========== index.html ========== ========== index.md ========== — layout: splash permalink: / title: “AI that works beyond the demo” excerpt: “Predictable behavior, every time, at scale, with real users.” description: “Consulting and services for teams shipping AI features into production.” author_profile: false classes: wide last_updated: 2025-12-28 header: overlay_color: “#0B1220” overlay_filter: 0.35 actions: - label: “Book intro call” url: “https://calendly.com/philipstevens4/intro” —

Most AI pilots start strong, then break in ways no one predicted.

Why teams get stuck

- Outputs vary too much to trust
- Hard to explain or sign off on what it produces
- Changes ship without knowing what they’ll break
- Too slow or expensive to scale up

The fix

1. Define what “good” looks like (and what can’t happen)
2. Test until it passes consistently
3. Add release checks so updates don’t quietly break it

OFFER	WHAT YOU GET	PRICING	DETAILS
Audit	Define the bar, find the gaps, get a plan	From $7,500	PDF
Build & Fix	Meet the bar. Reliable, scalable, trustworthy.	$25k–$90k	PDF
Keep it Stable	Keep meeting the bar. Predictable releases, fewer surprises.	From $3k/mo	PDF

Not sure which fits?

Book intro call{: .btn .btn–primary }

Offer 1: Audit

Get clarity + a plan

If you have a workflow that kind of works, this is the fast way to get clarity. Pull the system apart, look at real examples, and pin down what “good enough” actually means for your use case.

What you get:

- The main ways it fails (with real examples)
- A clear definition of “good enough” everyone can agree on
- A step-by-step plan to make it reliable

Book intro call{: .btn .btn–primary } Download offer details (PDF){: .btn .btn–inverse }

Offer 2: Build & Fix

Make it reliable

This covers implementation. Use an existing plan (or define “good”), build the tests, and iterate until behavior is consistent on real inputs.

Deliverables:

- A repeatable test set you can run before shipping changes
- A version that behaves consistently on real inputs
- A simple rollout + rollback plan

Book intro call{: .btn .btn–primary } Download offer details (PDF){: .btn .btn–inverse }

Offer 3: Keep it Stable

Keep it reliable as you ship

For teams already in production (or shipping frequently) who want fewer surprises. Make releases repeatable, catch regressions early, and keep the tests up to date as new edge cases show up.

Ongoing outputs:

- A lightweight release checklist
- Early warning when things start degrading
- New test cases added as edge cases show up

Book intro call{: .btn .btn–primary } Download offer details (PDF){: .btn .btn–inverse }

{% include lead-capture.html source=“landing-page” %}

FAQ

Can we do this under NDA?

Yes, an NDA can be signed before reviewing sensitive details.

What do you need from us?

Usually: a few example inputs/outputs, current prompts or workflow code, and enough context to define what “good” looks like. For Build & Fix, repo access and a way to run tests in CI may also be needed.

What about sensitive data?

Data exposure can be minimized. Redacted samples, synthetic test cases, and tight access controls all work. Work can happen in your environment if needed.

What does “success” mean?

Success means the workflow meets a clear definition of “good” and passes tests on representative cases, within agreed limits for cost, speed, and quality.

Who needs to be involved?

Typically: one engineering owner, one product/domain owner, and someone who can approve the definition of “good” and provide access.

How long does this usually take?

- Audit: 1-2 weeks
- Build & Fix: typically 3-8 weeks, depending on scope
- Keep it Stable: ongoing

How does it start?

Book an intro call. If it’s a fit, the next step is defining the workflow, scope, and success criteria, then starting with an Audit or jumping straight to Build & Fix.

{% include last-updated.html %}

========== _data/navigation.yml ========== main: - title: “Services” url: / - title: “Library” url: /library/ - title: “About” url: /about/

========== _includes/footer.html ========== {% unless site.atom_feed.hide %} {% assign show_atom = true %} {% endunless %} {% if site.footer.links or show_atom %}

{% if site.data.ui-text[site.locale].follow_label %}

- **{{ site.data.ui-text[site.locale].follow_label }}**

{% endif %}

{% if site.footer.links %} {% for link in site.footer.links %} {% if link.label and link.url %}

- {{ link.label | escape_once | strip }}

```
{% endif %}
  {% endfor %}
{% endif %}

{% unless site.atom_feed.hide %}
  <li><a href="{% if site.atom_feed.path %}{{ site.atom_feed.path }}{% else %}{{
'/feed.xml' | relative_url }}{% endif %}"><i class="fas fa-fw fa-rss-square" aria-
hidden="true"></i> {{ site.data.ui-text[site.locale].feed_label | default: "Feed" }}</a>
</li>
{% endunless %}
```

{% endif %}

© {% assign site_time = site.time | date: '%Y' %}{% if site.footer.since and site_time != site.footer.since %}{{ site.footer.since }} - {% endif %}{{ site_time }} {{ site.copyright | default: site.title | escape_once | strip }}.
========== _includes/footer/custom.html ==========

```
<span>Have an AI workflow stuck in pilot?</span>
<a href="https://calendly.com/philipstevens4/intro" style="margin-left: 0.3rem; text-
decoration: underline; font-style: normal;"">
  Book an intro call
</a>
```

========== _includes/head/custom.html ========== {% if page.profile_page %}

{% endif %}

========== _includes/last-updated.html ========== {% if page.last_updated %}

Last updated: {{ page.last_updated | date: "%Y-%m-%d" }}

{% endif %} ========== _includes/lead-capture.html ==========

Free: LLM Production Readiness Checklist

A fill-in workbook to assess your workflow before shipping. Covers failure modes, release checks, monitoring, and regression tests. Score your readiness and identify gaps.

```
  <input
    type="email"
    name="email"
    id="email"
    placeholder="you@company.com"
    required
    autocomplete="email"
  />
  <button type="submit" class="btn btn--primary">Submit</button>
</div>

<!-- honeypot -->
<input
  type="text"
  name="company"
  tabindex="-1"
  autocomplete="off"
  aria-hidden="true"
  style="position:absolute;left:-9999px;opacity:0;height:0;width:0;"
/>
<input type="hidden" name="k" value="formkey_QidffT6hpBTjEE38dkn2pbCvfCmebUJn" />
<input type="hidden" name="source" value="{{ include.source | default: 'unknown' }}" />
<input type="hidden" name="ua" id="lead_ua" />
```

========== _pages/about.md ==========

title: “About Me” layout: single permalink: /about/ description: “Philip Stevens: 10+ years applied ML and foundation model engineering. I focus on making high-stakes LLM workflows reliable, measurable, and safe.” author_profile: true profile_page: true last_updated: 2025-12-28 —

Email me{: .btn .btn–primary }

Hi, I’m Phil.

I’m a foundation model engineer focused on making LLM workflows reliable in production. I care about this because I’ve seen too many promising systems fail when they hit real users — and most of those failures were preventable with better evals and release discipline.

I bring 10+ years of applied ML across personalization, recommendations, NLP, and real-time decision systems, including production work at Agoda and Quantcast. I’ve been consulting since 2023.

I publish case studies and technical write-ups here to share how I approach this work.

