GUIDE · FREE

How to know whether your AI really does what you think it does

Almost every organization deploying AI validates on vibes, on a handful of examples that work. This guide explains what a serious evaluation is, why it shouldn't be handed to the people who built the system, and what it opens up once it exists.

SynaLinks is an applied research firm working on neuro-symbolic systems, based in Toulouse, France. We build the datasets and evaluations that decide whether an AI system holds up in production, as well as the open-source frameworks that produce them.

ONE CONFUSION FIRST

Three fake evaluations.

Public benchmark scores

They compare models against each other on generic tasks. They say nothing about what yours does on your records, or for your customers. A model that tops the leaderboard can be bad at your job.

Manual spot checks, or “vibe-testing”

Someone asks fifteen questions, reads the answers, decides it looks good. Not reproducible, not comparable between versions. And the fifteen questions are the ones the system already handles.

A model grading itself, or “LLM-as-a-judge”

Often the only workable method at scale, but an unvalidated judge is one more opinion. Calibrate it against human judgments and measure its agreement before trusting it.

Cases from reality
your records, real questions, failures included

Written criteria
what a correct answer is, case by case

A harness
that replays the whole thing on demand

=

A measurement
Reproducible. Comparable between versions.

Replaying cases is easy. Deciding whether an answer is correct, much less so.

That verdict decides everything, and most tooling can only reach it by asking another model for its opinion.

This is our speciality. We build **neuro-symbolic** systems: a language model backed by a symbolic layer made of code, logic and constraints. That layer is what makes verification automatic. An answer is no longer judged, it is checked: a rule passes or fails, a query returns the right row or it doesn't. The verdict is identical on every run, it explains itself, and it replays for a fraction of what a model judge costs.

THE UNCOMFORTABLE PART

Nobody grades their own homework.

The problem has nothing to do with honesty: it is structural, and it works against the most scrupulous people you can hire.

THE BUILDER GRADES ITSELF

Chooses the architecture →

Chooses the test cases

One set of assumptions about what matters

The case they didn't anticipate while building is the one they don't anticipate while testing.

A THIRD PARTY GRADES

Builds the system →

Asks what “correct” means

Two independent sets of assumptions

They can't read the intent from the code, so they have to ask the business the question.

And the incentive points the wrong way. An evaluation concluding "this doesn't work" costs its author the next phase of the engagement.

None of this is specific to AI. We don't let a company certify its own accounts or audit its own security.

Ask who wrote the evaluation of your system. If it's whoever wrote the system, you already know what it concludes.

WHAT IT UNLOCKS

Six things you can't do without a measurement.

It gets treated as an end-of-project formality, when everything else rests on it.

WITHOUT AN EVALUATION

- The system answers
- Someone finds it good
- ?

The chain stops. Nothing to optimize against, compare with, or prove.

WITH AN EVALUATION

- The system answers
- The suite measures
- Failures name the cause
- You fix it

↺ and measure again

The loop closes, and an agent can turn it for you.

01 **Switch models when you decide to**

Providers deprecate models on their calendar, not yours. Without an evaluation, migrating is a leap in the dark. With one, you replay the suite on the candidate and decide on numbers.

02 **Pay less for the same quality**

With no measurement, everyone stays on the biggest model out of fear. With one, you try the small open model running on your own hardware and see what quality it costs. Often, nothing.

03

Automate improvement instead of arguing about it

An evaluation is a signal. Without it, nothing to optimize against. With it, an agent searches in your place: prompts, examples, distilling onto your domain. Improvement becomes a loop that runs.

04

Run research on your own problem

Launch twenty architecture variants, keep the one that wins. Decisions get resolved by experiments, not by whoever speaks with most confidence.

05

Ship without holding your breath

Every change clears the suite before production: a prompt, an index, a source, a model version. Regressions surface at your end, not at your customers'.

06

Answer "how do you know?"

In front of a board, a major client or a regulator, the question is always the same. A replayable evaluation turns a conviction into evidence. It is what the EU AI Act is starting to require.

WHAT IT LOOKS LIKE

Five properties of a good evaluation.

Drawn from reality

cases come from your records and real questions, failures included. A test set of easy cases measures your optimism.

Explicit about criteria

"correct answer" is written case by case: the right source, the right amount, a refusal when the information is missing. A criterion you can't write down you can't measure.

Reproducible

run tomorrow, it gives the same result. Versioned with the code, run automatically.

Able to fail

it has to be able to fail. An evaluation everything passes at 98% reassures you instead of teaching you something.

Yours

the dataset, the criteria and the harness stay with you, and get reused on every new model.

WHERE TO START

Two steps to take without us.

1 **Collect thirty failures**

Not thirty questions: thirty cases where your system answered badly, in real use. If nobody wrote them down, that is already a finding, and the first thing to fix.

2 **Write what the right answer would have been**

Case by case, with the source that proves it. It's tedious, it is where the work is, and it is what surfaces the internal disagreements about what "correct" even means.

Stop there. Those two steps cost nothing but attention and produce the one thing everybody lacks: **labelled examples** from your own business, where someone wrote down what "correct" meant.

That is the raw material for what follows. They are what lets you calibrate an automated judge and check that it agrees with you before anyone trusts it at scale. That is where we take over: covering the real variety of cases, validating the judge, industrializing the harness, and fixing the data your thirty failures implicated.