INTELLIGENCE RESEARCH · MAY 2026

Token Maxxing Is Making Engineering Teams Slower

An analysis of 9M+ pull requests across 2,444 engineering organizations. More AI spend. More code volume. More production failures. We measured where AI engineering effort actually goes, how code review has responded to 2.6× volume growth, and why the reactive work treadmill keeps accelerating. The findings: $0.82 of every AI dollar is consumed before a single feature reaches users.

SCROLL

PRs analyzed

1M+

Across 2,444 engineering organizations

Reactive work, median org

44%

Bugs + maintenance — nearly half of all capacity

Reports vs PR growth

3.7× vs. 2.6×

Failures compounding faster than output

01

$0.18 reaches users

For every dollar a team spends on AI coding tools, just 18 cents becomes shipped product. The other **82 cents** is consumed by the maintenance cycle those same tools accelerate, not because engineers are slow, but because there is **no closed loop between production reality and the code being written.**

Reactive engineering $0.44 · Code rework $0.27 · Review friction $0.11 · Shipped product $0.18

Proportional allocation of all engineering spend, platform average. The green band is the only part that reaches users.

Reactive · P75 / P90

$0.62 / $0.76

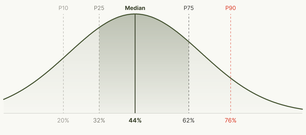
Shipped · P75 / P90

$0.10 / $0.06

02

Half of all engineering output is reactive.

At the median organization, **44% of every PR** is reactive — fixing existing code or keeping systems running. The distribution has a long tail: at the 90th percentile, more than three-quarters of all engineering effort produces no net-new product.



Share of engineering output classified as reactive (bugs + maintenance), by organization percentile. The shaded band is the interquartile range. Wins in spend accelerate volume in **both** sides — features and maintenance alike — so the tail only thickens.

03

1 in 4

lines written each week is **thrown away** before the week closes — not planned refactoring, but code that **did not survive the sprint it was written in.** The AI generates from local context, never from production reality, which patterns failed, which edge cases were already tried and reverted.

Weekly code churn by percentile

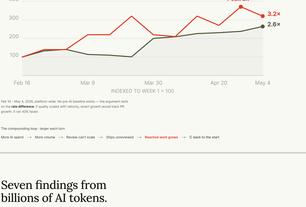
Industry 17%

Lines written and discarded within the same week. The median sits right at the Pluralsight/GitPrime industry benchmark. The **P90 runs at twice it.**

04

Reverts are outpacing output.

Over twelve weeks, weekly PR volume grew **2.6×** — but reverted PRs grew **3.7×**. The failure rate is climbing faster than the work itself. Each revert spawns a bug-fix PR, which feeds the reactive total. The 44% becomes 50%, then 56%.



Feb 16 – May 4, 2026, platform-wide. No pre-AI baseline exists — the argument rests on the **rate differences**: if quality scaled with velocity, revert growth would track PR growth. It can't.

The compounding loop · larger each turn

More AI spend → More volume → Review can't scale → Ships unreviewed → **Reactive work grows** → ↺ back to the start

Seven findings from billions of AI tokens.

Across 9M+ pull requests · 2,444 organizations · May 2026

01

44%

Median org reactive work share — bugs + maintenance consuming nearly half of all engineering capacity

02

25%

Weekly code churn — one in four lines written is discarded before the sprint closes

03

21.6%

Comments addressed — 4 in 5 review comments are never acted on, across 229,000+ records

04

48.5%

PRs under-changed in under 60 minutes — 10,588 with zero reviewer comments

05

3.7×

Reverted PRs grew 3.7× while PR volume grew 2.6× — failures compounding faster than output

06

11.6d

Average bug lifetime in production before being caught and reverted

07

11,033

High-risk flagged PRs approved and merged anyway — 52.7% of all automated risk flags were ignored

05

Half approved in under an hour. Four in five comments never acted on.

Review hasn't scaled with AI output. **48.5% of PRs are approved in under 60 minutes** — faster than any meaningful review. And bug & error comments, the highest-value category, are addressed **just 26% of the time.**

Avg comments / PR

20.8

Bot-generated

80%

Comments addressed

21.6%

Comment types — share of 223k comments, and how often each is acted on

	Share	Acted on
Bug / error	31.0%	34%
Other	20.3%	18%
Security	18.7%	21%
Testing	8.5%	21%
Performance	8.3%	21%
Code suggestion	7.3%	14%
Style / UI	5.5%	19%
Documentation	4.2%	21%
Refactor / design	3.0%	21%

Review turnaround time · 9M+ PRs

Under 1 hr	48.5%
1–4 hours	15.4%
4–24 hours	17.3%
Over 24 hrs	18.7%

06

759 of every 1,000 issues are critical or high.

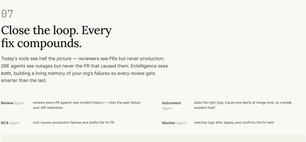
Across organizations with production error tracking connected, **three in four issues are serious enough to cause direct user impact.** Critical issues live **3.3 times on average** before anyone catches them — failures that already reached users, repeatedly, before being logged.

Critical — service-breaking, data loss, security · from 3.3× · **High** — significant functional failure · from 1.2× · **Medium** — non-blocking, limited blast radius · from 6.4× · **Low** — cosmetic, no direct user harm · from 8.2×

Per 1,000 production issues · 1,041 issues · 7,643 PR-month events "Fires" = average times an issue recurs before it is caught.

Top recurring error classes — flagged in review, what happened next

Fixed · Merged anyway



1,041 match events · 15 organizations · **Merged anyway** = flagged by Entelligence, then approved and merged without being fixed.

07

Close the loop. Every fix compounds.

Today's tools see half the picture — reviewers see PRs but never production; SRE agents see outages but never the PR that caused them. Entelligence sees both, building a living memory of your org's failures so every review gets smarter than the last.

Review Agent — reviews every PR against real incident history — cites the past failure your diff resembles

RCA Agent — root-causes production failures and drafts the fix PR

Instrument Agent — adds the right logs, traces and alerts at merge time, so a break explains itself

Monitor Agent — watches logs after deploy and confirms the fix held

REACTIVE WORK

44% → next 90 days

Measurable reduction

REVIEW QUALITY

21.4% addressed

Production-grounded

BUG LIFETIME

11.6-day avg → Caught at PR merge

Entelligence

Production reliability intelligence for AI engineering teams — closing the loop between code and what happens in production.

Home · Book a Call · Sign up