ROSE JONES · AN ENGINEERING WALKTHROUGH

Agent Evaluation

For engineers who have never evaluated an agent before: a fundamental mental model, not a tool tutorial.

If you can't define success, you can't measure it. If you can't measure it, you can't improve it.

01 · WHY BOTHER

Development without evaluation is guessing

WITHOUT EVALS

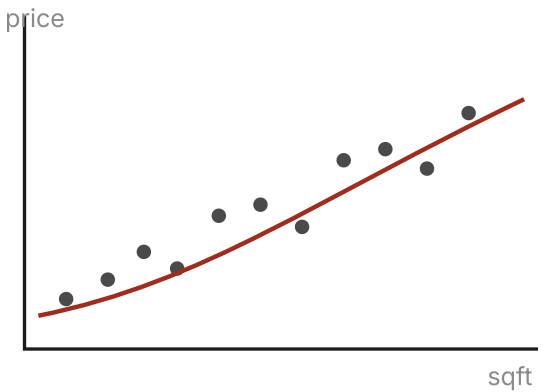
- "It seems better" after every prompt tweak
- Fix one case, silently break three others
- No way to compare models, prompts, or architectures
- Demos work; production surprises you

WITH EVALS

- Every change scored against the same yardstick
- Regressions caught before users see them
- An objective basis for choosing model, prompt, design
- Evaluation drives iteration, not the other way around

Classic ML: clean ground truth, clean metrics

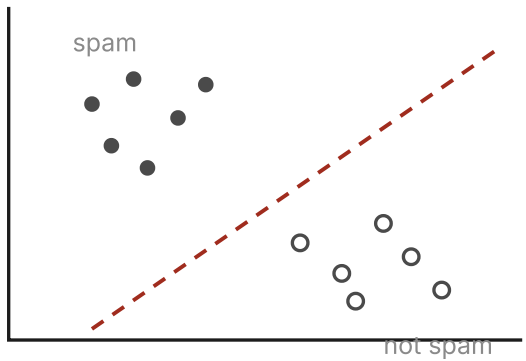
REGRESSION · PREDICT A NUMBER



House price from square footage, bedrooms, location. Fit a curve; measure the distance from reality.

MAE · MSE · RMSE · R^2

CLASSIFICATION · PICK A CATEGORY



Spam or not-spam, from keywords, sender, formatting. Every prediction is simply right or wrong.

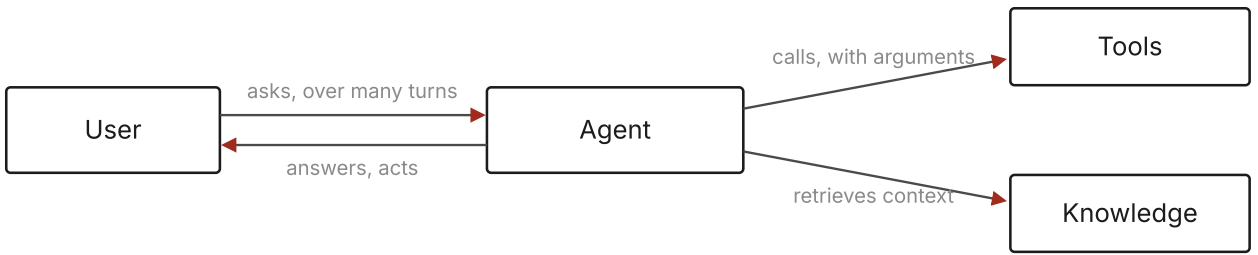
Accuracy · Precision · Recall · F1 · AUC-ROC

The luxury of this era was a **mathematically defined ground truth.** *Hold that thought: it is exactly what we lose next.*

03 · WHAT CHANGED

A model we didn't build, wrapped in an agent

We no longer train the model; our levers are context, prompts, and parameters. And "good" became subjective, graded by similarity scores, judges, and humans. Then agents raised the stakes further:



THEY ACT

Any tool call can fail even when the words sound right.

THEY CONVERSE

Context must survive the session; each turn should advance the task.

