@demotoprod · EVALS · also LEVERS

Eval checklist

One provider's evaluation guide lists vibe-based evals, judging by "it seems like it's working", as an anti-pattern. A change tried on a few examples is tested on those examples only. Four checks, each sized for an afternoon: a golden set, a CI run, a failing margin and a score history.

1 Build a golden set from real cases

One engineering post calls 20 to 50 simple tasks drawn from real failures a great start. A provider's evaluation guide says to log as you develop so you can mine the logs for cases, include edge and adversarial cases, and avoid datasets that do not match production traffic.

- Pull 20 to 50 real inputs from logs and tickets, favoring past failures; write each known-good answer first and commit it.
- Use an exact or string match where you can; check any model grader against your own labels before it can fail a build.

2 Run it on every prompt, model or retrieval change

One provider's guide advises pinning production applications to specific model snapshots for consistent behavior, and says even different snapshots within the same family could produce different results. A published CI example runs its evals only when its prompt files or its config file change.

- List the dated model ID, prompt and retrieval settings in files the workflow watches.
- Run the set on the new snapshot before switching, and keep the old one live until the score holds.

```
on:
  pull_request:
    paths: ['prompts/**', 'models.yaml', 'retrieval/**']
```

3 Fail the build on a drop larger than the noise

A task that passed on one eval run might fail on the next. One published power calculation needs about 969 independent questions to detect a 3-point difference 80% of the time at a 5% false-positive rate, so a 20 to 50 case set can flag only large drops.

- Run the unchanged prompt on the full set 5 times; set the failing margin above the spread of those scores.
- Fail the build below baseline minus margin; save a new baseline only after a person reviews the failing cases.

```
import json, sys
new = json.load(open("results.json"))["score"]
base = json.load(open("baseline.json"))["score"]
MARGIN = 0.05   # placeholder: set above your 5-run spread
if new < base - MARGIN:
    sys.exit(f"golden set fell: {base:.2f} to {new:.2f}")
```

4 Track the score and add every miss

A score only means something next to the last one, so log every run. An eval at 100% tracks regressions but gives no signal for improvement.

- Log one row per run (date, commit, model ID, score); chart it and compare with the last run.
- Turn each production miss into a new case that week; at 100%, add harder cases.

Nitish Gautam · indica-tech.com · hello@indica-tech.com

INDICA TECH · @DEMOTOPROD

Nitish Gautam

Founder-led AI engineering. Demo to production, under NDA, in regulated industries, on fixed scope.

Paste into your project

ONE FILE FOR THE WHOLE GOLDEN SET

```
# evals/golden.yaml: what the CI job reads
cases: evals/golden.jsonl       # 20 to 50 real inputs, answers written first
model: <dated model ID>         # a pinned snapshot, never an alias
watch: [prompts/, models.yaml, retrieval/]
baseline_runs: 5                # unchanged prompt, full set
margin: <above your 5-run spread>
fail_if: score < baseline - margin
history: evals/history.csv      # date, commit, model ID, score
on_miss: add the case that week
```

Where this list comes from

Ten-plus years taking products from zero to one across seven sectors, including defence, fintech, medical and telecom, with AI shipped into regulated and air-gapped environments. The @demotoprod channel takes one production AI failure apart each day. These checklists are the same work written down: every item is a dated failure or a documented default, with the control that stops it and the primary source.

If this list found a gap

1. ### Your demo works and production is stuck.

   Production Readiness Audit. Fixed price, two weeks. A five-dimension scorecard, cost and failure analysis, and a 90-day execution roadmap.

2. ### You need senior AI judgement without a full-time hire.

   Fractional AI Lead. Monthly: 90-day minimum, then 30 days' notice either side. Weekly 1:1 with the founder or CTO, architecture and code review, technical hiring support, investor-ready due diligence.

3. ### An AI feature has to work, by a date.

   AI Build Sprint. Fixed price, four to eight weeks. Scope signed off up front, weekly demos, CI/CD and observability included, 30-day production warranty.

4. ### A regulator, an auditor or a customer's security team is asking.

   AI Governance & Compliance. Fixed price, four weeks. ISO 27001 / SOC 2-aligned controls, a model risk framework, human-in-the-loop and audit logging, regulator-ready documentation.

BOOK A CALL

indica-tech.com/#contact

EMAIL

hello@indica-tech.com

WEBSITE

indica-tech.com

ALL CHECKLISTS

indica-tech.com/checklists/

YOUTUBE

youtube.com/@demotoprod

LINKEDIN

linkedin.com/in/startup-founderx

INSTAGRAM

instagram.com/demotoprod

TIKTOK

tiktok.com/@demotoprod

Free to share. The live list, with every source link: indica-tech.com/checklists/#eval-golden-set