Blue Camel Consulting

A FIELD GUIDE · NOVEMBER 2026

The current state of AI-assisted development

What coding agents do well today, the working method that makes them dependable, and how a team takes over an app built this way.



Bridging strategy to execution

BLUE CAMEL CONSULTING

JEFF PINTO · BLUECAMELCONSULTING.COM

WHERE THINGS STAND

From autocomplete to agents that finish tasks

Length of task an agent finishes alone

Doubled about every 7 months in METR's March 2025 analysis

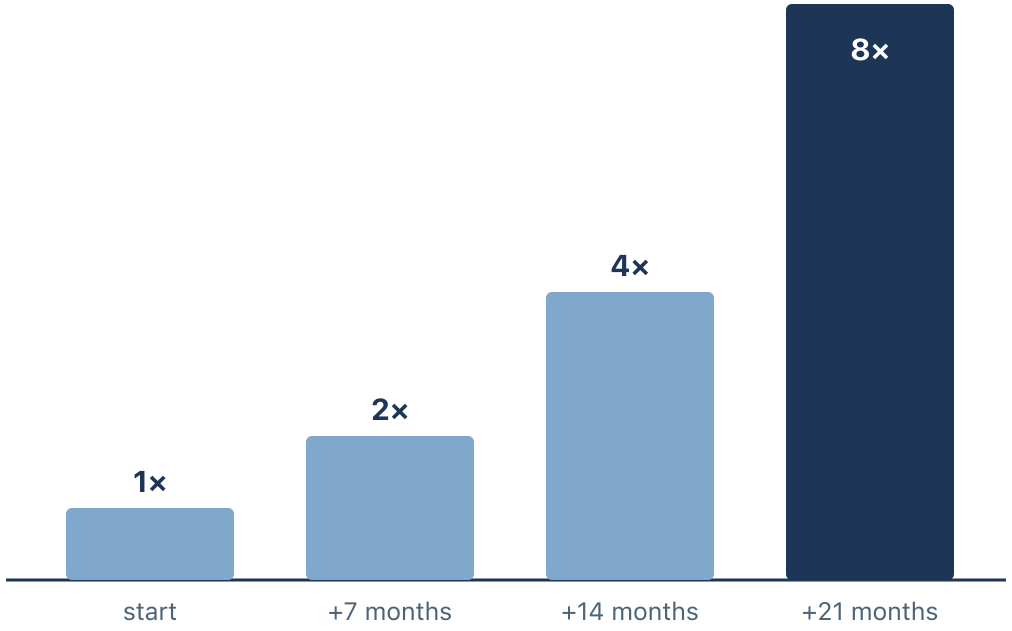


Illustration of the measured rate. The bars are the math of a 7-month doubling, not METR's data points.

~90%

of developers use some form of AI assistance (DORA report, via InfoQ).

60k+

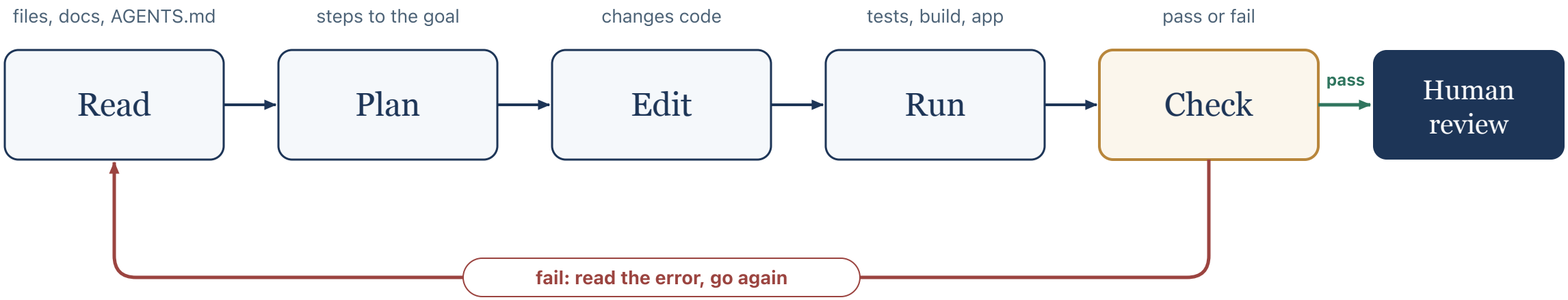
open-source projects carry an AGENTS.md, read by Codex, Copilot and Cursor.