Where to go next

- Services — engagement options and pricing
- Library — case studies and tutorials
- CV — full background

{% include last-updated.html %}

========== _pages/work.md ==========

title: “Library” layout: splash permalink: /library/ description: “Case studies and technical notes on LLM workflow reliability, evals, and release gating.” excerpt: “Case studies and technical notes.” classes: wide last_updated: 2026-01-01 header: overlay_color: “#0B1220” overlay_filter: 0.35 —

{% assign case_studies = site.library | where: “type”, “case-study” | reverse %} {% assign tutorials = site.library | where: “type”, “tutorial” | reverse %} {% assign total = case_studies.size | plus: tutorials.size %}

{% if total == 0 %}

Content in progress. Case studies and tutorials will appear here as published.

{% else %}

{% if case_studies.size > 0 %} ## Case studies

{% for item in case_studies %} - {{ item.title }}{% if item.excerpt %} — {{ item.excerpt }} {% endif %} {% endfor %} {% endif %}

{% if tutorials.size > 0 %} ## Tutorials

{% for item in tutorials %} - {{ item.title }}{% if item.excerpt %} — {{ item.excerpt }}{% endif %} {% endfor %} {% endif %}

{% endif %}

If any of this is relevant to what you’re working on, get in touch.

{% include last-updated.html %}

========== _pages/cv.md ==========

title: “CV” permalink: /cv/ description: “CV of Philip Stevens, applied ML + foundation model engineering across personalization, NLP, and production LLM workflows (evals, RAG, post-training).” toc: true toc_label: “Table of Contents” toc_icon: “bookmark” author_profile: true classes: wide last_updated: 2025-12-28 —

{% include last-updated.html %}

Download CV PDF{: .btn .btn–info}

Contact

- Email: philipstevens4@gmail.com
- LinkedIn: linkedin.com/in/philip-charles-stevens/

Experience

Self-employed

Foundation Model Engineer (Consultant)

Mar 2023 - Present

High-stakes, domain-adapted LLM workflows, made reliable.

*Selected outcomes:_

- Built spec-driven eval suites and regression gates and integrated them into release processes to prevent regressions.
- Stabilized RAG across updates with retrieval instrumentation, golden sets, and regression tracking.
- Shipped versioned LoRA/QLoRA adapters with curated data and training recipes, validated against task-specific evals.
- Improved tool and agent reliability with tool contracts, routing and guardrails, and scenario tests for recovery.
- Reduced serving cost and latency via profiling, batching, quantization, runtime selection, and caching under eval gates.

Agoda

Senior Data Scientist

Mar 2020 - Feb 2023, Bangkok, Thailand

Leading online travel agency, subsidiary of Booking Holdings.

*Accomplishments and Responsibilities:_

- Spearheaded several frontend personalization projects using contextual bandit algorithms (e.g., linear Thompson Sampling), dynamically adjusting content based on user data, boosting bookings by 500/day.
- Developed recommendation systems with Word2Vec/Doc2Vec embedding models, increasing daily bookings by hundreds.
- Enhanced systems to highlight key reviews using advanced BERT and LDA topic models, significantly boosting user engagement and resulting in additional bookings.
- Collaborated with the product team, offering data-driven strategic recommendations that improved business outcomes and informed key decision-making processes.

Quantcast

Data Scientist

Oct 2014 - Sep 2018, London, UK

Industry-leading AI-powered targeted advertising and audience measurement based in San Francisco. Joined as part of startup acquisition.

*Accomplishments and Responsibilities:_

- Directed many experiments to enhance core targeting models using advanced feature engineering, new data sources, refined model architectures, hyperparameter tuning, and domain drift monitoring, achieving 2-10% quarterly conversion rate improvements.
- Managed the end-to-end machine learning lifecycle and data pipeline for core targeting models, ensuring robust performance and consistency across data collection, processing, model training, deployment, and performance monitoring.

- Collaborated with external stakeholders to deliver custom projects and regularly communicated technology updates to advertising agencies, strengthening client relationships and enhancing project outcomes.

Struq

Data Scientist

Oct 2013 - Sep 2014, London, UK

A fast-paced AdTech startup, acquired by Quantcast.

*Accomplishments and Responsibilities:_

- Integrated user data into click, conversion, and revenue prediction models, enhancing accuracy through advanced feature engineering techniques, resulting in a ~20% increase in user clicks and conversions for clients.

Education

University of Auckland

Master of Science in Computer Science, 2012

- Graduated with 1st Class Honours
- Faculty of Science Master's Award
- Master's Scholarship funded though Royal Society of New Zealand Marsden Grant, Dr. Beryl Plimmer
- First in Course Award in COMPSCI 767 (Intelligent Software Agents)
- Faculty of Science Summer Research Scholarship

Bachelor of Arts in Mathematics and Philosophy (Dual), 2010

Publications

Stevens, Blagojevic, & Plimmer, 2013: "Supervised Machine Learning for Grouping Sketch Diagram Strokes." SBIM '13

Skills

- **Post-training and adaptation (core):**
  - Instruction tuning (SFT), task and domain adaptation
  - Preference optimization: DPO, ORPO, SimPO-style objectives
  - Preference data design: pairwise and single-response feedback, rubric design, consistency checks
  - Alignment and safety post-training: constitutional style critique and revision loops, RLAIF patterns when needed
  - PEFT: LoRA and QLoRA adapters, adapter packaging, versioning, merge and composition strategies

- **Data for post-training (what actually moves the needle):**
  - Dataset design and curation: filtering, dedup, quality gates, label guidelines, synthetic data with verification
  - Decontamination and leakage control: strict train and eval separation, contamination checks
  - Eval set construction: golden sets, stress sets, adversarial sets aligned to real failure modes
- **Evaluation and release engineering:**
  - Spec-driven evals: failure modes, acceptance criteria, scenario tests, regression harnesses
  - CI integrated eval gates, safe rollout patterns, rollback criteria, drift monitoring triggers
- **Grounded workflows and tool reliability:**
  - RAG design: chunking, hybrid retrieval, reranking, citation and attribution behavior
  - Retrieval instrumentation: coverage and recall proxies, regression tracking
  - Tool contracts, routing, guardrails, safe failure modes, recovery tests
- **Reliability contracts for production:**
  - Structured outputs, schema validation, constrained decoding for deterministic interfaces
  - Output validation that is separate from prompting, with explicit fallbacks
- **Serving and inference efficiency:**
  - Throughput and latency optimization: continuous batching, KV cache and prefix caching concepts
  - Quantization under quality gates, profiling driven capacity planning, caching strategies
- **Observability and security:**
  - OpenTelemetry-based tracing, latency and cost monitoring, error taxonomy
  - LLM security: prompt injection, insecure output handling, excessive agency controls, audit trails
- **Stack:**
  - Python, SQL (advanced); R, Scala, Java, C# (proficient)
  - PyTorch; Transformers ecosystem (Transformers, PEFT, TRL); scikit-learn, xgboost
  - Spark, PySpark, Hive, Hadoop

See supporting write-ups.

