Automate, Monitor, Escalate

Architecting Decision Systems at Scale

Philipp Eisenhauer

Philipp Eisenhauer

peisenha.github.io

I build production systems that translate scientific models into decision frameworks — so organizations can act on their evidence at scale.

THE QUESTION

How do you architect decision systems that compound impact at scale — automating execution, monitoring for failures, and escalating to human judgment if needed?

Decision systems are everywhere

- **Credit card fraud** — approve or decline transactions in milliseconds, monitor fraud rate and model drift, escalate flagged charges to a human analyst
- **Content moderation** — remove clear violations automatically, monitor false-positive rate, escalate borderline posts to a human reviewer
- **Online Catalog** — generate product page hypotheses automatically, monitor quality gates and A/B results, escalate failures to human auditors

What you learned

You can answer one question precisely:

Did this intervention cause that outcome?

- Randomized experiments
- Difference-in-differences
- Matching, synthetic control

That question is hard. The methods are rigorous.

But there is a problem.

Measurement is just the beginning

Operating Modes

Each stage of the pipeline — Measure · Evaluate · Allocate — runs in one of three modes.

Automate

system performs reliably
decisions run without review

Monitor

track whether performance holds
detect drift before it compounds

Escalate

performance has deteriorated
human judgment repairs the rule

The Tension

- **Automate** as much as possible — every decision that runs without human review is a gain
- **Monitor** everything that runs without you — every drift that goes undetected becomes a compounding failure
- **Escalate** when performance breaks — every rule a human repairs returns stronger than before

Automate · Monitor · Escalate

Three vantage points. Same pattern.

- **for a program** — Catalog AI
- **for a portfolio** — Impact Engine
- **as a career** — Decision System Architect

AUTOMATE · MONITOR · ESCALATE FOR A PROGRAM

Catalog AI

The Challenge

Hundreds of millions of product pages. Millions added or edited daily.

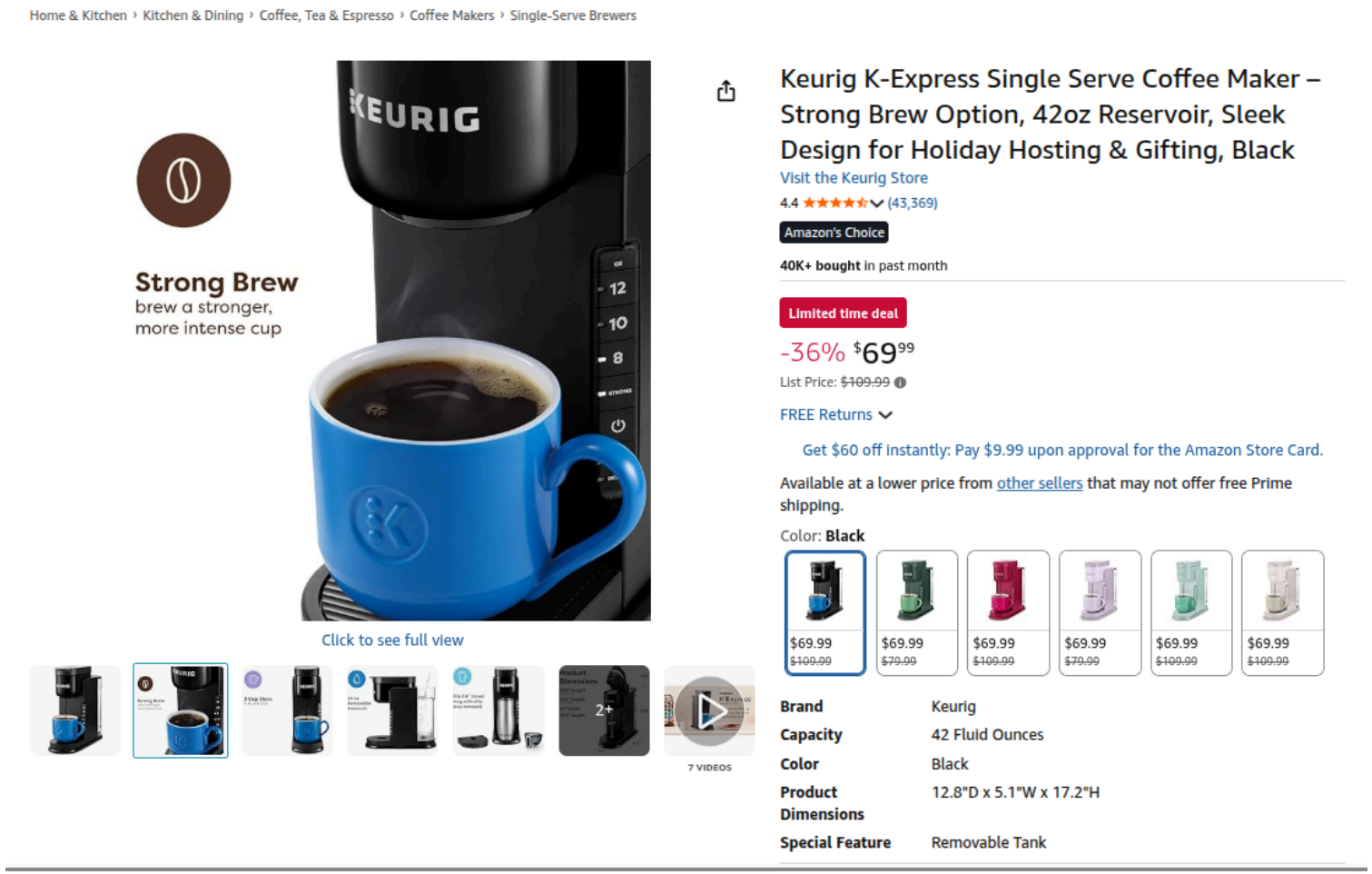
A human team reviews thousands. The catalog needs millions.

