Data Science Anti-Patterns and Better Alternatives

- Lifecycle-wide lens: plausible practices that reliably produce poor outcomes — and what replaces them
- For data scientists, analysts, ML teams, educators, and technical managers
- Anti-patterns are diagnosable: symptom → root cause → alternative → measurement
- Roadmap: framing, data, experimentation, modeling, deployment, tooling, governance, playbook
- Goal: recognize the pattern, choose a proportionate fix, make change measurable

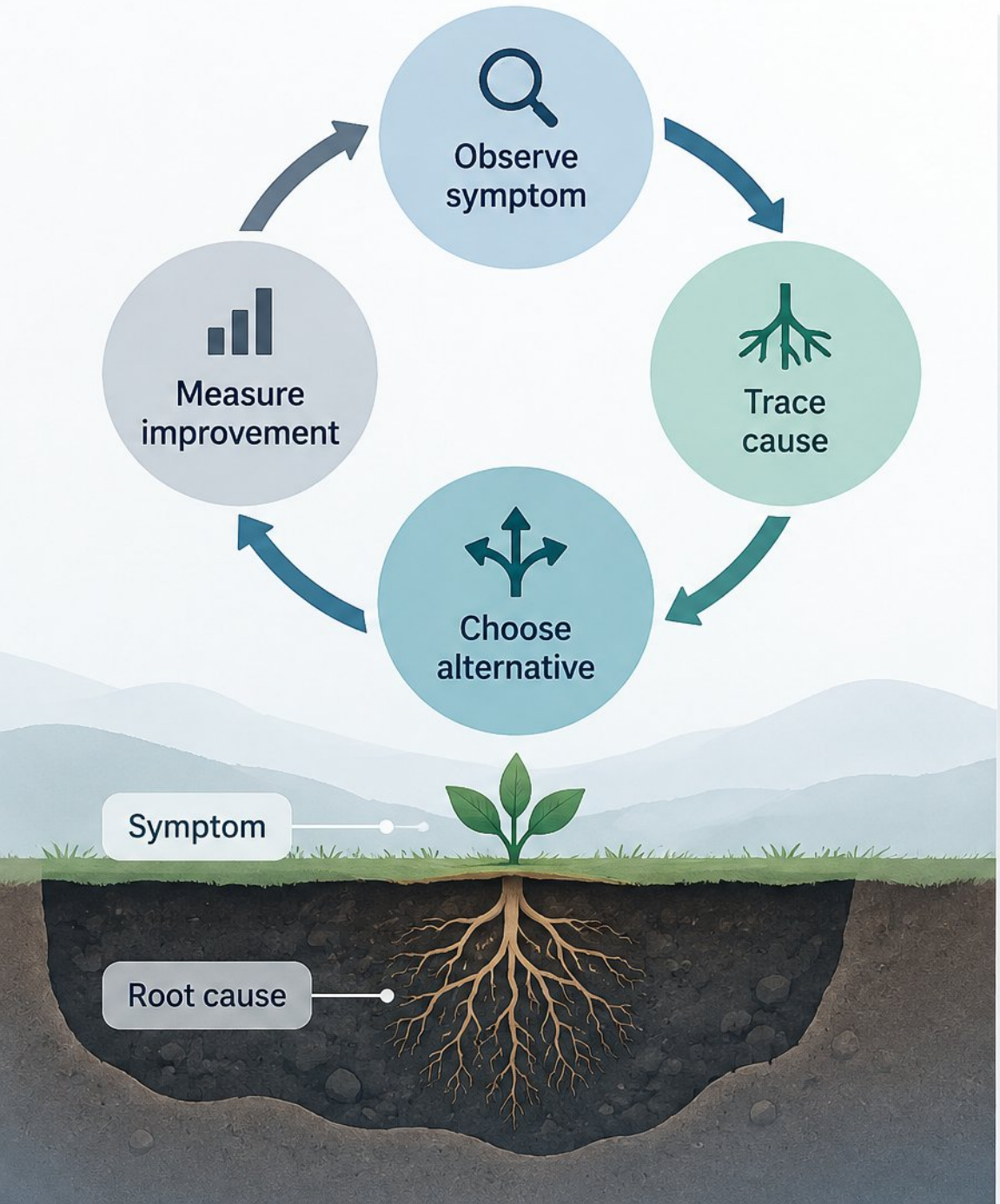


What Counts as an Anti-Pattern, and Why They Persist

- Anti-patterns: defective but plausible practices that recur across teams
- The term gives teams a shared, fast vocabulary for diagnosing weak practice
- Both technical errors and missing context on how ML results are used
- Why they persist: metric pressure, unclear decision ownership, tooling defaults
- Interviews with 65 practitioners: top failure cause is misunderstood problems
- Practitioners get a diagnostic habit; managers a review vocabulary



