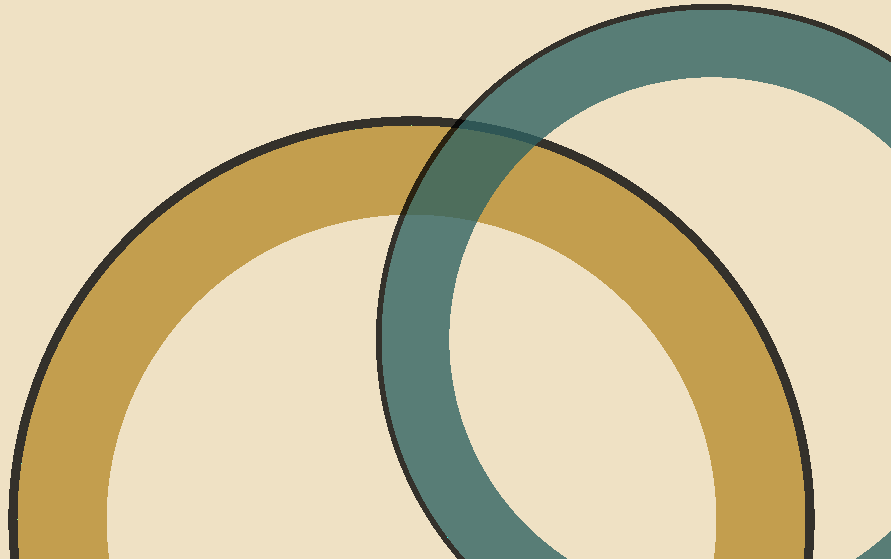
Engineering AI Agents · Week 8

Evaluating agents

Why the test habit gives way · an eval set you believe in · a judge you have calibrated · a gate that holds



Slides CC BY 4.0 · from the teaching kit of *AI Agents, Engineered* by Enrique Gutiérrez · aiagentsengineered.com/teach/

Your agent just handled a hard support ticket perfectly, in front of four colleagues. Someone asks: when can we roll this out?

Write down the evidence you hold about how it will behave tomorrow.

Chapter 16, “Why Evaluation Is Hard” · the chapter’s own exercise

By the end of today you can

Four objectives

1. Write an **eval set** whose every task a reference solution proves passable
2. Compute **pass@k** and **pass^k**, and put an **interval** on a pass rate
3. Build a **judge** and calibrate it against your own labels
4. Wire a **regression gate** over capability and regression suites

Part 1 · Chapter 16 (full book)

Why evaluation is hard

Inspection gives way

No single right answer

A plan, a summary, a reply: dozens of acceptable variants. An exact match fails good answers.

A judgment call

“Thorough”, “appropriate”: two competent reviewers can disagree, and the standard moves with context.

The ground shifts

A model update or a reworded prompt, and month-old tests fail with no code change.

Agents add: trajectories

One run is one path of many. And what does the grade attach to: the answer, or the route?

Chapter 16, “Why Evaluation Is Hard” · “tests over a model move on their own”

Predict before the next slide · one minute in pairs

An agent succeeds on 90 percent of attempts. Today it will act ten times, unattended.

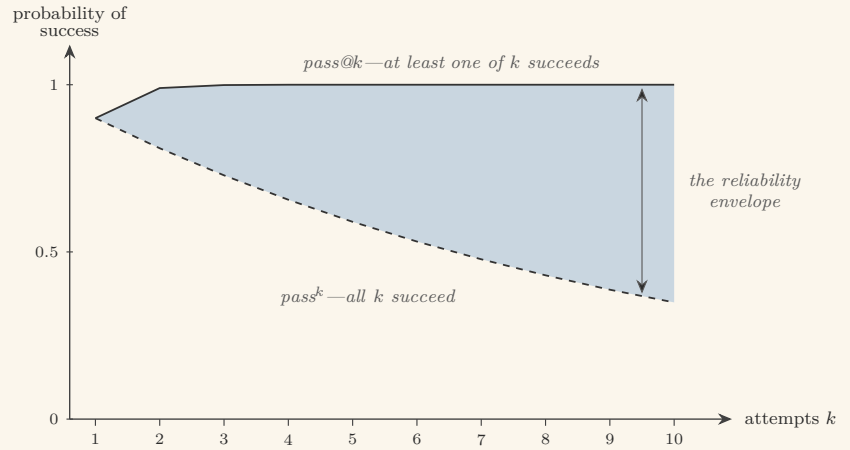
The chance that at least one action succeeds? That all ten do? Which of the two describes this product?

