FILE THE WORKFLOW, NOT THE CONVERSATION

Turn a reviewed pass into a workflow another writer can run

After the team has reviewed two successful runs of the same pass, document the files, modes, output, approval point, and stop conditions another writer needs. Test the workflow without the original chat. If the test exposes a gap in the procedure, fix the canonical file. *Anything set as dialogue is an instruction you can give the agent word for word; adapt paths and filenames to the production.*

PREPARED FOR THE AVS SCREENWRITING TEAM · JULY 2026

FADE IN:

THE SECOND REVIEWED SUCCESS BELONGS IN A FILE

One reviewed run may depend on tacit context. After a second reviewed success, compare the runs and document the sequence, constraints, and human decisions that produced the accepted output shape. The result is ready for a fresh-session test, but it is not yet trusted team procedure.

Rule 1: File only work the team expects to repeat.

Rule 2: State files, modes, outputs, approvals, and stop conditions.

Rule 3: When a run exposes a defect, correct the workflow.

1. **Choose a pass with two reviewed successes.** Start with self-coverage: a bounded task the team has run at least twice and whose outputs a writer reviewed. Wait to document prediction scoring, outline stress tests, panel reads, or condensing until each has also produced two reviewed results. Name the stable goal and the human decision the output supports. Put production-specific exceptions in `projects/[production]/project-instructions.md`.

   Document criteria the team has already reviewed. An agent finishing the run does not approve its output.

2. **Reconstruct the decision logic, not the transcript.** Ask the agent to identify what had to be known, what happened in what order, what was prohibited, and where a person chose the next move:

   The Writer
   (to the agent)

   ```
   PROPOSE only. Compare the reviewed runs in
   projects/return-visit/coverage/2026-07-03-draft-2.md and
   projects/return-visit/coverage/2026-07-10-draft-3.md.
   Report whether workflows/self-coverage.md and
   workflows/README.md exist. Preview the workflow's full
   proposed contents in chat, or a minimal diff if it
   exists. Include goal, named inputs, ASK-only steps,
   output shape, approval point, and stop conditions.
   Do not create or change files. Wait for a separate
   APPLY instruction naming the target.
   ```

   You should get: *an in-chat preview whose goal, inputs, modes, output, approval point, and stop conditions can be checked.* ***Next:*** *find hidden memory such as "as discussed," "the usual files," or "use the standard rubric."*

3. **Strip away conversational debris.** Delete greetings, retries, abandoned wording, and details that belong only to the completed production. Keep the sequence, constraints, definitions, evidence requirements, and decision gates that made the pass trustworthy. Replace "as discussed" with the actual rule and "the usual material" with named paths. If the purpose of a step cannot be stated, the procedure is not ready for team use.

4. **Make every dependency and mode explicit.** Name each required file with an exact path or a placeholder such as `[target-draft-path]`, `[rubric-path]`, or `[reviewer-profile-path]`. State what happens if one is missing, stale, contradictory, or unreadable. Begin every numbered step with **ASK**, **PROPOSE**, or **APPLY**. ASK reads and reports without file changes; PROPOSE previews in chat without file changes; APPLY changes only files named in a writer-approved instruction.

5. **Design the output before polishing the instructions.** "Up to three concerns supported by citations, with evidence, inference, and conflict separated" can be checked; "helpful coverage" cannot. Say whether the run reports in chat or writes to an approved path, where the writer decides, and what the run must never change. A one-page workflow should support one decision and one output shape. Split it when either changes.

   ONE-PAGE WORKFLOW ANATOMY

   GOAL *find cited concerns before the writer approves a draft*
   USE / NOT *use for preflight coverage; do not revise or predict notes*
   INPUTS *target draft path · rubric path · reviewer-profile path*
   STEPS *numbered; each begins ASK, PROPOSE, or APPLY*
   OUTPUT *in-chat coverage with citations and separated reasoning*
   APPROVAL *writer chooses whether a cited concern merits revision*
   STOP *stop for missing or unreadable input, or expanded scope*
   STATUS *owner, version, trust status, last-tested production and date*

6. **Test for hidden context.** The author of a workflow is the worst person to spot what it assumes. Before saving it as trusted, ask for an adversarial read:

   The Writer
   (to the agent)

   ```
   ASK only. Review the in-chat preview proposed for
   workflows/self-coverage.md. List each missing path,
   undefined term, hidden assumption, or unclear mode.
   Do not rewrite the preview or change any file.
   ```

   You should get: *a cited list of missing or ambiguous instructions.* ***Next:*** *correct the preview, approve it, apply only the named workflow files, then test the saved file in a fresh session.*

   After the writer approves the preview, use a separate APPLY instruction naming `workflows/self-coverage.md` and `workflows/README.md`; no other file may change. Save the approved procedure under `workflows/`. In `workflows/README.md`, record its goal, owner, version, and status: **trusted**, **untrusted**, or **retest required**. Inspect both diffs before trusting the workflow.

