EcomLabs

CHECKLIST AND TEMPLATES

Review an AI agent's changes to your tests

The three reviews that catch a deleted, skipped or weakened test, the git commands to run them, the rule ready for CLAUDE.md or AGENTS.md, and the hook and CI step that enforce it.

FOR	Vitest projects where an AI agent writes code
TESTED WITH	Node 22.22.3 · Vitest 5.1.0 · git 2.54 · Windows 11
UPDATED	November 2, 2026
DOCS	code.claude.com/docs/en/hooks
ARTICLE	ecomlabs.dev/g/trampa-en
AUTHOR	Luis Morales · @luismoralesdev

Contents

Step 01	Review checklist
Step 02	The rule for CLAUDE.md or AGENTS.md
Step 03	The PreToolUse hook
Step 04	The CI step

STEP 01

Review checklist

Before you accept an agent's change, check:

- [ ] You read `git diff HEAD --stat` (or `main...branch`) before the code.
- [ ] No test file shows up in `--diff-filter=D`.
- [ ] No new `.skip`, `.only`, `.todo`, `xit` or `xdescribe`.
- [ ] The test count is the same as before or higher, with no new `skipped`.
- [ ] Every expected value that changed has a business reason.
- [ ] No new mock accepts any argument where a specific one used to be checked.
- [ ] Without the fix (`git stash push -- src`), the new tests fail.

The commands, in this order:

```
# The agent's uncommitted changes
git diff HEAD --stat
git diff HEAD --diff-filter=D --name-only
git diff HEAD -- '*.test.*'

# Do the new tests fail without the fix?
git stash push -- src
npx vitest run
git stash pop
```

Once it has committed, replace `HEAD` with `main...its-branch`; without `HEAD` you miss files removed with `git rm`. If the fix created files in `src/`, add `-u` to the stash.

STEP 02

The rule for CLAUDE.md or AGENTS.md

Paste it into `./CLAUDE.md` or `./.claude/CLAUDE.md`, or into `AGENTS.md`, which Claude Code reads when there is no `CLAUDE.md` (from 2.1.277). It is context, not an enforced rule: that is step 03.

```
## Tests

- Never delete, skip (.skip, .only, .todo, xit) or weaken a test or an expected value to make the suite pass.
Fix the code.
- Do not replace a real check with a mock that accepts anything (expect.anything(), a mock that ignores its
arguments).
- If a test looks wrong, stop and ask me before you touch it. Explain why.
- When you finish, list every test file you changed and why.
```

STEP 03

The PreToolUse hook

Save it as `.claude/hooks/guard-tests.mjs`. It blocks (exits 2) an `Edit` or `Write` to a test file that adds `.skip`, `.only`, `.todo` or a Jest alias, or that leaves fewer `it(` or `test(` calls:

```
// PreToolUse hook for Edit|Write: blocks changes that skip or remove tests.
// Exit 0 lets the tool run; exit 2 blocks it and Claude reads stderr.
import { existsSync, readFileSync } from "node:fs";

const { tool_name, tool_input } = JSON.parse(readFileSync(0, "utf8"));
const file = tool_input.file_path ?? "";

if (!/\.(test|spec)\.[^\\/]+$|[\\/]__tests__[\\/]/.test(file)) process.exit(0);

const before = tool_name === "Edit" ? tool_input.old_string : existsSync(file) ? readFileSync(file, "utf8") :
"";
const after = tool_name === "Edit" ? tool_input.new_string : tool_input.content;

const SKIP = /\.(skip|skipIf|only|todo)\s*\(|\b(xit|xtest|xdescribe|fit|fdescribe)\s*\(/g;
const TEST = /(?<![.\w])[xf]?(it|test)(\.\w+)*\s*\(/g;
const count = (re, text) => (String(text ?? "").match(re) ?? []).length;

const problems = [];
if (count(SKIP, after) > count(SKIP, before)) problems.push("adds .skip, .only, .todo, xit or xdescribe");
if (count(TEST, after) < count(TEST, before)) problems.push("removes an it() or test() block");

if (problems.length) {
  console.error(
    `Blocked: this change to ${file} ${problems.join(" and ")}. ` +
      "Do not skip, delete or weaken a test to make it pass: fix the code. " +
      "If the test itself looks wrong, stop and ask the user.",
  );
  process.exit(2);
}
```