THEY RETRIEVE

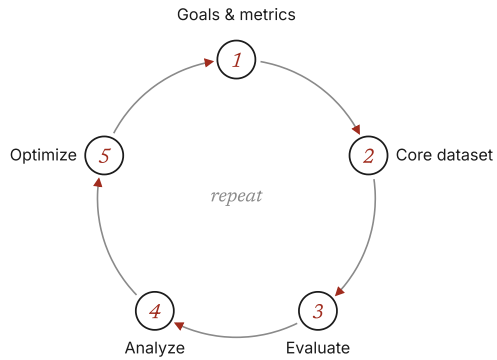
Wrong context in, wrong answer out, whatever the model.

So evaluation must mirror the full lifecycle: not was this sentence good? but did the agent do the right thing, the right way, end to end?

04 · THE CORE PRACTICE

Eval-driven development

1	**Goals & metrics**	Define what "good" looks like.
2	**Core dataset**	Real examples, expected outcomes.
3	**Evaluate**	Score against the dataset.
4	**Analyze**	What failed, and why?
5	**Optimize**	Adjust, then repeat the cycle.



Steps 1 and 2 are the whole game. *Without defined success and concrete examples, every iteration after them is just noise.*

05 · GROUND TRUTH, REBUILT

The core dataset

WHAT IT IS

- Input, context, and expected outcome, per example
- Drawn from real situations: past conversations, tickets, frequent questions
- Covers the frustrating and slow paths, not just the happy ones
- Paired with a rubric: a scoring guide that encodes expert judgment
- Small and honest beats large and synthetic; start with dozens

ONE EXAMPLE, IN FULL

INPUT	*"Where's my refund for order #8123?"*
CONTEXT	Order record (return received Jul 2) · refund policy (5 to 10 business days)
EXPECTED	Confirms the return was received; states the 5-to-10-day window; makes no promise outside policy
RUBRIC	Grounded in policy (0 to 2) · order facts correct (0 to 2) · tone (0 to 1) · offers next step (0 to 1)

A support-agent case. Note the expected outcome describes behavior, not exact wording.

05B · GROUND TRUTH, REBUILT

The regression dataset

Every fixed failure becomes a permanent test case, run on every change like a test suite. Here is one case being born:

1	**Incident**	Jul 3: the agent promises an *instant* refund. Policy says 5 to 10 days. A user screenshot circulates.
2	**Trace**	Pull the full interaction: the input, what was retrieved, which tools ran, the exact output.
3	**Distill**	Freeze it as a test case. Same input and context; expected: quote the policy window, never promise "instant".
4	**Guard**	The case joins the regression set. Any future prompt or model change that reintroduces the promise fails loudly.

Remember "fix one case, break three others"? **This is the set that catches the other three.**

06 · CHOOSING METRICS

Match the metric to the agent

AGENT TYPE	WHAT YOU MEASURE
RAG	Answer relevancy · faithfulness · contextual precision & recall · citation quality
Tool-using	Task completion · tool correctness · argument correctness
Multi-turn	Turn relevancy & efficiency · role adherence · knowledge retention · conversation completeness
Custom	LLM-as-a-judge with weighted criteria: helpfulness, reasoning, tone

Evaluation is not one-size-fits-all. Frameworks such as DeepEval and LangSmith ship these as ready-made scorers; choose yours before you build.

The same refund case, under each lens

METRIC	THE QUESTION IT ASKS	WHAT FAILING LOOKS LIKE
Answer relevancy RAG	Did the reply address what was asked?	*Asked "where's my refund?", got a tutorial on starting a return.*
Faithfulness RAG	Does the answer stick to the retrieved context?	*Policy says 5 to 10 days; the agent invents "usually within 48 hours".*
Contextual recall RAG	Did retrieval fetch everything the answer needs?	*Policy retrieved, order record missed: can't confirm the return arrived.*
Contextual precision RAG	Is the relevant material ranked on top?	*Eight shipping-label docs first; the refund policy buried at rank 7.*
Tool & argument correctness TOOLS	Right tool, right inputs?	*Calls lookup_order with #8132 instead of #8123.*
Knowledge retention MULTI-TURN	Does it remember what it was told?	*Order number given in turn 1, asked for again in turn 4.*
Turn relevancy MULTI-TURN	Does each turn move the task forward?	*Mid-case, the agent pivots to promoting the loyalty program.*
Role adherence MULTI-TURN	Does it stay the agent you defined?	*The refund assistant starts offering tax advice.*