Chapter 16, “Why Evaluation Is Hard” · the chapter’s worked example · attempts treated as independent

One agent, two honest summaries of the same attempts

pass@k, pass^k and the envelope

- **pass@k**: at least one of k succeeds. Right when a cheap check picks the good one
- **pass^k**: all k succeed. Right when every action has to be right
- Equal at k = 1; they pull apart as k grows
- The gap is the **reliability envelope**



The widening gap is the reliability envelope, and a demo only ever shows the top curve.

‹ 4 / 4 ›

Figure from Chapter 16 · 90 percent per attempt, illustrative

Chapter 16, “Why Evaluation Is Hard” · “The demo tells you what the agent can do. Shipping asks what it always does.”

Part 2 · Chapter 16 (full book)

Building an eval set

Three runs, read by hand

- The same task, three times; read all three end to end
- Ask of each one only: **“would I accept this?”**
- Repeated trials, a human grader, a criterion, a reading of the spread

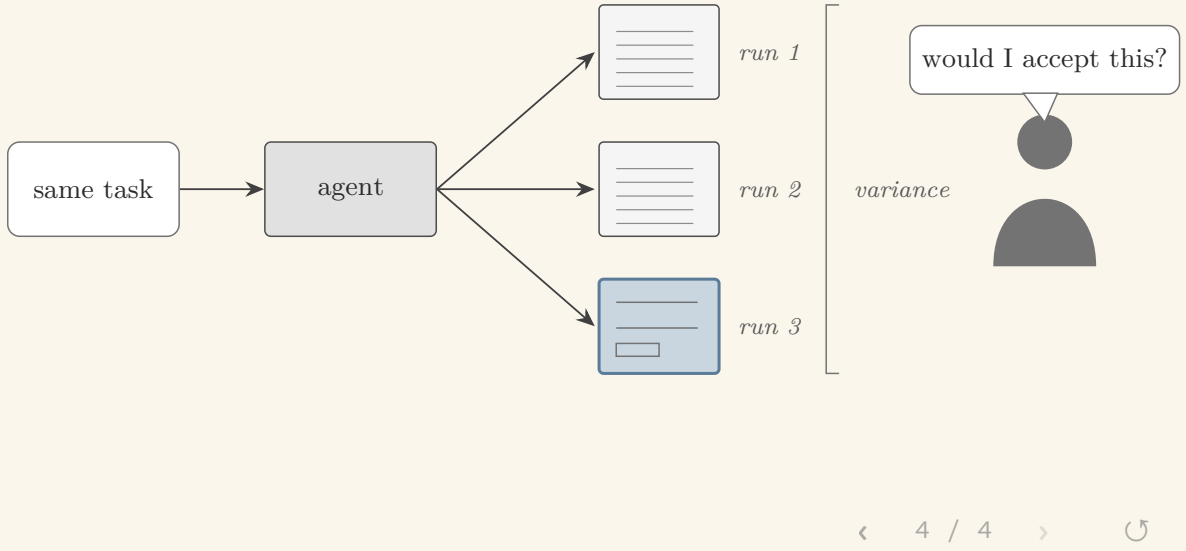


Figure from Chapter 16

Chapter 16, “Building an Eval Set” · the ritual follows one practitioner’s working method

Decide what “would I accept this?” means
before you measure anything

A criterion two experts agree on

Not testable

“Summarize the meeting well”

Testable

“Extract every action item, each under twenty words, with an owner named wherever the transcript names one”

