

THE OLD WAY



THE NEW WAY

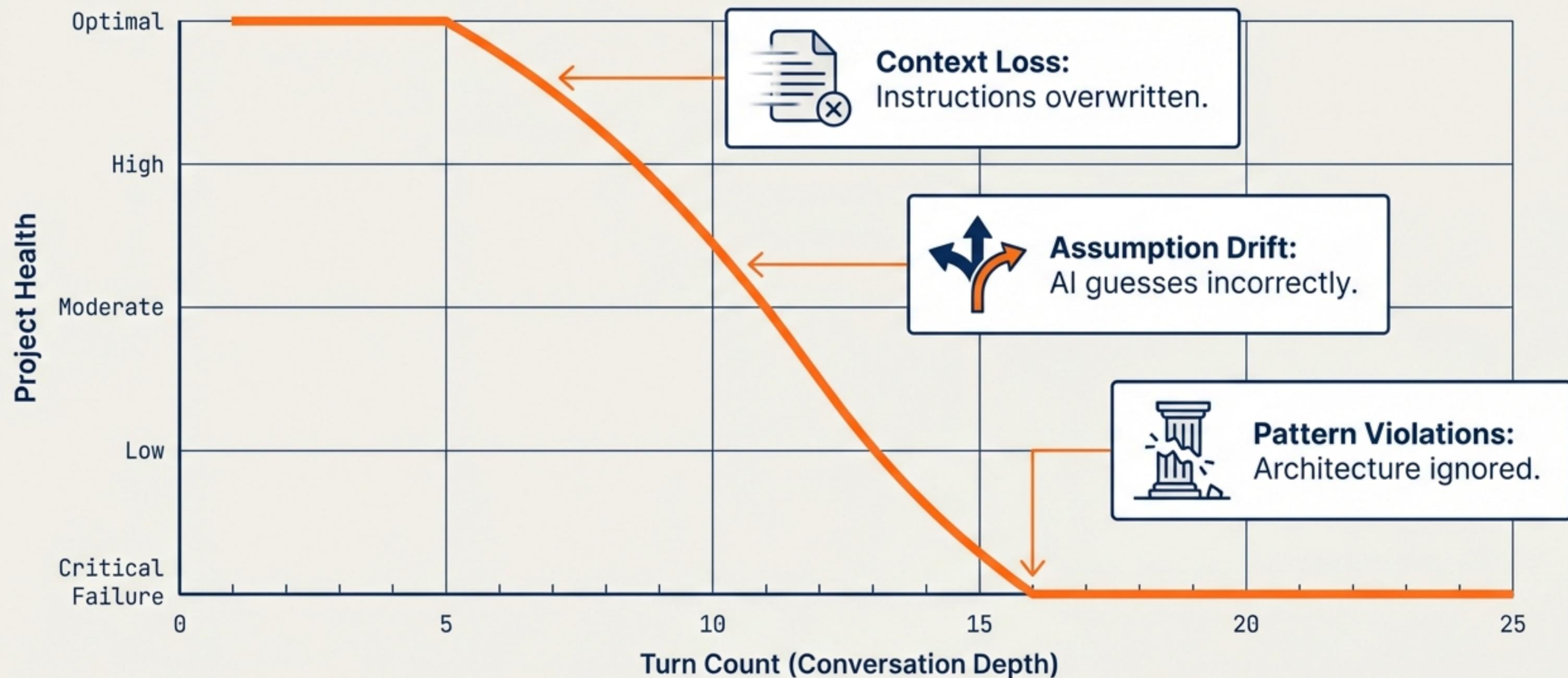


SPEC-DRIVEN DEVELOPMENT

Why “Vibe Coding” Fails at Scale—and How to Fix It.

A MANUAL FOR THE CLAUDE CODE ERA

The Diagnosis: The Vibe Dip



Iterative discovery mathematically guarantees degradation over time.

The Solution



Vibe Coding (Discovery)

I'll know what I want when I see it.



Result: Endless Iteration.