The Diagnosis Loop: From Symptom to Root Cause to Alternative



- Four-step loop: observe symptom → trace cause → choose alternative → measure improvement
- Common symptoms: brittle pipelines, non-reproducible analyses, models degrading after launch, unused dashboards, gains lost before business metrics
- Check these causes first: incentive misalignment, unclear decision ownership, weak data contracts, premature optimization
- In 96 production ML postmortems, most outages were distributed-systems, deployment, and data-join failures — not ML
- Deliberate tradeoff = known cost, an owner, an expiry date; unexamined default = anti-pattern

Problem Framing and Metric Anti-Patterns

Start with the right question, measure what matters.

Data-first start (anti-pattern)



Decision-first start (better)



- Anti-pattern: starting from available data or models, not the decision
- Anti-pattern: optimizing a proxy metric weakly tied to real outcomes
- Anti-pattern: vague success criteria and moving goalposts
- Better: decision-first framing and separate business from model metrics
- Proxies are provisional evidence — document assumptions and add guardrails

Why Proxy Metrics Betray Us: Goodhart in Practice

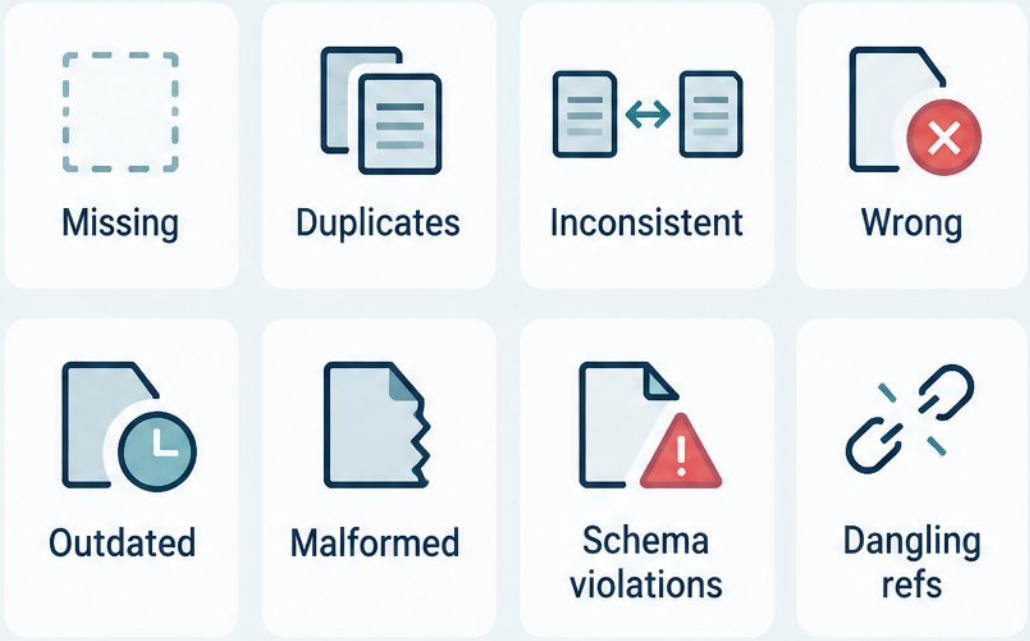
- Once a proxy becomes the target, experiments get selected for the proxy, not the product outcome
- Three failure shapes: incomplete coverage, component-level metrics, model-as-judge rewarding verbosity
- Correlation traps: user-level correlation and naive treatment-effect correlation both mislead
- Proxy assumptions — unconfoundedness, exclusivity, comparability — are almost certainly violated; ask how badly
- Better alternatives: metric-to-value mapping, written success contract, post-deployment comparison, guardrail metrics



