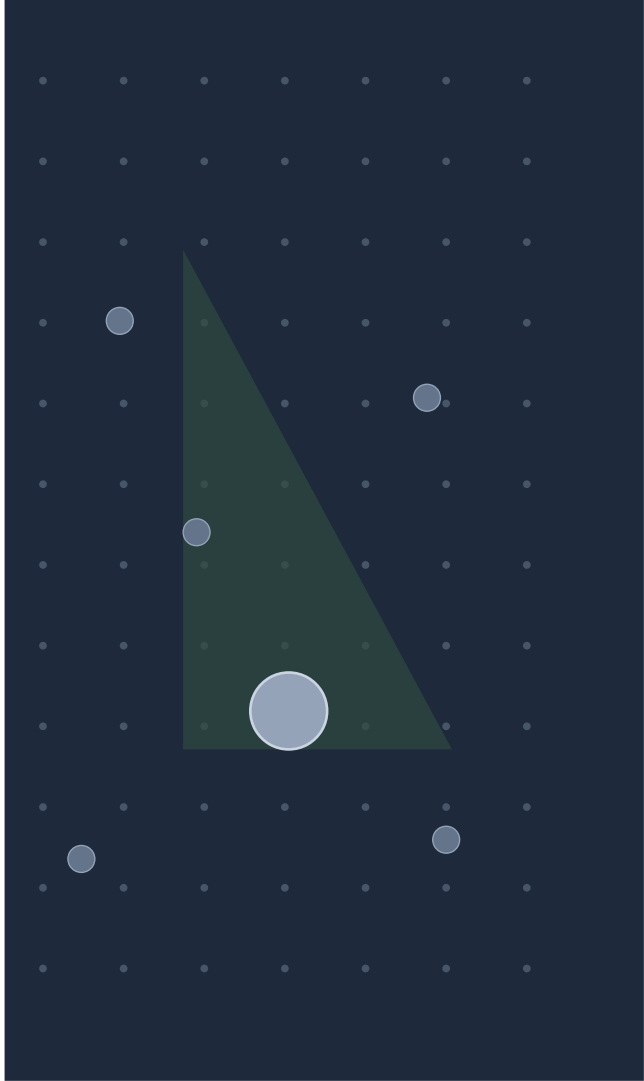
</> AIEWF 2026

YOUR CODING AGENT IS CREATING REVIEW DEBT

Sachin Gupta · Backend Engineer

AI Engineer World's Fair 2026



THE GAP NOBODY IS MEASURING

Coding agents ship PRs faster than humans can trust them.

+25%

Commits, year over year

GitHub Octoverse 2025

−27%

Comments on commits, same year

GitHub Octoverse 2025

And at high-AI-adoption teams, median PR review time is up 441.5% — and 31.3% of PRs are merged with zero review. *Faros AI 2026 benchmark*

⚠ **The gap between those two numbers is review debt.**

THE STORY EVERYONE TELLS

How most teams measure coding agents today

PRs per developer *Faros AI 2026, "Acceleration Whiplash"*	**+16%**	*Vanity*
Median PR size *DX 2026 longitudinal study (44 → 72 lines)*	**+63%**	*Vanity*
Cycle time, open → merge *DX 2026 (PR throughput +8% as AI usage rose 65%)*	**Modestly down**	*Misleading*

THE STORY NOBODY TELLS

What those numbers quietly stop measuring

Reviewer fatigue

Senior engineers are carrying dramatically more review load than a year ago. They are not happier about it.

Late-night merges

PRs unreviewed for 4+ days get a thumbs-up at 11pm before a Friday deadline.

Test theater

Tests get added — but they assert what the code did, not what it should do.

Architectural drift

The same problem solved three different ways in three files. Nobody noticed.

Incident lag

Bugs traced back to AI PRs land weeks or months later. Nobody connects the dots.

DEFINITION

Review debt — definition + why it compounds

Review debt *(noun):* the accumulating gap between the code a coding agent has produced and the code humans have actually reviewed, trusted, and understood.