How do you close that gap — without sacrificing quality?

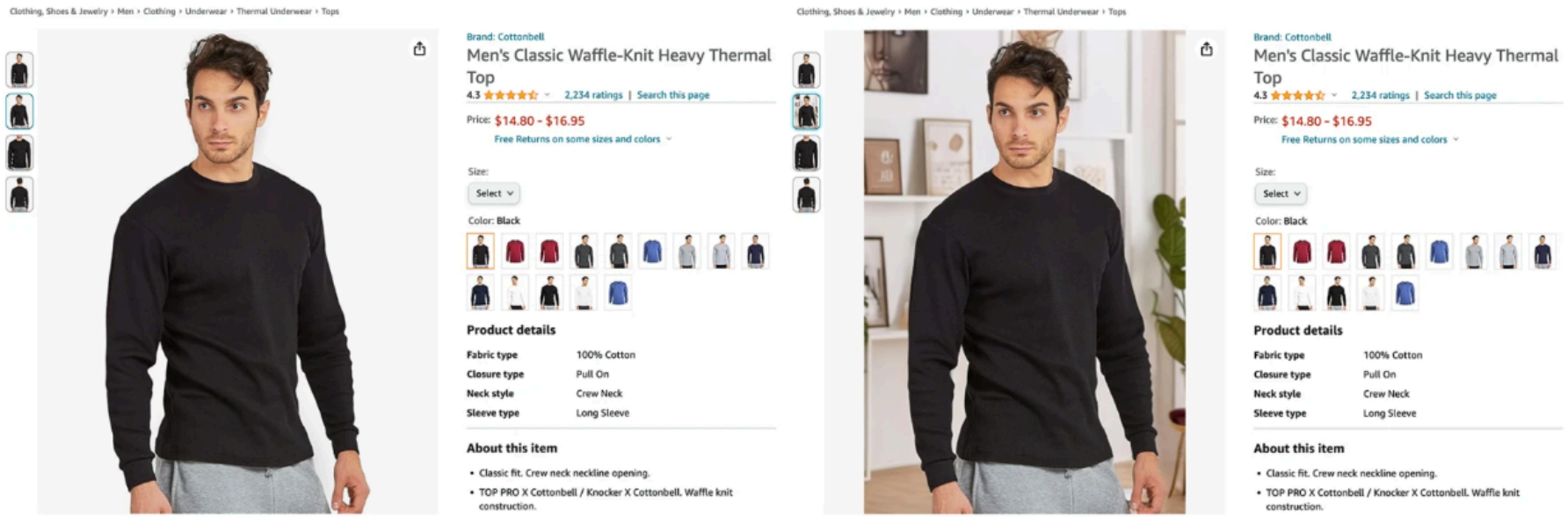
Catalog AI

We built a generative AI system that creates, tests, and improves product pages — end to end.