Data Collection and Preparation Anti-Patterns

- Raw data treated as ground truth: provenance gaps breed costly wrong conclusions
- Silent schema drift, undocumented joins, and quality issues sink projects early
- Name and triage issues: missing, duplicates, inconsistent, wrong, outdated, malformed, schema violations, dangling refs
- Big data paradox: volume never guarantees validity; capture metadata is a prerequisite
- Better: data contracts, provenance tracking, automated tests on ingest and transform, close producer–user collaboration

Common Data-Quality Faults



Know the issues. Build better data.

Data Leakage: The Silent Inflation of Model Quality

- **Leakage:** prediction-time-unavailable information shapes training — validation looks great, production fails
- **Top route:** scaling, imputation, feature selection, or augmentation fitted before the split
- **Check the family explicilty:** similarity, group, and temporal leakage across splits
- **Better alternatives:** split first, never fit on test data, pipeline preprocessing, use grouped and time-aware splits
- **Guardrails:** automated leakage checks, provenance-based screening, documented preprocessing contract per model



EDA and Experimentation Anti-Patterns

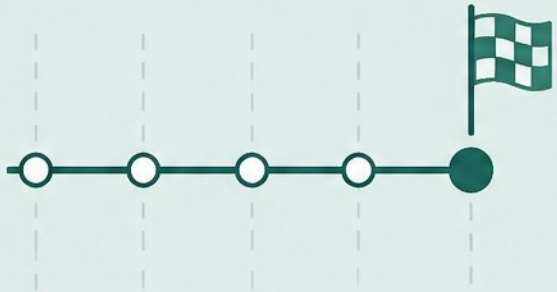
BETTER QUESTIONS. MORE RELIABLE INSIGHTS.

- Hypothesis fishing: report only attractive findings, forget how many tests ran
- Peeking: stopping when p clears 0.05 can inflate false positives 5–10x
- Field evidence: ~75% of effects in p-hacked experiments were truly null
- Underpowered tests and observational correlation mistaken for causal effect
- Fix: pre-register hypotheses, pre-agree sequential stopping rules, run power analysis
- Keep operational checks separate from success monitoring

Peek Repeatedly and Stop Early



Run to a Pre-Agreed Horizon

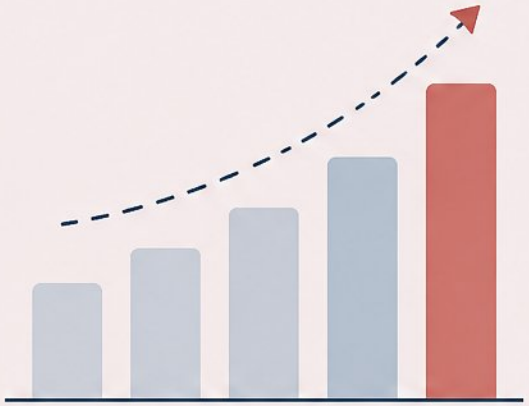


CURIOSITY DISCIPLINE BETTER DECISIONS

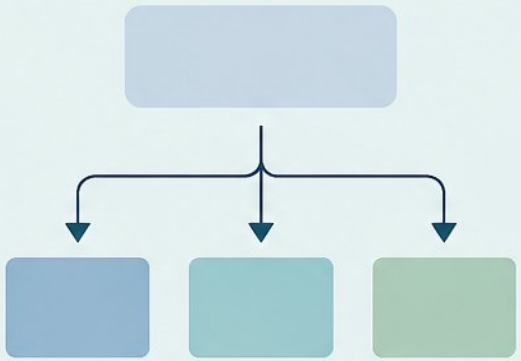
Modeling and Evaluation Anti-Patterns

- Leaderboard chasing without a simple baseline hides weak models
- Accuracy misleads under class imbalance; use MCC or PR AUC
- Repeated test-set use makes tuning decisions leak into evaluation
- Baseline first, three-way splits, slice-based reporting
- Check calibration curves and proper scoring rules for overconfidence