========== assets/css/main.scss ==========

@import “minimal-mistakes/skins/{{ site.minimal_mistakes_skin | default: ‘default’ }}”;
@import “minimal-mistakes”;

hr { border: 0; border-top: 2px solid #494e52; margin: 3em 0; }

.notice h2 { font-size: 1.5em; margin: 0; }

.faq-section h3 { font-style: italic; font-size: 1.1em; }

.template-box { border: 2px solid #494e52; border-radius: 4px; padding: 1.5em; margin: 1.5em 0; background-color: #f9f9f9; }

.template-title { font-size: 1.25em; margin-top: 0; color: #494e52; border-bottom: 1px solid #ddd; padding-bottom: 0.5em; }

.template-section { font-size: 1.1em; margin-top: 1.5em; margin-bottom: 0.5em; color: #494e52; }

.dashboard-example { border: 2px solid #494e52; border-radius: 6px; padding: 1.5em; margin: 1.5em 0; background: linear-gradient(135deg, #1a1a2e 0%, #16213e 100%); color: #eee; }

.dashboard-title { font-size: 1.2em; color: #fff; border-bottom: 1px solid #444; padding-bottom: 0.75em; margin-bottom: 1em; }

.dashboard-subtitle { font-size: 1em; color: #ccc; margin-top: 1.5em; margin-bottom: 0.5em; }

.metric-grid { display: grid; grid-template-columns: repeat(auto-fit, minmax(200px, 1fr)); gap: 1em; margin-bottom: 1em; }

.metric-card { background: rgba(255, 255, 255, 0.05); border: 1px solid #444; border-radius: 4px; padding: 1em; }

.metric-card-title { color: #7ec8e3; font-weight: bold; font-size: 0.95em; margin-bottom: 0.5em; }

.metric-card table { width: 100%; font-size: 0.85em; border-collapse: collapse; }

.metric-card td { padding: 0.3em 0.5em; color: #ddd; border-bottom: 1px solid #333; }

.metric-card td:first-child { color: #aaa; }

.metric-card td:last-child { text-align: right; color: #888; }

.status-ok { color: #4ade80 !important; font-weight: bold; }

.alerts-table { width: 100%; font-size: 0.9em; border-collapse: collapse; }

.alerts-table td { padding: 0.4em 0.75em; color: #ddd; border-bottom: 1px solid #333; }

.alerts-table td:first-child { color: #888; font-family: monospace; font-size: 0.9em; }

.alert-info { color: #60a5fa !important; font-weight: bold; }

.alert-warn { color: #fbbf24 !important; font-weight: bold; }

// Eval Report Styles .eval-report { border: 2px solid #e5e7eb; border-radius: 6px; margin: 1.5em 0; background: #fff; overflow: hidden; }

.eval-report-header { display: flex; align-items: center; gap: 1em; padding: 1em 1.5em; background: #f9fafb; border-bottom: 1px solid #e5e7eb; }

.eval-status-pass { background: #22c55e; color: #fff; font-weight: bold; font-size: 0.85em; padding: 0.25em 0.75em; border-radius: 4px; }

.eval-status-fail { background: #ef4444; color: #fff; font-weight: bold; font-size: 0.85em; padding: 0.25em 0.75em; border-radius: 4px; }

.eval-title { font-weight: bold; color: #374151; }

.eval-date { color: #9ca3af; font-size: 0.9em; margin-left: auto; }

.eval-results-table { width: 100%; border-collapse: collapse; font-size: 0.95em; }

.eval-results-table th { background: #f9fafb; color: #6b7280; font-weight: 600; text-align: left; padding: 0.75em 1em; border-bottom: 1px solid #e5e7eb; }

.eval-results-table td { padding: 0.75em 1em; border-bottom: 1px solid #f3f4f6; color: #374151; }

.eval-results-table tbody tr:hover { background: #f9fafb; }

.trend-up { color: #22c55e; }

.trend-down { color: #ef4444; }

.trend-neutral { color: #9ca3af; }

.eval-summary { padding: 1em 1.5em; background: #f0fdf4; border-top: 1px solid #bbf7d0; color: #166534; font-size: 0.95em; }

// Case Snippet Styles (proof of work examples) details.case-snippet { border: 1px solid #e5e7eb; border-radius: 8px; margin: 1.5em 0; background: #fff; overflow: hidden; box-shadow: 0 1px 3px rgba(0, 0, 0, 0.1); }

details.case-snippet[open] { box-shadow: 0 2px 8px rgba(0, 0, 0, 0.15); }

summary.case-snippet-header { display: flex; align-items: center; gap: 0.75em; padding: 0.75em 1.25em; background: linear-gradient(135deg, #1e3a5f 0%, #2d5a87 100%); cursor: pointer; list-style: none; }

summary.case-snippet-header::-webkit-details-marker { display: none; }

summary.case-snippet-header::after { content: "▶"; color: rgba(255, 255, 255, 0.6); font-size: 0.7em; margin-left: auto; transition: transform 0.2s ease; }

details.case-snippet[open] summary.case-snippet-header::after { transform: rotate(90deg); }

.case-snippet-type { background: rgba(255, 255, 255, 0.2); color: #fff; font-size: 0.7em; font-weight: bold; padding: 0.25em 0.5em; border-radius: 3px; letter-spacing: 0.05em; }

.case-snippet-title { color: #fff; font-weight: 600; font-size: 0.95em; }

.case-snippet-body { padding: 1em 1.25em; }

.case-snippet-section { padding: 0.75em 0; border-bottom: 1px solid #f3f4f6; }

.case-snippet-section:last-child { border-bottom: none; }

.case-snippet-label { font-size: 0.75em; font-weight: 600; color: #6b7280; text-transform: uppercase; letter-spacing: 0.05em; margin-bottom: 0.35em; }

.case-snippet-content { font-size: 0.9em; color: #374151; line-height: 1.5; }

.case-tag { display: inline-block; font-size: 0.7em; font-weight: bold; padding: 0.2em 0.5em;

border-radius: 3px; margin-right: 0.35em; }

.case-tag-high { background: #fef2f2; color: #dc2626; }

.case-tag-med { background: #fffbeb; color: #d97706; }

.case-tag-warn { background: #fffbeb; color: #d97706; }

.case-tag-ok { background: #f0fdf4; color: #16a34a; }

.case-metric { color: #6b7280; }

.case-metric-before { color: #dc2626; text-decoration: line-through; }

.case-metric-after { color: #16a34a; font-weight: 600; }

// Template Dropdown Styles (for artifact templates) details.template-dropdown { border: 1px solid #e5e7eb; border-radius: 8px; margin: 1.5em 0; background: #fff; overflow: hidden; box-shadow: 0 1px 3px rgba(0, 0, 0, 0.1); }

details.template-dropdown[open] { box-shadow: 0 2px 8px rgba(0, 0, 0, 0.15); }

summary.template-dropdown-header { display: flex; align-items: center; gap: 0.75em; padding: 0.75em 1.25em; background: linear-gradient(135deg, #2d4a3e 0%, #3d6a56 100%); cursor: pointer; list-style: none; }

summary.template-dropdown-header::-webkit-details-marker { display: none; }

summary.template-dropdown-header::after { content: “▶”; color: rgba(255, 255, 255, 0.6); font-size: 0.7em; margin-left: auto; transition: transform 0.2s ease; }

details.template-dropdown[open] summary.template-dropdown-header::after { transform: rotate(90deg); }

.template-dropdown-body { padding: 1em 1.25em; }

// Lead Capture Box .lead-capture-box { border: 2px solid #7c3aed; border-radius: 8px; padding: 1.5em; margin: 3em 0 1.5em 0; background: linear-gradient(135deg, #faf5ff 0%, #f3e8ff 100%);

.lead-capture-description { margin: 0 0 0.75em 0; color: #374151; line-height: 1.5;

strong { color: #5b21b6; } }

.lead-capture-list { margin: 0 0 0.75em 0; padding-left: 1.25em; color: #374151;

li { margin-bottom: 0.25em; } }

.lead-capture-note { margin: 0 0 1.25em 0; font-style: italic; color: #6b7280; font-size: 0.95em; }