Register it in `.claude/settings.json`:

```
{
  "hooks": {
    "PreToolUse": [
      {
        "matcher": "Edit|Write",
        "hooks": [
          {
            "type": "command",
            "command": "node",
            "args": ["${CLAUDE_PROJECT_DIR}/.claude/hooks/guard-tests.mjs"]
          }
```

```
        ]
      }
    ]
  }
}
```

> **CAUTION**
>
> Tested by piping JSON into it, not inside Claude Code. It does not see changes Claude makes through `Bash` or `PowerShell` (an `rm`) and cannot judge expected values: that is step 04 and your review.

STEP 04

The CI step

Save it as `scripts/check-tests.mjs`. It fails when the branch deletes tests, adds a modifier, or has fewer tests than its base (per `vitest list --json`):

```
// Fails when this branch deletes test files, adds .skip/.only/.todo, or has fewer
// tests than the commit it branched from. Usage: node scripts/check-tests.mjs origin/main
import { execFileSync } from "node:child_process";

const base = process.argv[2] ?? "origin/main";
const git = (...args) => execFileSync("git", args, { encoding: "utf8" }).trim();
const TESTS = ["*.test.*", "*.spec.*", "*__tests__/*"];
const SKIP = /\.(skip|skipIf|only|todo)\s*\(|\b(xit|xtest|xdescribe|fit|fdescribe)\s*\(/;
// vitest list --json prints one entry per test it would run (skipped and todo are left out).
const countTests = () =>
  JSON.parse(execFileSync(process.execPath, ["node_modules/vitest/vitest.mjs", "list", "--json"], { encoding:
"utf8" })).length;

if (git("status", "--porcelain")) {
  console.error("Commit or stash your changes first: this script checks out the base commit.");
  process.exit(1);
}

const problems = [];
const deleted = git("diff", "--diff-filter=D", "--name-only", `${base}...HEAD`, "--", ...TESTS);
if (deleted) problems.push(`Deleted test files:\n${deleted}`);

const added = git("diff", "-U0", `${base}...HEAD`, "--", ...TESTS)
  .split("\n")
  .filter((line) => line.startsWith("+") && !line.startsWith("+++") && SKIP.test(line));
if (added.length) problems.push(`Added skip/only/todo:\n${added.join("\n")}`);

const branch = git("rev-parse", "--abbrev-ref", "HEAD");
const back = branch === "HEAD" ? git("rev-parse", "HEAD") : branch;
const headCount = countTests();
git("checkout", "--quiet", "--detach", git("merge-base", base, "HEAD"));
let baseCount;
```

```
try {
  baseCount = countTests();
} finally {
  git("checkout", "--quiet", back);
}
console.log(`Tests: ${baseCount} on ${base}, ${headCount} on this branch.`);
if (headCount < baseCount) problems.push(`Fewer tests: ${baseCount} -> ${headCount}`);

if (problems.length) {
  console.error(problems.join("\n\n"));
  process.exit(1);
}
console.log("OK: no test was deleted, skipped or lost.");
```

In `.github/workflows/tests.yml` (with `node_modules` in `.gitignore`):

```
name: Tests

on:
  pull_request:

jobs:
  test:
    runs-on: ubuntu-latest
    timeout-minutes: 10
    permissions:
      contents: read
    steps:
      - uses: actions/checkout@v6
        with:
          fetch-depth: 0
      - uses: actions/setup-node@v7
        with:
          node-version: 22
      - run: npm ci
      - name: No deleted, skipped or lost tests
        run: node scripts/check-tests.mjs origin/${{ github.base_ref }}
      - run: npx vitest run
```