

Product Management Frameworks:

Comparing Approaches and Use Cases



- A repeatable way to decide what to build, in what order, and how to measure success



- Reduces ambiguity, aligns stakeholders, and speeds up decisions



- Frameworks are tools, not religions—use the lightest one that works



- Roadmap: compare major frameworks, map to real situations, build a selection plan



Why Frameworks Matter: Roles, Contexts, and Core Trade-offs



- PMs, teams, and founders use frameworks differently by stage



- Wrong framework costs: process overhead, false precision, speeding up errors



- Core questions: problem, action, order, and measurement



- Course principle: diagnose the situation before choosing a tool



The Framework Landscape: Discovery, Prioritization, Delivery, and Strategy



Most effective teams use 3-5 frameworks, not one

A Comparison Map: What Each Framework Is Actually For

- Grouped by job: discovery, prioritization, delivery, strategy, measurement



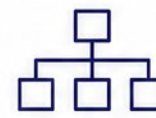
Discovery



Lean Startup



JTBD



Opportunity
Solution Tree



Double Diamond



Prioritization



RICE



ICE



MoSCoW



Kano



Value vs Effort



WSJF



Delivery



Scrum



Kanban



Dual-Track Agile



Strategy & Measurement



OKRs



North Star Metric



HEART

Selection Criteria: Matching Framework to Situation



- **Key factors:** team size, product maturity, uncertainty, reversibility, market speed, data



- **Reversibility:** one-way vs two-way doors



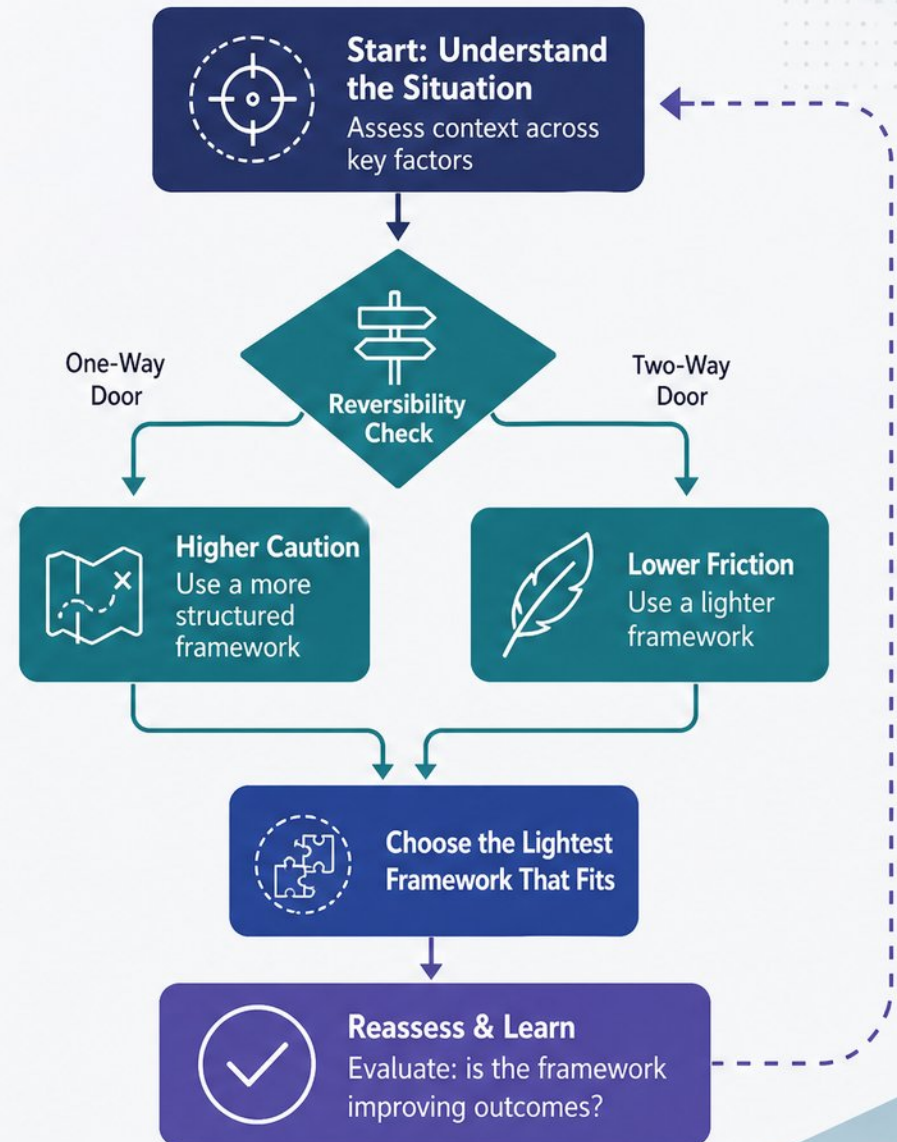
- **80/20 rule:** reading the situation is the hard part; choosing the framework is the easy part