   READY WHEN

   In a fresh session, a second writer can identify the goal, required files, prohibited actions, output shape, stop conditions, and approval point from the workflow before running a step.

CUT TO:

RUN THE WORKFLOW BY NAME

Once a procedure passes the fresh-session test, run it by filename and target. The workflow states the reviewed method; the request names this production's draft. The agent must still stop at every documented approval point and before any unapproved file change.

The Writer
(to the agent)

```
ASK only. Run workflows/self-coverage.md on
projects/return-visit/draft-3.pdf. Use the rubric and
reviewer profile paths named by the workflow.
Return the documented output in chat. Stop on any
missing or unreadable input. Do not create or
change files.
```

You should get: the same output shape promised by the workflow. Next: compare the run with the procedure before judging the prose.

Review on two levels. Separate whether the agent followed the procedure from whether the procedure described quality well enough:

- **Execution failure:** the run skipped or misread a clear instruction, or used the wrong mode. Cite the workflow line, restore any unauthorized change, and rerun from inspected inputs without changing the procedure.
- **Procedure failure:** the run followed the written instructions, but those instructions allowed a required check or output to be omitted, blurred, or contradicted. Repair the canonical workflow before rerunning it.

Use the failed check to decide which case occurred. A longer one-off prompt or hand-repaired output does not correct a defect in the canonical workflow.

Repair the line that allowed the failure. Treat a failed run as evidence about the procedure. Locate the instruction that permitted the result, then request a minimal diff:

The Writer
(to the agent)

```
PROPOSE only. The output at [failed-output-path] did
not meet this requirement:
[missing-or-incorrect-result]. Compare that output with
[workflow-path]. Cite the workflow line that allowed
the failure and preview the smallest preventive
diff in chat. Do not change either file.
```

You should get: one diagnosed defect and a minimal procedure diff. Next: confirm `[workflow-archive-path]` is unused, then issue a separate APPLY naming that archive and `[workflow-path]`. Inspect both changes and rerun from inspected inputs.

Keep one owner for the canonical file. Use another APPLY limited to `workflows/README.md` to record the change, version, and last test. If a rubric, reviewer profile, or folder schema changes, mark affected workflows for retest; a prior result does not test the changed dependency.

Split workflows that require two decisions. A one-page workflow should support one goal and one writer decision. If coverage instructions also ask for prediction filing, revision, or another decision, identify the boundary:

The Writer
(to the agent)

```
ASK only. Review workflows/self-coverage.md. Does it
mix coverage with prediction filing, revision, or
another decision? If so, show where the boundary
belongs and why. Do not change any file.
```

You should get: a reasoned keep-or-split decision. Next: split only when the reader would otherwise make two different decisions.

Not every task needs a shared workflow. Discovery, creative judgment, and unusual production needs may remain direct conversations. Document only a bounded, repeated team pass that becomes easier to run and inspect from a canonical file.

Keep workflows ready to run.

- **Name workflows by outcome, not tool:** `self-coverage.md`, not `claude-prompt.md`.
- **Use named inputs.** Paths and source roles replace "the usual files."
- **Require writer approval.** Every APPLY instruction names the approved file and change.
- **Record version and last test.** Include production, date, and the person who reviewed the result.
- **Retest dependencies.** Rubric, profile, policy, or folder changes can invalidate a procedure.
- **Keep exceptions local.** Put them in `projects/[production]/project-instructions.md`, not the shared workflow.

WHEN A WORKFLOW BECOMES UNTRUSTED

For a wrong input, wrong mode, silent file change, or dependence on unstated chat, stop the run. Inspect the sources and diff. Restore only unauthorized hunks, preserving later human work. Mark the workflow untrusted, repair it, and run a fresh-session test before reuse.

Guardrail. Repeatability is not correctness. A procedure can reproduce a bad judgment perfectly, and consistent output is not evidence that the underlying criterion is sound. Keep approved sources, explicit uncertainty, writer judgment, and periodic fresh-session tests in the process. The writer retains authorship and approval.

READY WHEN

A writer names the workflow and target, receives the documented output, and reaches the approval point without recovering the original conversation.

FADE OUT.