

PUBLISHED: NOVEMBER 4TH · BY SHREYAS KALYANPAD

An eval is a test set for an AI feature: real inputs, plus a written rule for what counts as a good answer.

Almost nobody has one. LangChain surveyed 1,340 people and published the results in June. 89% had set up observability, so they can watch every step their agent takes. Only 52% run evals against a test set before they ship. About a third named quality as the main thing keeping them out of production.

Teams can see exactly what their AI did. They cannot say whether it was good.

> Observability tells you what happened. An eval tells you whether it should have.

Why this lands on the designer

Deciding what counts as a good answer is not an engineering question. Teresa Torres puts it well. Correctness depends on context, so do not let a vendor define it for you. A student asking about astrophysics wants a long teaching answer. A professor asking the same question does not. Same input, different right answer.

Anthropic's own eval guidance lands in the same place from the other side: the people closest to the product and its users are best placed to define success. That is your chair.



This is what a design review looks like once somebody has written the test set.

1. Twenty inputs beat one good demo

What teams do: judge the feature on the demo that went well in the review.

Why it fails: you chose that input. Anthropic warns that teams burn weeks on features that pass early testing, then miss expectations nobody wrote down.

Do instead: 20 to 50 inputs pulled from real failures. That is their starting point, not hundreds.

2. Write tasks two people would score the same

What teams do: "the answer should be helpful."

Why it fails: Hamel Husain and Shreya Shankar call out scores like helpfulness and coherence. They measure abstract qualities that may not matter to your product.

Do instead: use Anthropic's bar. A task is written well when two people who know the domain reach the same verdict on their own.

3. Pass or fail, never one to five

What teams do: rate each output out of five.

Why it fails: a 3 means something different to every reviewer, and people park in the middle to avoid deciding.

Do instead: go binary. If you need shades, split one question into several yes or no checks.

4. Read thirty outputs yourself first

What teams do: ask the model to find its own failures.

Why it fails: its guesses shape what you then notice. Husain and Shankar say keep the first pass manual for exactly this reason.

Do instead: read at least 30 yourself and write rough notes. Then let the model hunt for more.

5. Test the cases that should not happen

What teams do: check the feature does its job.

Why it fails: one-sided evals create one-sided optimization. Anthropic hit this building web search for Claude. Test only whether it searches when it should, and you get a thing that searches for everything.

Do instead: make half your list cases where the feature should stay quiet.

6. Passing once is not passing

What teams do: run each input a single time.

Why it fails: the same input gives different answers. At a 75% success rate per try, the odds of passing three runs in a row are about 42%.

Do instead: run each input a few times. Say out loud whether you need it right once or right every time.

STOP SAYING	WRITE DOWN INSTEAD
"The demo felt good"	20 to 50 inputs from real failures
"Make it more helpful"	A rule two domain experts would agree on
"It scores 4 out of 5"	Pass or fail, split into separate checks
"The model flagged these"	Thirty outputs you read yourself
"It does the job"	The cases where it should do nothing
"It worked when I tried it"	Three runs, same input, same verdict

The audit

Open the last AI feature your team shipped. Write the twenty inputs you would bet on. Score today's output pass or fail against each one. If you cannot fill that list, your team is shipping on feel, and the review you sat through was never a design review.

Checked against

Read on 4 November 2026.

- LangChain, State of Agent Engineering: 1,340 responses, 89% have observability, 52.4% run offline evals on a test set, quality is the top barrier to production
- Anthropic, Demystifying evals for AI agents: 20 to 50 tasks from real failures, tasks two domain experts would score the same, balanced problem sets, pass^k
- Hamel Husain and Shreya Shankar, AI Evals FAQ: binary pass/fail over Likert scales, review at least 30 traces yourself, why generic metrics mislead
- Teresa Torres, Product Talk, AI Evals: A Hands-On Guide for Product Teams: correctness is context dependent, do not outsource what good looks like