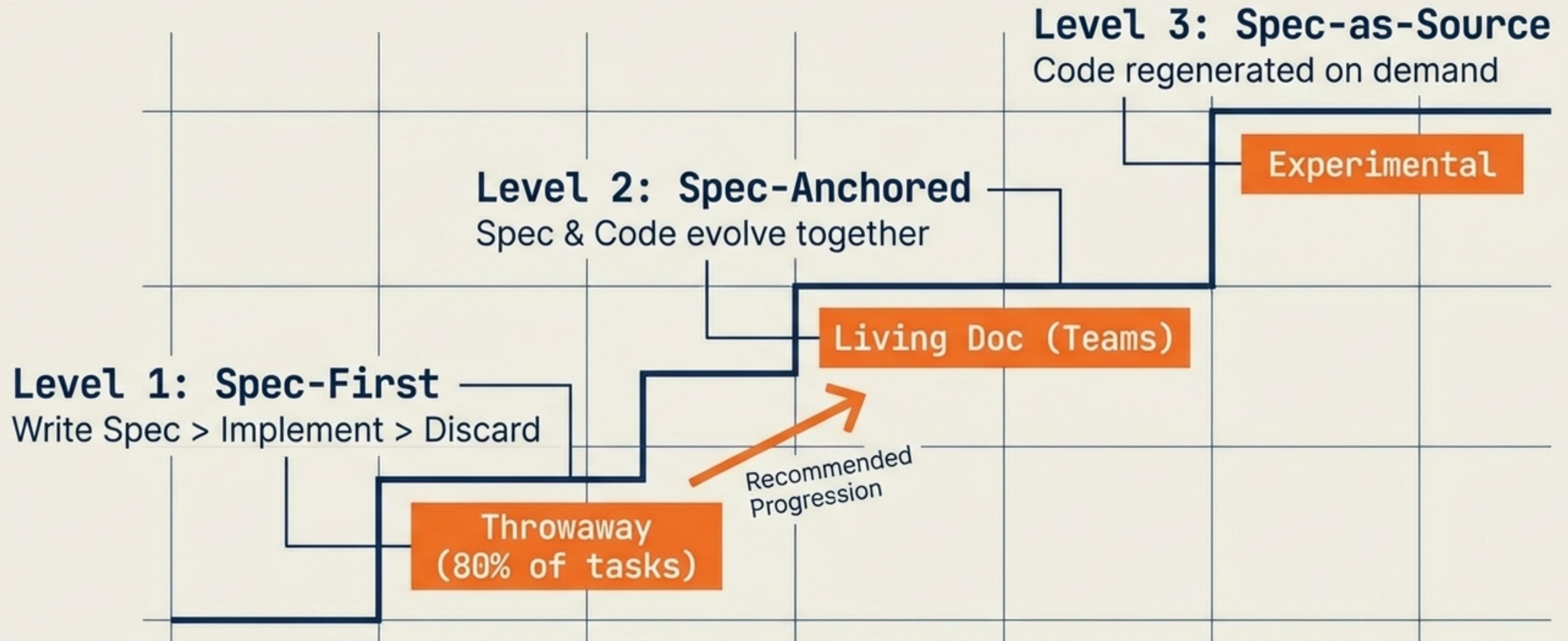
SDD (Declaration)

Here is exactly what exists, what to build, and what NOT to do.



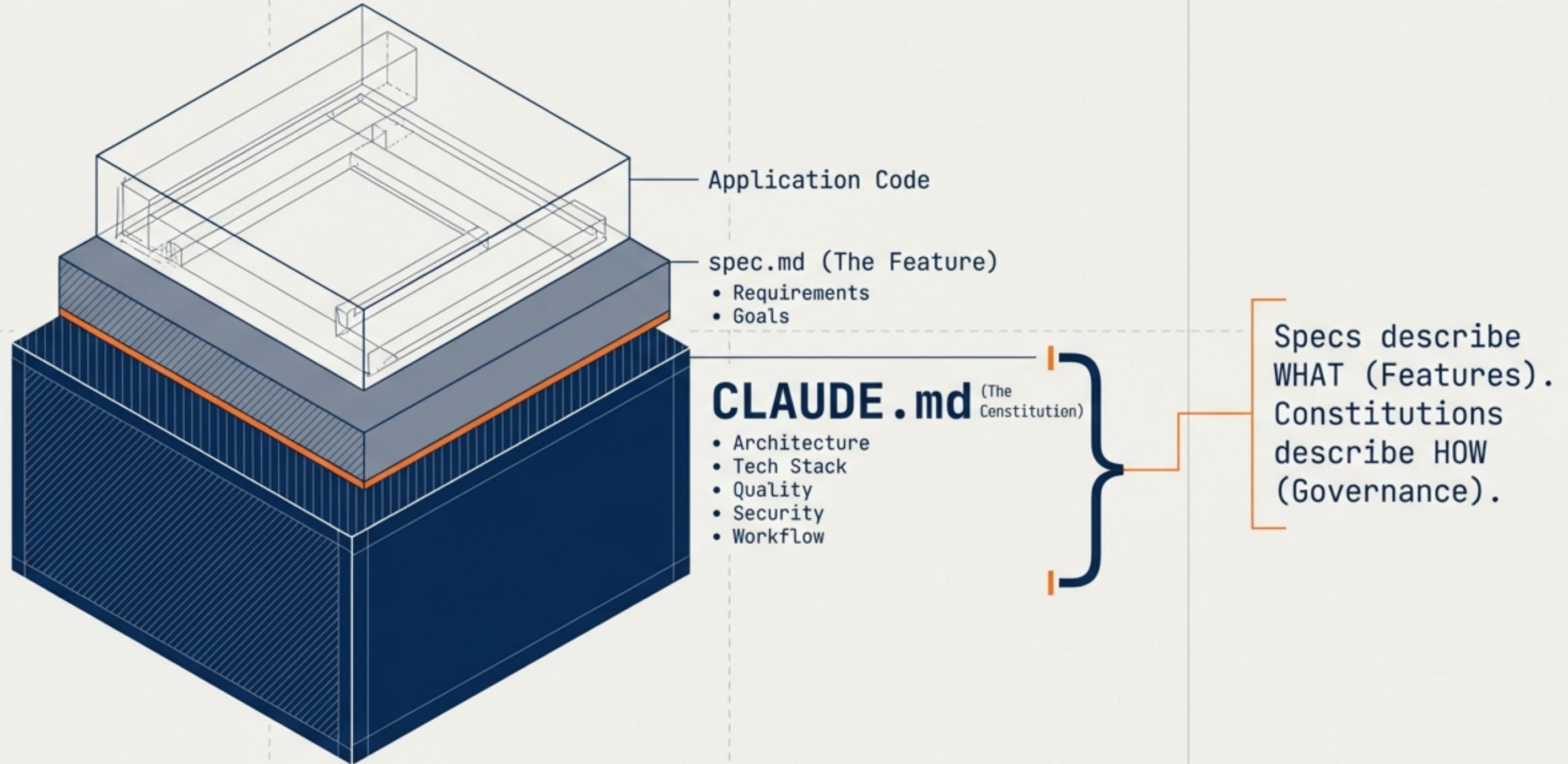
Result: Correct on First Try.