- **Diagnostic:** name the decision you get wrong, then pick the lightest framework that fixes it



- **Evaluate:** is the framework improving outcomes?



Use Case 1: Early-Stage Discovery and Validation



- Validate real customer problems before building the wrong solution



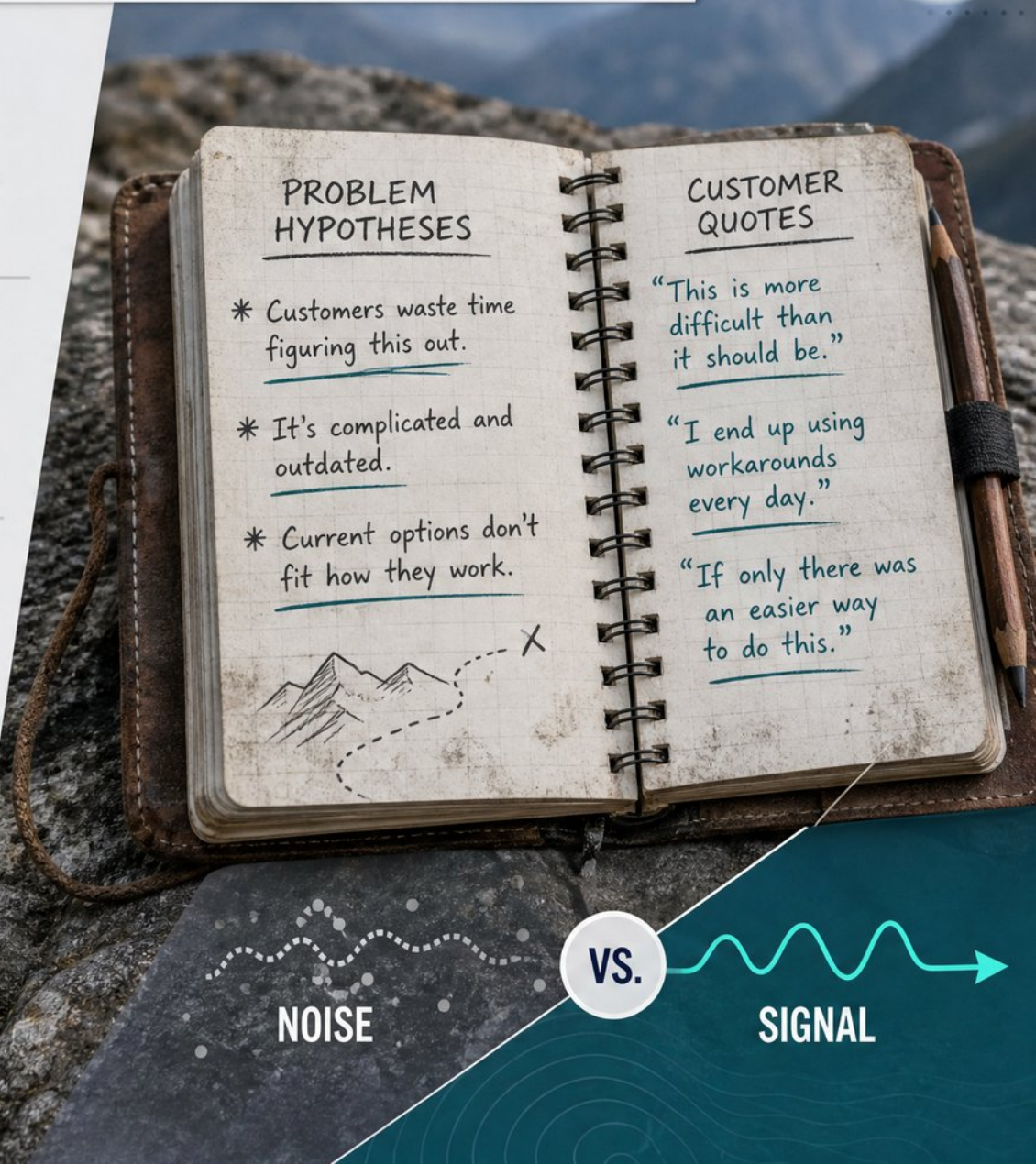
- Apply Lean Startup, customer interviews, and JTBD frameworks



