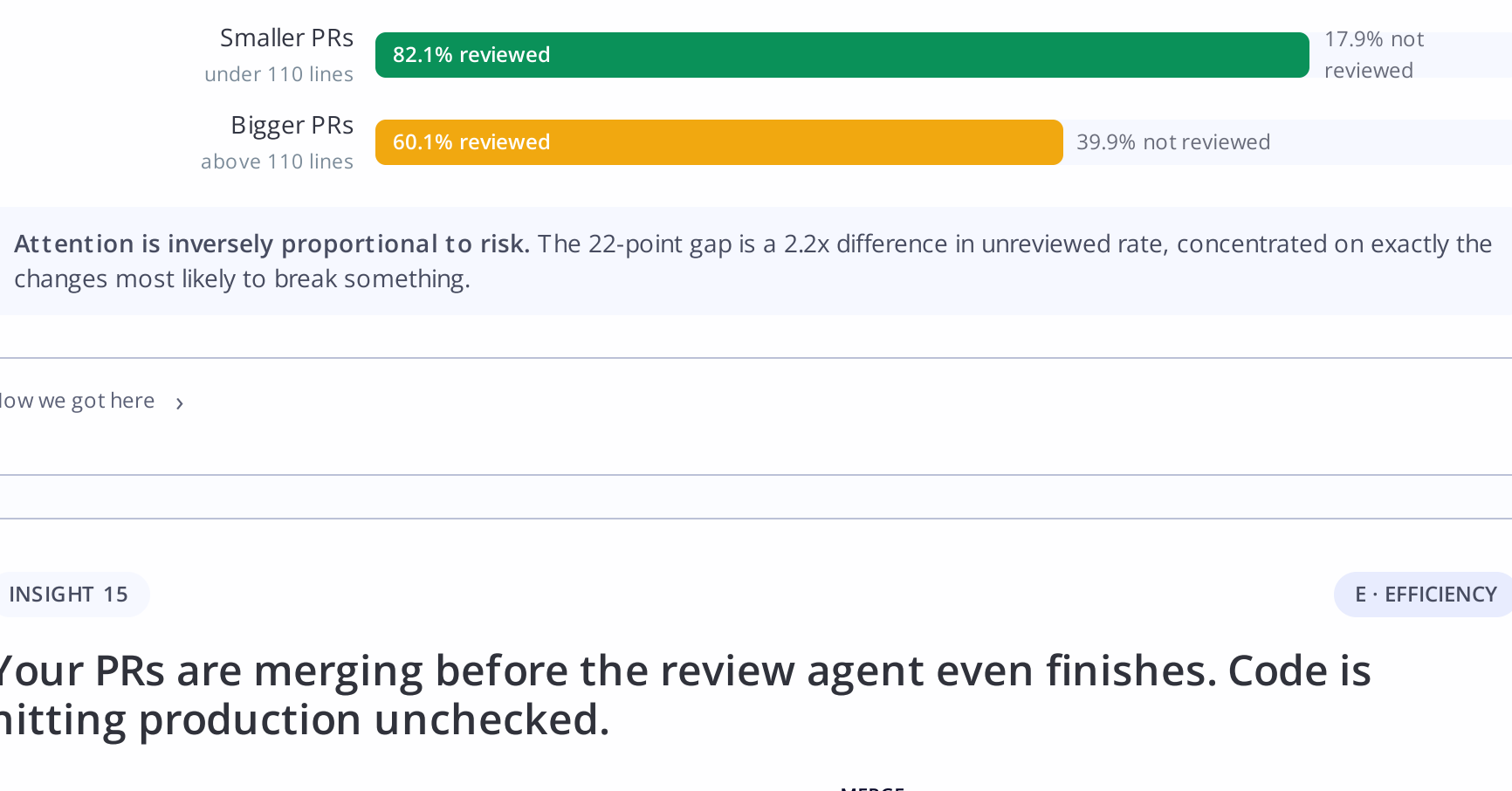
☒ Higher cadence in heavier-AI months ☐ Flat or lower

How we got here &gt;

## INSIGHT 08

G - GAINS

Q2 '26 vs Q1 '26: cycle time down 17%, merges needing a hotfix down 40%, real feature work up 45%.