.checklist-form { .form-row { display: flex; gap: 0.75em; flex-wrap: wrap; }

input[type=“email”] { flex: 1; min-width: 200px; padding: 0.75em 1em; border: 1px solid #d1d5db; border-radius: 6px; font-size: 1em;

&:focus { outline: none; border-color: #7c3aed; box-shadow: 0 0 0 3px rgba(124, 58, 237, 0.1); } } }

.btn–primary { background-color: #7c3aed; border-color: #7c3aed; white-space: nowrap;

&:hover { background-color: #6d28d9; border-color: #6d28d9; } }

.lead-capture-message { margin: 0; min-height: 1.5em; }

.lead-capture-success { margin-top: 0.5em; padding: 1em; background: #f0fdf4; border: 1px solid #bbf7d0; border-radius: 6px; color: #166534; text-align: center; } }

// Checklist Landing Page .checklist-landing { max-width: 600px; margin: 0 auto;

h2 { color: #5b21b6; margin-bottom: 0.25em; }

.checklist-subtitle { color: #6b7280; font-size: 1.1em; margin-bottom: 2em; }

.checklist-preview { background: #f9fafb; border: 1px solid #e5e7eb; border-radius: 8px; padding: 1.5em; margin-bottom: 2em;

ul { margin: 0.5em 0 1em 0; padding-left: 1.25em; }

li { margin-bottom: 0.4em; color: #374151; }

.checklist-note { font-style: italic; color: #6b7280; margin-bottom: 0; } }

.checklist-form-box { border: 2px solid #7c3aed; border-radius: 8px; padding: 1.5em; background: linear-gradient(135deg, #faf5ff 0%, #f3e8ff 100%); }

.checklist-form { label { display: block; font-weight: 600; color: #374151; margin-bottom: 0.75em; }

.form-row { display: flex; gap: 0.75em; flex-wrap: wrap; }

input[type=“email”] { flex: 1; min-width: 200px; padding: 0.75em 1em; border: 1px solid #d1d5db; border-radius: 6px; font-size: 1em;

&:focus { outline: none; border-color: #7c3aed; box-shadow: 0 0 0 3px rgba(124, 58, 237, 0.1); } }

.btn–primary { background-color: #7c3aed; border-color: #7c3aed; white-space: nowrap;

&:hover { background-color: #6d28d9; border-color: #6d28d9; } } } }

// Checklist Thanks Page .checklist-thanks { max-width: 600px; margin: 0 auto; text-align: center;

.thanks-message { font-size: 1.1em; color: #374151; margin-bottom: 1.5em; }

.btn–large { padding: 1em 2em; font-size: 1.1em; }

.print-tip { margin-top: 1.5em; color: #6b7280; font-size: 0.9em; }

hr { margin: 2.5em 0; }

h3 { color: #374151; } }

========== assets/downloads/llm-production-readiness-checklist.html ==========
<!DOCTYPE html>

LLM Workflow Production Readiness Checklist

Version 1.0 | Philip Stevens | philipstevens4@gmail.com

Workflow name: _____	**Review date:** ___/___/_____
Team / Owner: _____	**Reviewer(s):** _____
Workflow description (1 sentence): _____	

How to Use This Checklist

This checklist is designed for engineering teams preparing to ship an LLM-powered workflow to production. Work through each section before release. Items marked with **[CRITICAL]** are non-negotiable; shipping without them significantly increases the risk of production incidents.

Score Your Readiness

After completing the checklist, tally your results here:

CATEGORY	PASSED	TOTAL
[CRITICAL] items (Parts 1–3)	____ / 7	Must be 7/7 to ship
Failure mode coverage (Part 2)	____ / 19	Target: 15+
Monitoring configured (Part 4)	____ / 14	Target: 10+
Harness requirements (Part 5)	____ / 7	Target: 5+

Interpretation:

- **All CRITICAL + targets met:** You're ready to ship. Proceed with staged rollout.
- **CRITICAL pass, targets partial:** Ship with caution. Prioritize gaps in first week post-deploy.
- **Any CRITICAL fail:** **Do not ship.** Fix these first:
  1. Define and run a golden set eval (≥50 cases)
  2. Add safety checks for unsafe inputs and PII
  3. Test your rollback procedure once

```
</li>
```

Part 1: Eval Gates

Before any release, run these evaluations and verify thresholds are met.

1.1 Accuracy & Quality

CHECK	YOUR THRESHOLD	MEASURED	PASS?
[CRITICAL] Task accuracy on golden set (n≥50)	____%	____%	☐
Accuracy on edge cases subset	____%	____%	☐
Accuracy on adversarial/malformed inputs	____%	____%	☐
Human preference score (if applicable)	____/5	____/5	☐

Setting thresholds: Start with your current baseline. If you don't have one, run the eval and use current performance minus a small buffer (e.g., if you measure 94%, set threshold at 92%). Raise the bar over time.

1.2 Safety & Compliance

CHECK	THRESHOLD	MEASURED	PASS?
[CRITICAL] Refusal rate on unsafe inputs	100%	____%	☐
[CRITICAL] No PII leakage on test set	0 instances	____	☐
Hallucination rate (factual claims)	≤____%	____%	☐
Compliance with domain-specific rules	100%	____%	☐

Unsafe input test set: Include prompt injections, attempts to extract system prompts, requests for harmful content, and attempts to bypass guardrails. Minimum 20 cases; 50+ recommended.

1.3 Performance & Cost

CHECK	THRESHOLD	MEASURED	PASS?
Latency p50	≤___s	___s	☐
[CRITICAL] Latency p95	≤___s	___s	☐
Latency p99	≤___s	___s	☐
Cost per request (avg)	≤ </td> < td>___	☐	
Token efficiency (output/input ratio)	≤___	___	☐

> **Latency measurement:** Measure end-to-end, not just model call time. Include retrieval, preprocessing, validation, and any retries.

1.4 Regression Check

CHECK	THRESHOLD	MEASURED	PASS?
[CRITICAL] Regression suite pass rate	100%	___%	☐
No new failures on previously-passing cases	0	___	☐
Performance delta vs. previous version	≤___%	___%	☐

Part 2: Failure Mode Coverage

Verify you have detection and mitigation for each failure mode category.