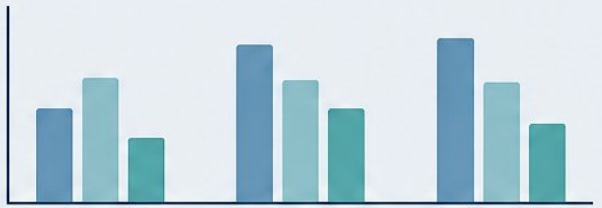
Anti-Pattern



Better Approach



Performance Across Slices



Calibration Curve



Deployment and Monitoring Anti-Patterns

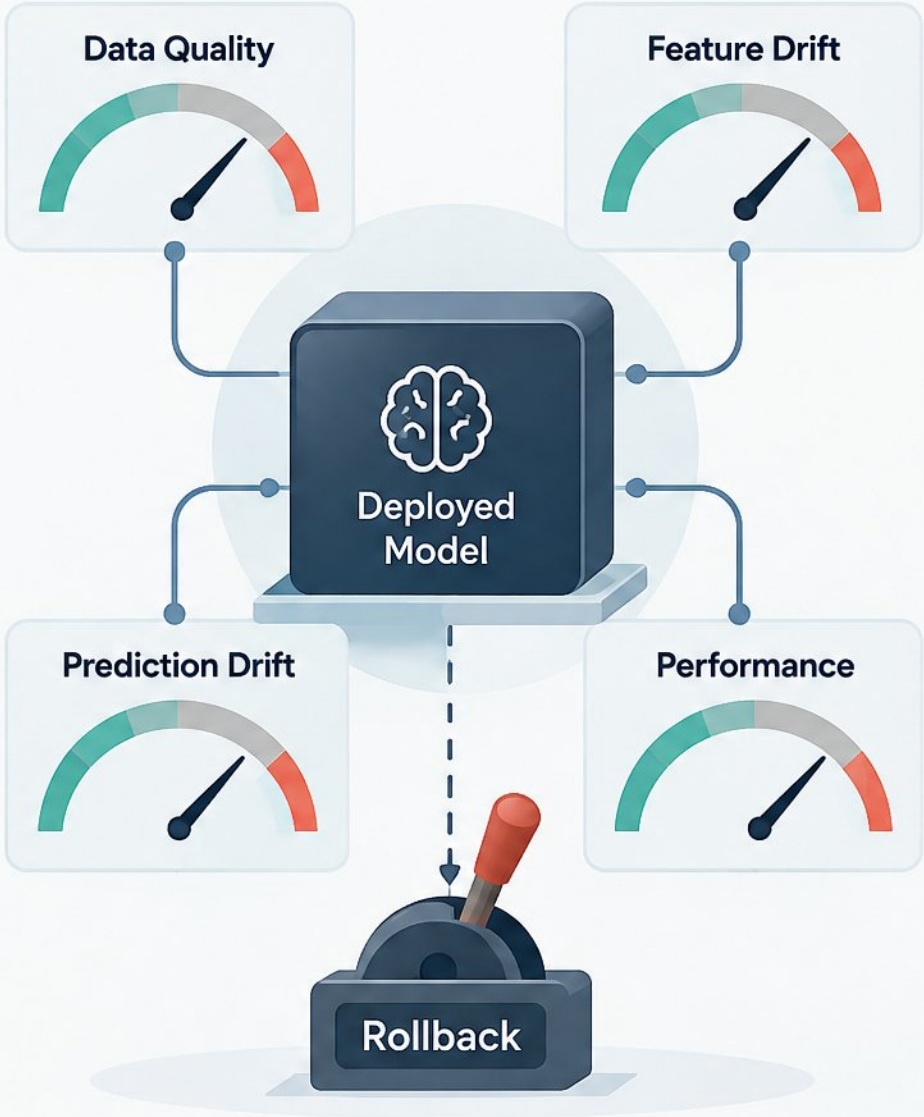
Anti-pattern: deploy-and-forget — no monitoring, no owner, no alert thresholds

Anti-pattern: no rollback path or incident playbook; first failure becomes a crisis

Drift types: data drift (inputs change), concept drift (input-target relationship changes), prediction drift