20% slower → likely faster

METR's early-2025 study measured experienced developers about 20% slower with AI tools. Its February 2026 update expects newer tools to speed them up, on early evidence.

Sources: METR time horizons · METR uplift update, Feb 2026 · InfoQ on DORA, Mar 2026 · agents.md

A loop of small steps, each one checkable



The agent is only as good as what it can run.

A test suite and a working app turn each loop into evidence. With nothing to run, the loop has no Check.

Source: GitHub Copilot docs: "Copilot can research a repository, plan changes, and implement them in the background."

STRENGTHS AND LIMITS

Where agents shine and where they slip

STRONG TODAY

- A well-specified change to an existing pattern
- Reading an unfamiliar codebase and explaining it
- Writing and running tests, then fixing what fails
- Upkeep: upgrades, refactors, documentation
- Parallel work on separate branches

NEEDS A PERSON

- Vague goals: the agent fills gaps with guesses
- Output that looks right and is subtly wrong
- Business meaning: is it the right number?
- Large changes in one go, hard to review
- Anything with no test or check to run against

Developers naming "almost right, but not quite" as their top frustration with AI tools

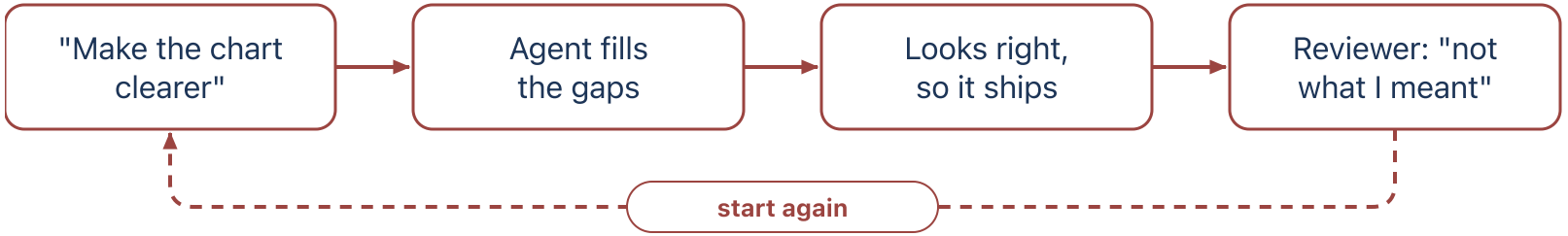


Source: Stack Overflow Developer Survey 2025, AI section

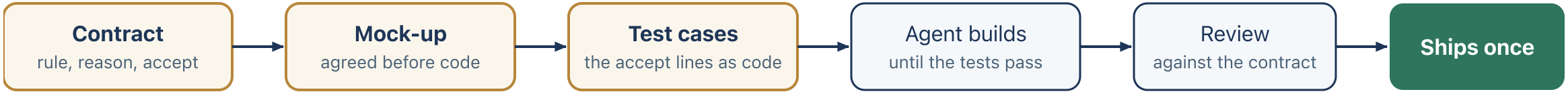
THE BIGGEST GAP

Most teams skip the contract, the test cases and the mock-up

FROM A PROMPT ALONE



FROM A CONTRACT



PEOPLE DECIDE HERE, BEFORE ANY CODE

Contract

What the screen shows, for whom, and why. Each item ends in an accept line: the observable fact that makes it done.

Mock-up

A picture with sample figures, agreed with the person who asked. Choices are made on the picture, where they are cheap.

Test cases

Each accept line written as a check the agent can run. Each check is proven by breaking the code once and seeing it fail.

A CONTRACT ITEM, START TO FINISH

One reviewer comment, carried from words to proof

contract.md · sample item · Actual and Plan in two hues

SAMPLE REQUEST FROM A REVIEWER

"Make the chart easier to read:
Actual in green, Plan in blue"

RULE

Plan is a solid blue. Actual is a solid ink.
Neither uses a status colour.

REASON

Green already means a good reading on these screens. A green bar every month would call every month good, including months below plan.