- Track learning velocity and validated risk reduction, not output



- Skip heavyweight prioritization and OKR processes until PMF



Use Case 2: Prioritization and Roadmap Decisions



SPECIALIZED TOOLS

Different lenses.
One roadmap.
Better decisions.

RICE



Ranks by score:
Reach × Impact × Confidence ÷ Effort

MoSCoW



Scopes releases:
Must, Should, Could, Won't-have

KANO



Classifies value:
basics versus delighters



- **RICE** ranks by score: $\text{Reach} \times \text{Impact} \times \text{Confidence} \div \text{Effort}$
- **MoSCoW** scopes releases: Must, Should, Could, Won't-have
- **Kano** classifies value: basics versus delighters
- Combine them: **MoSCoW** to filter, **RICE** to rank, **Kano** to balance
- Avoid false precision, easy picks, and HiPPO-driven choices

Use Case 3: Delivery, Iteration, and Team Alignment



- Scrum: predictable sprints;
Kanban: continuous flow;
Dual-Track: parallel discovery and delivery



- Align product, design, engineering on shared outcomes, not feature lists



- Connect delivery cadence to measurable results, avoiding dysfunction



- Discovery and delivery run in parallel, not as sequential phases



- Discovery validates next work while delivery ships current work



Use Case 4: Strategy, Goals, and Measurement



- **North Star Metric:** single metric for core product value



- **OKRs:** quarterly goals that target input metrics driving the North Star



- **Product strategy:** choices linking vision to measurable outcomes



- **Avoid vanity metrics;** select leading indicators that influence business results



- **Review cadence:** weekly check-ins, quarterly grading, annual North Star reassessment

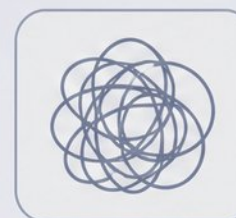
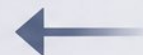


SIGNAL

Leading indicators that inform and drive outcomes



VS.



NOISE

Vanity metrics that distract and don't influence results

Combining Frameworks in Practice



- Targets different decisions; avoid running the same item through multiple tools



