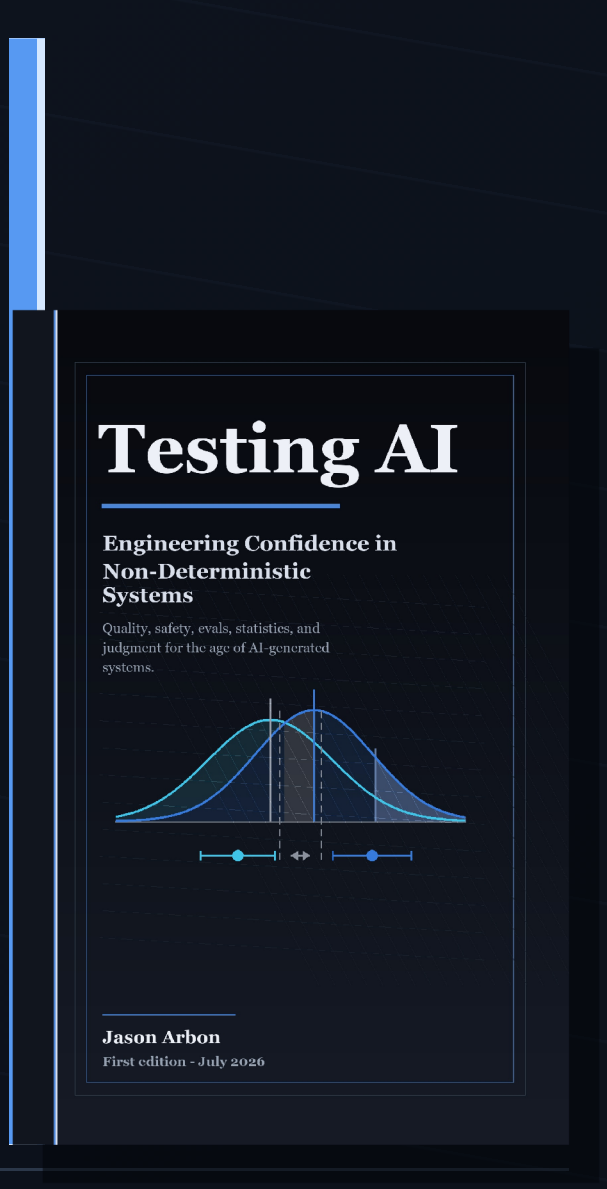
THE FIRST MENTAL SHIFT

Stop testing one answer.

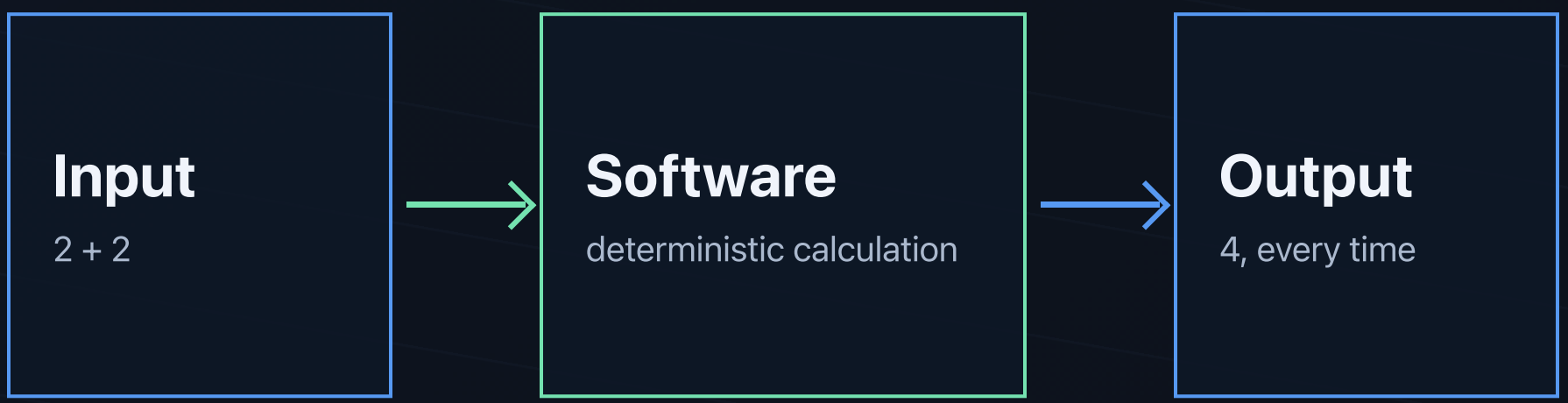
AI quality is not whether one run looked good. It is whether the behavior is good enough, often enough, in the real world.