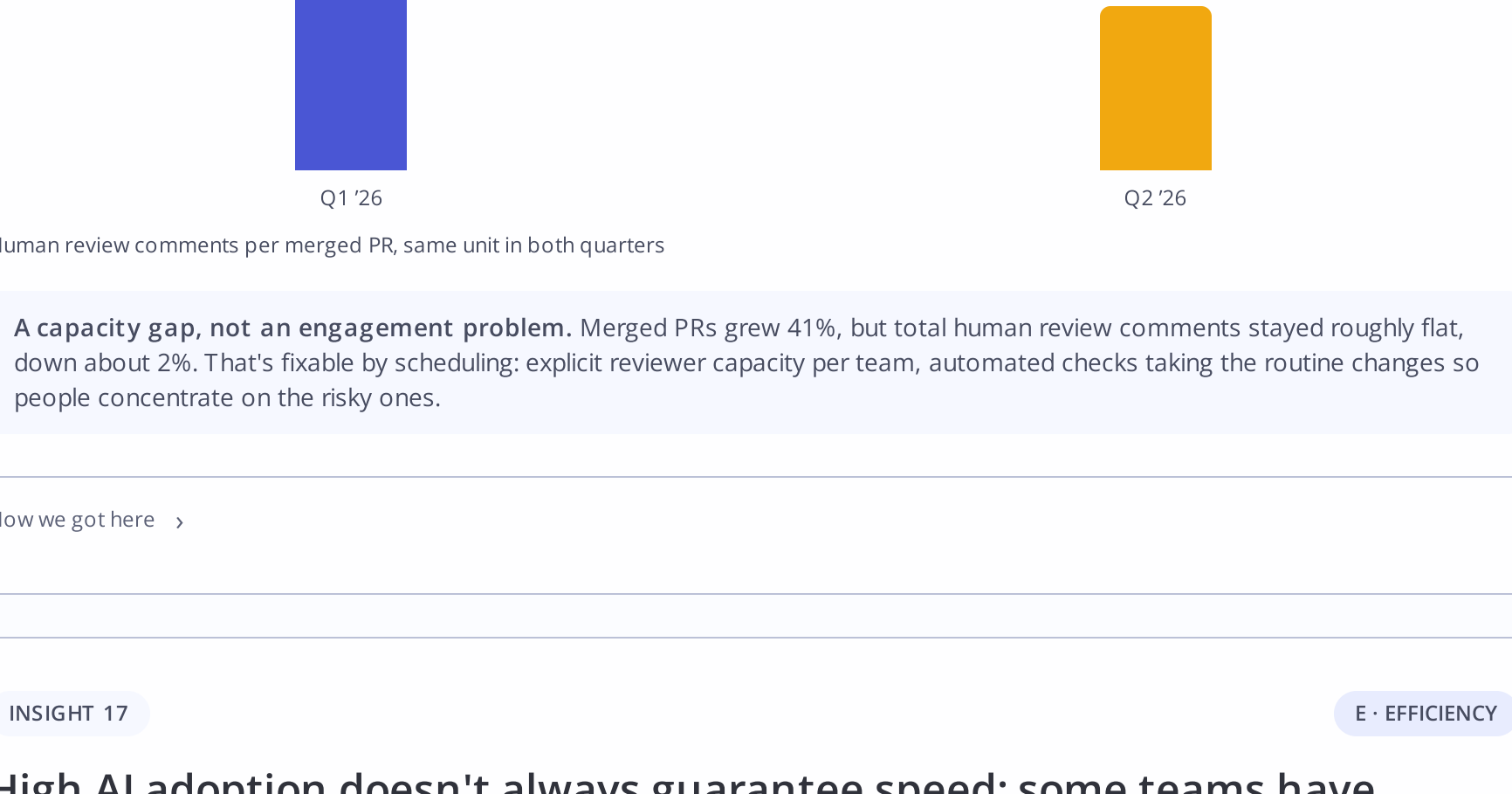
Report real work, not raw traffic. Most of the raw doubling was the release process shuffling code between branches on its own. Strip it out and feature work still grew 45% quarter on quarter, built by a team that grew 27%.