- Two experts could disagree in good faith? **Sharpen the criterion, not the agent**
- Ambiguous cases go to **one arbiter**, not a committee

Chapter 16, “Building an Eval Set” · both criteria are the chapter’s

A curated collection of tasks, each an input plus a way to grade the output

The eval set

Each task	an input, and a reference answer, a checkable condition or a rubric
Passable	a reference solution proves it; a task nobody can pass measures nothing
Size	twenty to fifty hand-checked tasks
Source	your failures, from the traces: week 7's flywheel

Chapter 16, “Building an Eval Set” · “Twenty to fifty tasks you have hand-checked and believe in beat a few hundred synthetic ones nobody has read”

What to grade: the default the field has converged on

Grade the outcome, in the state

The outcome

- Query the calendar; run the generated code; read the row back
- “Done!” is a sentence, not evidence

The route, only for

- **Efficiency**: a three-step task that took fifty
- **Safety**: the destructive tool was never touched

Chapter 16, “Building an Eval Set” · “The transcript can claim success while the world is unchanged, and only one of them is your product.”

How to grade: climb from cheap to expensive, and stop as low as you can

The grader ladder

Rung	Use it when	For example
Program	an objective answer exists	it parses, the tests pass, nothing internal leaked
Binary rubric	quality splits into yes/no questions	names the customer? every claim sourced?
Judge	no code can decide	a research brief, a delicate reply

- Several small graders, one dimension each, tell you **what** broke
- **Negative cases**: tasks that assert what the agent must not do

Chapter 16 · the best judge, in the chapter's phrase, "is the one you did not need"

The one discipline to keep if you could keep only one

Read the failures

Fair failures

Pull up failing runs: is it clear what the agent got wrong? If not, suspect the grader.

Infrastructure apart

Timeouts, crashed sandboxes, rate limits: a separate status, never the quality number.

Designs, not scores

Borrow public benchmarks' designs and tasks. Their leaderboard is someone else's product.

Chapter 16, "Building an Eval Set" · "If a failing case leaves you confused about what was wanted, the case is the bug."

Predict before the next slide · commit to an answer

Version A passes 43 of 50 tasks. Version B, a prompt edit, passes 44 of 50.

Ship B because it is better? What would you want to know first?

A scenario for this check · counts invented, one run per task

A pass rate is an estimate; put an interval on it

How sure is a pass rate?

Version	Passed	Pass rate	95% interval
A	43 / 50	86.0%	73.8% to 93.0%
B	44 / 50	88.0%	76.2% to 94.4%

- The gap could easily be noise: two-sided p = 0.77; detecting it would take about 4,438 tasks per version
- Both versions ran the **same** 50 tasks: compare per task, which ones B won and which it lost; and run each task several times

Chapter 16 · “a two-point shift on a fifty-task suite is well inside the noise” · intervals and test: standard statistics from the course’s eval sample-size calculator, not formulas from the book

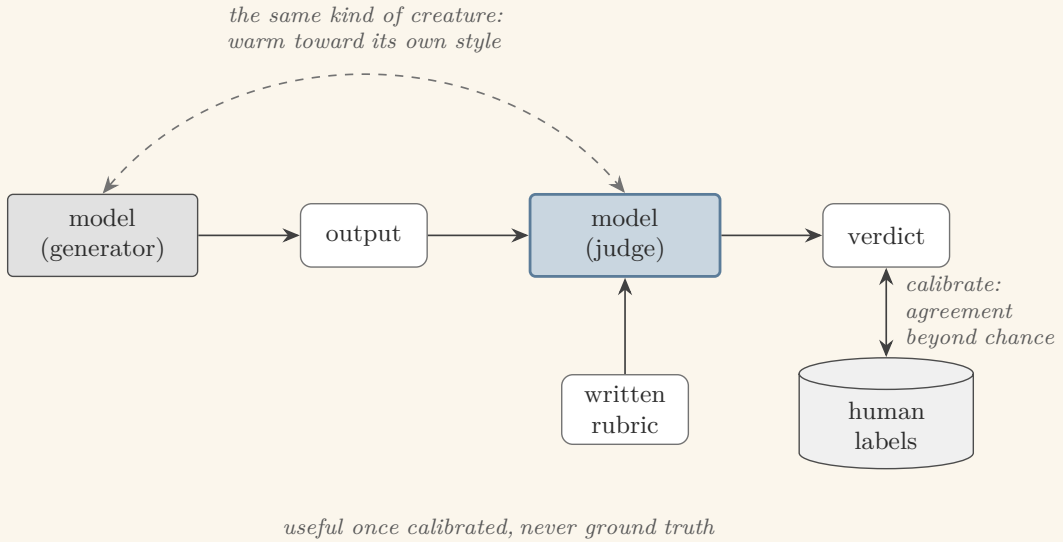
Part 3 · Chapter 16 (full book)

