

Debugging Workflows

A cheat sheet. Five flows for the bugs that show up most. Pick by the symptom, not by habit.

01 Trace it forward

When the error has a clear entry point.

- Paste the failing call + stack trace into Ask.
 - Ask: "walk me through how this gets called and what state it touches."
 - Don't ask for a fix yet — find the wrong assumption first.
- > "Here's the call: [paste]. Trace the data through to the error. Where could state diverge?"

Watch out: fixing before tracing leads to whack-a-mole.

02 Diff it red→green

When tests just started failing on a known-good branch.

- git diff against last green commit.
 - Paste diff + failure message into Agent.
 - Constrain it: only revert / patch within the diff.
- > "Tests started failing after this diff: [paste]. Failure: [paste]. Find the cause inside the diff. Don't touch unrelated code."

Watch out: let Agent rewrite half the file and you've lost the bisect signal.

03 Reproduce, then fix

When the bug is reported but you can't see it.

- Spend the first 10 min on a failing test, not on a fix.
 - Once red, switch to TDD-with-Agent flow.
 - If you can't reproduce, the report is incomplete — go back.
- > "Here's the bug report: [paste]. Write the smallest test that reproduces it. Don't implement the fix yet."

Watch out: fixing what you can't reproduce wastes everyone's time.

04 Print, observe, prune

For state and timing bugs the model can't see.

- Ask Agent for instrumentation only — log statements, not fixes.
 - Run, capture output, paste back into chat.
 - Now Agent has the context that was missing.
- > "Add console.log around [function]. Don't change behavior. I'll run it and paste the output back."

Watch out: leaving log statements in. Have Agent rip them out at the end.

05 Bisect by hand

When AI guesses the wrong cause repeatedly.

- git bisect — do it yourself, faster than convincing the agent.
 - Land on the offending commit, then narrow to a hunk.
 - Paste just that hunk + symptom into Ask.
- > "This commit caused [symptom]. Here's the diff: [paste]. What's the most likely connection?"

Watch out: starting with bisect on small bugs. Use it when other flows have failed twice.

06 Step through with Debug Mode

Cursor's purpose-built failure helper.

- Trigger from the error itself — "Debug with AI" on red output.
- More focused than Ask: it sees the failure context automatically.
- Underused. Reach for it before Ask on test failures.

Watch out: Debug Mode is read-only. Switch to Agent when you've found the fix.

Triage by symptom

First 30 seconds. Pick the flow before opening a chat.

Stack trace, single file, named line	Debug Mode — let it diagnose at the failure point
Tests are red, multiple files involved	Agent + failing test prompt — let it iterate
Behavior is wrong but no error	Ask first — have the model trace the data path
Heisenbug — passes alone, fails together	Plan Mode — state-leak hunt needs strategy first
Fixed it once, now broken again	Bisect manually, write a test — stop the cycle

Anti-patterns

If the bug isn't dying, often it's the workflow, not the model.

- Fixing without reproducing.** If you can't reliably hit the bug, you can't verify the fix. Demand a failing case first.
- Letting Agent rewrite the function.** Big diff hides the bug. Constrain to the smallest patch that turns red → green.
- Losing the diff signal.** If git diff was your friend, don't accept changes that nuke the diff. Checkpoint each turn.
- Same prompt, twice.** If a fix attempt failed, change the prompt or the flow — not just the model.

The diff is the spec. Every debug session should end with a smaller, narrower diff than you'd write by hand. If Agent's patch is larger than yours would be, ask why before you accept.