2.1 Output Quality Failures

FAILURE MODE	DETECTION METHOD	MITIGATION	☐
Hallucinated facts	Citation verification, factual consistency check	Ground with retrieved docs, add confidence thresholds	☐
Incomplete output	Required field validation, length checks	Structured output schema, retry logic	☐
Wrong format	Schema validation, regex checks	Strict output parsing, fallback formatting	☐
Inconsistent with context	Semantic similarity to input, contradiction detection	Re-ranking, chain-of-thought verification	☐
Outdated information	Timestamp checks on retrieved content	Source freshness filters, recency weighting	☐

2.2 Safety Failures

FAILURE MODE	DETECTION METHOD	MITIGATION	☐
Prompt injection executed	Input classification, output anomaly detection	Input sanitization, output filtering, system prompt hardening	☐
PII in output	Regex + NER detection on outputs	PII scrubbing layer, training data audit	☐
Harmful content generated	Content classification on outputs	Output filtering, refusal training	☐
System prompt leaked	Pattern matching for prompt fragments	Instruction hierarchy, output filtering	☐
Unauthorized capability use	Action logging, capability boundaries	Explicit allow-lists, confirmation steps	☐

2.3 Reliability Failures

FAILURE MODE	DETECTION METHOD	MITIGATION	☐
Model API timeout	Request timing, circuit breaker triggers	Timeouts, retries with backoff, fallback responses	☐
Rate limit exceeded	429 response tracking	Request queuing, rate limiting at app layer	☐
Context window exceeded	Token counting before calls	Truncation strategy, summarization, chunking	☐
Retrieval returned no results	Empty result detection	Fallback to broader query, graceful degradation	☐
Retrieval returned irrelevant results	Relevance scoring threshold	Re-ranking, score cutoffs, “I don’t know” responses	☐

2.4 Upstream Dependency Failures

FAILURE MODE	DETECTION METHOD	MITIGATION	☐
Model behavior changed (silent update)	Eval suite drift detection, output distribution monitoring	Version pinning where possible, automated regression alerts	☐
Embedding model changed	Similarity score distribution shift	Re-index on change, version tracking	☐
Vector DB unavailable	Health checks, latency monitoring	Caching layer, graceful degradation	☐
Source data stale or missing	Freshness checks, data pipeline monitoring	Staleness alerts, fallback sources	☐

Part 3: Release Decision Framework

3.1 Ship / No-Ship Criteria

SHIP if all of the following are true:

☐	All **[CRITICAL]** eval gates pass
☐	No regressions on the regression suite
☐	All failure modes have detection or mitigation in place
☐	Rollback tested and verified working
☐	Monitoring and alerting configured
☐	Required sign-offs collected

NO-SHIP if any of the following are true:

☐	Any **[CRITICAL]** eval gate fails
☐	New regression introduced
☐	Unmitigated high-severity failure mode discovered
☐	Rollback not tested or broken
☐	Missing required sign-off

3.2 Release Artifacts Checklist

Before release, verify these artifacts exist and are versioned:

ARTIFACT	LOCATION	VERSION	VERIFIED?
Prompt(s)	_____	v____	☐
System configuration	_____	v____	☐
Model identifier	_____	_____	☐
Eval suite	_____	v____	☐
Regression test set	_____	v____	☐
Retrieval index (if applicable)	_____	v____	☐

3.3 Rollback Verification

CHECK	STATUS
Previous version artifacts accessible	☐
Rollback procedure documented	☐
Rollback tested in staging	☐
Rollback time estimate: ____ minutes	☐
Rollback owner identified: _____	☐

Part 4: Post-Deploy Monitoring

4.1 Real-Time Signals

Configure alerts for these signals before going live:

SIGNAL	ALERT THRESHOLD	CURRENT VALUE	CONFIGURED?
Error rate (5xx, exceptions)	>___%	___%	☐
Latency p95	>___s	___s	☐
Request volume anomaly	±___% from baseline	___	☐
Cost per hour	> /td > < td>___	☐	
Empty/null response rate	>___%	___%	☐

4.2 Quality Monitoring (Sampled)

SIGNAL	SAMPLE RATE	CHECK FREQUENCY	CONFIGURED?
Human review of random outputs	___%	Daily / Weekly	☐
Automated quality scoring	___%	Continuous	☐
User feedback/thumbs tracking	100%	Continuous	☐
Hallucination spot-check	___%	Daily / Weekly	☐

4.3 Drift Detection

SIGNAL	DETECTION METHOD	CHECK FREQUENCY	CONFIGURED?
Output length distribution	Statistical test on rolling window	Daily	☐
Output sentiment/tone	Classifier on sampled outputs	Daily	☐
Refusal rate	Threshold on rolling average	Continuous	☐
Latency trend	Regression on 7-day window	Daily	☐
Eval score trend	Weekly eval run, track over time	Weekly	☐

Part 5: Regression Harness Structure

5.1 Test Case Categories

A complete regression suite should include cases from each category:

CATEGORY	DESCRIPTION	MINIMUM CASES	YOUR COUNT
Golden set	Representative inputs with verified correct outputs	50	____
Edge cases	Boundary conditions, unusual but valid inputs	20	____
Adversarial	Prompt injections, malformed inputs, attack attempts	20	____
Historical failures	Cases that broke in previous versions	All	____
High-stakes	Cases where errors have significant consequences	10	____

5.2 Test Case Structure

Each test case should include:

```
{
  "id": "unique-identifier",
  "category": "golden|edge|adversarial|regression|high-stakes",
  "input": { ... },
  "expected_output": { ... } | null,
  "evaluation": {
    "method": "exact_match|semantic_similarity|llm_judge|custom",
    "threshold": 0.95,
    "custom_evaluator": "path/to/evaluator" | null
  },
  "metadata": {
    "added_date": "2024-01-15",
    "source": "production_failure|synthetic|user_reported",
    "severity": "critical|high|medium|low",
    "notes": "..."
  }
}
```

5.3 Harness Requirements

REQUIREMENT	IMPLEMENTATION	DONE?
Single command to run full suite	`make eval` or equivalent	☐
Parallelized execution	Configurable concurrency	☐
Deterministic where possible	Fixed seeds, temperature=0	☐
Results persisted	Database or versioned files	☐
Diff against previous run	Automated comparison	☐
CI/CD integration	Runs on PR, blocks on failure	☐
Human-readable report	Summary + drill-down	☐

Part 6: Quick Reference

Red Flags That Should Block Release

1. **Regression on any previously-passing test case** — Something broke
2. **Safety eval failure** — Non-negotiable
3. **Latency p95 above threshold** — Will affect users
4. **Untested rollback** — You will need it eventually
5. **"We'll fix it after launch"** — You probably won't

Common Mistakes

MISTAKE	WHY IT HURTS	WHAT TO DO INSTEAD
Testing only happy paths	Real traffic includes edge cases and adversarial inputs	Build adversarial test set from day one
Threshold set to current performance	Any variance causes false failures	Set threshold below current with small buffer
Eval suite in notebook, not CI	Gets skipped under deadline pressure	Integrate into PR workflow from start
No rollback testing	Rollback fails when you need it most	Test rollback monthly, after every infra change
Ignoring cost until bill arrives	Budget surprises, rushed optimization	Track cost per request from day one
"Model X is better" without eval	Vibes don't catch regressions	Always run full eval before switching

First 24 Hours Post-Deploy

HOUR	ACTION
0-1	Watch error rate, latency, request volume
1-4	Spot-check 10 random outputs manually
4-8	Review any user feedback/complaints
8-24	Compare quality metrics to pre-deploy baseline
24+	Run full eval suite, compare to release eval

Example: Invoice Data Extraction Workflow

This example shows how to fill out key sections for a common workflow. Copy this pattern for your own.

Workflow name: Invoice field extraction	**Review date:** 01/15/2025
Team / Owner: Finance Automation / J. Chen	**Reviewer(s):** M. Park, S. Lee
Workflow description: Extract vendor, amount, date, and line items from uploaded PDF invoices into structured JSON.	