Swipe for the confidence-engineering version of testing.



THE OLD CONTRACT

Classic software gives you a stable target.



> Same input, same conditions, same result.

> Exact assertions are still excellent for hard rules.

Keep this discipline wherever the product can be deterministic.

WHAT CHANGES WITH AI

The same prompt can produce a range of behavior.

Input: “Summarize this report for the board.”

- “Revenue rose 11%, driven by enterprise renewals.”
- “The report shows 11% growth, led by enterprise renewals.”
- “Revenue rose 11%.” A key fraud-risk caveat is missing.

Different wording may be fine. Missing the decision-changing risk is not.

EVIDENCE OVER ANECDOTES

One good run is an observation.

It can demonstrate possibility. It cannot estimate reliability.

ONE RUN

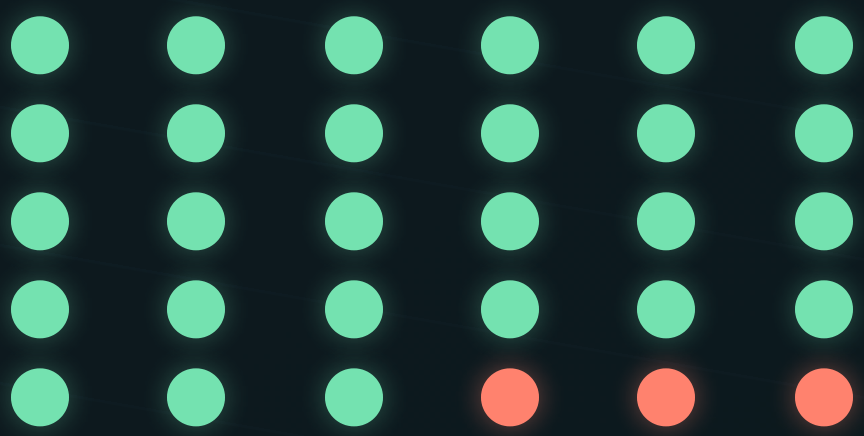
N = 1

One good outcome.
Nothing more.

SAMPLE

REPRESENTATIVE SAMPLE

N = 30



3 high-risk failures.
Now you can see a pattern and investigate the slice.

Most runs can look fine while the tail still matters.

TEST PROPERTIES, NOT PROSE

Move from strings to criteria.

Brittle check

"The answer must equal this exact sentence."

Useful criterion

States the 30-day policy, does not invent an exception, and gives a clear next step.

Weak result

Different wording, automatic fail.

Meaningful result

Policy is correct, safe, clear, and actionable.

Hard boundaries stay exact. Flexible quality needs a rubric.

GOOD IS NOT ONE NUMBER

Quality is a set of tradeoffs.

Correctness
Did it do the job?

Safety
Did it avoid unacceptable harm?

Groundedness
Can the answer be supported by evidence?

Usability
Can a real person act on it?

Reliability
Does it work when production gets messy?

Cost and latency
Can the product sustain the behavior?

The right measurement is the one that changes a responsible decision.

BEFORE THE EVAL

Ask whether this should be AI at all.

Use deterministic software

When the product needs exactness, auditability, fixed logic, low latency, or clear accountability.

OR

Use AI deliberately

When flexible language, perception, retrieval, personalization, or fuzzy judgment creates real user value.

The first confidence-engineering decision is not model choice. It is use-case choice.

A PRACTICAL SEQUENCE

First stabilize. Then restore reality.

1

Get a baseline

Freeze seeds, configurations, retrieval snapshots, and tool responses to find ordinary product bugs.

2

Measure variability

Turn production conditions back on: live data, real timing, changing user state, and model variation.

> Failures in both passes are likely ordinary bugs. Failures only in production-like runs tell you where variability changes quality.

Debug deterministically. Validate under realistic variation.

THE RELEASE QUESTION

Every eval must lead to a decision.

Is this behavior safe and useful enough for this release?

Not yet

Fix a blocking failure, add a constraint, gather more evidence, or reduce exposure.