Like financial debt — it compounds. The interest is paid in human attention.

Why it compounds — three feedback loops:

1	**Agents learn from your codebase**	Yesterday's un-reviewed code becomes tomorrow's AI suggestion.
2	**Reviewers cede the architectural call**	Attention contracts to syntax; big-picture decisions move to never.
3	**Velocity expectations reset**	Throughput rises; headcount-per-review falls. No slack to pay debt back.

Five signal families. Ten deterministic checks. No LLM in the loop.

Diff size & coupling

How big, how spread

Test evidence gap

Code added vs. tests added

AI-authorship indicators

Metadata signals of AI authorship

Evidence & rationale gaps

What the PR doesn't explain

Directory & ownership spread

How many CODEOWNERS touched

Deterministic = the score is defensible in a real engineering review. No model-of-the-week.

Each family rolls up multiple checks; ten checks total. Full mapping in Q&A reference.

SIGNAL 1 / 5

Diff size & coupling

What it measures

Net lines changed, plus how many files are touched and whether they cluster (one module) or sprawl (across modules).

Why agents struggle here

Agents bias toward *"just fix the failing call site."* Human engineers route fixes to the *cause*. Agent PRs sprawl across files searching for the same problem.

SIGNAL 2 / 5

Test evidence gap

AI-authored PRs ship with a far lower test-to-code ratio than human-authored ones.

Test LOC added ÷ production LOC added — per PR, not aggregate.

And the tests that do show up tend to assert what the code did, not what it should do — locking in behavior, including bugs.

⚠ **Why this number is brutal:** Agents happily generate tests — but they assert what the code *did*, not what it *should do*. The pattern doesn't even capture that quality gap.

SIGNAL 3 / 5

Directory & ownership spread

Distinct CODEOWNERS teams or individuals whose files appear in the diff.

FEW

HUMAN PR

Concentrated ownership

Reviewer holds the whole mental model. One approval, one context.

MANY

AI PR

Spread ownership

Multiple reviewers, multiple contexts. No single human holds the whole thing.

SIGNAL 4 / 5

AI-authorship indicators

What it measures

Detects metadata signals of AI-assisted authorship. Not based on parsing the code itself — based on what the author published about the code.

Coauthor footer — likely

```
Co-authored-by: Copilot <copilot@github.com>
```

Branch name pattern — possible

```
codex/, copilot/, cursor/ prefixes
```

PR body / commit phrases — possible

```
"generated by", "assisted by", "powered by Claude"
```

Real-data check

Scanned 3 public repos, 524 PRs

Steady-state firing rate: **5-20%** of PRs per week

Repos: A, B, C — public OSS, names withheld

Highest signal: **likely** via coauthor footer (Repo A)

Lowest signal: 0% on Repo C (no convention)

⚠ **Amplifier only.** Score impact +2 to +5 points (max), and never fires score impact when tests pass and CI is green.

SIGNAL 5 / 5

Evidence & rationale gaps

Whether the PR explains why — not just what.

HIGH GAP

"Fix flaky tests"

PR body length

18 characters

Commit message

"updates"

LOW GAP

"Stop using getRawHeader for SNI..."

PR body length

420 characters + repro

Linked

Issue · Design doc · Bench

THE SCORING RUBRIC

How the five signals combine

```
ReviewDebt(PR)

= 0.25·DiffCoupling  +  0.25·TestEvidenceGap  +  0.20·OwnerSpread
+ 0.15·AI-indicators  +  0.15·RationaleGap
```