The Maturity Spectrum

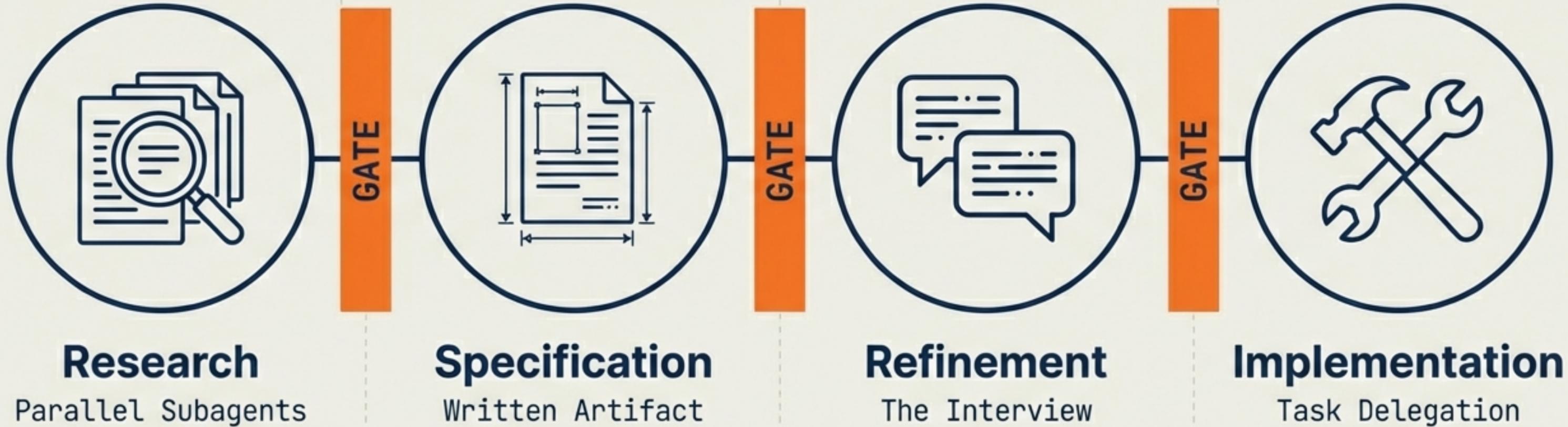


Default to Level 1. Graduate to Level 2 for longevity.