ACCEPT

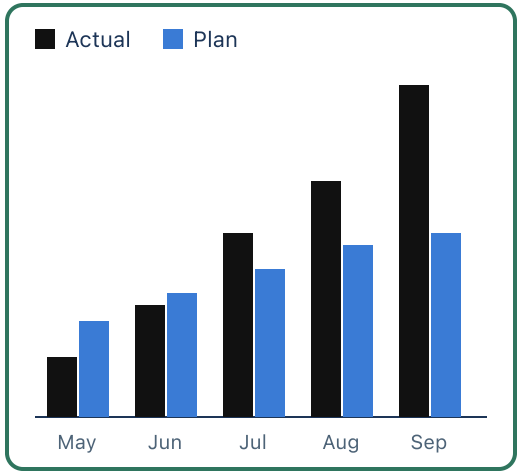
```
fill of rect[actual] = --ink; fill of rect[plan] = --plan-blue;
neither equals any --status-* value;
each measures at least 3:1 on the card
```

PROOF THE CHECK WORKS

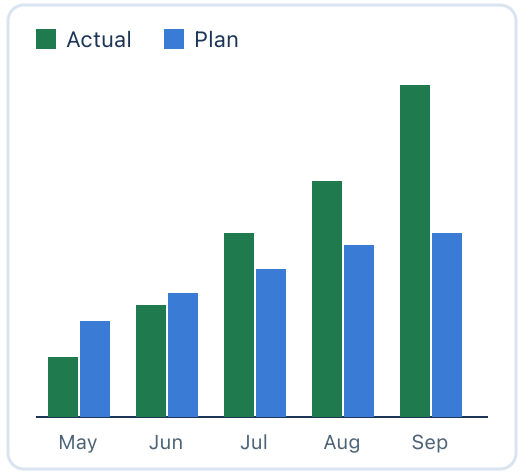
```
mutant: set Actual back to the old gray
✗ fill check goes red   (the check can fail, so its pass means something)
✓ restored: fill check passes
```

Mock-up for discussion. Layout only: figures are samples.

OPTION 1 · CHOSEN



OPTION 2 · AS WRITTEN



Both options are drawn. The owner chooses on the picture, before any code changes.

Every item

a rule, a reason and an accept line

Picture first

choices made on the mock-up

Accept = check

each accept line runs as a test

Proven

each check is broken once to show it can fail

THE WORKING METHOD

Six habits that make agent output dependable

1



A written contract

What the view shows, for whom, and how to tell it is done.

The one most teams skip.

2



Instruction files

AGENTS.md holds setup, test commands and house rules.

3



Tests as the gate

Done when the checks pass, not when the agent says so.

4



Small diffs

Smaller changes improve review and lower deployment risk (DORA).

5



Staging before production

Every change runs on a preview copy first.

