

FOR CTOS · IT'S NOT THE MODEL

Lessons from the best AI engineering teams.

Top teams run the same models you do and get cheaper, more autonomous, lower-rework results. The gap comes from three things a CTO controls, and all three can be measured before the bill arrives.

WHAT GETS DEBATED

Model choice

Decided a few times a year

WHAT GETS DISCUSSED ↑

WHAT DECIDES THE OUTCOME ↓

01

Intent

How clearly the work is specified

02

Environment

Whether the agent can check itself

03

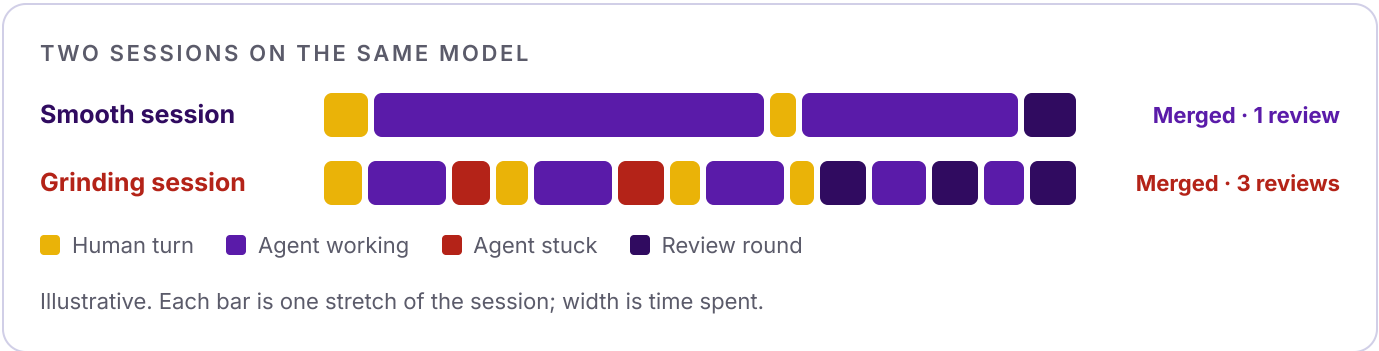
Stewardship

Who finishes the job before review

Shape every task, every day

Same model. Same task. Very different bill.

Most of the AI conversation in engineering is about which model to buy. Yet two developers can hand the same model the same ticket and get opposite results. One session ships code that survives review. The other loops through clarifications, broken tooling, and review rounds that redo the agent’s work.



Three numbers a CTO should watch

COST

Token spend per merged AI line

What each line of shipped agent code actually cost.

AUTONOMY

Merged AI lines per human turn

How far the agent gets before it needs a person.

REWORK

Review rounds per 1,000 AI lines

How much of the work comes back after the PR opens.

OUTER LOOP ENGINEERING

All three numbers are lagging. Top teams verify while the work is still happening.

PR Reviews

Checks every change for bugs, conventions, and blast radius before a person reviews it.

End-to-end tests

The Synthetic Users Agent walks real user flows and records what breaks.

Verification Agent

Checks the work against the original requirement, separate from the agent that wrote it.

SRE Agent

Watches production, ties alerts to the change that caused them, and starts the fix.

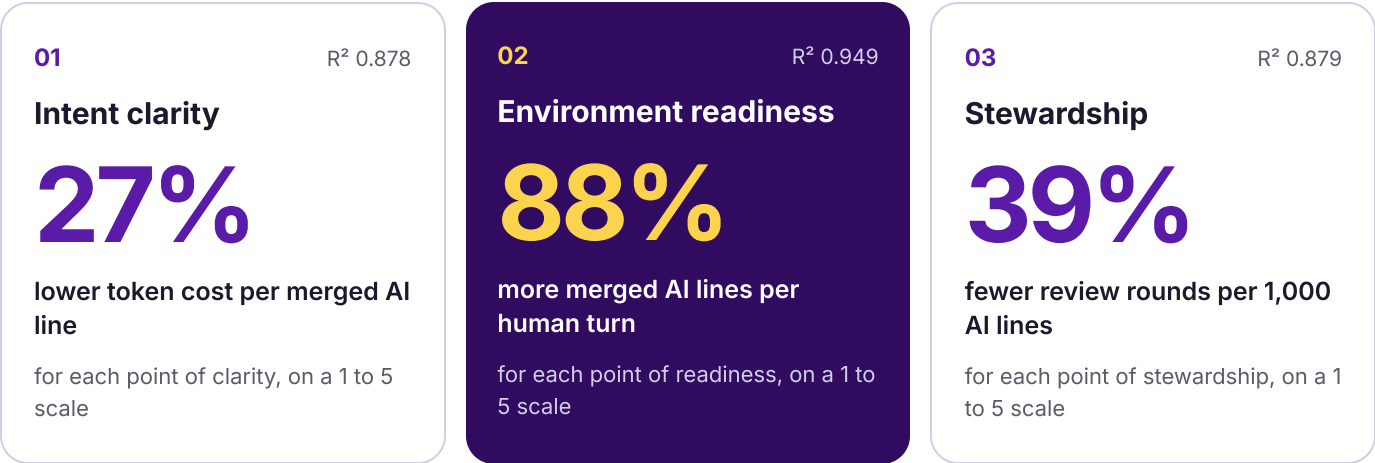
CodeWiki

Answers from the current code, so agents and new engineers start with the right context.

