ILLUSTRATIVE SAMPLE · A FICTIONAL SYSTEM, NOT A CLIENT'S

Workshop pack: AI coding agents in a five-person team

Prepared by John Karanja · fixed scope · Sample, November 2026

A one-day, hands-on workshop for a five-person team. The team maintains a web application and has started using AI coding agents without agreed rules. The team builds one agent workflow from start to finish on its own code, and writes its own rules and review gate. This pack holds everything used on the day. Not included: building your product, tool licences, model training or certification.

What you leave with

1. **A shared rule file for your agents.** Written together on the day, for your own repository: what agents may change, what they must ask about, and what they must never touch.
2. **One working workflow.** The team builds it from start to finish on a real task, as a pattern to copy for the next one.
3. **A review gate.** Agent-written changes pass the same checks as human ones: a pull-request checklist for agent work, fitted to the checks your pipeline already runs.

Agenda

Morning: how agents fail, and one workflow built together

1. 09:00 to 09:30. Introductions, the chosen task, and what "done" means for it.
2. 09:30 to 10:30. How agents fail: changes nobody asked for, confident wrong answers, tests edited to pass, and work that cannot be traced.
3. 10:45 to 12:30. Build one workflow together, hands-on, on the chosen task.

Afternoon: rules and review gates on your own codebase

1. 13:30 to 14:30. Write the rule file for your repository.
2. 14:30 to 15:30. Build the review gate, and run it on the morning's change.
3. 15:45 to 16:30. A second, smaller task, run by a different pair using the rules and the gate.
4. 16:30 to 17:00. The one-page plan for the next month: who does what.

Before the day: planning-call questions

A 30-minute call, about a week before, to choose the task and the codebase.

- Which repository will we work in, and can everyone run it locally on the day?

- Which AI coding agents does the team use now, and who uses them most?
- What is one real task, about half a day's work for one person, that is ready to start?
- What checks does your pipeline run on every change: tests, formatting, type checks, anything else?
- Which parts of the code does nobody want an agent to touch, and why?
- Has an agent-written change ever caused a problem? What happened?
- Who reviews changes today, and how long does a review usually wait?

Rule file template

One file at the root of the repository, read by every agent and every person. Keep it short enough to read in two minutes. The team fills in the brackets on the day.

Agents may change

- Code inside the folders named for the task: [folders].
- Tests for the code they change, adding cases but never removing one.
- Documentation that describes the code they changed.

Agents must ask before

- Adding, removing or upgrading a dependency.
- Changing a database schema or a migration.
- Changing a public interface that other code or other teams rely on.
- Touching more than [number] files in one change.

Agents must never touch

- Secrets, keys, settings files with passwords, or anything in [secrets folder].
- Release, deployment or pipeline settings.
- Payment, authentication or permission code: [folders].
- An existing test, just to make it pass.

How agents work here

- One task per change, described in one sentence at the top.
- Run all the tests before proposing a change, and say which ones ran.
- When unsure, stop and ask. A question is better than a guess.

Review checklist for agent-written changes

The same checks as a human change, plus six questions the reviewer answers before approving.

1. Does the change do only what the task asked? Any file outside the task's folders needs a reason.
2. Do all pipeline checks pass, and were no tests removed, skipped or weakened?

3. Is every new dependency approved, as the rule file requires?
4. Could the reviewer explain each changed line to a colleague without the agent's help?
5. Is anything in the "never touch" list changed? If so, the change is refused.
6. Is the change small enough to undo on its own?

The hands-on exercise

The team works in pairs on the task chosen on the planning call. One person directs the agent. The other notes every point where the agent needed steering or correcting.

1. Before the agent sees the task, write it as one sentence and three "done when" checks.
2. Give the agent the task and the draft rules. Ask it for a plan first, and review the plan before any code is written.
3. Let it work in small steps. After each step, run the tests and read the change.
4. When it goes wrong, note what happened and which rule would have prevented it.
5. Run the finished change through the review checklist, as a second pair would.
6. Done when all three checks pass and the reviewer approves without the agent's help.

The notes from step 4 become new lines in the rule file in the afternoon.

Next-month plan template

One page, written in the last half hour, and reviewed on the follow-up call.

- **Owner of the rule file:** [name]. Reviews proposed changes to it each week.
- **Owner of the review gate:** [name]. Turns manual checklist items into pipeline checks.
- **Next three tasks for agents:** [task], [task], [task], each with its "done when" checks.
- **How we share lessons:** [where and when], for example ten minutes at the weekly meeting.
- **What we will measure:** [for example, changes refused at review, and why].
- **What we stop doing:** [one habit the day showed to be risky].

After the day

Two weeks later, one follow-up call reviews the plan, the rule file and anything the gate caught. If you hire me for team coaching within 60 days, the workshop's full fee comes off its price.

This is a sample: a fictional system, written like a real report. A real report follows the scope we agree in writing.

John Wainaina Karanja · UTC+3 · info@johnkaranja.com · johnkaranja.com/book