

The ten-point checklist

Plausible is the default. Quality is a setup. Your agents fight each other so you don't have to fight the slop. Tick what you have; the first unticked box is Monday's job.

PLAUSIBLE

Looks right. CI is green, the diff reads fine at a glance, the agent says it works. Three problems hiding under it.

QUALITY

Still right when someone actually checks. Something caught it before you did. Nothing passed because it looked fine.

THE LOOP Make it work

Without these, nothing else has anything to run on.

☐ 01

Deterministic rules

Everything a machine can decide, a machine decides. Nothing mechanical reaches a human reviewer.

`tsc --strict` · `eslint` · `knip` for dead files, exports and dependencies · `dependency-cruiser` for circular imports and module boundaries · `zod` at the edge, inferred types inside (`drizzle`)

ASK Can a PR with an unused export or a circular import go green? If yes, the rule is advice, not a gate.

☐ 02

A simulator, or a behavioural oracle

Code that follows every rule can still fail in practice. The rules check the words; something has to check the world.

Playwright from the UI down to the database · `msw` for a deterministic network · for hardware: the real device, or a faithful simulator of it

ASK Can the agent get a yes/no about behaviour, not just about types, without a human in the loop?

☐ 03

Quality context

What to do, what not to do, and why. Skills for the repeatable procedures, decision records for the reasons.

No decision records? Mine Confluence, Jira history, PR history, `git log`. Have something.

ASK Is the agent figuring out the same things from scratch every session? That is the problem, not the model.

THE SIGNAL Keep it honest

Agents optimise for the oracle's verdict. If the oracle can be edited, it will be.

☐ 04

A tamper-proof oracle

Tests that pin behaviour (`toBe(42)`, `never toBeDefined()`), mutation testing that tests the tests, and an independent reviewer bound by the anti-gaming charter (overleaf).

`stryker` on core modules, 70–85% mutation score as the target · the reviewer is a different model from the writer, and cannot edit tests or thresholds

ASK Would anyone notice if it were wrong? Run Stryker once on your core module and look at what survives.

☐ 05

A deterministic environment

If a failure is not reproducible, the agent cannot learn from it, and neither can you. It will “fix” phantom problems, or learn to ignore red.

Pinned dependencies and lockfile · fake clocks and seeded randomness in tests · no real network in CI · flaky tests quarantined, not retried

ASK When something fails, can you tell in one look whether it was the environment or the agent?

☐ 06**Small PRs**

AI pushed PR size up 154%. Big diffs exhaust context, invite shortcuts, and get rubber-stamped.

Split by layer: schema · API · logic · UI · tests. One concern per PR, one PR per agent session.

ASK Could a reviewer hold the whole diff in their head? If not, it is not reviewable, only approvable.

☐ 07**Reversible, limited blast radius**

Cheap rollback is what lets you accept less than certainty on each change.

Sandboxed agents · least-privilege tokens · worktrees per task · feature flags · one-command rollback

ASK What is the worst thing this agent can do with the credentials it has right now?

☐ 08**Review capacity treated as scarce**

Everything mechanical goes to machines. Human judgement goes to intent, architecture and business logic.

A third more PRs now merge with no review at all. When review becomes the queue, people rubber-stamp.

ASK What did the last human review actually catch that a machine could not have?

COMPOUND Make it improve

Without these you fight the same failure modes forever.

☐ 09**Ratchet**

Every escaped bug becomes a permanent rule, invariant, or line of context. Thresholds go up when you can, never down.

One adversarial test per incident · raise coverage and mutation thresholds after every green week

ASK What did the last incident leave behind in the repo?

☐ 10**Production feedback**

The outermost oracle, for everything the simulator did not model.

Observability · SLOs · canaries · error budgets that feed back into point 09

ASK How would you know, within an hour, that an agent-written change made things worse?

THE ANTI-GAMING CHARTER · WHAT THE REVIEWER ENFORCES

1. Never lower a coverage or mutation threshold to make a change pass.
2. Never rewrite or delete a test to match the behaviour that broke it.
3. Never silence the type system: no `as unknown as`, no `!` on a nullable value, no `@ts-ignore` without a linked issue.
4. The reviewer is a different model from the writer. It reads this charter with examples, and it cannot edit the tests.

MONDAY · 1 HOUR

npx stryker run on one core module. Look at what survives.

MONDAY · 1 HOUR

One boundary rule in dependency-cruiser, failing the build.

MONDAY · 1 HOUR

One decision record, mined from the last contested PR.

If you cannot name the problem a tool solves for you, you bought the feeling.