XHawk AI employees that verify agent work at every step, from ticket to production.

Three levers predict agent results better than the model does.

Span analysed complete agent sessions from 103 engineering teams over 12 weeks, May to July 2026: every prompt, human and agent turn, tool call, diff, and review through merge. Each session was scored on three properties and compared with cost, autonomy, and rework. Sessions on the same models diverged sharply, and the divergence tracked these three scores.



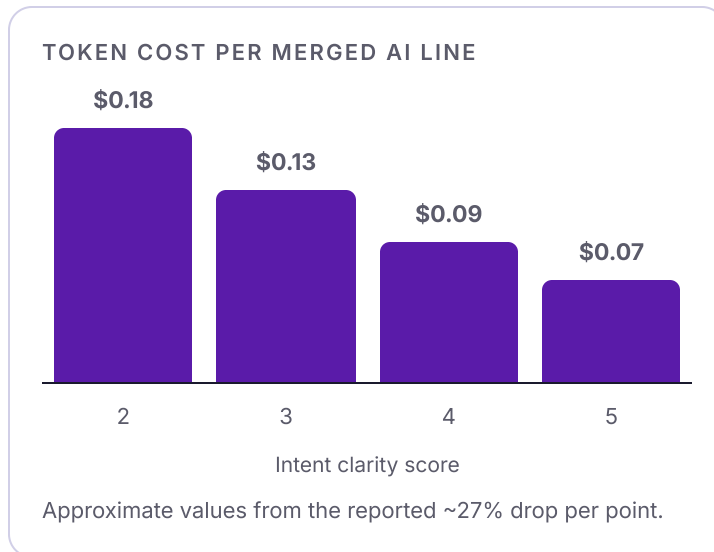
LEVER	WHAT IT MEANS IN PRACTICE	WHAT IT MOVES
Intent clarity	The task arrives with scope, constraints, and a definition of done	Token cost
Environment readiness	The repo builds, tests run, and the agent can verify its own work	Autonomy
Stewardship	Conventions, failure handling, and rollout safety are handled before the PR	Review rework

READ CAREFULLY

These are correlations from observational data, not a controlled experiment. They are strong enough to manage by, not proof of cause.

A vague ticket is the most expensive line item in your AI budget.

Unclear prompts send agents exploring: wrong-direction implementations, regenerations, and long searches for context a single sentence could have supplied. The largest single source of wasted tokens was the agent guessing what “done” meant, and guessing wrong.



A TYPICAL PR

Clarity 3.0

\$6.43

Clarity 4.0

\$4.72

One point of clarity, multiplied across every PR your teams open each week.

What a clear task looks like

01

Context

Link the issue, doc, or error

02

Goal

The behavior that should change

03

Done when

Tests that pass, cases that keep working

04

Constraints

Conventions, security, performance

05

Out of scope

Files and APIs not to touch

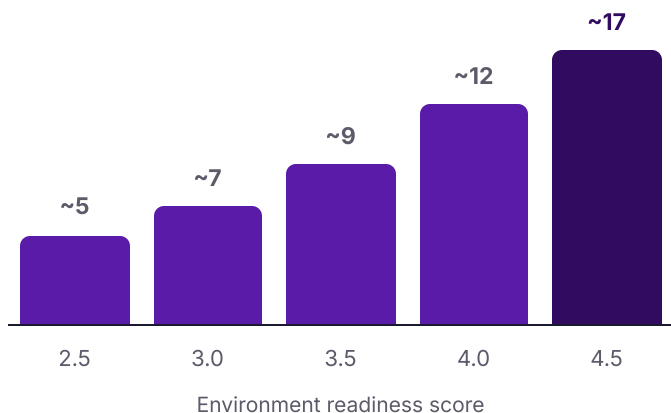
CTO MOVE

Make this a team template for recurring work like bug fixes and new endpoints. Clarity stops depending on who wrote the ticket, and it becomes something you can review.

Autonomy belongs to the workspace, not the model.

When the repository builds cleanly and the tests run first time, an agent can act, check its result, and correct itself before it ever needs a person. When the lint config is broken or the tests are flaky, it cannot tell its own mistakes from the environment's, so it hands the problem back. Every handback costs a human turn.

MERGED AI LINES PER HUMAN TURN



Approximate. Span reports roughly 5 lines per turn at 2.5 rising to 17 at 4.5.

THE SURPRISE

A flaky test suite was worse than no test suite.

With no tests the agent knows it is guessing.
With flaky tests it trusts a signal that lies.

The readiness checklist



One-command setup

Install, build, and test with a single documented command.



Reliable, fast tests

Fix flaky tests and broken harnesses first.



Fast feedback beyond tests

Type checks, linters, and builds with machine-readable errors.