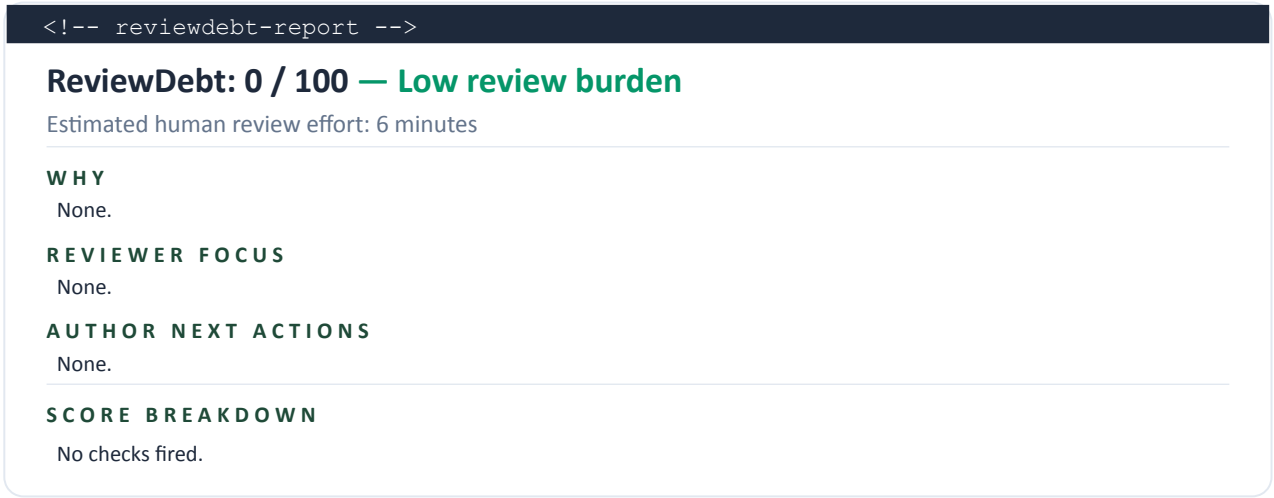
0 – 25	26 – 50	51 – 75	76 – 100
Healthy	Watch	High debt	Reject / refactor

Weights are starting defaults. Calibrate them against your last 200 PRs.

WALKTHROUGH 1 / 3

Clean PR — this is what healthy looks like

Scenario: 01-small-safe-tested · small focused change with matching tests.



Healthy floor: no signals firing → no advice generated. This is the report shape for a well-shaped PR.

WALKTHROUGH 2 / 3

High-debt AI PR — needs evidence

Scenario: 02-ai-broad-rewrite · large agent-assisted refactor, suspect claims, no tests.

```
<!-- reviewdebt-report -->
```

ReviewDebt: 60 / 100 — Needs evidence

Estimated human review effort: 86 minutes

WHY

1. PR is large — review will take longer.
2. Claims in the PR description are not supported by visible evidence.
3. Source changed but no tests were added or updated.
4. PR has soft indicators of AI-assisted authorship. This is informational only — not a definitive claim, and not a penalty on its own.

REVIEWER FOCUS

- Sample a few representative files first; ask the author to split if scope is unclear.
- Verify or push back on the claims before approving.
- Ask the author to attach test evidence.
- Ask the author to confirm tests exercise the new behavior end-to-end.

AUTHOR NEXT ACTIONS

- If multiple unrelated changes are bundled, consider splitting the PR.
- Either back up the claim with evidence or remove it from the description.
- Add or update tests for the changed behavior.
- Add tests that exercise the new behavior, or link to existing coverage.

