

is not a signature.

AI reviews. Evidence proves. A named human signs.

THE CHALLENGE

Agents now write more code than people can read.

237
files changed by an agent,
in one pull request

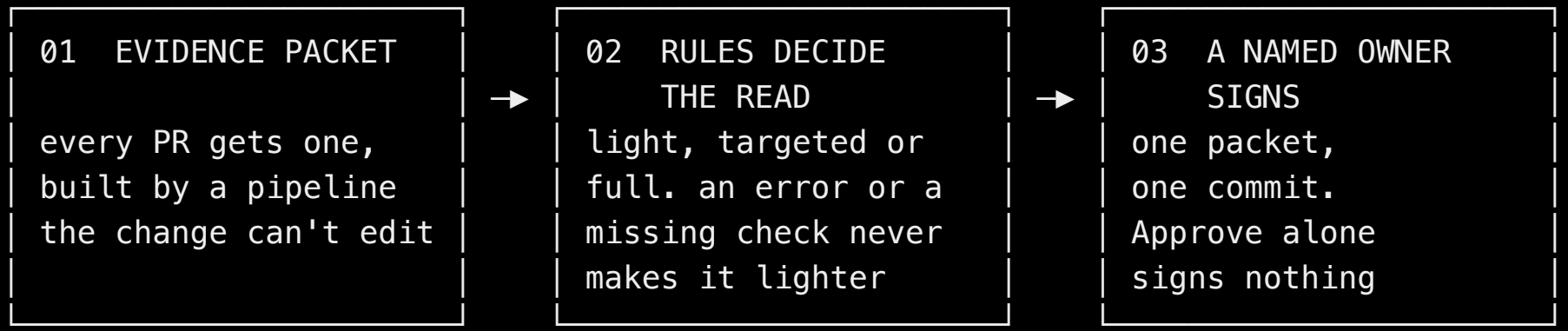
1
name on the approval:
yours

+1,383 -9

> reviewer: ~~"the AI said it was fine."~~ ✗ NOT ACCOUNTABILITY

RECOMMENDATION

Sign the evidence, not the diff.

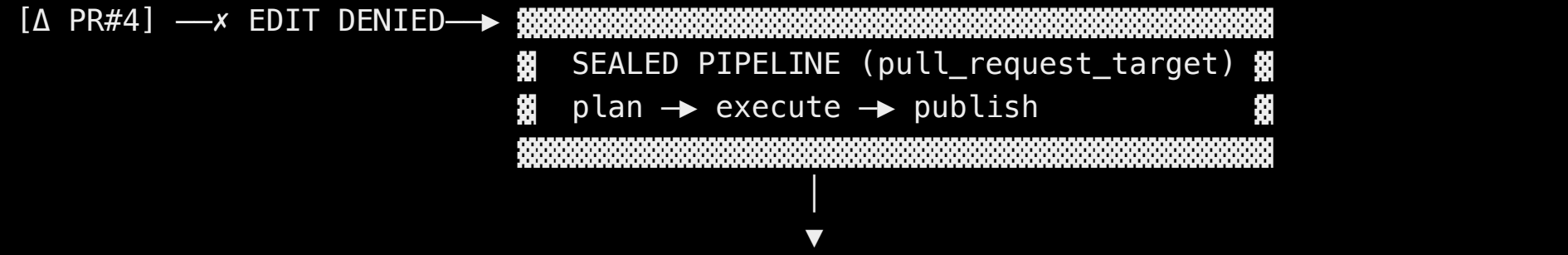


RULE #1 no line cited → no claim. A script checks every citation.

We test the reviewers with planted bugs they are not told about.

HOW IT RUNS

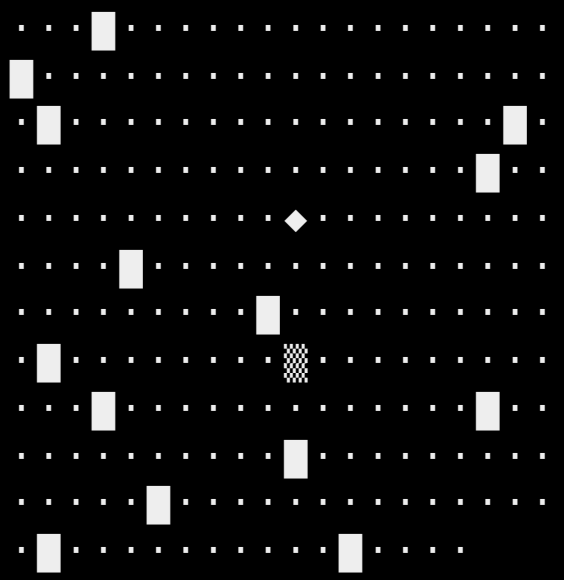
The change can't grade its own homework.