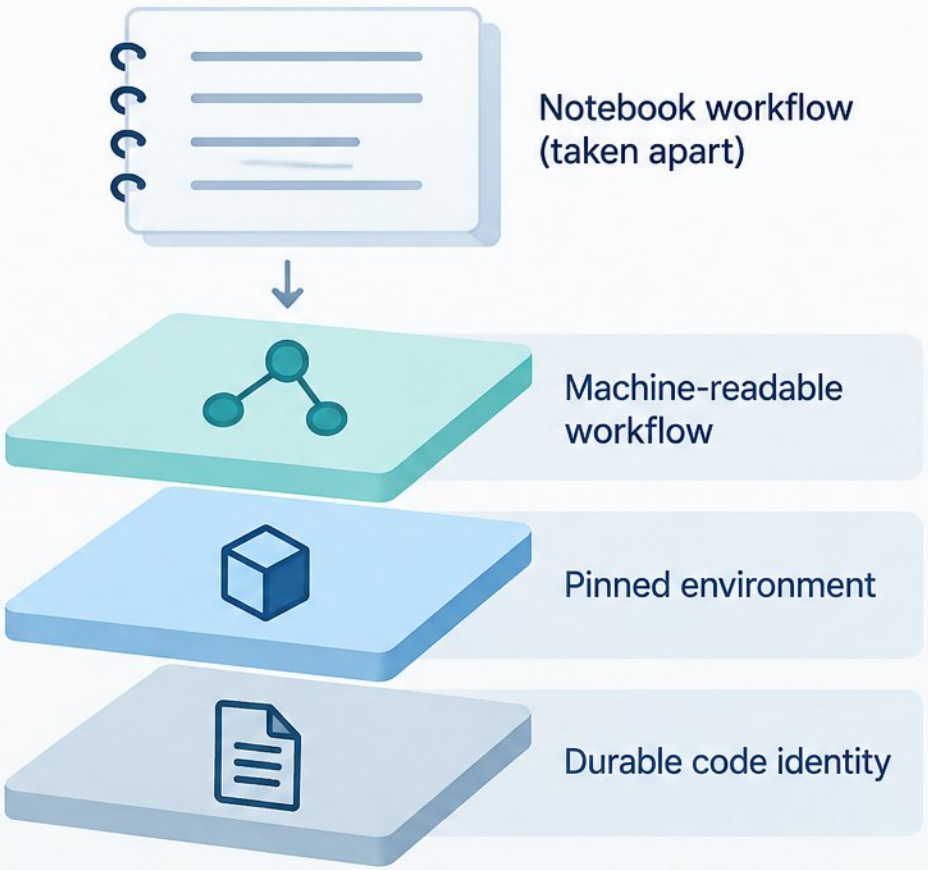
Better alternatives: capture baselines at deployment, layered monitoring, shadow or canary releases

Retrain on drift triggers, not fixed schedules; calibrate alerts with domain knowledge to avoid fatigue



Tooling, Automation, and Reproducibility Anti-Patterns

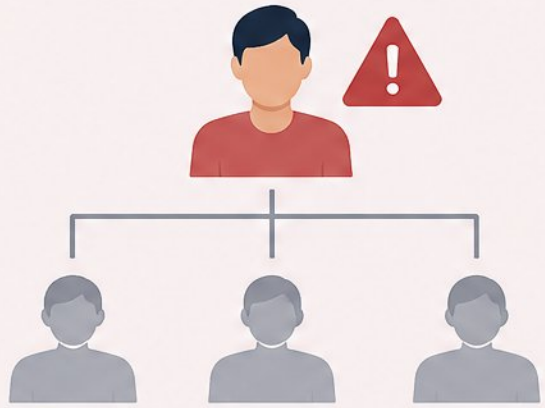
- Notebook-only workflows with hidden state; anything done by hand should be done by syntax
- One-person environments don't reproduce: needs workflow logic, pinned environment, durable code identity
- Over-engineering: buying a heavy platform for a small analytical question
- Better: version control tied to commits, parameterized one-command pipelines, lockfiles or containers, CI tests
- Apply rigor proportionally to scale and impact; test by rebuilding from scratch on a clean machine