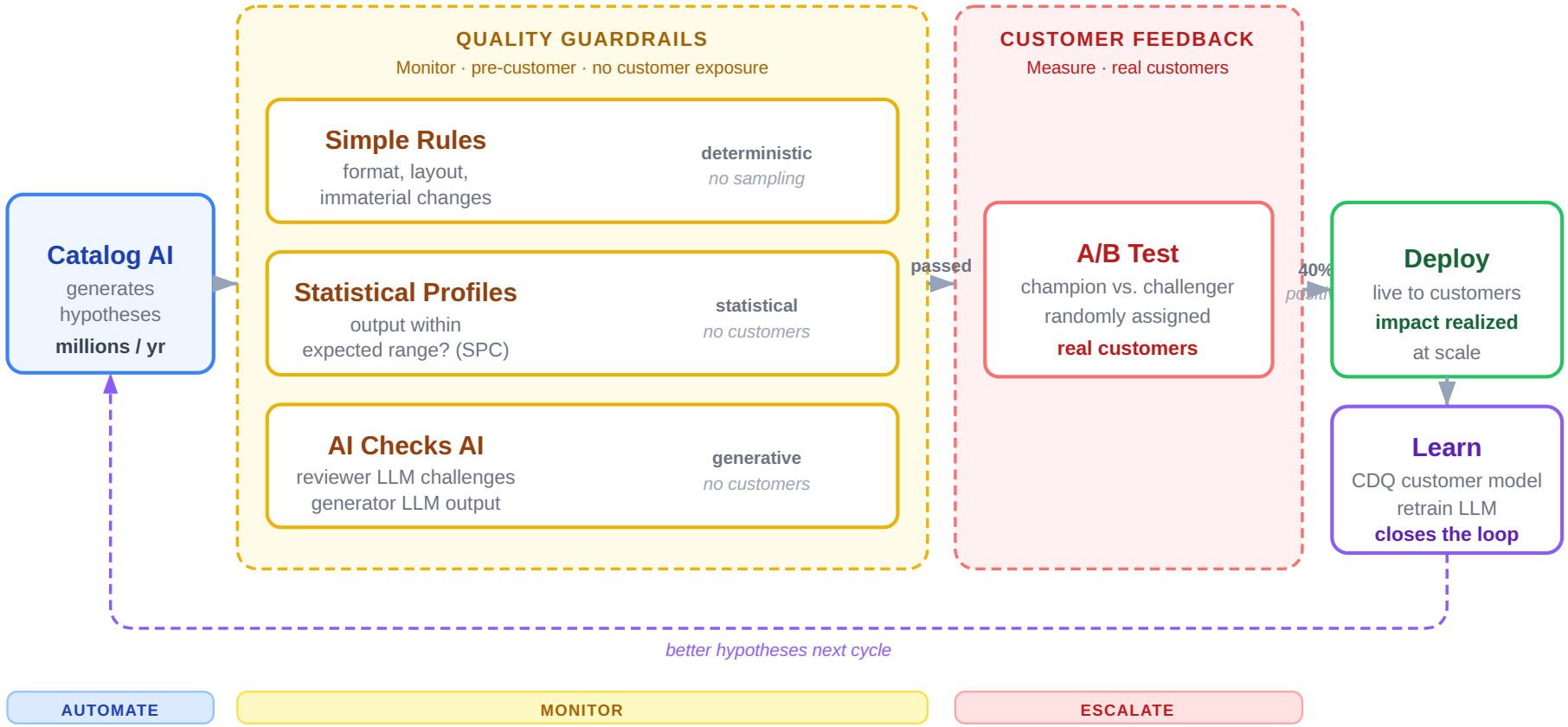
Millions of product pages



Concept Test



The system in practice



The result

Humans generating thousands of hypotheses per year.