The Foundation: Project Constitution

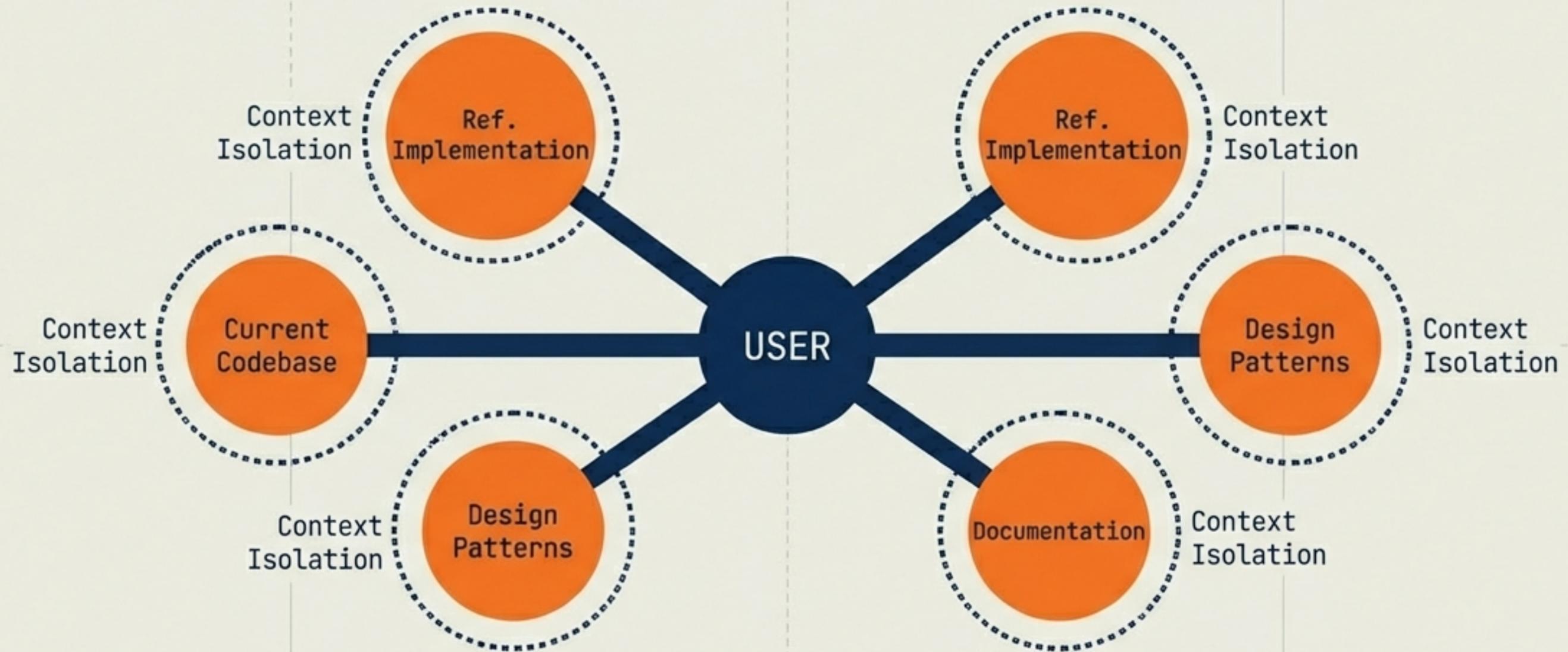


The 4-Phase Workflow



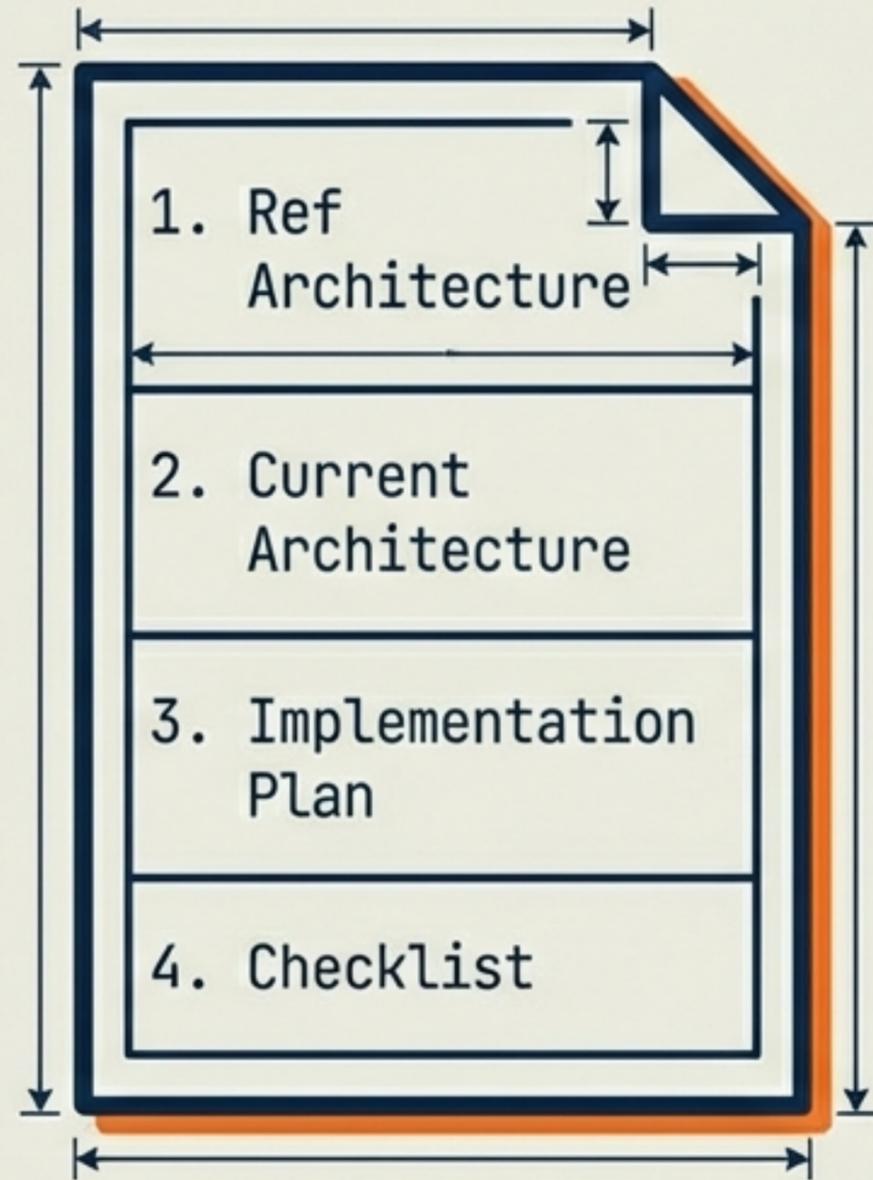
“Review the plan, not just the code.”