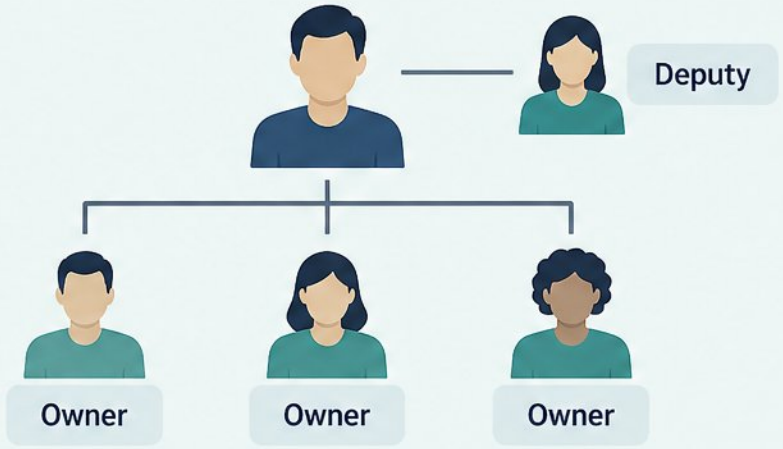
Team Process, Governance, and Communication Anti-Patterns

Anti-Patterns



- Hero culture: critical pipelines depend on one person, no deputy recorded
- No decision owner: if nobody decides with a dashboard, it goes unused
- Governance bolted on as a final gate, not designed into the workflow

Better Approach



- Name one accountable owner per data domain or model; document decision rights
- Cadence: always-on alerts, weekly triage, monthly owner review, quarterly risk review
- Measure behavior: owned products, overdue quality issues, repeated incident causes

PEOPLE + PROCESS + OWNERSHIP = MORE RELIABLE OUTCOMES

Better Alternatives Playbook: One Reference Table

From common pitfalls to stronger practices across the ML lifecycle.

Lifecycle Stage	Common Anti-Pattern	Better Alternative
Framing		Framing: decision-first intake; documented decision, owner, and success contract
Data		Data: contracts, automated quality tests, leakage-aware preprocessing
Experimentation		Experimentation: pre-registered hypotheses; pre-agreed sequential stopping rules
Modeling		Modeling: baseline first, slice-based evaluation, calibration checks
Operations		Operations: monitoring baselines, rollback paths, named owners and deputies

Your 30-60-90 Day Plan, for Individuals, Teams, and Educators

PRACTICE
BETTER
LEARN
TOGETHER

30 Days

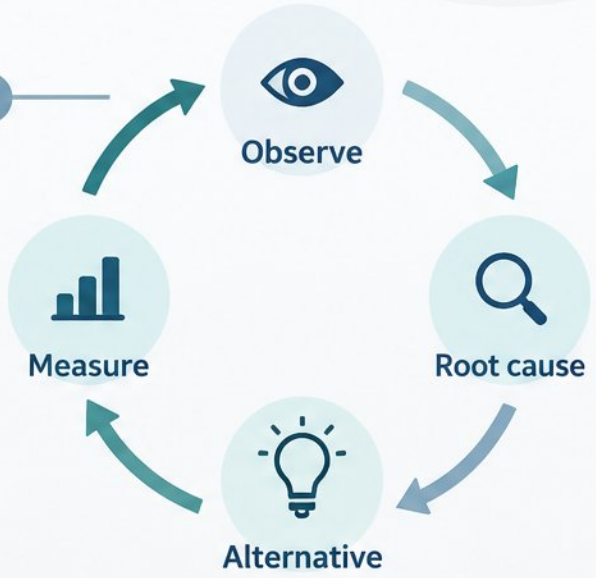
Pick three anti-patterns, assign owners, define success evidence

60 Days

Add one data contract, a leakage check, a monitoring baseline

90 Days

Run case reviews and build a pattern library



- Prioritize upgrades by impact, effort, and reversibility
- Days 1–30: pick three anti-patterns, assign owners, define success evidence
- Days 31–90: add one data contract, a leakage check, a monitoring baseline
- Managers and educators: run case reviews and build a pattern library
- Goal: deliberate, measurable, repeated improvement — not perfection

PersonWise.

PersonWise • AI digital-human course platform



Scan the code or click anywhere to watch this course live, with voice Q&A

personwise.ai/t/a39cee68/public