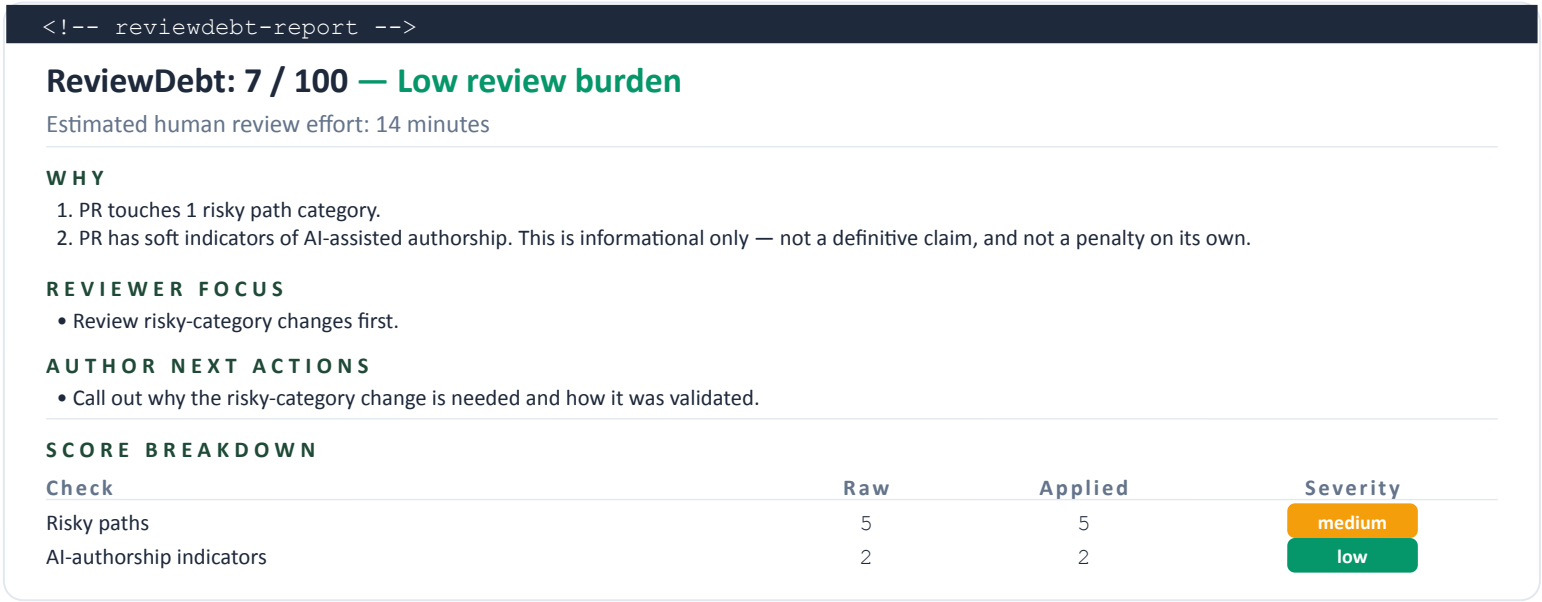
SCORE BREAKDOWN

Check	Raw	Applied	Severity
Diff size	25	25	high
Claim/evidence mismatch	18	18	high
Missing test evidence	12	12	medium
AI-authorship indicators	5	5	medium

WALKTHROUGH 3 / 3

AI-authored, well-shaped — score 7

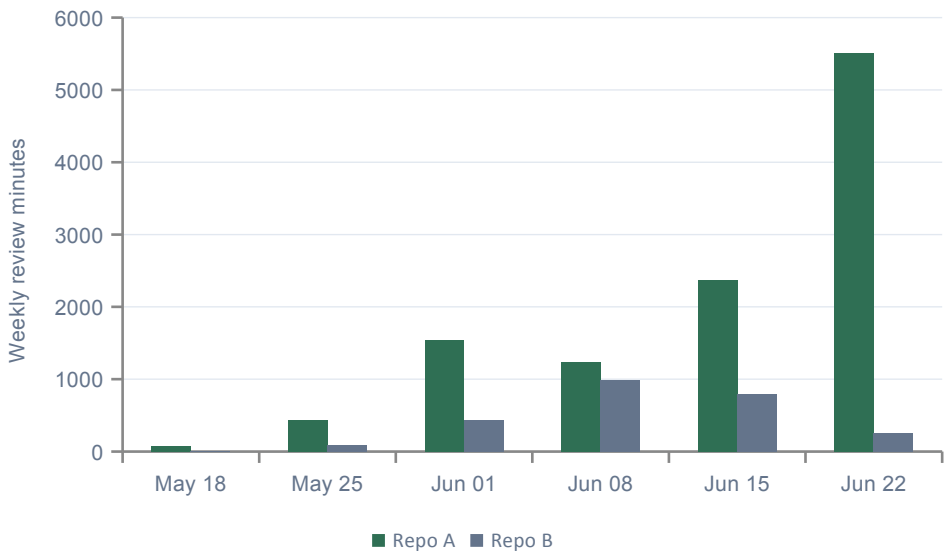
Scenario: 09-ai-likely-pr · agent-authored PR but with matching tests and green CI.