LLM-as-a-judge

A language model with a written rubric,
scoring outputs as a reviewer would

A model grading a model

- Thousands of outputs in minutes; never tired at four in the afternoon
- One study: agreement with humans **over 80 percent**, about what two humans manage
- An **instrument**: useful once calibrated, biased in known directions



Calibrated against human labels, the judge becomes a useful instrument, never ground truth.

‹ 4 / 4 ›

Figure from Chapter 16

Chapter 16, “LLM-as-a-Judge” · the agreement figure is one study’s, cited in the chapter

Two framings, four habits

Framing	Good for
Pointwise: one output, pass/fail or a score	an absolute signal, dashboards
Pairwise: A or B?	head-to-head decisions; steadier, but swayed by surface features

1. A rubric for a new hire: criteria, grade meanings, a pass and a fail example, binary sub-questions
2. Reasoning first, verdict last
3. The reference answer, whenever one exists
4. A structured verdict your code can parse

Chapter 16, “LLM-as-a-Judge” · “a verdict stated first and justified after is a snap judgment wearing a lab coat”

All four will be waiting in your first judge

Four ways a judge goes wrong

Bias	What it does	What you do
Position	prefers an answer for its slot	run both orders; keep verdicts that survive the swap; the rest are ties
Verbosity	longer reads as better	say length is not quality; check length against score
Self-preference	warms to its own style, and its family's	a different model judges when the stakes are real
Leniency	everything is a four or a five	reserve the top grade for flawless; prefer pairwise

Chapter 16, "LLM-as-a-Judge" · in the study the chapter cites, some judges favored the first answer in about three of four comparisons

Predict before the next slide · one minute in pairs

Your judge agrees with your own labels on 42 of 50 outputs: 84 percent. You labeled 40 of the 50 good.

Is it a good judge? What would a judge that says “good” every time score?

A scenario for this check · counts invented

Before a judge grades anything unsupervised, calibrate it against ground truth: you

Calibrate it like an instrument

- Hand-label a few dozen real outputs, some of them truly bad
- **Cohen's kappa**: agreement beyond what chance gives
- **Precision and recall**: missed bad outputs, failed good ones
- The **level**: harsher or kinder than you
- Spot-check on a schedule; recalibrate when the judge's model changes

The chapter's example

70 percent of outputs are good. A judge answers "good" every time: accuracy 70 percent, kappa zero.

Chapter 16, "LLM-as-a-Judge" · "A judge you have not calibrated is an opinion you have automated." · human graders on such tasks: kappa around 0.7 to 0.8, a practitioner yardstick

Part 4 · Chapter 16 (full book)

