

What we will cover

How to organize checks, from generated code to merge.

- | | | |
|-----------|----------------------------|--|
| 01 | Check early | Test locally and prepare a clear PR. |
| 02 | Organize review | Coordinate the bots and sort their feedback. |
| 03 | Decide on the merge | Fix, verify and know when to stop. |
| 04 | Improve the harness | Learn from problems and check whether the team saves time. |

Projects and resources

Claude Code Ultimate Guide

cc.bruniaux.com

Claude Cowork Guide

cowork.bruniaux.com

ccboard

ccboard.bruniaux.com

CC Copilot Bridge

ccbridge.bruniaux.com

StarMapper

starmapper.bruniaux.com

The newcomers

Search Console MCP

search-console.bruniaux.com

wt-sweep

wt-sweep.bruniaux.com

github.com/FlorianBruniaux



florian.bruniaux.com



More agents. Who reviews their work?

When PRs arrive faster than we can verify them...

IN PARALLEL



More code alone does not mean faster delivery.

Organize checks with the harness



4 · COORDINATION

Queues · budgets · responsibilities

3 · EXECUTION

The bottleneck moves here



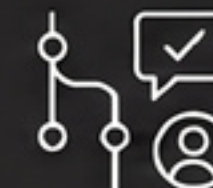
ON THE WORKSTATION

Local tests and review



IN CI

Checks and previews



ON THE PR

Reviews and decision

Permissions · evidence · costs



2 · REPO

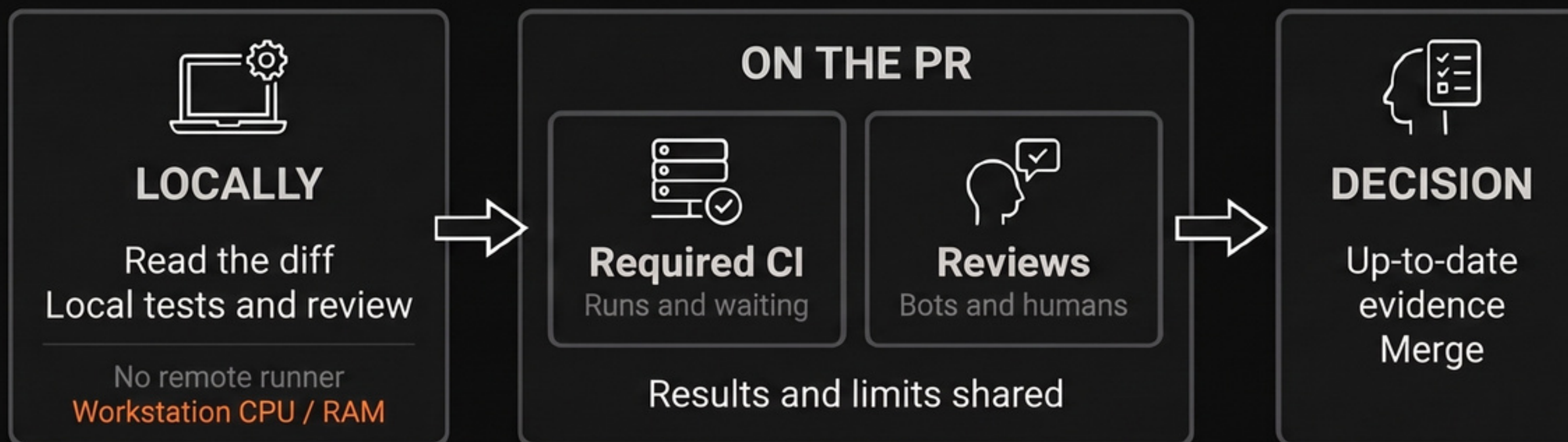
Versioned instructions and tests



1 · WORKSTATION

Tools and configuration

Check early, validate in CI



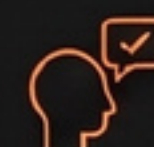
Finding an error later can multiply costs and delays



Additional runs



Waiting



Review to repeat

Local checks prepare the PR. Required CI remains mandatory.