Example: Eval Gates (Section 1.1)

CHECK	YOUR THRESHOLD	MEASURED	PASS?
[CRITICAL] Task accuracy on golden set (n≥50)	≥92%	94.2%	✓
Accuracy on edge cases subset	≥85%	87%	✓
Accuracy on adversarial/malformed inputs	≥80%	82%	✓

Why these thresholds: *92% baseline from current prod performance minus 2% buffer. Edge cases lower because they include handwritten invoices. Adversarial includes corrupted PDFs and injected text.*

Example: Failure Mode Coverage (Section 2.1)

FAILURE MODE	DETECTION METHOD	MITIGATION	
Wrong amount extracted	Regex check: amount matches currency pattern	Flag for human review if confidence <0.9	✓
Missing required field	Schema validation on output JSON	Retry once, then escalate to human queue	✓
Hallucinated line item	Compare line item count to OCR bounding boxes	Reject if count mismatch >1	✓

Example: Test Case (Section 5.2)

```
{
  "id": "inv-edge-003",
  "category": "edge",
  "input": {
    "file": "handwritten_invoice_blurry.pdf",
    "expected_fields": ["vendor", "amount", "date"]
  },
  "expected_output": {
    "vendor": "ABC Supplies",
    "amount": 1234.56,
    "date": "2024-12-01",
    "confidence": 0.85
  },
  "evaluation": {
    "method": "custom",
    "threshold": 0.9,
    "custom_evaluator": "evals/invoice_field_match.py"
  },
  "metadata": {
    "added_date": "2025-01-10",
    "source": "production_failure",
    "severity": "high",
    "notes": "Handwritten invoices frequently miss date field"
  }
}
```

This checklist is provided as a starting point. Adapt thresholds, categories, and checks to your specific workflow and domain requirements.

========== assets/downloads/llm-workflow-audit.html ========== <!doctype html>

```
    <h1>LLM Workflow Audit</h1>
    <p class="subtitle">Define the production bar: failure modes, acceptance criteria,
and an eval plan you can implement.</p>
  </div>
  <div class="header-right">
    <div class="name">Phil Stevens</div>
    <div>philipstevens.github.io</div>
  </div>
</header>

<dl class="overview">
  <dt>Scope</dt>
  <dd>One LLM workflow end-to-end (prompting/routing, RAG/tools if present, output
constraints, and operational controls).</dd>
  <dt>Primary output</dt>
  <dd>Contracts + gates + failure-mode coverage + eval plan + ops plan (ready to
implement).</dd>
</dl>

<section>
  <h2>Best Fit</h2>
  <ul>
    <li>You need clear ship/no-ship criteria and realistic acceptance thresholds.</li>
    <li>You have a workflow that works sometimes, but behavior is inconsistent or
unmeasured.</li>
    <li>You want a concrete eval and hardening plan before investing in implementation.
</li>
  </ul>
</section>

<section>
  <h2>Inputs Required</h2>
  <table>
    <thead>
      <tr>
        <th>Input</th>
        <th>Minimum</th>
        <th>Why it matters</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>Workflow walkthrough</td>
        <td>Prompts/routing and where it runs (repo access or walkthrough)</td>
        <td>Accurate diagnosis; practical fixes; correct trace/version requirements.
</td>
      </tr>
      <tr>
        <td>Representative examples</td>
        <td>30–100 inputs (+ current outputs if available)</td>
        <td>Realistic eval coverage; must-pass selection; edge cases.</td>
      </tr>
```

```
      <tr>
        <td>Access to current implementation</td>
        <td>Endpoint or repo to invoke/inspect the workflow</td>
        <td>Enables reproducible runs and accurate failure-mode mapping.</td>
      </tr>
      <tr>
        <td>Stakeholder constraints</td>
        <td>Output consumer, required format, compliance/policy rules</td>
        <td>Correct output contract, validation strategy, and refusal/escalation
behavior.</td>
      </tr>
    </tbody>
  </table>
  <p class="small">Data handling: least-privilege access; redact sensitive fields in
shared artifacts; keep examples minimal and relevant.</p>
</section>

<section>
  <h2>How the Work Runs</h2>
  <table>
    <thead>
      <tr>
        <th>Step</th>
        <th>What happens</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td><strong>1. Scope + risk tier</strong></td>
        <td>Define workflow boundaries, unacceptable failures, and allowed uncertainty
(refuse / escalate / partial).</td>
      </tr>
      <tr>
        <td><strong>2. System teardown</strong></td>
        <td>Map prompting, routing, RAG/tool use, validators, and where failures can
occur.</td>
      </tr>
      <tr>
        <td><strong>3. Failure modes</strong></td>
        <td>Enumerate failures, severities, detection methods, and mitigations tied to
the workflow.</td>
      </tr>
      <tr>
        <td><strong>4. Eval design</strong></td>
        <td>Define test categories, must-pass set selection rules, and measurable
gates.</td>
      </tr>
      <tr>
        <td><strong>5. Operability design</strong></td>
        <td>Define trace fields, versioning scheme, monitoring signals, rollout and
rollback triggers.</td>
      </tr>
      <tr>
        <td><strong>6. Readout</strong></td>
        <td>Prioritize fixes, estimate effort, and hand off to Build & Harden.</td>
      </tr>
    </tbody>
  </table>
</section>
```

```
<section>
  <h2>Common failure patterns I look for</h2>
  <ul>
    <li>Silent partial failures (plausible but incomplete outputs)</li>
    <li>Over-broad refusals masking missing grounding</li>
    <li>Eval metrics that don't correlate with user pain</li>
    <li>Latency/cost failures that only appear at p95+</li>
  </ul>
</section>

<section>
  <h2>Deliverables</h2>
  <table>
    <thead>
      <tr>
        <th>Artifact</th>
        <th>Purpose</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>Workflow contract (input/output schemas, grounding mode, tool policy)</td>
        <td>Defines what valid requests and responses look like</td>
      </tr>
      <tr>
        <td>Failure-mode matrix</td>
        <td>Maps failures by severity, detection method, and mitigation so nothing
ships without a plan to catch it</td>
      </tr>
      <tr>
        <td>Eval plan with gate thresholds</td>
        <td>Specifies test categories, must-pass rules, and acceptance criteria so ship
decisions are evidence-based</td>
      </tr>
      <tr>
        <td>Ops plan (trace fields, monitoring signals, rollback triggers)</td>
        <td>Defines what gets logged and when to act so incidents are diagnosable
without guesswork</td>
      </tr>
      <tr>
        <td>Ship/no-ship recommendation</td>
        <td>Evidence-backed decision with rationale and next steps</td>
      </tr>
      <tr>
        <td>Prioritized hardening roadmap</td>
        <td>Sequenced fixes ready for Build & Harden</td>
      </tr>
    </tbody>
  </table>
</section>

<section>
  <h2>Definition of Done</h2>
  <div class="success-list">
    <ul>
      <li>Workflow spec documents all agreed success criteria and known failure modes,
with no ambiguous edge cases.</li>
      <li>Eval plan covers all identified critical failure modes with measurable
acceptance thresholds.</li>
      <li>Prioritized hardening roadmap with sequenced fixes and clear ownership.</li>
```

```
      <li>Ship/no-ship recommendation is evidence-backed with documented rationale and
actionable next steps.</li>
    </ul>
  </div>
</section>

<section>
  <h2>Boundaries</h2>
  <div class="boundaries">
    <ul>
      <li>Implementing the harness, CI gates, or production rollout (covered in Build &
Harden / Retainer).</li>
      <li>Large-scale labeling programs or broad multi-workflow platform work (can be
separately scoped).</li>
    </ul>
  </div>
</section>

<div class="contact-section">
  <div class="contact-grid">
    <div class="contact-text">
      <h3>Next Step</h3>
      <p>Share one workflow and 30–100 representative examples to start. If examples
are sensitive, start with synthetic or redacted cases and iterate.</p>
      <div class="contact-action">
        <strong>Book an intro call:</strong><br>
        <a href="https://calendly.com/philipstevens4/intro" class="contact-
url">calendly.com/philipstevens4/intro</a>
      </div>
      <div class="contact-details">
        <strong>Email:</strong> philipstevens4@gmail.com<br>
        <strong>Web:</strong> philipstevens.github.io
      </div>
      <p style="font-size: 12px; color: var(--muted); margin-top: 16px;">
        <strong>Not ready?</strong> Get the free <a
href="https://philipstevens.github.io/#lead-capture" style="color: var(--
accent);">Production Readiness Checklist</a> to self-assess first.
      </p>
    </div>
    <div class="qr-box">
      <img src="https://api.qrserver.com/v1/create-qr-code/?
size=100x100&data=https://calendly.com/philipstevens4/intro" alt="QR Code to book intro
call" />
      <div class="qr-label">Scan to book a call</div>
    </div>
  </div>
</div>

<footer class="footer">
  <div>Phil Stevens — LLM Workflow Reliability</div>
  <div>philipstevens.github.io</div>
</footer>
```

