

# LLM Workflow Audit

Phil Stevens

philipstevens.github.io

Define the production bar: failure modes, acceptance criteria, and an eval plan you can implement.

---

## SCOPE

One LLM workflow end-to-end (prompting/routing, RAG/tools if present, output constraints, and operational controls).

## PRIMARY OUTPUT

Contracts + gates + failure-mode coverage + eval plan + ops plan (ready to implement).

## Best Fit

- You need clear ship/no-ship criteria and realistic acceptance thresholds.
- You have a workflow that works sometimes, but behavior is inconsistent or unmeasured.
- You want a concrete eval and hardening plan before investing in implementation.

## Inputs Required

---

INPUT	MINIMUM	WHY IT MATTERS
Workflow walkthrough	Prompts/routing and where it runs (repo access or walkthrough)	Accurate diagnosis; practical fixes; correct trace/version requirements.
Representative examples	30–100 inputs (+ current outputs if available)	Realistic eval coverage; must-pass selection; edge cases.
Access to current implementation	Endpoint or repo to invoke/inspect the workflow	Enables reproducible runs and accurate failure-mode mapping.
Stakeholder constraints	Output consumer, required format, compliance/policy rules	Correct output contract, validation strategy, and refusal/escalation behavior.

*Data handling: least-privilege access; redact sensitive fields in shared artifacts; keep examples minimal and relevant.*

## How the Work Runs

---

STEP	WHAT HAPPENS
<b>1. Scope + risk tier</b>	Define workflow boundaries, unacceptable failures, and allowed uncertainty (refuse / escalate / partial).
<b>2. System teardown</b>	Map prompting, routing, RAG/tool use, validators, and where failures can occur.
<b>3. Failure modes</b>	Enumerate failures, severities, detection methods, and mitigations tied to the workflow.
<b>4. Eval design</b>	Define test categories, must-pass set selection rules, and measurable gates.
<b>5. Operability design</b>	Define trace fields, versioning scheme, monitoring signals, rollout and rollback triggers.
<b>6. Readout</b>	Prioritize fixes, estimate effort, and hand off to Build & Harden.

## Common failure patterns I look for

---

- Silent partial failures (plausible but incomplete outputs)
- Over-broad refusals masking missing grounding
- Eval metrics that don't correlate with user pain
- Latency/cost failures that only appear at p95+

## Deliverables

---

ARTIFACT	PURPOSE
Workflow contract (input/output schemas, grounding mode, tool policy)	Defines what valid requests and responses look like
Failure-mode matrix	Maps failures by severity, detection method, and mitigation so nothing ships without a plan to catch it
Eval plan with gate thresholds	Specifies test categories, must-pass rules, and acceptance criteria so ship decisions are evidence-based
Ops plan (trace fields, monitoring signals, rollback triggers)	Defines what gets logged and when to act so incidents are diagnosable without guesswork
Ship/no-ship recommendation	Evidence-backed decision with rationale and next steps
Prioritized hardening roadmap	Sequenced fixes ready for Build & Harden

## Definition of Done

- Workflow spec documents all agreed success criteria and known failure modes, with no ambiguous edge cases.
- Eval plan covers all identified critical failure modes with measurable acceptance thresholds.
- Prioritized hardening roadmap with sequenced fixes and clear ownership.
- Ship/no-ship recommendation is evidence-backed with documented rationale and actionable next steps.

## Boundaries

- Implementing the harness, CI gates, or production rollout (covered in Build & Harden / Retainer).
- Large-scale labeling programs or broad multi-workflow platform work (can be separately scoped).

## Next Step

Share one workflow and 30–100 representative examples to start. If examples are sensitive, start with synthetic or redacted cases and iterate.

### Book an intro call:

[calendly.com/philipstevens4/intro](https://calendly.com/philipstevens4/intro)



Scan to book a call

**Email:** philipstevens4@gmail.com

**Web:** philipstevens.github.io

**Not ready?** Get the free [Production Readiness Checklist](#) to self-assess first.