A PR the team and agents can understand

TL;DR

After a `git pull` that changes `pnpm-lock.yaml` or `pnpm-workspace.yaml`, the `post-merge` and `post-rewrite` hooks failed during `ai:sync` because pnpm refuses scripts until `node_modules` is up to date. Both hooks now run `pnpm install --frozen-lockfile` first, as post-checkout already does.

- ▶  Modeling
- ▶  Business rules: the rule that changes
- ▶  Business: what changes if it is wrong
- ▶  Tech: what changes in the code
- ▶  Tests: what is checked and what is not
- ▶  Docs: what must follow
- ▶  Checklist
- ▶  AI and details
- ▶  Local harness details

For people

Understand the problem and the change.

For agents

My skill reads “AI and details”: a section optimized for AI.

For everyone

Tests performed, limits and open questions.

What could this change break?

Precise checks, observable results.

WHAT CHANGES

AI CAN CHECK

PEOPLE CAN ALSO

**A button
or interface**

Inspect the DOM and test clicks.
Check mobile rendering.

Check readability
and the actual journey.

**A shared
component**

Write / run E2E tests
on affected pages.

Try sensitive journeys.
Find overlooked cases.

**Access
rights (RBAC)**

Test roles: allowed and denied
access, through the UI and API.

Confirm who needs access.
Review risky cases.

Complementary roles, not exclusive ones. The usual tests remain required.

GitHub labels to route review

Who should review this PR?

EXAMPLE LABELS



Review AI

Review assigned to AI

A bounded, understood change that tests can verify.
Within a scope authorized by the team.



Review Human

Human review required

Access rights, private data, calculations, payments or unclear consequences.
When in doubt, ask for human review.

FOR PEOPLE: PRS TO REVIEW

is:pr is:open label:"Review Human"

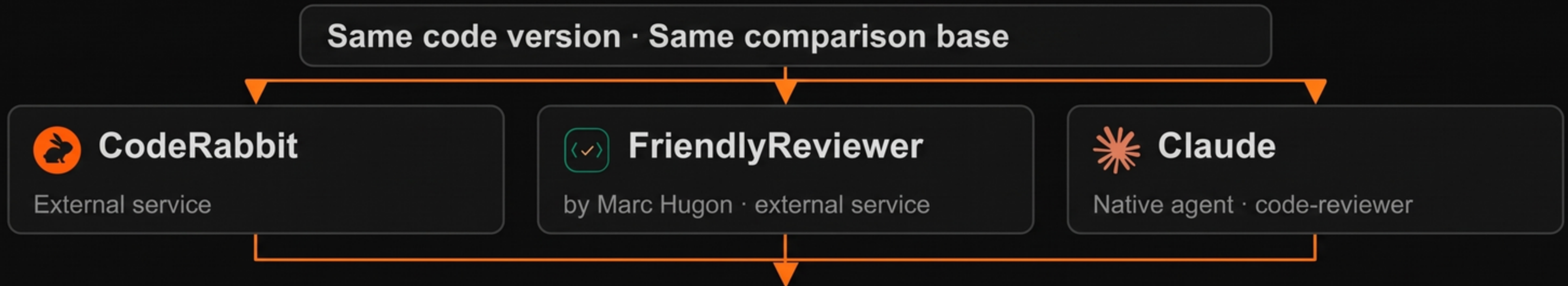
FOR AI: PRS TO PROCESS

is:pr is:open label:"Review AI"

A cue for people. A filter for AI.

Required checks remain mandatory.

3 readers / 1 review to consolidate



/tech:pr-feedbacks

My skill for processing all feedback on the PR

1 · Collect

All feedback
on the PR

→ 2 · Validate

Read, cross-check and verify
against current code

→ 3 · Correct

Run fixes with
multiple agents / LLMs

After fixes: rerun the relevant tests and reviews.

For people: disagreements, limits and decisions to make.



A good diagnosis does not guarantee a good fix

The bot is right about the problem. Its solution still needs checking.

SIMPLIFIED EXAMPLE: 2 CASES, 2 EXPECTED ERRORS

1 · PROBLEM FOUND

The test accepts any error.

- Missing document → an error
- Forbidden access → an error

It can pass for the wrong reason.

2 · BOT SUGGESTION

Expect the same message in both cases.