- OKRs + RICE + JTBD + HEART: direction, backlog, discovery, UX measurement



- MoSCoW then RICE: scope the release, then rank the 'Musts' internally



- Add, replace, or remove one piece at a time; adapt inputs and cadence



- Avoid framework churn, cargo-culting, and process theater





Anti-Patterns: How Frameworks Fail in Real Teams



- Subjective scores create the illusion of objectivity



- Framework theater: process over outcomes



- Prioritization as opinion laundering



- Enterprise cadence applied to startups



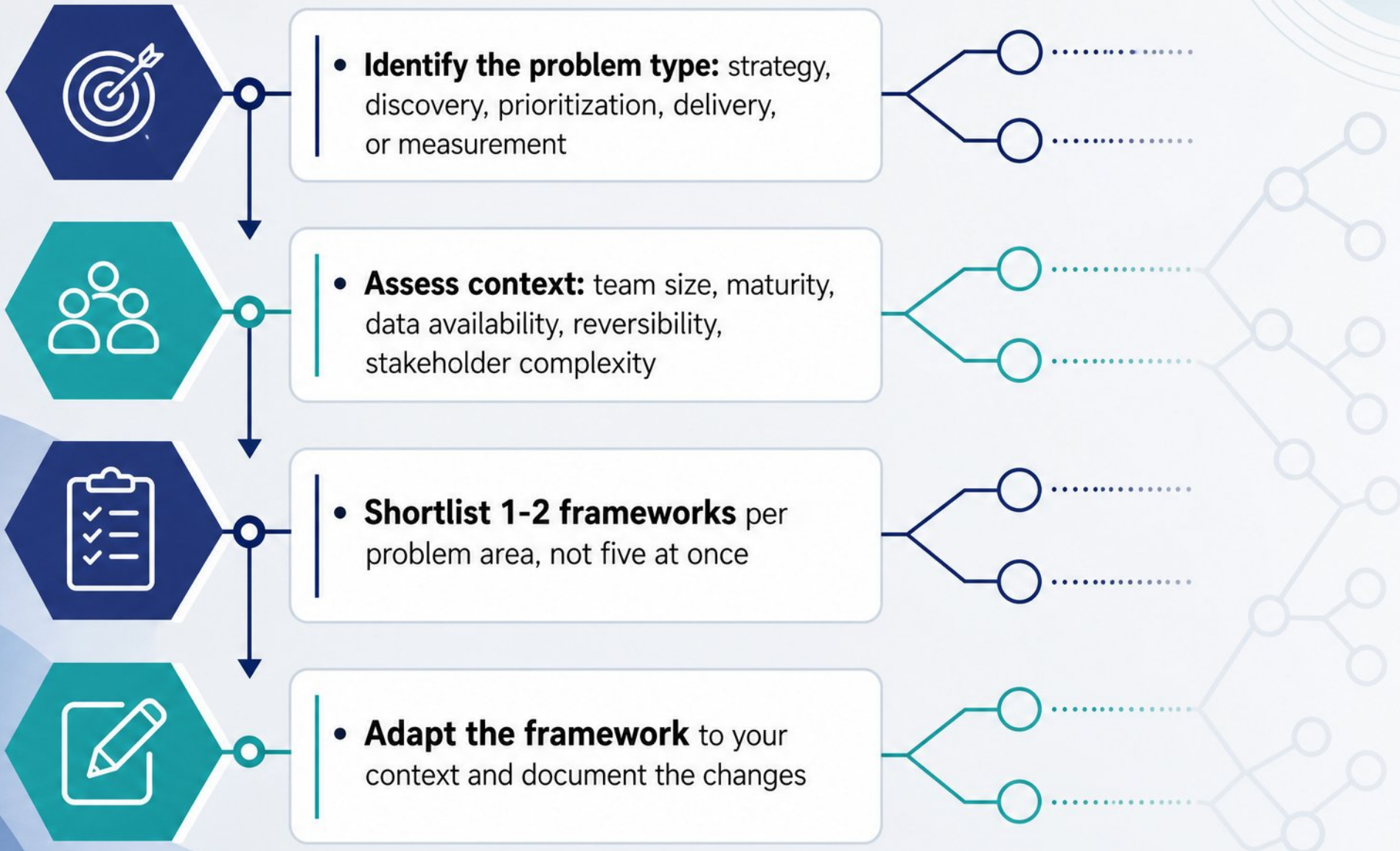
- Copying systems without their context



- Antidote: treat frameworks as decision aids



A Reusable Decision Flow for Choosing a First Framework





Action Plan: Selecting and Testing a Framework for Your Team



- Assess team context and pain points first



- Choose one improvement area and one starter framework



- Define a short experiment with clear success signals



- Set kill criteria to know when to stop



- Avoid pilot-test pitfalls; evaluate feedback loops

PILOT EXPERIMENT

CONTEXT SIGNALS



Understand the team, current challenges, and what matters most.

STARTER FRAMEWORK



Select one framework that fits the improvement area and team context.

KILL CRITERIA



Define clear conditions that will end the experiment early.

Wrap-Up and Next Steps



- Four question families: problem, action, sequence, measurement



- Lightweight system: 2-3 outcomes, weekly discovery, one prioritization method, metrics review



- Run one framework for a quarter; then keep, adapt, or drop



- The framework is not the work: reading the situation is



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/f0014aeb/public