Phase 1: Parallel Research



```
> Spin up multiple subagents to investigate [topic]...
```

Phase 2: Effective Specs



The Template

Constraints > Requirements

VAGUE (Bad)	CONSTRAINT (Good)
Make it fast. ❌	Do NOT pre-fetch > 3 items. ✅
Handle errors. ❌	Retry 3x with exponential backoff. ✅

Phase 3: Refinement (The Interview)

Read this spec and ask me every question that could cause failure.

Ambiguity
Categories

Data
Decisions

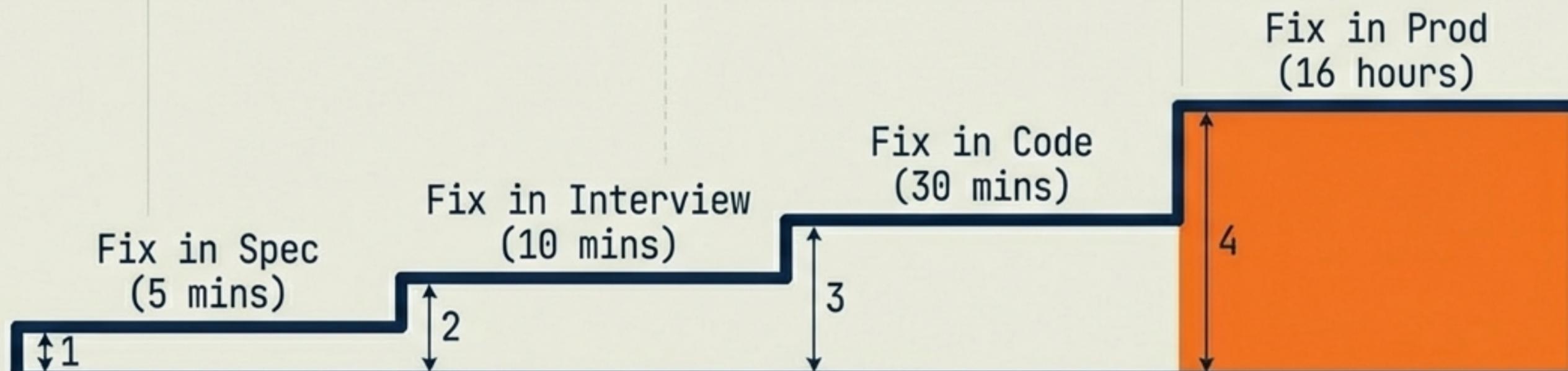
Conflict
Resolution

Patterns

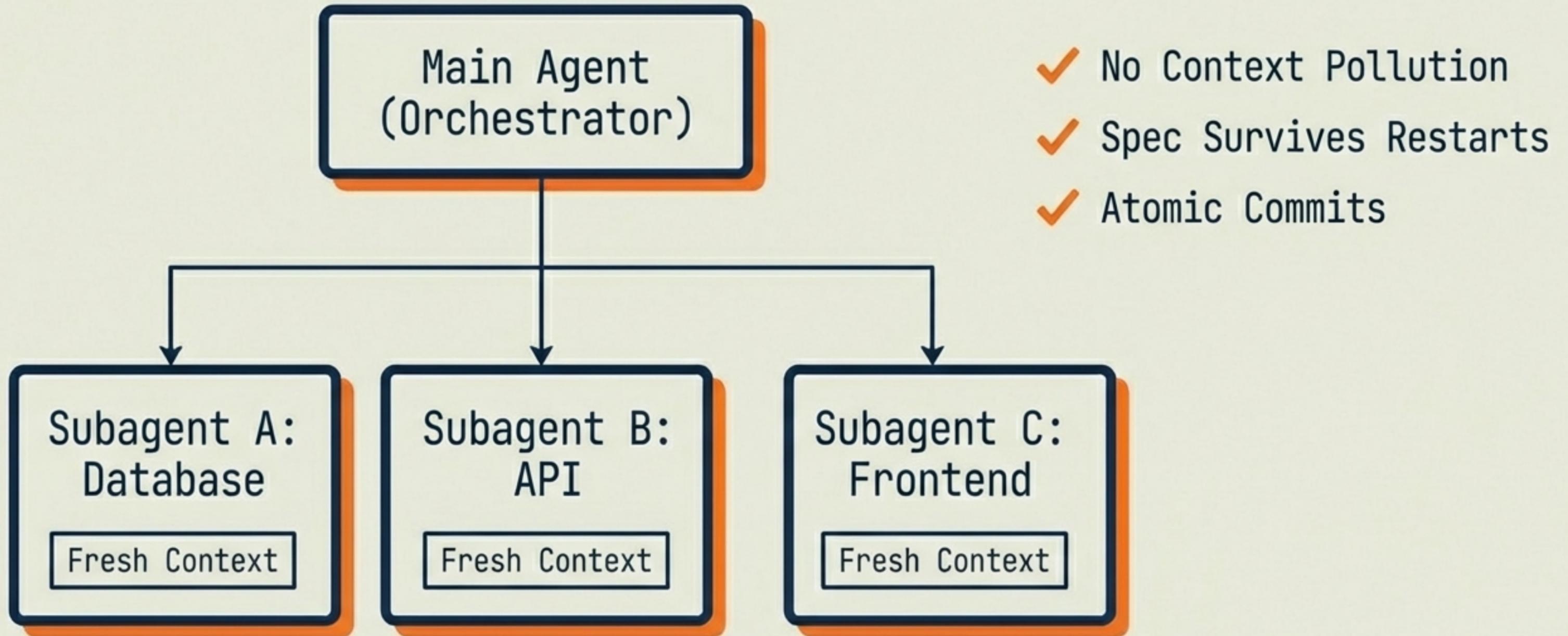