- Missing document → “Denied”
- Forbidden access → “Denied”

More precise, but still incorrect.

3 · FIX TO MAKE

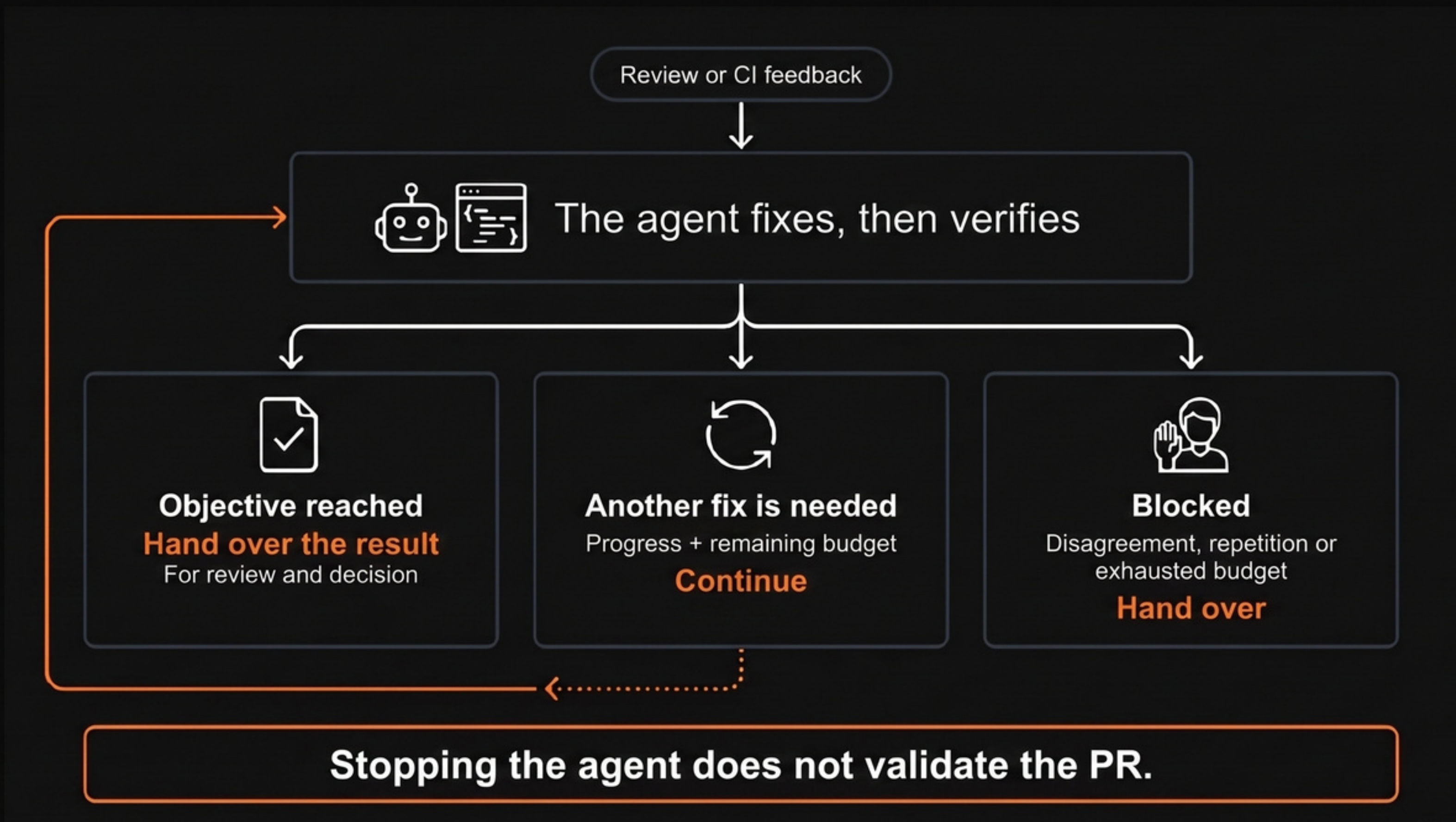
Check the error for each case.