How we got here &gt;

## INSIGHT 09

G - GAINS

AI-assisted code isn't just faster to write, it performs better across all 4 quality metrics: less rework, less refactor churn, fewer hotfixes, and faster recovery.



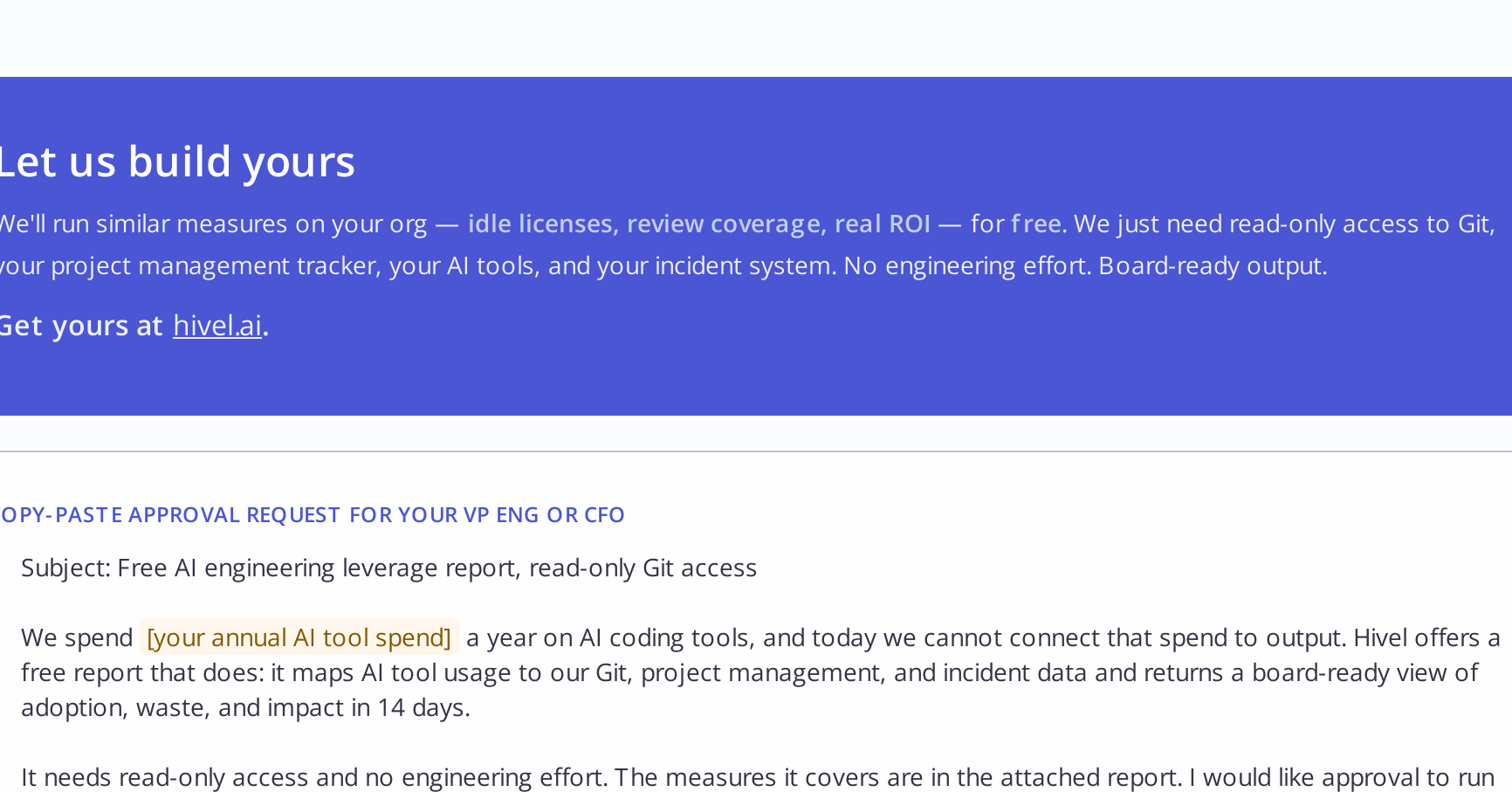
One thing to take away: on all four quality measures, AI-assisted code is at least as good as the rest. The speed gains in Insight 08 did not come out of quality.