Failure
Recovery

Boundaries

Escalator
of Cost

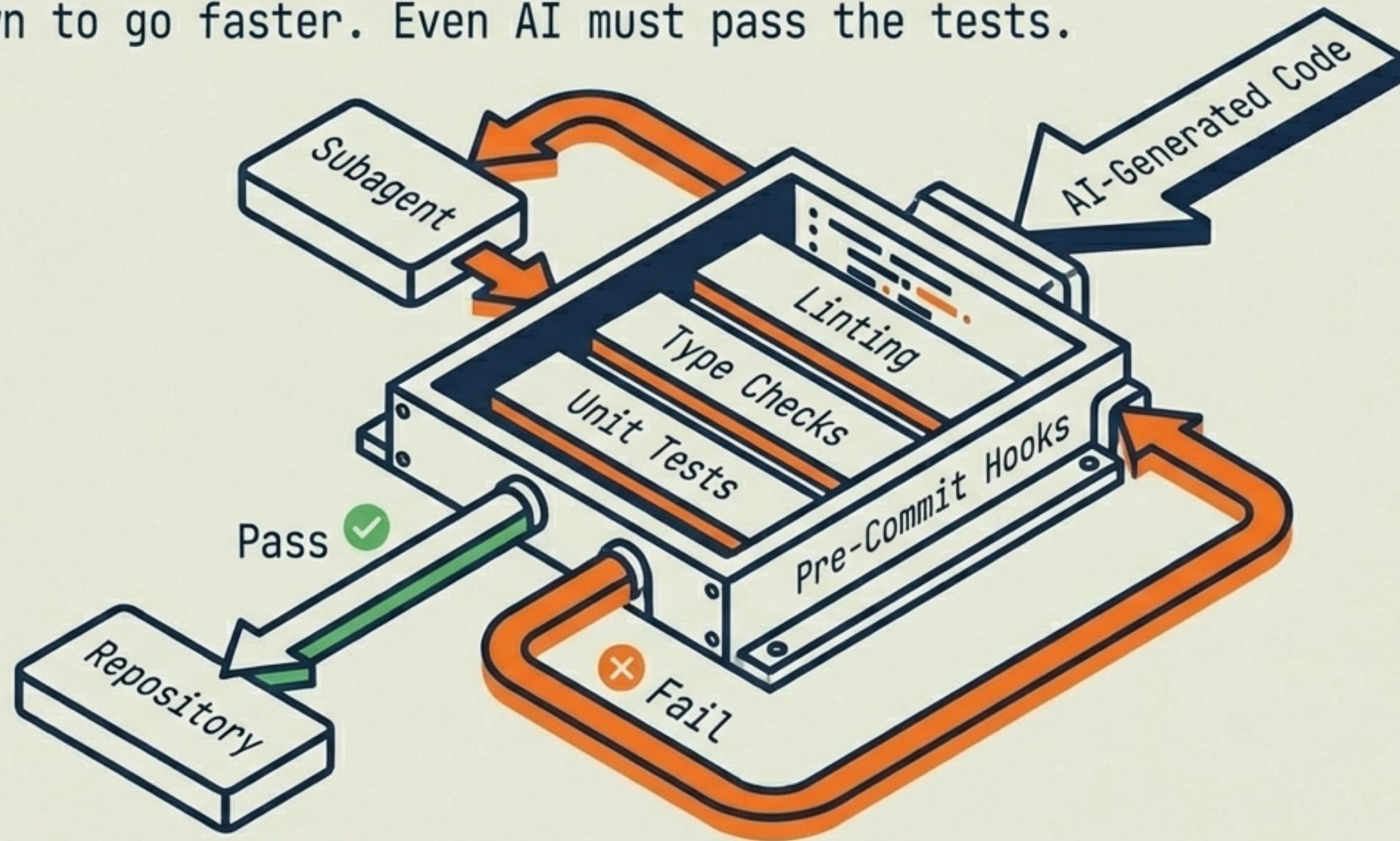


Phase 4: Task-Based Implementation

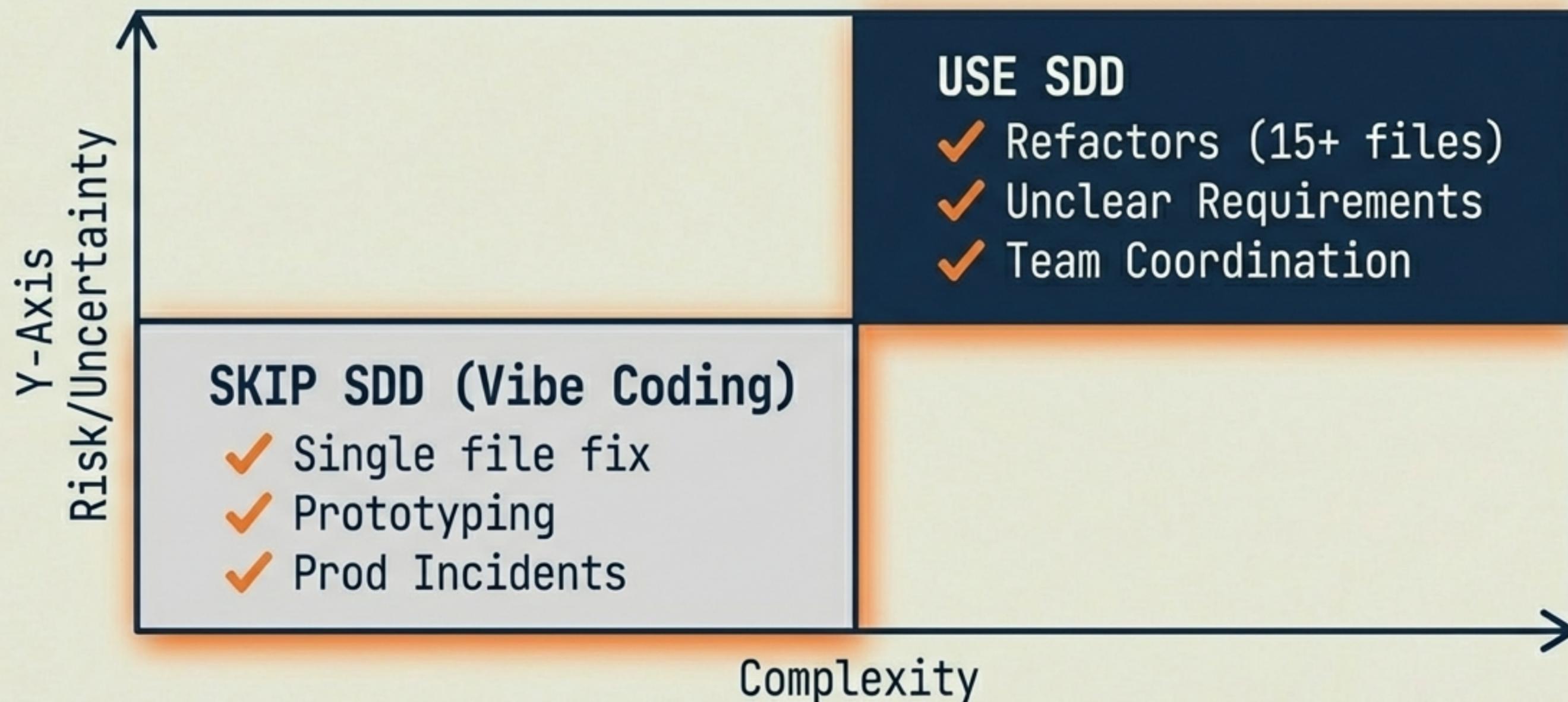


The Backpressure Mechanism

Slowing down to go faster. Even AI must pass the tests.