A metric is just a question you ask of the same trace. If the name sounds fuzzy, **say the failure out loud.**

LLM-as-a-judge, demystified

WHAT IT IS

- A model handed your rubric and asked to grade another model's output
- Exists because subjective criteria don't scale to humans reading every case
- Returns scores *with reasons*, so failures stay debuggable
- The judge is a model too: it has biases, and it needs spot-checks against human labels before you trust it

THE RUBRIC, APPLIED

JUDGE PROMPT	You are grading a refund-support reply. Policy: refunds take 5 to 10 business days; return received Jul 2. Score the reply: grounded in policy (0 to 2), order facts (0 to 2), tone (0 to 1), offers next step (0 to 1). Give a one-line reason per score.
REPLY UNDER TEST	*"Your return arrived July 2, so your refund is on the way! It usually lands within 48 hours."*
JUDGE OUTPUT	grounded **0/2** ("48 hours" contradicts policy) · facts 2/2 · tone 1/1 · next step **0/1** (none offered) → **3 of 6, fail**

The rubric from the core-dataset slide, executed by a model instead of a human.

One question per criterion, scores with reasons, and **a human audit of the judge itself**. *That's the whole trick.*

08 · READING THE NUMBERS

What a run actually looks like

SUITE	PROMPT V3	PROMPT V4
Core set · 50 cases	39 / 50 (78%)	42 / 50 (84%)
Regression set · 12 cases	12 / 12	**11 / 12**
Case #12 · "instant refund"	pass	**fail**
Verdict	shipped	**held**

Pass rates are over three runs per case. Agents are nondeterministic: the same eval run twice gives different numbers, so score repeated runs, never a single pass.

v4 wins the average and still can't ship: it re-broke case #12. **Aggregates hide regressions; the regression set catches them.**

09 · AFTER YOU SHIP

Online evaluation: production signals

DID IT WORK?

- Resolution rate, thumbs up or down
- Escalation rate: did they ask for a human?
- Turns to resolution

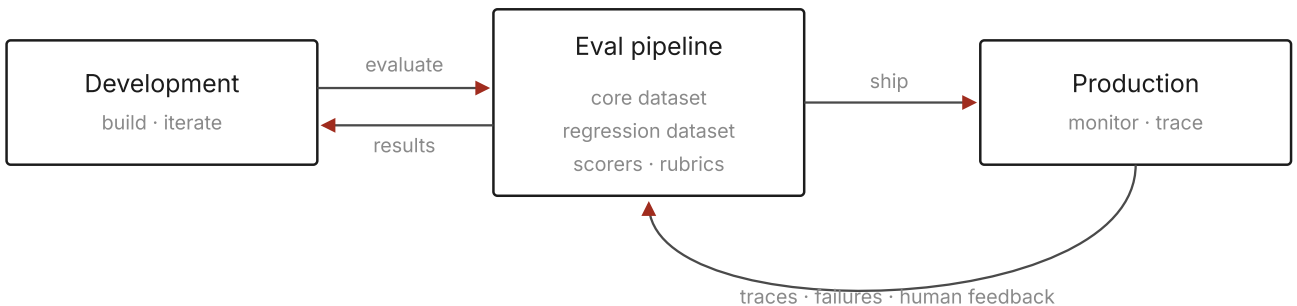
WAS IT WORTH IT?

- Token usage, and so cost per resolution
- Latency and time-to-first-token; no one likes a slow agent

WHAT DO USERS SAY?

- Qualitative feedback: likes, dislikes, expectations
- User feedback is critical; it defines what "good" means next

Close the loop



Failures become regression cases; strong real examples enrich the core set. Both feed the next development cycle.

When something breaks, first ask: **did our evals miss it?** *Fix the eval, then the prompt.*

11 · TAKEAWAYS

If you remember four things

1	**Define success first**	Actionable, measurable goals before any building. Vague goals produce unmeasurable agents.
2	**Build the datasets**	A core set for ground truth, a regression set for past failures. Start small, keep them honest.
3	**Metrics follow the agent**	RAG, tools, and multi-turn each need their own scorers, plus audited judges for the subjective parts.
4	**Evaluation never ends**	A continuous loop between model behavior, user experience, and system performance.

The goal, in one line: agents that are **reliable, measurable, and improving over time.**