An agent-authored PR with tests and green CI scores 7 — low review burden. AI-likely is an amplifier, never a standalone penalty.

SCALING UP

3 public repos. 524 PRs. Review burden climbs even when AI authorship doesn't.



Volume is the actual variable

AI authorship was flat — total review burden was not.

Amplifier-only holds up under real data

5-20% AI-indicators per week, never landed in high band.

Complexity drives burden, not authorship

The 4 highest-burden PRs were large structural changes — migrations, SDK rewrites, multi-team refactors. AI-driven volume creates the conditions; complexity creates the outliers.

**Repo C included in the 524-PR total but too sparse to chart at this scale (24 merged PRs over 90 days, 0% AI-indicators throughout).*

Methodology: 524 merged PRs · 3 public OSS repos · names withheld for this presentation · CSV available after session.

IN ABSOLUTE TERMS

What 524 PRs looked like under the lens

228

hours of cumulative review burden

Across 3 public OSS repos in 27-90 day windows.

9 PRs/day

sustained merge rate at the high-velocity repo

Volume — not authorship — drives the burden.

5,036

minutes on a single outlier PR

The framework caught a 1,163-file migration that should have shipped as 12.

5-20%

weekly AI-indicators firing rate

Steady-state across all weeks. AI signal is constant. Volume is what changes.

Methodology: 524 merged PRs · 3 public OSS repos · names withheld for this presentation · CSV available after session.

WHAT TO DO ABOUT IT

How to move the needle

Each signal has an inverse. Five craft moves your team already knows.

1	One logical change per PR	Diff size & coupling ↓
2	Tests ship with the change	Test evidence gap ↓
3	Stay in one owner's territory	Directory & ownership spread ↓
4	Author writes the "why" — humans, not agents	Evidence & rationale gaps ↓
5	Same review standard for AI PRs as human PRs	AI-authorship indicators ↓

"None of these require new tools. They're craft moves your team already knows. The scanner just gives you the number to measure whether they're working."

HOW TO ADOPT THIS

Five steps from zero to a defensible number

1	**Backfill**	Score your last 200 merged PRs. Calibrate the weights against your gut-feel ranking.
2	**Threshold**	Set a 'must justify' line. Default: PR ≥ 50 needs a comment explaining why it's allowed to merge.
3	**Surface**	Post the score as a PR comment. No blocking, just visibility. Behavior shifts on its own.
4	**Aggregate**	Roll it up weekly per team. Slope of the team's debt line is the management signal.
5	**Talk about it**	Bring the number to every retro, every roadmap review, every promotion case. Make it boring.

THE STRATEGIC CASE

From measurement to governance

What this number lets you say in the next planning review:

"Our coding-agent rollout has added X% throughput."

"It has also added Y points of average review debt over Z weeks."

"That's roughly N senior-engineer-hours per week of review tax."

"Here's what changing the slope costs vs. not changing it."

2026 · adoption

Are we using agents? What's our throughput? Adoption rate? Do developers like the tool?

2027 · governance

Can we trust what we ship? Who's accountable? What did the reviewer verify? Where's the audit trail?

TAKEAWAYS

Three things to take into Monday morning

1

Measure the gap

Score your next twenty PRs by hand using the five signals. You'll feel the shape of the number before you build any tooling.

2

Make the slope visible

It's the slope of review debt over time that tells you whether your team's discipline is keeping pace with adoption.

3

Have the conversation

Bring the number to your next engineering review. "We have X throughput, Y debt, here's our slope." Move from vibes to measurement.

Anti-patterns to avoid: *approved-with-comments merges, the "AI did the boring parts" excuse, "we'll catch it in QA," and "PRs are smaller now" theater.*

Thank you.

Sachin Gupta

Backend Engineer · AIEWF 2026

Score your last 20 PRs by hand this week.
If the number feels right, you have your framework.