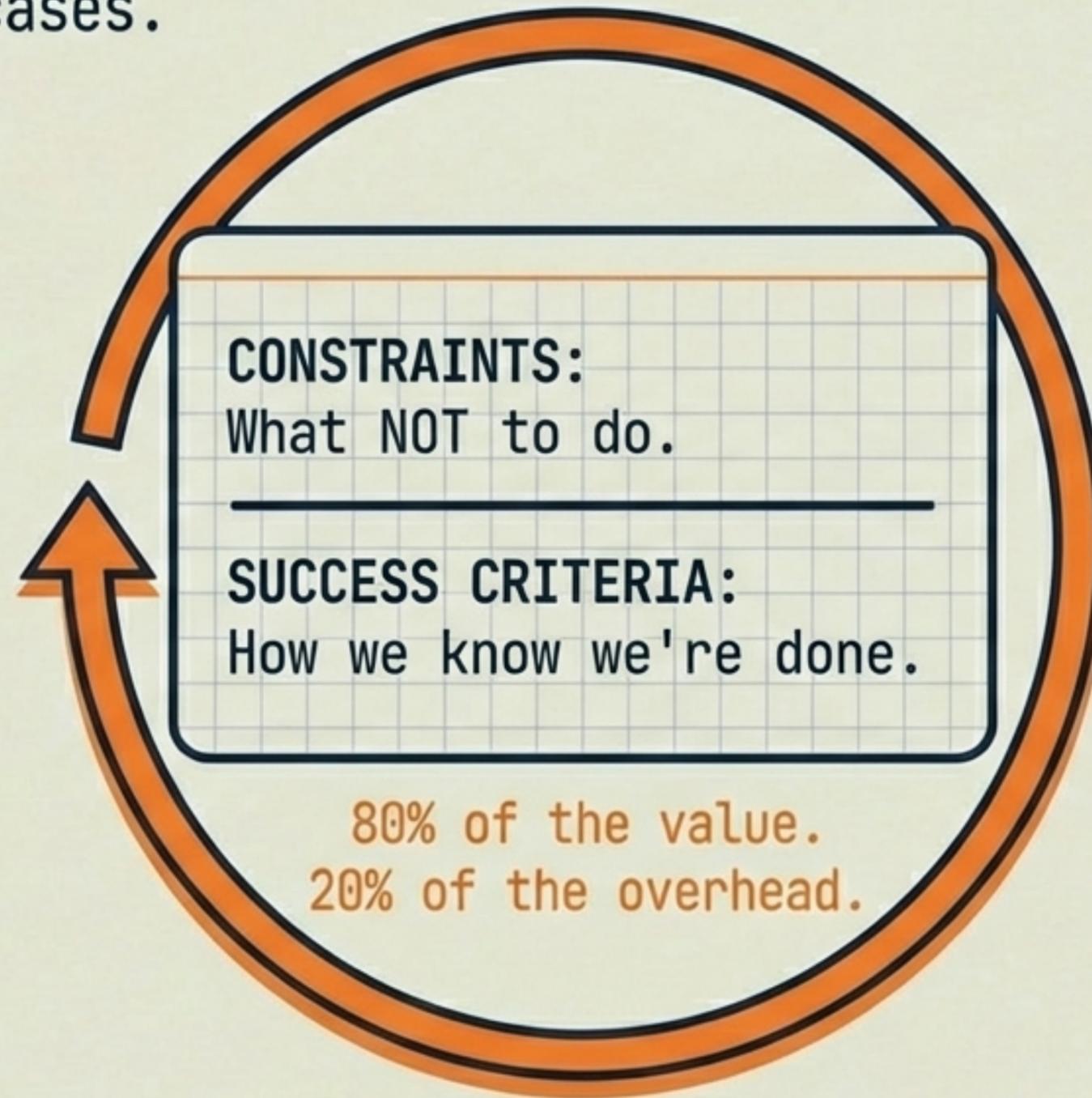
Decision Framework



IF (files_affected > 5) OR (requirements_unclear == TRUE) -> Use SDD

The Lightweight Spec Pattern

For the borderline cases.



Summary & Toolkit

The Philosophy

- ✓ Vibe Coding = Discovery (Prototyping)
- ✓ SDD = Engineering (Production)

The Toolkit



CLAUDE.md (Constitution)



Parallel Research



The Interview



Context Isolation

“A great software project is shaped by extracting the ambiguity before writing the code.”