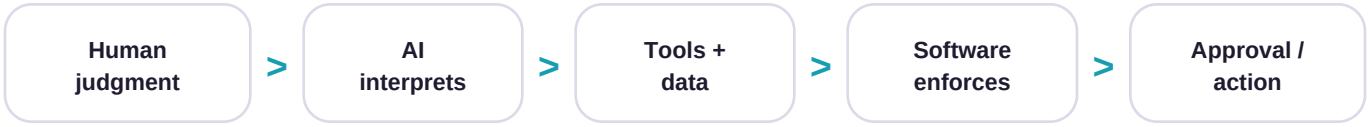
Human + AI Workflow Design

Make judgment, authority, uncertainty, and recovery explicit.

PM CHEAT SHEET

VISUAL ANCHOR



Not every workflow needs final approval. Raise control as data sensitivity, authority, or impact rises.

1. ASSIGN EACH LAYER ONE JOB

Human judgment

- Set goals, risk tolerance, evidence bar, exceptions, and what must stay under human control.

AI

- Interpret, summarize, classify, compare, synthesize, and surface uncertainty.
- Suggest rather than silently authorize.

Tools + data

- Retrieve, query, or execute through bounded tools with scoped inputs, permissions, and observable outcomes.

Software

- Enforce exact rules: permissions, schemas, allowed transitions, version checks, limits, dedupe, and required approvals.

People / action

- Approve consequential mutations, resolve tradeoffs, correct wording, reject proposals, or keep uncertainty open.

2. SUGGEST VS. ENFORCE VS. AUTHORIZE

Rule of thumb: If you can write the rule precisely, do not make the model enforce it.

AI should suggest

- Semantic judgments with real ambiguity
- Summaries, comparisons, draft recommendations
- Potential changes a person can inspect

Software should enforce

- Rules you can state exactly
- Permissions, limits, validation, versioning
- Safe defaults when authority or inputs are missing

People should authorize

- High-impact or irreversible actions
- Policy exceptions and tradeoffs under uncertainty
- What becomes authoritative when mistakes materially matter

3. MAKE UNCERTAINTY VISIBLE

Evidence is not truth. Pending is not accepted. Unknown is not zero.

- Separate accepted facts, pending proposals, and open questions.
- Preserve source and freshness when they matter.
- Let users adjust, reject, leave unchanged, or keep tracking uncertainty.

TAKEAWAY

The best AI workflow makes the authority boundary obvious before anything goes wrong.

Failure Diagnosis + Recovery

Debug the layer that failed, then make the recovery path visible.

PM CHEAT SHEET

4. DIAGNOSE THE FIRST OBSERVABLE FAILURE

Do not default every AI failure to a prompt problem.

Layer	Failure signal	First thing to inspect
Instructions	Same evidence is misread across cases	Framing, examples, ambiguity, conflicting instructions
Retrieval	Right source is missing, stale, or conflicting	What was actually retrieved before judging generation
Tools	Call fails, times out, or returns partial/wrong data	Inputs, outputs, retries, dependency failure
Logic	Model output is reasonable but product behavior is wrong	Validation, branching, versioning, deterministic state rules
Permissions	System can see/do too much or too little	Auth, scopes, isolation, approval requirements
UX	User cannot tell what happened or recover	Explanation, correction, and recovery in the interface

Debugging principle: Trace backward to the first place reality diverges from the expected workflow.

5. DESIGN RECOVERY BEFORE LAUNCH

Model/provider fails

Fail closed or use an intentionally limited fallback. Never invent an answer just to keep the flow moving.

Evidence is weak

Show insufficient support, ask for more evidence, or keep a Question open.

Tool call fails

Retry only when safe; otherwise expose a manual or escalated path.

State is stale/conflicting

Block the mutation and require re-review against the current version.

Permission is missing

Deny the action, explain the boundary, and request approved access.

Bad action slips through

Preserve history, support reversal/escalation where possible, and add a regression case.

6. PM DESIGN CHECKLIST

Before design

- ☐ Final authority mapped
- ☐ AI read/write/tool scope bounded
- ☐ Reversibility + approvals explicit
- ☐ Uncertainty that must stay visible

Before release

- ☐ Evidence vs. proposal is visible
- ☐ Correction / rejection path exists
- ☐ Failure + fallback + escalation designed
- ☐ Silent misses are traceable

During iteration

- ☐ Retest exact failure + boundaries
- ☐ Use holdouts/repeated runs
- ☐ Measure false certainty + review burden
- ☐ Fix the failing layer, not the nearest prompt

TAKEAWAY

Human-in-the-loop is not “put a person at the end.” It is explicit authority, uncertainty, correction, and recovery end to end.