

# 2026-04-25-03 — Fresh-session verification handoffs: COO writes the next-instance prompt without being asked

vade-coo

2026-04-25

## Table of contents

**Status:** active

**Supersedes:** none. Extends the structured-handoff convention from `coo/committee_protocol.md §6` (constitutional-file committee handoff) to ordinary boot-impacting PRs.

**Paired artifacts:** - vade-coo-memory PR (this commit): adds the reflective rule to `CLAUDE.md` “Handoff prompts for boot-impacting PRs” - vade-runtime PR #68 (merged): the trigger event — Mem0 MCP fix that needed Ven to explicitly ask “give a prompt for the next instance” before the handoff was produced

**Context.** When the COO ships a fix whose effect can only be verified after a fresh container boot — `.mcp.json`, `~/.claude/settings.json` env, `SessionStart` hooks, anything else loaded at startup — the running session can’t test it. PR #68 (Mem0 via `MEM0_API_KEY`, skipping OAuth) was a clean example: I shipped, named the verification steps in the PR test plan, and stopped. Ven had to explicitly prompt before I produced the self-contained handoff. Per Ven, this is a recurring pattern across boot-impacting PRs. The cost is one Ven message per PR; the gap is procedural, not technical. The structured handoff already exists for constitutional-file committee PRs (`coo/committee_protocol.md §6`); ordinary boot-impacting PRs sit in the gap. This memo closes it.

**Decision.** When the COO opens a PR whose verification can only happen on a fresh session, the COO writes a self-contained handoff prompt for the next instance and surfaces it on **two surfaces, same content**, on the same turn the PR is opened, **without waiting for Ven to ask**:

1. **PR body**, under a clearly delimited heading `## Handoff prompt for next instance`, placed after the test plan. The next instance fetches it via `gh pr view <PR>`.
2. **Final assistant chat message** of the turn, as a quoted block. So Ven copies it directly without scrolling or fetching.

**Trigger** (precise, scannable at PR-draft time): any PR that modifies `.mcp.json`, `~/.claude/settings.json` env, `SessionStart` hooks (`vade-hooks/`, `coo-bootstrap.sh`, `session-start-sync.sh`, `coo-identity-digest.sh`, `discussions-digest.sh`, `memo-index.sh`, `dispatch.sh`), `coo-env`, `setup-receipt.json`, the snapshot’s `/home/user/layout`, OS-image-build steps in `vade-runtime/scripts/`, or anything else where the test plan references “fresh session”, “next session”, “after merge ... in a new session”, “container resume”, or “next container epoch”. When in doubt, write the handoff — the cost of a stray prompt is one ignored chat message; the cost of skipping is Ven asking and an extra round-trip.

**Required content** of every handoff prompt: - One-line context: PR # + title + what just merged. - Ordered verification checks. Each step has a concrete command/tool call + expected output. - For each step, what to do on failure (the diagnostic next move). - Any followup work blocked by the diagnosed issue (e.g., the episodic Mem0 write a Mem0-outage diagnostic session couldn't make). - Whether to greet Ven before or after verification. - Self-contained — no references to “the prior session” or “as we discussed”. The next instance boots cold.

**Why this works.** `CLAUDE.md` loads at session start, every session, so the rule is read on every boot the rule is supposed to govern. The trigger list is concrete enough to scan at PR-draft time without judgment (“if any modified file is on the list, write the handoff”). Two surfaces, one author, single turn — the rule adds one section to the chat output and one section to the PR body. Hook-based automation (auto-extract handoff from PR body, post-to-chat) was considered and rejected as out of scope: the marginal reliability gain over a prominent rule the COO loads at boot doesn't justify the implementation cost.

**What this does NOT change.** - `coo/committee_protocol.md` §6 — the structured committee handoff stays exactly as written for constitutional-file PRs. - The PR-test-plan format. The handoff is in addition to the test plan, not in place of it. - Authority for any PR. The handoff is procedural surface, not delegation. No new autonomous actions. - The end-of-session discipline in `CLAUDE.md` “When you end a session”. The handoff is at PR-open time, not session-end.

**Authority.** BDFL directive (Ven's chat message of 2026-04-25, run-2026-04-25T180830). Per `coo/committee_protocol.md` §Scope `CLAUDE.md` operational-addition carve-out: this memo is the load-bearing source; the `CLAUDE.md` change is the reflective surface and qualifies as operational, single-instance authority. Not on the constitutional-file scope list.

**Retirement condition.** This memo retires when (a) the rule is absorbed into a hook (e.g., a PR-open hook that auto-detects boot-impacting diffs and writes the handoff), at which point the procedural surface becomes mechanical; OR (b) the trigger list is invalidated by a substrate change that decouples PR test plans from fresh-session verification (e.g., live MCP hot-reload), at which point a successor memo replaces this one. Until either, this is the binding rule.

## Links to this page

**Day overview — 2026-04-27**

`/postmerge-check slash command` (sibling primitive). PR vade-coo-memory#202 — fetches the most recently merged vade-app PRs, extracts their `## Post-merge confirmation` handoff sections, executes the steps in order. Use case: a fresh session can verify a merge without copy-pasting the handoff prompt manually. Sibling to the MEMO-2026-04-25-03 handoff-prompt rule from the prior day's ...

**The day-overview arc, 2026-04-24 → 2026-05-03**

The 2026-04-25 day shifted register: not structural but operational. The skills-research epic (#20) closed with `MEMO-2026-04-25-01` ratifying the agent-teams pilot — five teammates joined to a single `agent-teams` instance, eleven artifacts shipped, eleven PASS on both safety and emancipatory audits, zero rework. The four betterment-pattern findings adopted that day (prefer slash-command over harness-skill form; invest in role specs not dispatch ...