An agent guide in the repo

A short file covering build, test, conventions, and sharp edges.



No hidden manual steps

Script seed data, fixtures, stubs, and local credentials.



Legible code

Consistent structure and naming are what the agent searches.

Every review comment is work that arrived late.

Strong sessions did not stop when the code ran. The developer pushed the change toward local conventions, made failure handling explicit, thought through blast radius and rollback, and left the context a reviewer needs. Weak sessions pushed all of that downstream, where it involves more people and more rounds.

LOW STEWARDSHIP

The reviewer finishes the job

This module names things differently.

What happens on an empty input?

How would we undo this in production?

Where are the logs for this path?

~7 review rounds per 1,000 AI lines

HIGH STEWARDSHIP

The author finishes the job

✓ Matches the conventions of nearby files

✓ Handles errors and edge cases in the code

✓ States reach, scale, and how to roll back

✓ Ships logs, metrics, and alerts with the change

~4 to 5 review rounds per 1,000 AI lines

Example review comments, paraphrased. Round counts are Span's estimates at stewardship scores of 3.0 and 4.0.

Four habits that move work left

Read the diff first

Check for scope creep, deleted tests, and odd rewrites before anyone else sees it.

Make it prove the work

End every task by running the suite and showing the failing case now passes.

Ask for a self-critique

Have the agent list assumptions, missed edge cases, and what it could not verify.

Treat rollout as part of the change

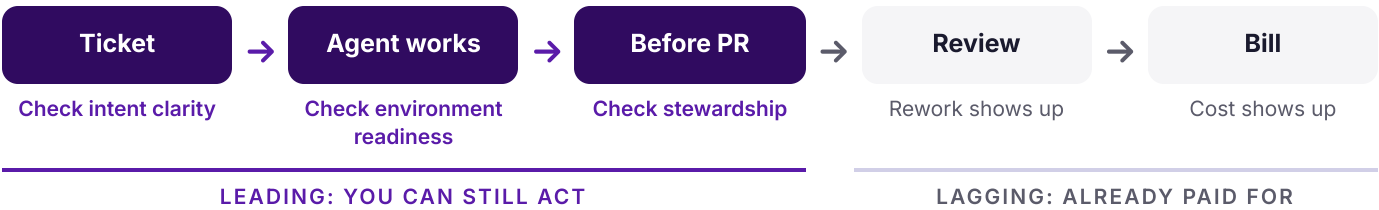
Feature flags, reversibility, and migration order belong in the PR.

THE POINT

Stewardship does not remove the work of making a change safe. It moves that work into the session, before it becomes a reviewer's problem.

Catch the expensive session before it gets expensive.

Cost, autonomy, and rework are scorecards you read after the fact. The three levers are visible inside the session itself, which means they can be checked while there is still time to act: an unclear task before tokens are spent, a broken environment before it forces another handback, a thin change before the PR opens.



How XHawk works each lever

LEVER	XHAWK	WHAT CHANGES FOR YOUR TEAM
Intent clarity	Company Brain, Product Manager	Tickets arrive with linked context, past decisions, and acceptance criteria, so agents start with the definition of done
Environment readiness	Platform Engineer, Verification Agent	Agents run in sandboxes where they can build, test, and check their own work before asking a person
Stewardship	PR Reviews, Synthetic Users Agent	Conventions, blast radius, and rollout safety are checked on every change before a human reviewer sees it
Measurement	Company Brain	Each session leaves a record, so you can see which teams and repos need which lever

WHY IT COMPOUNDS

Model choice is a decision you make a few times a year. These levers apply to every task, every day, and every fix to them carries over to the next session.

Start with the cheapest lever and build up.

Intent clarity needs no new infrastructure, only a better way of writing work down. Environment readiness takes some team investment. Stewardship is the habit that makes both stick. Sequence them in that order.

DAYS 0 TO 30

Clear intent

- › Adopt a task template for your three most common ticket types
- › Require a definition of done on every agent task
- › Resolve open questions before handing work to an agent

DAYS 30 TO 60

Ready environments

- › Get every core repo to one-command setup
- › Fix or quarantine flaky tests
- › Add a short agent guide at each repo root

DAYS 60 TO 90

Stewardship and measurement

- › Make verification and self-critique the last step of every task
- › Add rollout and observability to the PR checklist
- › Track cost, autonomy, and rework by team

WHAT TO REPORT TO THE CTO

Cost per merged AI line Trending down	AI lines per human turn Trending up	Review rounds per 1,000 AI lines Trending down
--	--	---

A NOTE ON MODELS

None of this says models do not matter. Better models raise the ceiling. These levers decide how close each team gets to it.

QUESTIONS FOR THE CTO

- 01 Do you know what one merged line of agent code costs your company today?
- 02 How far does an agent get in your main repo before it has to ask someone for help?
- 03 How many of your review comments are work that should have happened before the PR?
- 04 Is your next AI decision a model choice, or a fix to the system around the model?

**Everyone can buy
the same model.
The system around it is yours.**

Build the system with **XHawk.**

AI Employees for Product, Engineering & Analytics.

Learn more →
xhawk.ai