How we got here &gt;

## INSIGHT 10

G - GAINS

Your AI tools returned \$374K in value against \$120K in spend. For every dollar invested in AI, you got 3.1x back.



Directional, not final: the \$24K recovered-license line is modeled from Insight 11 and still needs finance confirmation, and no renewal or seat-reduction saving is booked yet. Read the rollup as directional until the cost-recovery data lands.

How we got here &gt;

## Improve it

RECOVERY + EFFICIENCY - ACT ON WHAT YOU FOUND

## INSIGHT 11

R - RECOVERY

48 of 239 licenses, 20% of the estate, were used on ten days or fewer across Q2 '26.



The clearest cost lever in the report. Redeploy the 48 dormant and rarely-used licenses before renewal, and reclaim the duplicates: 28 engineers hold two or more paid AI licenses and use one.

How we got here &gt;

## INSIGHT 12

E - EFFICIENCY

Coding time fell 54% quarter over quarter, but review is now the longest stage in your cycle. The win in coding speed is being absorbed by a growing review bottleneck.



The bottleneck moved. Review time grew 36% while coding time fell 54%. The next unlock is review capacity, not more AI seats. Insights 13 to 16 are what is happening inside that review stage.

How we got here &gt;

## INSIGHT 13

E - EFFICIENCY

45.3% of Jun '26 merges were not reviewed by anyone, worse than May '26 (41.1%).



The trap: agent comments count as reviewed in raw dashboard numbers, so review coverage looks fine while people stop reading the code. Insight 16 shows why: capacity, not engagement.

How we got here &gt;

## INSIGHT 14

E - EFFICIENCY

The bigger the change, the less likely it gets reviewed by a human. Your largest PRs are 2.2x more likely to skip review than small ones.



Attention is inversely proportional to risk. The 22-point gap is a 2.2x difference in unreviewed rate, concentrated on exactly the changes most likely to break something.

How we got here &gt;

## INSIGHT 15

E - EFFICIENCY

Your PRs are merging before the review agent even finishes. Code is hitting production unchecked.



One decision: run the reviewer the moment a PR opens and hold the merge until it finishes. It takes 104 seconds.

How we got here &gt;

## INSIGHT 16

E - EFFICIENCY

Review comments per merged PR fell from 1.3 to 0.9 between Q1 '26 and Q2 '26.



Human review comments per merged PR, same unit in both quarters

A capacity gap, not an engagement problem. Merged PRs grew 41%, but total human review comments stayed roughly flat, down about 2%. That's fixable by scheduling: explicit reviewer capacity per team, automated checks taking the routine changes so people concentrate on the risky ones.

How we got here &gt;

## INSIGHT 17

E - EFFICIENCY

High AI adoption doesn't always guarantee speed: some teams have converted it into real gains, others haven't.



Adoption isn't the win condition: some teams turned heavy AI use into real speed gains, one adopted just as hard and gained nothing.

How we got here &gt;

## Let us build yours

We'll run similar measures on your org — idle licenses, review coverage, real ROI — for free. We just need read-only access to Git, your project management tracker, your AI tools, and your incident system. No engineering effort. Board-ready output.

Get yours at [hivel.ai](#).

## COPY-PASTE APPROVAL REQUEST FOR YOUR VP ENG OR CPO

Subject: Free AI engineering leverage report, read-only Git access

We spend [your annual AI tool spend] a year on AI coding tools, and today we cannot connect that spend to output. Hivel offers a free report that does: it maps AI tool usage to our Git, project management, and incident data and returns a board-ready view of adoption, waste, and impact in 14 days.

It needs read-only access and no engineering effort. The measures it covers are in the attached report. I would like approval to run it.

Report basis: Q2 '26 (Apr to Jun 2026), compared against Q1 '26 only. Trends across the two quarters are shown monthly. Figures are grounded in real, anonymized Hivel client data; some values are illustrative, adjusted to present a clean sample of the reporting format.