Catalog AI generating tens of millions.

80% unreliable at launch. Monitoring earned the right to automate.

40% of content passing reliability checks improves sales or is neutral. 60% filtered before it reaches a customer.

Numbers at time of publication — HBR, Sept–Oct 2025

We closed the loop. Our system learns. Success compounds.

AUTOMATE · MONITOR · ESCALATE FOR A PORTFOLIO

Impact Engine

The Challenge

Dozens of initiatives. Each one generating its own evidence.

One team runs a clean RCT. Another runs a pre-post with a seasonal confound. Most pipelines treat them the same.

How do you allocate budget when you can't trust the estimates equally?

Impact Engine

An open-source pipeline that measures impact, scores evidence, and allocates budgets — end to end.

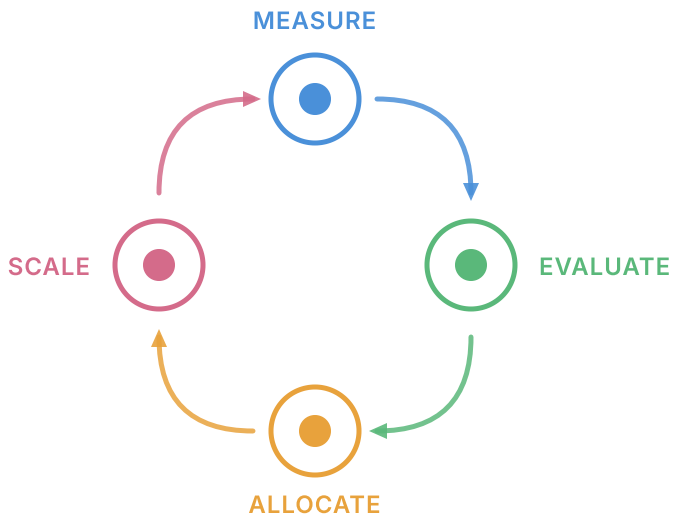
Three initiatives, one allocation

PORTFOLIO ALLOCATION · ONE CYCLE

Three initiatives. Same dollars to allocate. Evidence quality decides who gets funded.

INITIATIVE	METHOD	EST. IMPACT	CONFIDENCE	WEIGHTED IMPACT	DECISION
A · Pricing optim. *flagship checkout experiment*	Randomized A/B test `n = 240k · 6 weeks · clean diagnostics`	**$2.4M** annual	**0.85** high	**$2.04M** 2.4 × 0.85	FUND
B · Reco overhaul *large estimated lift, weak evidence*	Pre/post observational `no control group · seasonal confound`	**$1.8M** annual	**0.30** low	**$0.54M** 1.8 × 0.30	DEFER
C · Search ranking *smaller lift, well measured*	Diff-in-diff, matched controls `parallel trends hold · placebo passes`	**$1.1M** annual	**0.75** high	**$0.83M** 1.1 × 0.75	FUND

