FREE PRACTITIONER PACK

Agentic AI Deployment Pack

What to measure, how to build the golden set, how to judge, and how to gate releases. If you cannot measure the agent, you cannot ship it.

Pack	Agentic AI Deployment Pack
Document	Evaluation guide (3 pages)
Use	Quality engineering · release gates
Date	2026-09

AnovaCloud — Engineering the AI-Native Enterprise. Frisco, Texas. This pack is a working instrument, not a brochure: complete it with your team, score honestly, and bring it to your discovery call.

AnovaCloud — anovacloud.com

EVALUATION DIMENSIONS

Dimension	What it measures	Why it matters
Task success	Did the agent achieve the outcome? Graded against the golden set's expected result.	Primary release gate
Tool-use correctness	Right tool, right arguments, right sequence — graded per step, not just final answer.	Catches silent wrong-tool failures
Groundedness	Claims traceable to retrieved sources or tool outputs; no unsupported assertions.	Critical for RAG and research agents
Instruction following	Respected constraints: scope, format, policy, and explicit user instructions.	Policy compliance signal
Safety & refusal	Refused or escalated disallowed requests; handled adversarial inputs per policy.	Red-team dimension
Latency	Time to first useful output and to completion, p50/p95 per task class.	UX and cost driver
Cost per task	Model + tool + infra cost per completed task, tracked per task class.	Unit economics
Human effort	Review time, override rate, escalation rate — the hidden cost of the tier.	Oversight sustainability

BUILDING THE GOLDEN SET

- Start with 30–50 tasks sampled from real usage — real tickets, real questions, real edge cases — not synthetic happy paths.
- Stratify deliberately: routine, ambiguous, adversarial, and high-stakes tasks each represented. The set should embarrass a weak agent.
- Record expected outcomes, not just expected text. Grade the result (record created? correct answer? right tool called?) — text similarity is a weak proxy.
- Version the set. Every eval run records the set version; changes to the set are reviewed like code changes.
- Keep a held-out slice the team does not tune against. Tuning to the test is the oldest failure in evaluation.

JUDGING: HUMAN, MODEL, OR BOTH

- Human grading is ground truth. Calibrate graders with a rubric and examples; measure inter-grader agreement before trusting the labels.
- LLM judges scale grading — but a judge must itself be evaluated: measure judge-vs-human agreement on a labeled sample before relying on it.
- Use deterministic checks wherever possible: schema validation, expected tool calls, database state, exact-match on structured outputs. Deterministic beats judged.
- Separate the judge from the agent: different model or version where feasible, and never let the agent see the judge's rubric as an optimization target.

GATING RELEASES

- No green suite, no deploy. The golden set runs in CI on every prompt, model, tool, or policy change — regressions block release.
- Set per-dimension thresholds in writing with the business owner (e.g. minimum task-success rate, maximum cost per task). Thresholds are release criteria, not aspirations.
- Canary new versions on a slice of traffic with human review before full rollout; compare against the incumbent on the same tasks.
- Every production incident becomes eval cases. The golden set grows from failures — that is how the suite earns its keep.

REVIEW CADENCE

Weekly: eval dashboard review with the technical owner (success rate, cost, overrides, escalations). Monthly: golden-set health — retire stale tasks, add incident-derived cases, re-calibrate judges. Quarterly: threshold review with the business owner — are the gates still the right gates?