

THE 8 AGENTS I USE ANYTIME I BUILD SOMETHING

The command is `/build`. They run in order, each with a narrow job and a list of things it is not allowed to touch. The constraints are the part that makes it work.

THE RULE THE WHOLE CREW IS BUILT AROUND

No agent is allowed to check its *own* work.

THE FIRST FOUR

1 architect

Reads the actual codebase first, then writes the plan: ordered steps, the exact files to touch, and the risks a reviewer should check later.

Not allowed to write any code.

2 developer

Implements the plan end to end, matching the conventions already in the repo. Reports what changed and the exact command to test it.

3 review:simplify+e2e

Runs the test suite, then hunts redundancy and dead complexity. Told to leave the bug hunting to number 4.

4 review:adversarial

Told to break the change, not praise it. Goes after empty input, race conditions, permission gaps, off-by-one, injection. Runs at the same time as 3 and never sees its findings.

THE OTHER FOUR

5 review:screenshot

Launches the app, walks every affected screen, and flags layout breakage.

Only spawns when the build touched the frontend.

6 arbiter

Reads every finding, merges the ones pointing at the same root cause, drops the bikeshedding, and turns what survives into surgical instructions.

7 fixer

Applies only what the arbiter ordered, in priority order.

Not allowed to refactor past the instruction or add new behavior.

8 verifier

Re-runs the suite to confirm the fixes hold, and lists anything still broken.

Had no hand in the fixing. That is the point.

WHY IT IS BUILT THIS WAY

The one that wrote the code already decided. Everything after that is a defense. So nothing here reviews itself.

4 is the one that saves me