6



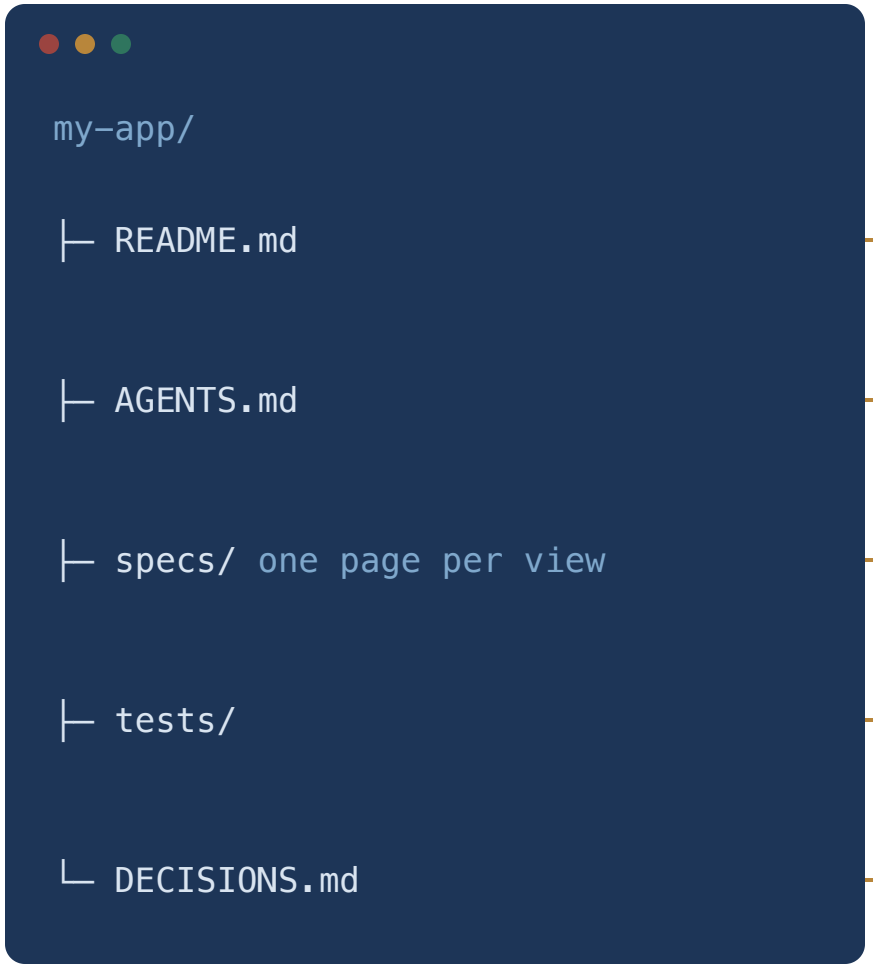
A human reviewer

A named person reads the diff and checks staging.

Sources: agents.md · InfoQ on DORA, Mar 2026

PROJECT FILES

What lives in the repository, and who reads it

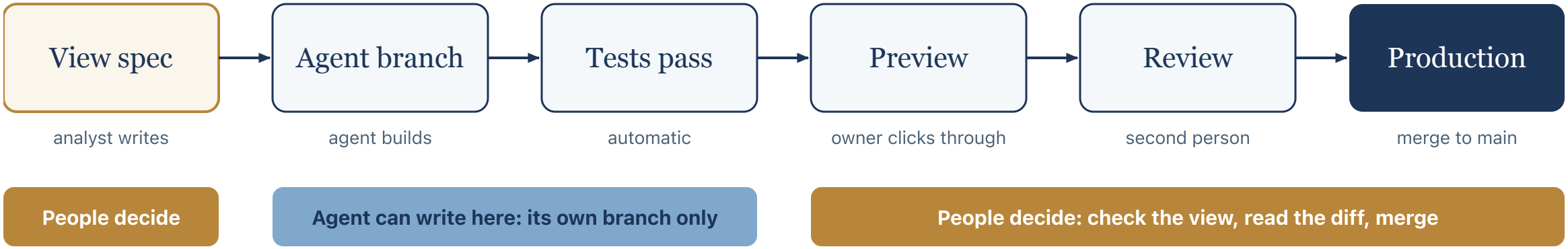


	MAIN READER
What the app is and how to run it. Where the data comes from.	People
Build and test commands, rules, never-do list. The closest file to the work wins.	Agents
Purpose, users, fields, filters, accept lines. The contract for each view.	Both
The accept lines as code. Run on every change.	Both + the build
Why things are the way they are. One line each.	The next person

Sources: Codex docs, AGENTS.md · Claude Code docs, memory · agents.md

FROM A DEFINED VIEW TO PRODUCTION

One path for every change



Testing and production stay apart

Only a reviewed merge reaches the live app.

Two reviewers

The asker checks the view. A second person reads the diff.

Agent permissions

The agent opens the pull request. A person merges it.

Source: GitHub Copilot docs: "Review and test the output before using it in production."

What a team needs on day one

IN THE REPOSITORY

README

Runs the app on a fresh laptop in under an hour.

AGENTS.md

The exact build, test and deploy commands.

specs/

A contract for each existing view: the written baseline.

tests/

A suite that passes today, run on every pull request.

staging

A preview copy and a one-command production deploy.

IN THE HANDOVER SESSIONS

1. **Walk the data path**
   source, refresh, screen
2. **Make one change together**
   end to end, with the agent
3. **Name owners and reviewers**
   per view, per change
4. **Agree the wrong-number route**
   who reports, who fixes

