Arko Labs

// ARKO LABS REPORT

Who Grades the Agent?

What 149 independent verdicts on AI-agent work taught us about letting agents ship code

Arko Labs · October 2026
by Camilo Girardelli · arkolabs.ia.br

Key findings

An agent should never be the one who decides its own work is done. Across two projects built entirely by AI agents, a separate evaluator caught problems the builder would have shipped.

01 **1 in 7 deliveries did not pass clean.** Of 149 independent verdicts on SHIELD-ID, 23 (15.4%) were failed, amended or passed with caveats by an agent that did not write the work.

02 **Hitting the metric is not the same as doing the job.** Twice, a change met its target while breaking its purpose. Only the evaluator, checking a floor set before any measurement, stopped it.

03 **Automated proof is not real use.** 30+ automated verification stages passed; one morning of real use by a non-technical person found 7 problems none of them caught.

04 **The gates paid off in production.** On ZECA, tests went from 3,770 (186 failing) to 17,815 all passing, and hotfixes on main dropped from ~9 to ~1 a month, while the team shipped 947 PRs in 26 weeks.

05 **It was cheap.** The whole AI cost on ZECA was a flat US$200/month subscription: US$1.27 per PR.

Setup

The data comes from two codebases built by the Guild, Arko Labs' team of AI agents, with a human owner who approves what ships.

	SHIELD-ID	ZECA
What it is	Synthetic-identity fraud detection prototype	Fuel marketplace for truck drivers, in production
Period	Jun 22 – Oct 26, 2026	Apr – Oct 2026 (26 weeks)
Unit of work	Round with an independent verdict (no PRs)	Pull request through a staged ritual
Size	1,075 commits; 20,264 lines of code, 40,102 of tests	947 PRs; 17,815 tests
Data	All synthetic, by construction	Production

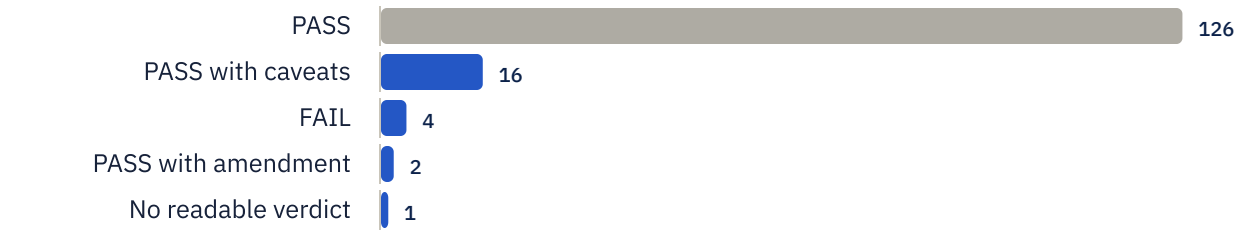
Roles. A developer agent builds; a QA agent tests; an evaluator agent (the Oracle) reviews each stage adversarially and records a verdict. The owner opens rounds, signs push windows and makes the calls logged in an approval log.

How verdicts are recorded. Each round leaves a verdict file on disk, linked to the round that produced it. Every number in this report was read from the repositories on Oct 26, 2026 (SHIELD-ID at commit 6c2f218), not typed from memory.

Finding 1: one in seven deliveries did not pass clean

1 in 7 rounds did not pass clean

23 of 149 independent verdicts (15.4%) failed, amended or passed with caveats



Blue = not clean. Each verdict was written by an agent that did not build the round.

SHIELD-ID VERDICT FILES ON DISK, JUN 22 – OCT 26, 2026 · 149 ROUNDS

This is not a human reviewing agents; it is one agent rejecting another. Most rejections were caveats, not failures: the evaluator rarely blocked outright, but it made the builder fix or disclose something in roughly one round out of seven.

Finding 2: the metric was met, the purpose was broken

Twice, a change hit its number while undoing the reason the number existed. Both times the evaluator caught it; the report alone would have said "done".

Case 1: the 2% that would have killed fraud detection. In user story US-007, the behavioral layer held back 114 of 2,400 legitimate users (4.75%) against a 2% ceiling. Raising the threshold to 0.65 would have delivered 1.46%. But a floor registered before any measurement said no fraud family may drop. A sweep of 62 thresholds found zero points meeting both.