Eval-driven development and regression gates

Treat the suite, like a test suite, as the definition of progress

Every change reruns the suite

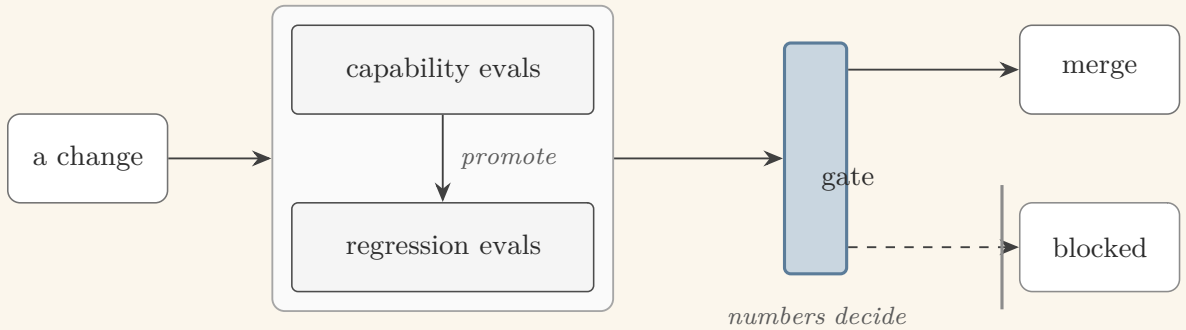
`a change` → `the suite, several runs per task` → `the numbers decide`

- The numbers, plural: pass rate, **cost**, **latency**, **steps**
- Prompts and tool definitions **versioned** with the code
- A model update becomes a decision, not a surprise users report
- Compare distributions, never single runs

Chapter 16, “Eval-Driven Development and Regression Gates” · “every change to a prompt, a tool definition, or the model reruns the suite, and the numbers decide”

Two suites and a gate

- **Capability**: the hill; mostly failing, on purpose
- **Regression**: ground taken; near 100 percent, loud on any slip
- A capability task that stabilizes is **promoted**
- **Saturation**: everything green, always, and no signal



The numbers, not moods, decide whether the change merges or is blocked.

Figure from Chapter 16

‹ 5 / 5 ›

Chapter 16 · “a capability suite passing at, say, eighty percent is no longer pulling you anywhere”

Two layers, one circuit, and a threshold that is enforced

Offline, online, and the gate

Layer	Answers
Offline: a fixed set, in CI, before shipping	did this change help?
Online: feedback, retries, abandonment, a sampled judge, read traces	what is happening out there?

- A gate: below the floor, no merge; **any safety failure** stops it
- **Fail closed**: uncertainty blocks
- A smoke subset per commit; the full battery nightly and before release

Chapter 16 · “A user who immediately rephrases the same request has graded the first answer, for free.”

In class · teams of two or three · 25 minutes · three browser tools

Measure the measurement

1. **A · pass@k (6 min).** The pass@k calculator, 22 of 25 runs: pass@k, pass^k and the envelope at k = 3 and 10; which number two products quote
2. **B · a pass rate (7 min).** The eval sample-size calculator: the interval for 24 of 30; a jump from 38 to 45 of 50; the tasks for ±5 points
3. **C · a judge (9 min).** The judge agreement calculator on 60 labels, a swap sample and a length sample: may this judge gate a suite?

Every link on the lecture page opens prefilled. Debrief, 3 minutes

Tools after Chapter 16 · intervals, tests and kappa are standard statistics computed by the tools · samples invented for the exercise

Lab 7, and reading

Lab 7 · the eval set v1

- 20–50 tasks, each **proven passable**; graders on the lowest rung; negative cases
- Three runs or more per task; **infrastructure apart**
- Pass rates with **intervals**; a judge calibrated on 30+ labels (the course's floor)
- A **gate** that fails closed

Reading for week 9

- Ch. 17, Security, Safety, and Guardrails
  `full book`

This week's reading: Ch. 16 (full book) · glossary: App. B (free) · no paid API needed

Five things to keep

- Quote the bound your product lives on: **pass@k** or **pass^k**
- Every task **passable**, graded on the **state**, on the lowest rung
- Put an **interval** on every pass rate; small gaps are noise
- A judge is an **instrument**: calibrate it with kappa against your labels
- Every change reruns the suite; the **gate fails closed**

Engineering AI Agents · Week 8

Intuition proposes; measurement decides.

Slides from the teaching kit of *AI Agents, Engineered* by Enrique Gutiérrez, CC BY 4.0 · aiagentsengineered.com/teach/

Reuse, adapt and translate with attribution · creativecommons.org/licenses/by/4.0/