- Missing document → “Not found”
- Forbidden access → “Denied”

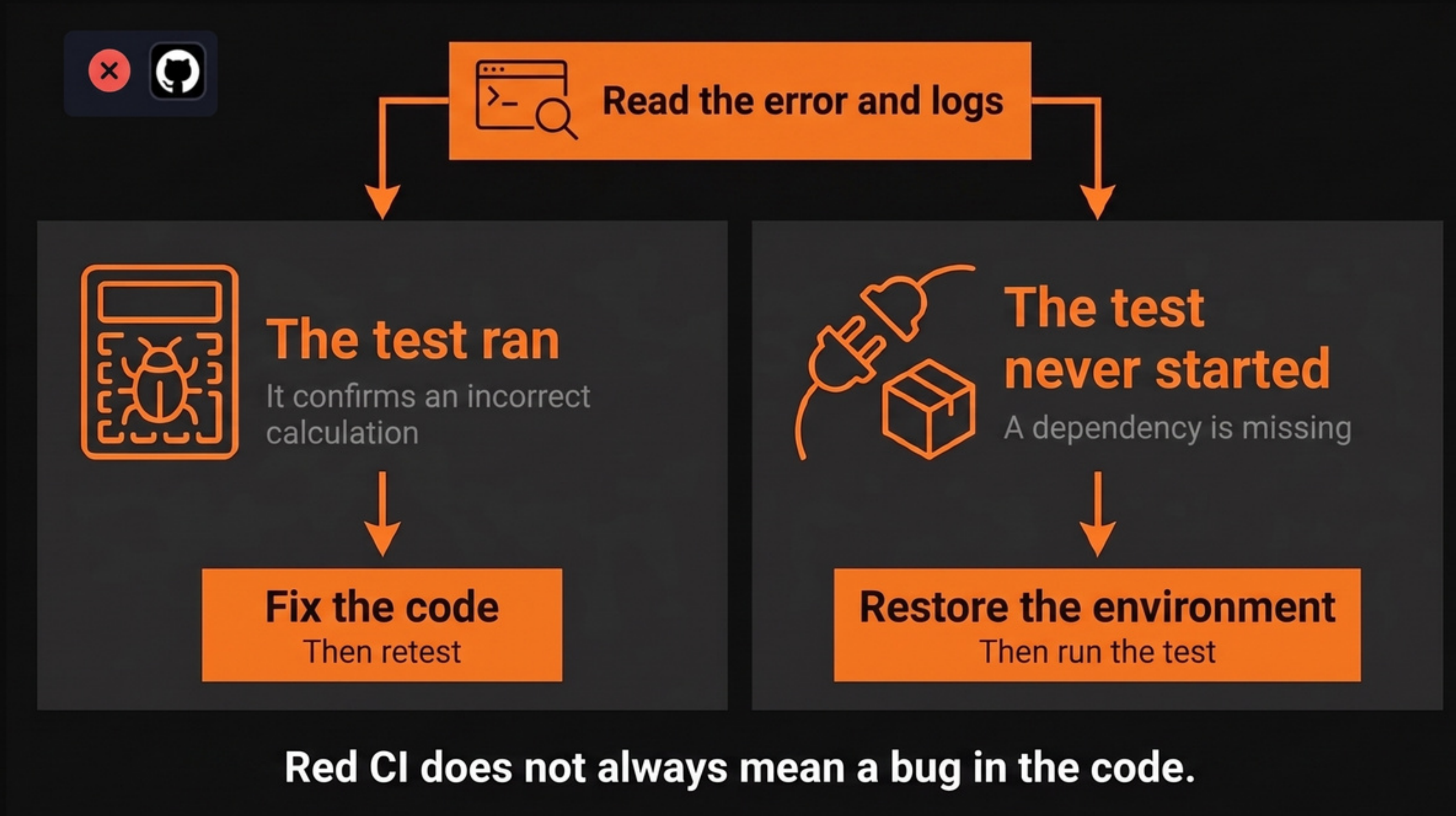
Adapt assertions, then retest.

Verify the reported problem. Then verify the proposed fix.

When should the agent stop fixing?

