

LLM Workflow Audit

Phil Stevens
philipstevens.github.io

Define the production bar for one workflow and produce an implementation-ready plan to ship it safely.

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 HAVE

- A demo or pilot that "works," but is brittle or hard to trust.
- Stakeholders asking: "How do we know it's safe to ship?"
- Frequent changes (prompt/model/docs/tools) causing behavior drift.

YOU NEED

- Explicit ship/no-ship criteria and a must-pass set.
- A failure-mode map that's testable and operational.
- An eval + monitoring plan tied to rollback triggers.

Inputs Required

INPUT	MINIMUM	WHAT IT ENABLES
Representative examples	30–100 inputs (+ current outputs if available)	Realistic eval coverage; must-pass selection; edge cases.
Current implementation overview	Prompts/routing and where it runs (repo access or walkthrough)	Accurate diagnosis; practical fixes; correct trace/version requirements.
Integration constraints	Output consumer + required format + downstream constraints	Correct output contract and validation strategy.
Policy/compliance constraints	PII rules, safety constraints, brand/voice rules (if any)	Explicit refusal/escalation behavior and safety gates.
Optional accelerators	Logs, known incidents, RAG details (chunking/retrieval), tool specs	Higher precision failure modes; stronger regression harness design.

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

Process

STEP	WHAT WE DO	ARTIFACT PRODUCED
1. Scope + risk tier	Define workflow boundaries, unacceptable failures, and allowed uncertainty (refuse / escalate / partial).	Workflow contract draft + risk tier defaults
2. System teardown	Map prompting, routing, RAG/tool use, validators, and where failures can occur.	Architecture + dependency map
3. Failure modes	Enumerate failures, severities, detection methods, and mitigations tied to the workflow.	Failure-mode coverage matrix
4. Eval design	Define test categories, must-pass set selection rules, and measurable gates.	Eval plan + gate thresholds proposal
5. Operability design	Define trace fields, versioning scheme, monitoring signals, rollout and rollback triggers.	Ops plan + release readiness requirements
6. Readout	Prioritize fixes, estimate effort, and define "done" for Build & Harden.	Implementation roadmap (sequenced, with acceptance criteria)

Deliverables

Workflow Contract Pack

- **Input contract:** fields, validation rules, max sizes.
- **Output contract:** schema/format + refusal/escalation structure.
- **Grounding mode:** grounded-only / mixed / ungrounded, with evidence rules.
- **Tool policy:** allow-list + argument validation + failure handling (if tools).

Gates + Coverage

- **Failure-mode matrix:** blocker/major/minor, detection (auto vs review), mitigations.
- **Eval blueprint:** categories, minimum counts, must-pass definition.
- **Gate thresholds:** risk-tier defaults for quality/safety/perf/regression.
- **Release requirements:** what evidence must exist to ship.

Ops & Auditability Plan

- **Trace requirements:** versions (model/prompt/retrieval), tool calls, retrieved IDs, latency/cost.
- **Monitoring plan:** alert thresholds and drift signals tied to rollback triggers.
- **Rollout plan outline:** shadow/canary/ramp recommendation and approval steps.

Success Criteria

- Everyone can answer: "What does 'correct' mean for this workflow?" in measurable terms.
- There is a must-pass set and a gate definition that would prevent unacceptable releases.
- There is an implementable plan to add traceability, monitoring, and rollback triggers.

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

Email: philipstevens4@gmail.com

Web: philipstevens.github.io



Scan to book a call