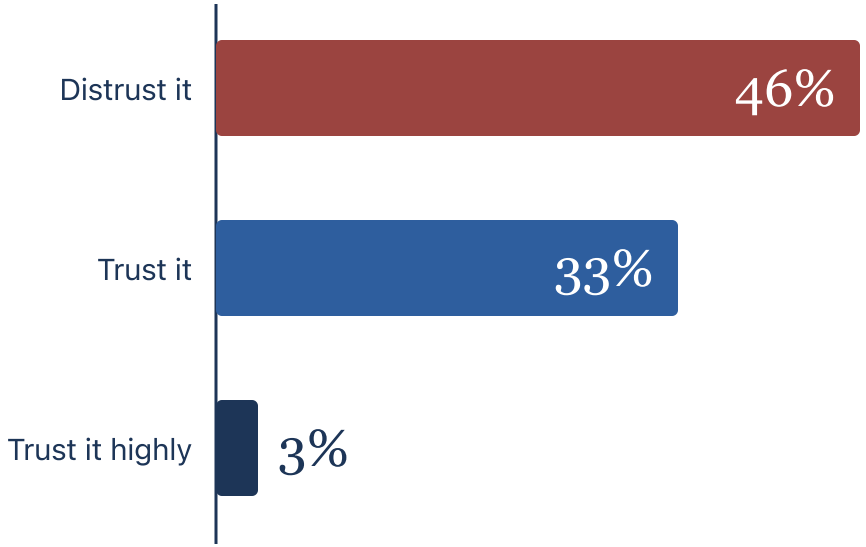
> The knowledge lives in files, so the agent works the same for the new owner as for the builder.

JUDGING TRUST

Trust the evidence, not the confidence

Developers on the accuracy of AI output

Share of respondents, Stack Overflow 2025



Agents sound just as sure when they are wrong.

Trust grows per task type: start with low-stakes views.

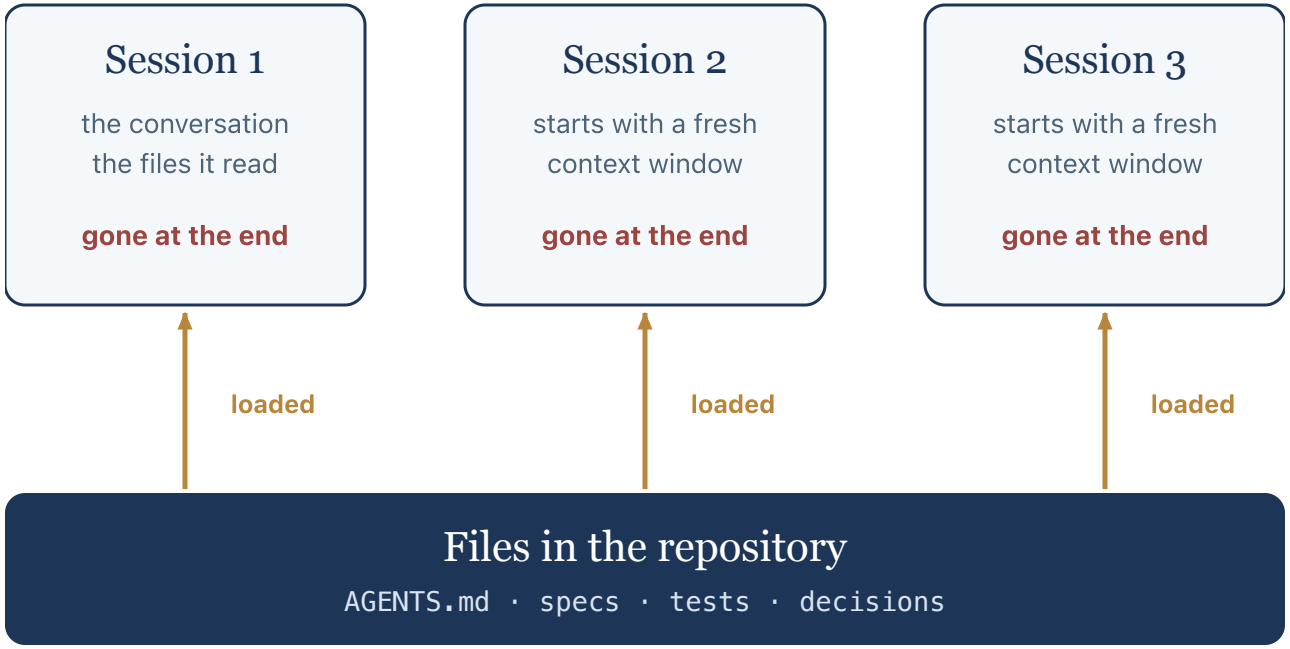
FIVE SIGNALS THAT EARN TRUST

- **The tests that cover the change ran and passed**
  evidence the build produced
- **The diff is small enough to read in ten minutes**
  evidence a person can check
- **The figure on screen matches the source system**
  evidence from outside the app
- **The agent's summary matches what the diff does**
  evidence the agent is describing its own work
- **Staging behaves as the contract says**
  evidence against the written baseline

Source: Stack Overflow Developer Survey 2025, AI section. "Trust it highly" is part of "Trust it".