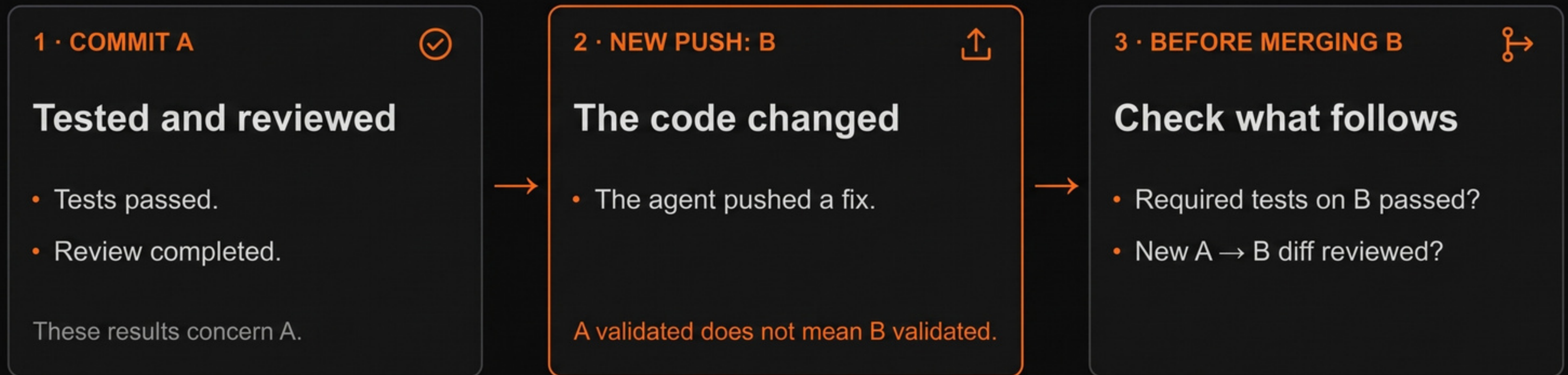


CI is red. Should we change the code?



A new push changes what still needs checking

Tests and review concern a specific code version.

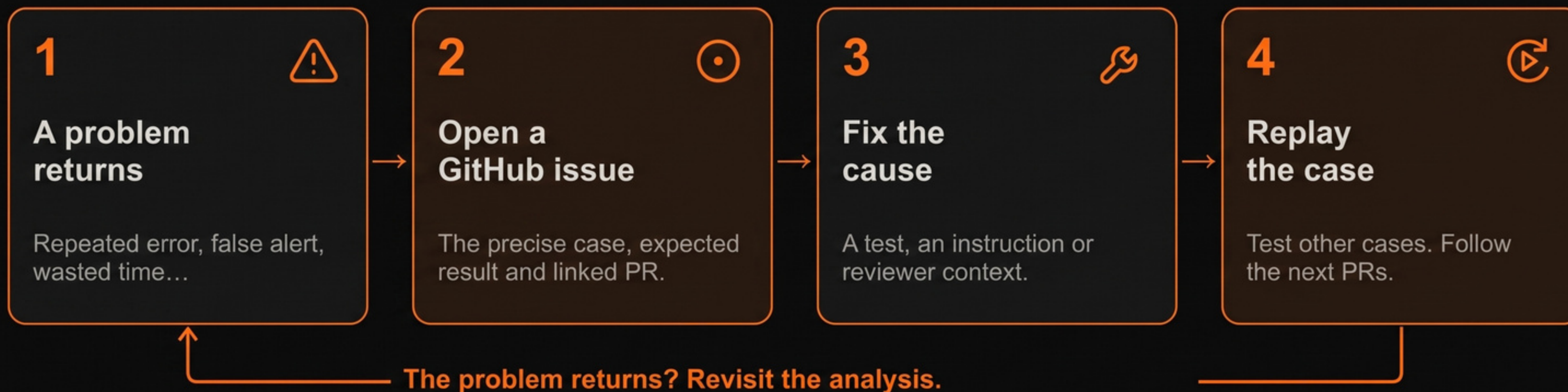


After every push: identify what needs retesting and review.

A running, skipped or canceled test is not a passed test.

When a problem returns, improve the harness

An issue to preserve evidence. A change to verify.



EXAMPLE

The agent misses the “empty list” case.