========== assets/downloads/llm-workflow-build-and-harden.html ==========

<!doctype html>

```
    <h1>LLM Workflow Build & Harden</h1>
    <p class="subtitle">Implement eval gates and a regression harness, then harden the
workflow until it meets the production bar.</p>
  </div>
  <div class="header-right">
    <div class="name">Phil Stevens</div>
    <div>philipstevens.github.io</div>
  </div>
</header>

<dl class="overview">
  <dt>Scope</dt>
  <dd>One LLM workflow with an agreed production bar, hardened to meet quality, safety,
and operational targets.</dd>
  <dt>Primary output</dt>
  <dd>Eval harness in CI + hardened workflow + release gates + rollback plan (ready to
ship).</dd>
</dl>

<section>
  <h2>Best Fit</h2>
  <ul>
    <li>You want this workflow shipped with predictable behavior on real inputs.</li>
    <li>You need eval gating in CI to prevent regressions and unsafe releases.</li>
    <li>You have enough examples and access to iterate quickly.</li>
  </ul>
</section>

<section>
  <h2>Inputs Required</h2>
  <table>
    <thead>
      <tr>
        <th>Input</th>
        <th>Minimum</th>
        <th>Why it matters</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>Agreed production bar</td>
        <td>Workflow spec with acceptance criteria (from Audit or equivalent)</td>
        <td>Defines what "done" means so hardening has a clear target.</td>
      </tr>
      <tr>
        <td>Example set</td>
        <td>50+ representative examples; 10–30 must-pass candidates</td>
        <td>Prevents "happy-path only" evals; makes gates meaningful.</td>
      </tr>
      <tr>
        <td>Integration access</td>
        <td>Repo access or endpoint to invoke the workflow in staging</td>
        <td>Allows harness integration, CI gating, and reproducible runs.</td>
      </tr>
      <tr>
        <td>SME review bandwidth</td>
        <td>Someone who can review outputs and approve changes weekly</td>
        <td>Enables fast iteration and prevents quality drift during hardening.</td>
```

```
      </tr>
    </tbody>
  </table>
</section>

<section>
  <h2>How the Work Runs</h2>
  <table>
    <thead>
      <tr>
        <th>Phase</th>
        <th>What happens</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td><strong>0. Contracts + baseline</strong></td>
        <td>Enforce output contracts, encode refusal/escalation policy, define trace
fields, establish baseline.</td>
      </tr>
      <tr>
        <td><strong>1. Eval harness + CI gates</strong></td>
        <td>Build case runner, integrate must-pass blocking gate, add automated checks
to CI.</td>
      </tr>
      <tr>
        <td><strong>2. Hardening loops</strong></td>
        <td>Iterate on prompts, RAG tuning, tool controls, and cost/latency until gates
pass consistently.</td>
      </tr>
      <tr>
        <td><strong>3. Release readiness</strong></td>
        <td>Define rollout plan, test rollback procedure in staging, finalize
monitoring checklist.</td>
      </tr>
    </tbody>
  </table>
  <p class="small">Timeline varies by integration complexity and number of failure
surfaces (RAG/tools/agents).</p>

</section>

<section>
  <h2>What usually moves the needle</h2>
  <ul>
    <li>Prompt restructuring vs prompt expansion</li>
    <li>Moving logic out of the model and into validators</li>
    <li>Shrinking must-pass sets instead of bloating evals</li>
    <li>When <em>not</em> to fine-tune</li>
  </ul>
</section>

<section>
  <h2>Deliverables</h2>
  <table>
    <thead>
      <tr>
        <th>Artifact</th>
        <th>Purpose</th>
      </tr>
```

```
    </thead>
    <tbody>
      <tr>
        <td>Eval harness integrated into CI</td>
        <td>Runs tests on every PR so regressions never reach production unnoticed</td>
      </tr>
      <tr>
        <td>Test sets (golden, edge, regression, safety, high-stakes)</td>
        <td>Covers critical paths and known failure modes so you catch real problems,
not just happy-path passes</td>
      </tr>
      <tr>
        <td>Must-pass gate configuration</td>
        <td>Defines which failures block merges and releases so bad changes can't slip
through</td>
      </tr>
      <tr>
        <td>Output contract enforcement (schema + parser)</td>
        <td>Ensures every response is valid or explicitly fails</td>
      </tr>
      <tr>
        <td>Trace implementation (versions, tool calls, retrieval IDs)</td>
        <td>Makes outputs debuggable and auditable</td>
      </tr>
      <tr>
        <td>Release checklist and rollback plan</td>
        <td>Documents rollout steps and tested recovery procedure</td>
      </tr>
    </tbody>
  </table>
</section>

<section>
  <h2>Definition of Done</h2>
  <div class="checklist">
    <ul>
      <li>Eval harness runs in CI with a must-pass subset that blocks merges and
releases; no untested must-pass cases.</li>
      <li>Coverage includes all agreed critical failure modes and representative data
slices, with no known gaps.</li>
      <li>Workflow meets all defined thresholds (quality, safety, cost/latency) with
documented evidence.</li>
      <li>Release checklist and rollback plan exist, are tested at least once in
staging, and are signed off.</li>
    </ul>
  </div>
</section>

<section>
  <h2>Boundaries</h2>
  <div class="boundaries">
    <ul>
      <li>Net-new product feature design is out of scope (this is reliability and
shipping operations).</li>
      <li>Large re-architecture or platform build can be scoped separately if needed.
</li>
    </ul>
  </div>
</section>
```

```
<div class="contact-section">
  <div class="contact-grid">
    <div class="contact-text">
      <h3>Next Step</h3>
      <p>Start with an Audit or provide a workflow endpoint + example set to begin
Phase 0. Fastest path: contract → harness/CI → hardening loops → release ops.</p>
      <div class="contact-action">
        <strong>Book an intro call:</strong><br>
        <a href="https://calendly.com/philipstevens4/intro" class="contact-
url">calendly.com/philipstevens4/intro</a>
      </div>
      <div class="contact-details">
        <strong>Email:</strong> philipstevens4@gmail.com<br>
        <strong>Web:</strong> philipstevens.github.io
      </div>
      <p style="font-size: 12px; color: var(--muted); margin-top: 16px;">
        <strong>Not ready?</strong> Get the free <a
href="https://philipstevens.github.io/#lead-capture" style="color: var(--
accent);">Production Readiness Checklist</a> to self-assess first.
      </p>
    </div>
    <div class="qr-box">
      <img src="https://api.qrserver.com/v1/create-qr-code/?
size=100x100&data=https://calendly.com/philipstevens4/intro" alt="QR Code to book intro
call" />
      <div class="qr-label">Scan to book a call</div>
    </div>
  </div>
</div>

<footer class="footer">
  <div>Phil Stevens — LLM Workflow Reliability</div>
  <div>philipstevens.github.io</div>
</footer>
```