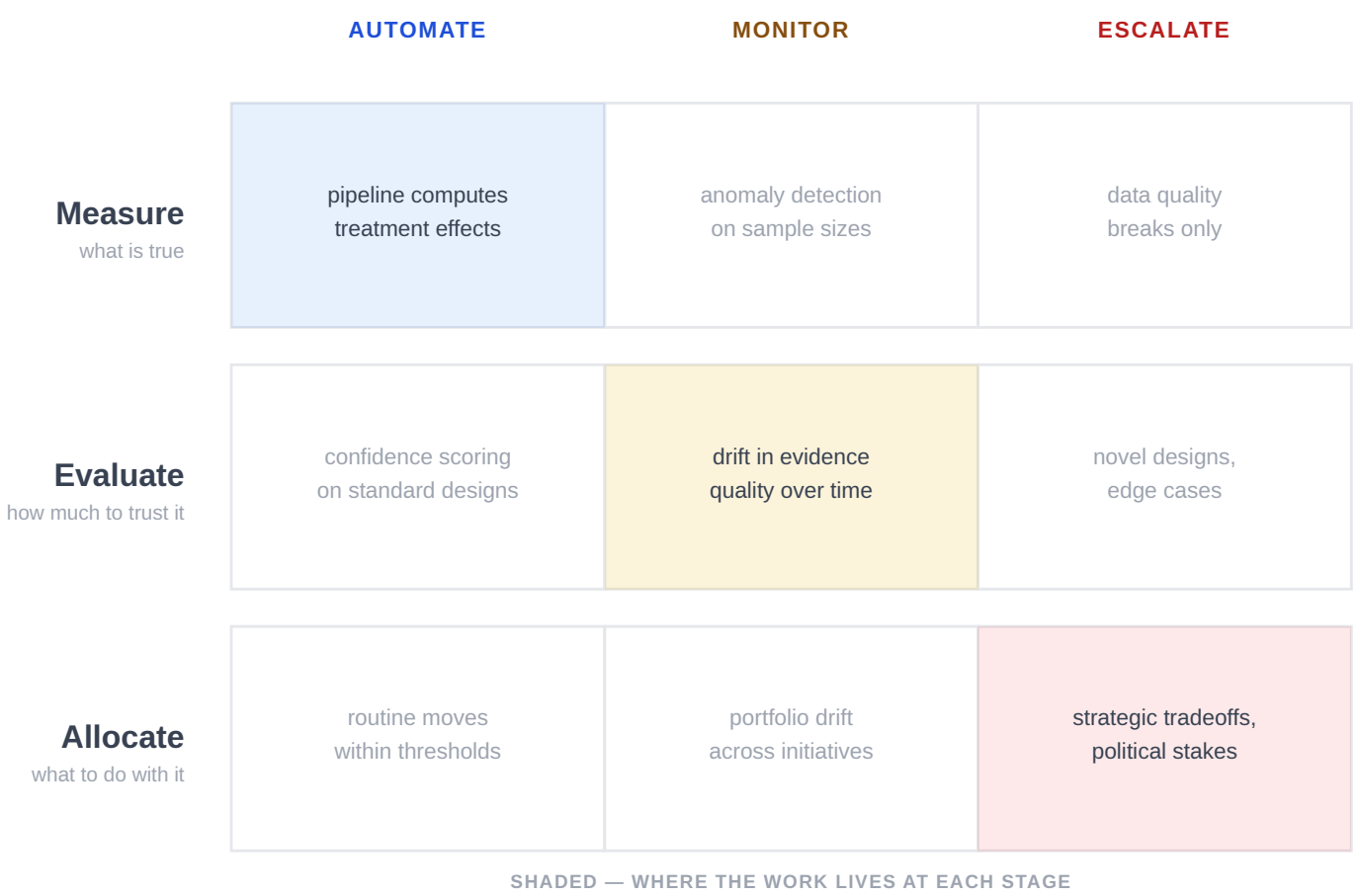
Impact Engine



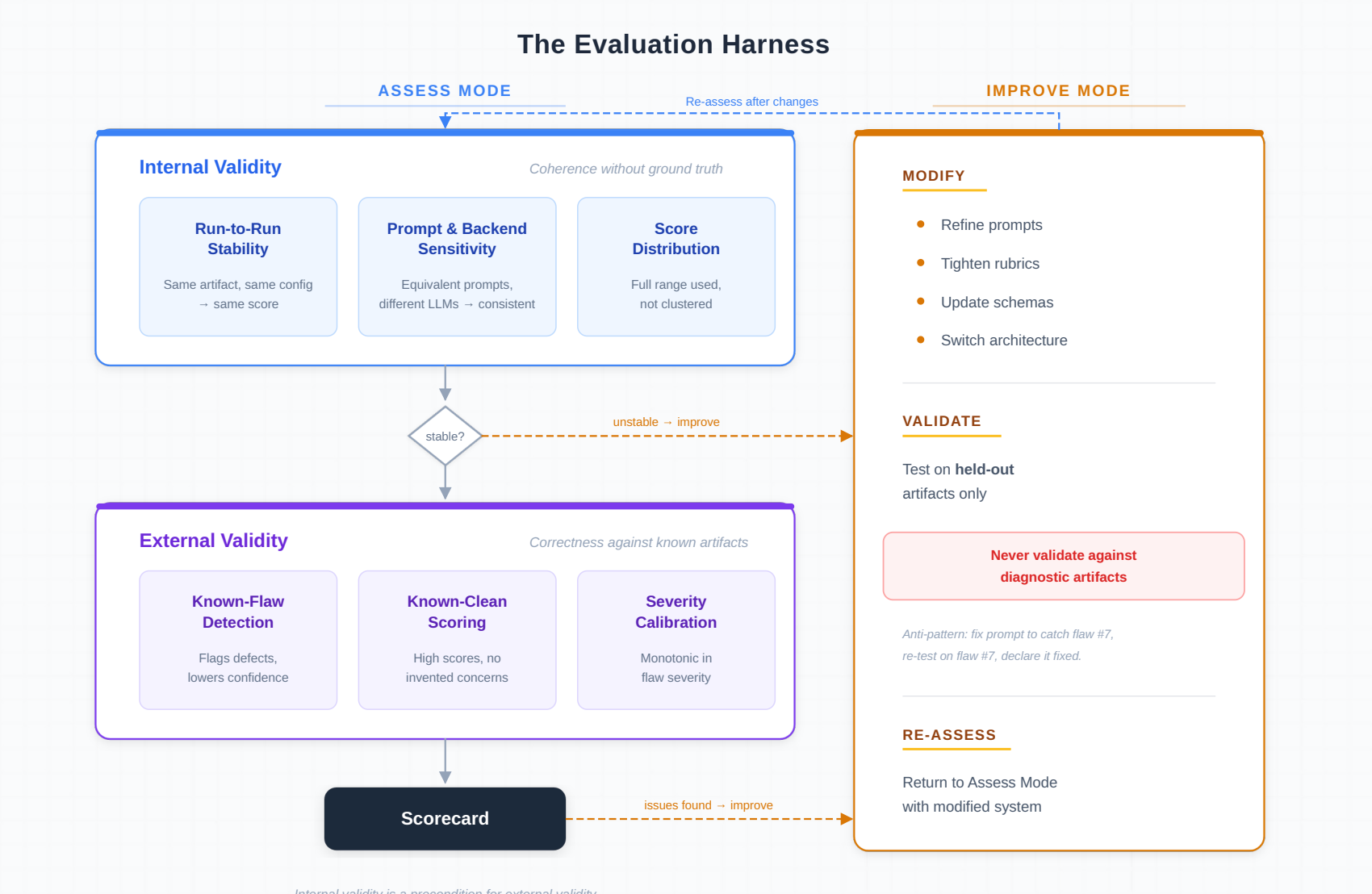
Start with a cohort of candidate initiatives and run small-scale pilots. **Measure** the impact of each pilot. **Evaluate** how much to trust each estimate based on methodological rigor and remaining uncertainty. **Allocate** budget to the most promising initiatives through constrained portfolio optimization. **Scale** the winners, monitor their performance, and feed the learnings back into the next cycle.

The loop only closes when all stages are connected.

Pipeline operating modes



Where science is most interesting



The value is in the system

The loop closes only when all stages remain connected and automated — when measurement feeds evaluation, evaluation feeds allocation, and allocation feeds the next measurement cycle without a human carrying the handoff.

A team doing this work creates a dependency. A system doing this work creates an asset.

AUTOMATE · MONITOR · ESCALATE AS A CAREER

Decision System Architect

The Opportunity

These systems don't build themselves. They require three disciplines that rarely develop in the same person — Causal Inference, Decision Theory, Software Engineering.

Most scientists develop one. Some develop two. Very few develop all three.

If you're working on it, you're early.

PIPELINE STAGE	SCIENCE DOMAIN
Measure what is true	**Causal Inference** estimating treatment effects from data
Evaluate how much to trust it	**Causal Inference** evaluating assumptions and robustness
Allocate what to do with it	**Decision Theory** optimization under uncertainty
Build how to make it repeatable	**Software Engineering** systems that run without human handoff