THRESHOLD	LEGIT USERS HELD	F-BOT	F-COMBINED	TARGET (2%)	FLOOR
`0.65`	1.46%	0.535	0.425	met	broken
`0.561 (current)`	4.75%	0.617	0.442	missed	held

At the 1.46% point, the layer caught the F-combined fraud family at 1.99% while holding back 1.67% of fluent legitimate users: the two came from the same behavior generator. That is not bad calibration; it is no separation at all. Without the evaluator, the report would have said "2% achieved" and been technically correct.

Case 2: a quiet guard is not a working guard. The product owner proposed deriving a freshness stamp from the board instead of writing it. The goal was met: the guard stopped rejecting the owner's push. But a derived stamp is always fresh by construction, so the guard could no longer warn. The evaluator failed it in round 157; it was fixed in round 158.

> **Why it works:** the separation is of interests, not of people. The builder wins by closing the round; the evaluator only wins by finding something.

Finding 3: automated proof is not real use

In the week of Oct 20–26, the Guild's own installer passed more than 30 automated verification stages. The owner's first real test, as a non-technical user on Windows, found 7 problems in one morning. None showed up in the automated checks. Among them:

- Smart App Control blocked the unsigned installer.
- Windows' `python3` was a fake Microsoft Store shortcut.
- The screen hung on "Opening..." because an old process kept running underneath.
- Accents came out broken in the installer.
- The app opened as a browser page instead of an app.

Every one became a new test and was fixed the same day, except code signing, left as an owner decision. The same week, the independent reviewer issued 12 rejections, all for real defects, including two security holes in the desktop app's window permissions and redirects.

> An evaluator agent checks the work against the spec. Only a human using the product checks the spec against the world.

Finding 4: in production, the gates paid for themselves

On ZECA, the same gates ran on a live product and quality went up while speed went up.

MEASURE	BEFORE THE GATES	WITH THE GATES (AS OF OCT 26, 2026)
Time per feature	3 to 5 days (founder's estimate)	~5 h median, kickoff to PR opened
Website tests	3,770, 186 failing	17,815, all passing
Hotfix, revert or rollback commits on main	~9 a month (Mar–May)	~1 a month (Jul–Oct)
AI cost	—	US$200/month flat (US$1.27 per PR)

The trade-off was real: with the full ritual, PRs got larger and the median from first commit to merge rose from ~1 h to ~20 h, because every PR goes through the reviewer, a test ratchet and a second key. Fewer, safer deliveries. Full case: ZECA case study.

What we recommend

If your team lets AI agents write code, five rules cover most of what we saw.

1. **Separate the builder from the judge.** The agent that writes the change never approves it. Give the evaluator its own prompt, its own context and a reason to find problems.

2. **Register the floor before you measure.** Write down what must not get worse (fraud caught, latency, coverage) before the first run. A target without a floor invites the 1.46% answer.

3. **Make the verdict a file, not a feeling.** Every round leaves a verdict linked to the work. When someone asks "who approved this?", the answer is on disk.

4. **Ratchet the tests.** Count tests and never let the number drop. Write the failing test before the fix.

5. **Keep a human who uses the product.** Automated gates check the work against the spec; only a person checks the spec against the world. Budget one real-use session per release.

Limits of this data

This is what two projects measured, not a benchmark.

- **Synthetic data.** Every SHIELD-ID figure comes from synthetic corpora, by construction. Nothing here proves performance on real fraud.
- **Not the "right" rate.** 15.4% is the rate measured in this project, with these roles, in this period.
- **Authorship by role, not by git.** All agent sessions commit under one git identity, so `git log --author` shows one author. Traceability lives in 516 round-named commit messages, 149 verdict files and the approval log. In-commit role tags only started on Oct 20, 2026.
- **Correlation in production.** ZECA's quality numbers improved alongside new process, new fleets and seasonality; the gates are one cause among several.
- **Before vs after on ZECA** uses the founder's estimate for feature time before the gates.

Sources

- ZECA case study
- SHIELD-ID case study
- Guild Diary, week of Oct 20–26, 2026 (internal)