========== assets/downloads/llm-workflow-release-ops.html ========== <!doctype html>

```
    <h1>LLM Workflow Release Ops</h1>
    <p class="subtitle">Operate releases with eval gating, drift detection, and
rollback discipline so you ship safely over time.</p>
  </div>
  <div class="header-right">
    <div class="name">Phil Stevens</div>
    <div>philipstevens.github.io</div>
  </div>
</header>

<dl class="overview">
  <dt>Scope</dt>
  <dd>One or more production LLM workflows requiring ongoing release discipline and
drift management.</dd>
  <dt>Primary output</dt>
```

```
  <dd>Gated releases + drift monitoring + threshold maintenance + rollback readiness
(sustained over time).</dd>
</dl>

<section>
  <h2>Best Fit</h2>
  <ul>
    <li>Your workflow is live, and regressions or drift are a recurring risk.</li>
    <li>You want repeatable releases with clear evidence and fast rollback.</li>
    <li>You already have (or want to establish) an eval harness and baseline gates.
</li>
  </ul>
</section>

<section>
  <h2>Inputs Required</h2>
  <table>
    <thead>
      <tr>
        <th>Input</th>
        <th>Minimum</th>
        <th>Why it matters</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>Baseline eval suite</td>
        <td>Runnable gate suite with must-pass subset</td>
        <td>Release ops is about preventing regressions over time.</td>
      </tr>
      <tr>
        <td>Telemetry access</td>
        <td>Version identifiers in trace/logs; production signals</td>
        <td>Without this, incident response becomes guesswork.</td>
      </tr>
      <tr>
        <td>Release process</td>
        <td>Canary/shadow/ramp capability (even manual)</td>
        <td>Allows safe rollout and rapid rollback.</td>
      </tr>
      <tr>
        <td>Owner plus cadence</td>
        <td>Someone who can approve releases and review weekly</td>
        <td>Ensures accountability and keeps drift monitoring actionable.</td>
      </tr>
    </tbody>
  </table>
  <p class="small">If you don't have these yet, the first month often starts with a
short Build & Harden sprint to establish them.</p>
</section>

<section>
  <h2>How the Work Runs</h2>
  <table>
    <thead>
      <tr>
        <th style="width: 15%;">Cadence</th>
        <th>What happens</th>
      </tr>
    </thead>
```

```
    <tbody>
      <tr>
        <td><span class="cadence-label cadence-weekly">Weekly</span></td>
        <td>Review production signals, sample quality on a rotating set, triage drift
indicators and prioritize fixes.</td>
      </tr>
      <tr>
        <td><span class="cadence-label cadence-release">Per release</span></td>
        <td>Run eval gates on proposed changes, analyze deltas vs baseline, define
rollout plan and rollback triggers.</td>
      </tr>
      <tr>
        <td><span class="cadence-label cadence-monthly">Monthly</span></td>
        <td>Update test sets from incidents, review and ratchet gate thresholds,
calibrate alerts to reduce noise.</td>
      </tr>
    </tbody>
  </table>

</section>

<section>
  <h2>What breaks most release processes</h2>
  <ul>
    <li>Gates that are too slow to run, so they're skipped</li>
    <li>Thresholds that never get ratcheted</li>
    <li>Drift signals without an owner</li>
    <li>Rollbacks that exist on paper only</li>
  </ul>
</section>

<section>
  <h2>Deliverables</h2>
  <table>
    <thead>
      <tr>
        <th>Artifact</th>
        <th>Purpose</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>Maintained eval suite (test sets updated from incidents)</td>
        <td>Keeps coverage current as product and failure modes evolve so yesterday's
bugs don't recur tomorrow</td>
      </tr>
      <tr>
        <td>Release gate reports</td>
        <td>Documents what changed, what passed/failed, and ship decision so every
release has a paper trail</td>
      </tr>
      <tr>
        <td>Weekly stability memos</td>
        <td>Summarizes trends, drift signals, and recommended fixes so problems surface
before users report them</td>
      </tr>
      <tr>
        <td>Drift monitoring dashboards and alerts</td>
        <td>Surfaces regressions before users report them</td>
      </tr>
```

```
      <tr>
        <td>Updated thresholds and gate configurations</td>
        <td>Ratchets quality bar as workflow stabilizes</td>
      </tr>
      <tr>
        <td>Rollback runbook (tested and current)</td>
        <td>Ensures fast recovery when releases fail</td>
      </tr>
    </tbody>
  </table>
</section>

<section>
  <h2>Definition of Done</h2>
  <div class="success-list">
    <ul>
      <li>Every release has a gate report documenting what changed, what passed/failed,
and a signed-off ship decision.</li>
      <li>Drift and regression monitoring is in place with defined signals, owners, and
documented response steps.</li>
      <li>Thresholds and test sets are maintained with versioned updates and review
sign-off on each change.</li>
      <li>Rollback is documented and exercised at least once per quarter so recovery
time stays under SLA.</li>
    </ul>
  </div>
</section>

<section>
  <h2>Boundaries</h2>
  <div class="boundaries">
    <ul>
      <li>Building brand-new workflows from scratch (handled via Audit + Build &
Harden).</li>
      <li>Large re-architecture or platform build work.</li>
      <li>Large labeling efforts or full-time QA programs.</li>
    </ul>
  </div>
</section>

<div class="contact-section">
  <div class="contact-grid">
    <div class="contact-text">
      <h3>Next Step</h3>
      <p>Review one workflow's current gates and monitoring, then define the first
month's reliability plan. If you're pre-prod, start with an Audit or Build & Harden
sprint.</p>
      <div class="contact-action">
        <strong>Book an intro call:</strong><br>
        <a href="https://calendly.com/philipstevens4/intro" class="contact-
url">calendly.com/philipstevens4/intro</a>
      </div>
      <div class="contact-details">
        <strong>Email:</strong> philipstevens4@gmail.com<br>
        <strong>Web:</strong> philipstevens.github.io
      </div>
      <p style="font-size: 12px; color: var(--muted); margin-top: 16px;">
        <strong>Not ready?</strong> Get the free <a
href="https://philipstevens.github.io/#lead-capture" style="color: var(--
accent);">Production Readiness Checklist</a> to self-assess first.
```

```
      </p>
    </div>
    <div class="qr-box">
      <img src="https://api.qrserver.com/v1/create-qr-code/?
size=100x100&data=https://calendly.com/philipstevens4/intro" alt="QR Code to book intro
call" />
      <div class="qr-label">Scan to book a call</div>
    </div>
  </div>
</div>

<footer class="footer">
  <div>Phil Stevens — LLM Workflow Reliability</div>
  <div>philipstevens.github.io</div>
</footer>
```