Maybe, with controls

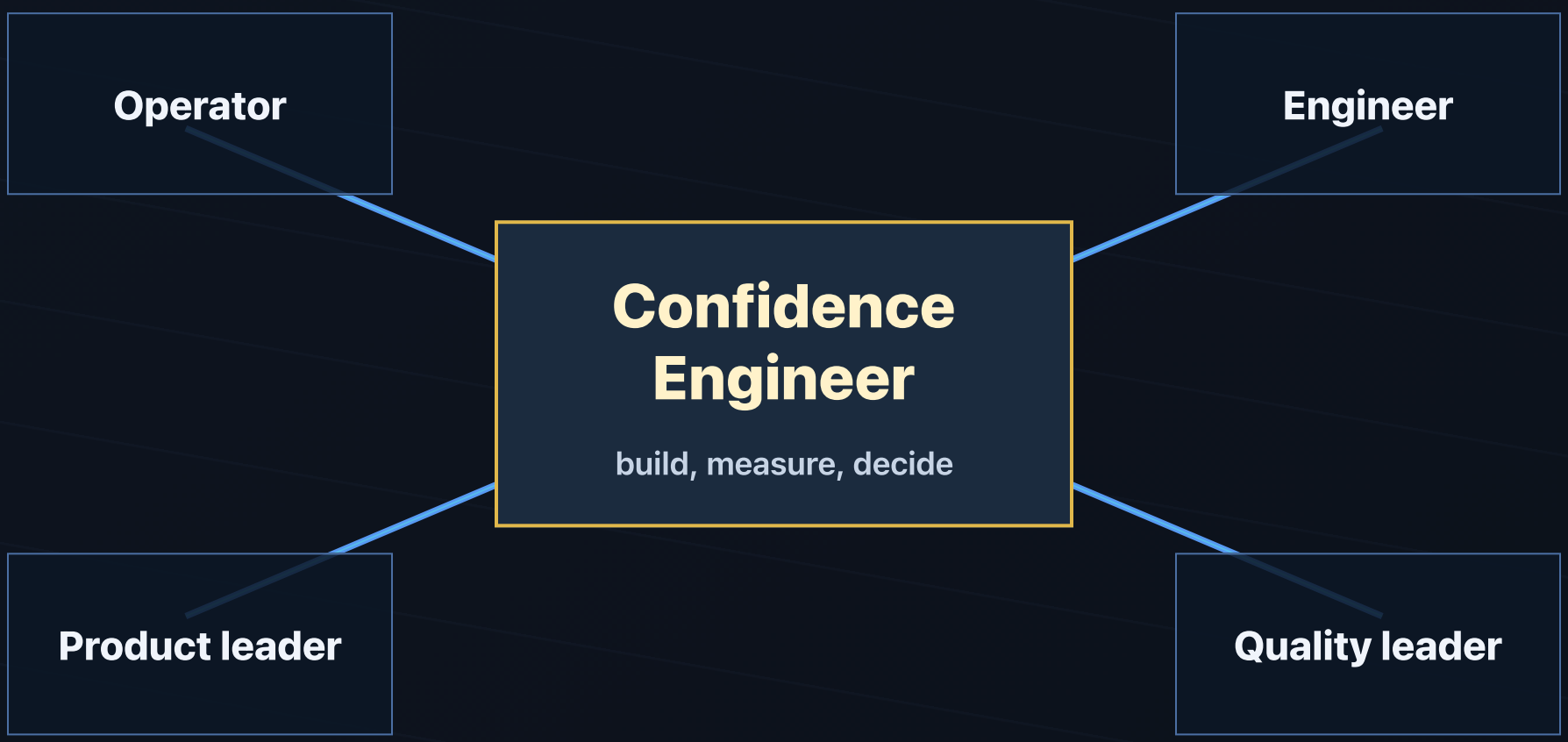
Canary to a risky slice, monitor it, and keep rollback ready.

What will we watch after it reaches real users?

A metric without a decision boundary is dashboard decoration.

THE NEW ROLE

Software work is becoming confidence engineering.



The job is not only to make the thing. It is to know when it deserves to ship.

Testing AI

READ THE BOOK

THE PRACTICAL OPERATING MANUAL

Testing is the doorway. Confidence is the destination.

Testing AI is for developers, quality leaders, product builders, and operators who need evidence that an AI system deserves to reach production.

Get Testing AI on Amazon

Chapter 1: The next-generation AI builder will measure uncertainty.