What the model retains, and how far it acts alone

MEMORY: ONLY WHAT IS WRITTEN DOWN CARRIES OVER



Instruction files are guidance.
Hard limits belong in permissions and branch rules.

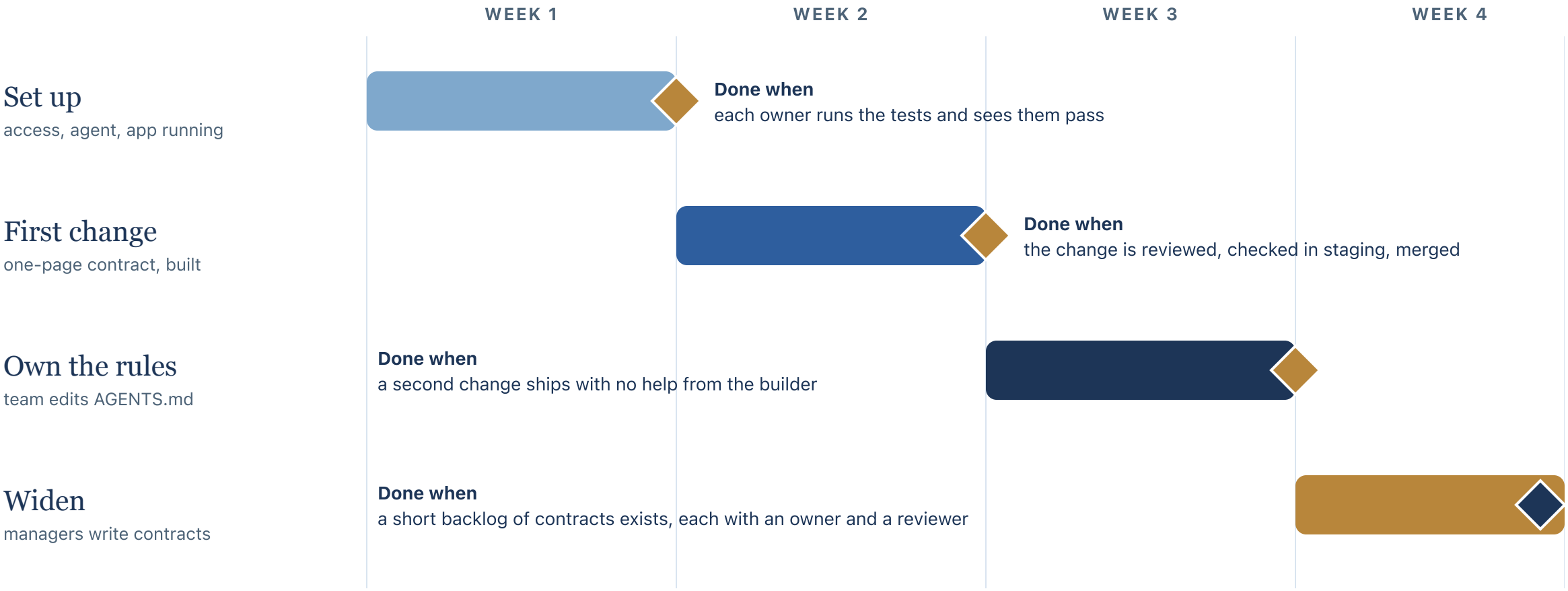
AUTONOMY: INSIDE THE REACH THE TEAM SETS



Sources: Claude Code docs, memory · Claude Code docs, subagents · Codex docs, AGENTS.md

A SUGGESTED 30-DAY START

Four weeks to a team that ships changes safely



Every week ends on something observable. That is the same rule as an accept line.

NEXT STEP

Starting an app this way, or taking one over?

What Blue Camel does

Builds data and AI applications with this method, and hands them over so the team that owns the app can change it safely.

Talk it through

jeff@bluecamelconsulting.com

bluecamelconsulting.com/work/ai-assisted-development