ACTOR	EDIT PIPELINE	BUILD PACKET	SIGN
the change	✗	✗	✗
coding agent	✗	✗	✗
pipeline	—	✓	✗
5 AI reviewers	✗	cite only	✗
named owner	—	✗	✓

RULES ROUTE ATTENTION

Written rules decide how much a human reads.



SYM	VERDICT	FILES
▉	read	15
▒	optional	1
·	skip with proof	221
?	decide	0
▒	not checked	6*

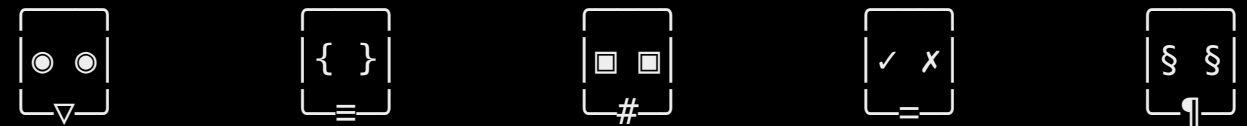
◆ why this depth: 1 generated file doesn't fit the pattern of the rest.

∗ signed as residual risk

without lgtm: read 237 files · with lgtm: read 15

NO LINE, NO CLAIM

Five reviewers. Every claim cites a line.



GENERALIST LOGIC SECURITY TESTS CONTRACTS

- claim + cited line → citation check (script) → second check → packet
- claim, no line → ✗ dropped before a human ever sees it
- can't decide → listed as refused, routed to the owner

On PR #4: 3 findings · 0 refused · 0 failed the citation check · all 3 confirmed.

WHAT THE OWNER SIGNS

```
EVIDENCE PACKET                    9ceb21e3f20a
commit a0aa5ab · pr #4 · targeted read
──────────────────────────────────────────────
requirement    REQ-MPC-2297
protected      generator-templates
files          ▌ 15   ▒ 1   · 221
ai findings    3 · all confirmed
refused        0
not checked    6 → residual risk
──────────────────────────────────────────────
signed by      ✓ named owner · a0aa5ab
```

One owner.
One packet.
One commit.

- ✓ what was checked, and by what
- ✓ what was not checked, and why
- ✓ the residual risk they accept

[Approve] ≠ a signature

New commit → new packet → new signature.

MEASURED ON FOUR AGENT-WRITTEN CHANGES

15

of 237 files to read.
221 skipped, with proof.



11/12

planted bugs caught by
five reviewers.



21/23

findings confirmed by a
second check.

wrongly disputed: 0

Limit: four changes on one repository show the method, not a result at scale.

TRUST IS EARNED PER AREA

Measure outcomes, not review activity.

SIGNAL	WHAT WE TRACK	ON THE SAMPLE
reviewer quality	planted bugs caught, per run/area	11/12
finding quality	findings confirmed + accepted	21/23 confirmed
human load	share routed to a human, time→merge	15/237 ≈ 6%
outcomes	reverts + incidents per area	proposed gate
escapes	bug after merge → trust withdrawn	proposed gate

"Skip with proof" widens in an area only while reverts and incidents don't rise as review time falls.

FAILURE MODES → DECISIONS

What went wrong, and what we decided.

* inconsistent catches — the reviewer missed a planted bug in an earlier run on the same commit
  → rerunning until it's caught does not count

* approve ≠ accountable — a GitHub Approve can be clicked without reading
  → only a sign-off for packet hash + commit counts

* uncited claims — plausible statements with no line behind them
  → dropped before the human sees them

* outliers in bulk — 1 of 237 generated files broke the pattern
  → "doesn't fit the pattern" forces a targeted read

HONEST LIMITS

Trade-offs

- ▒ skip-with-proof is only as good as the rules
- ▒ 5 reviewers + 2nd check cost more than one pass
- ▒ planted bugs test what we can imagine
- ▒ 6 items stayed unchecked: real, signed risk
- ▒ a reference implementation, not a product

Judges' questions

? what would make us not trust it

catch rates drop, reverts rise, owners keep rejecting findings, bugs only caught on rerun

? who owns the decision

one named engineer per commit. the AI never approves

? how does it scale

human time follows risk, not size: 237 files → 15. trust widens per area by track record

LGTM IS NOT A SIGNATURE

Line-by-line review becomes the fallback, not the default.

play	lgtm.pgf.lv
demo	youtube.com/watch?v=l-QzFClfZCU
repo	github.com/progresful/junction26-messagepack-csharp
packet	pull request #4, the evidence packet in review