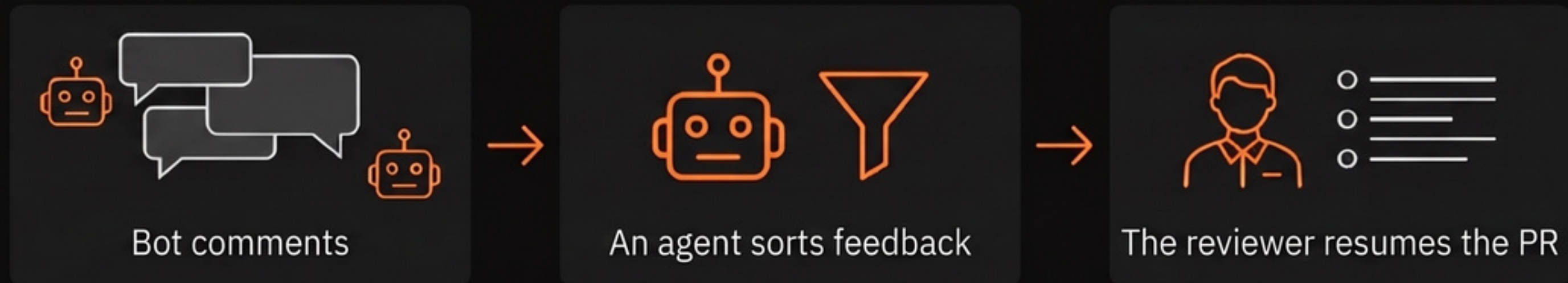
Regression test
+ clearer instruction if needed.



Check this case
and other behaviors.

At Aristote, we deliver faster

A feature in a few hours. An epic in a few days.



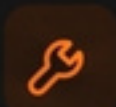
Compare similar PRs, before and after

	Less human time?	Reading and sorting comments
	Less waiting?	From review request to merge
	Less rework?	Corrections and bugs after delivery
	At what cost?	Models and CI, including failed runs

What could we do better?

Return to a recent PR. Choose 1 improvement to try.

REVISIT A PR

-  What went wrong?
-  What took the most time?
-  What will we change on the next PR?

1 Try 1 change



Choose a concrete action.
Test it on the next PR.

2 Check whether it helps



Less waiting? Less rework?
Without letting more bugs through.

Keep what helps the team. Adjust or abandon the rest.

The harness, from need to delivery

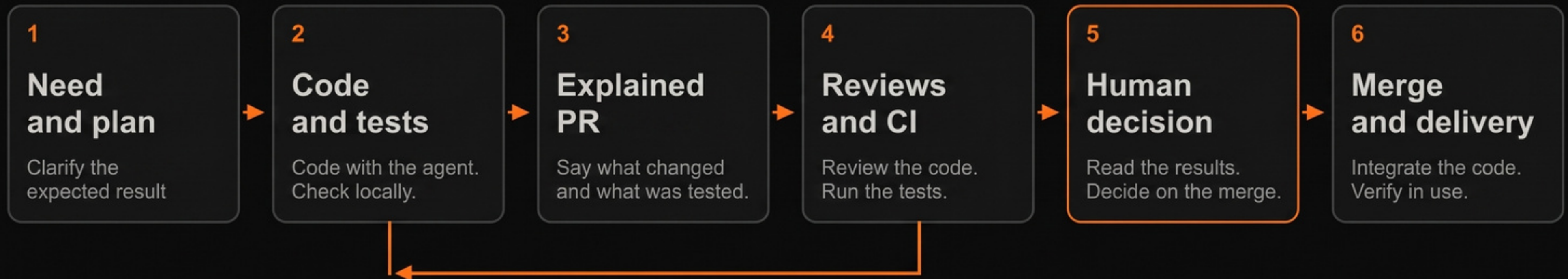
The harness **evolves** with the code, tools and team. It needs **technical skills** and **organization**.

THE SHARED FRAMEWORK Instructions and rules · skills and tools · permissions and budgets

 ON THE WORKSTATION

 ON THE PR AND IN CI

 WITH THE TEAM



Fix, then rerun the relevant tests and reviews

AFTER DELIVERY

Bugs, feedback, time and costs → improve rules, tests and tools

Questions?



[Portfolio](#)



[GitHub](#)