```
> execute ways_of_working_s07e00.sh
```

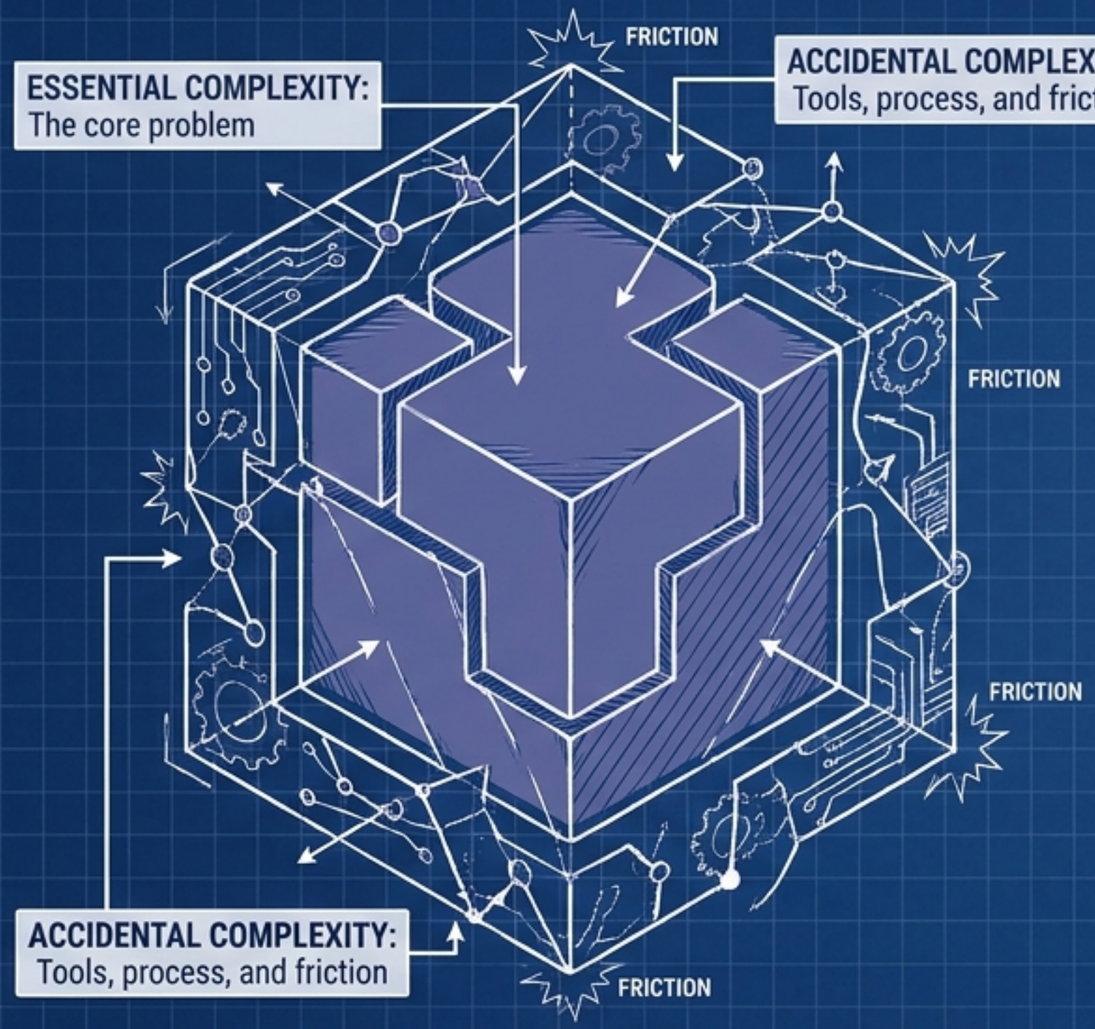
Ways of Working — S07E00: How Software Teams Organise Work

From Waterfall to AI-Native: Separating empirical evidence from practitioner opinion.

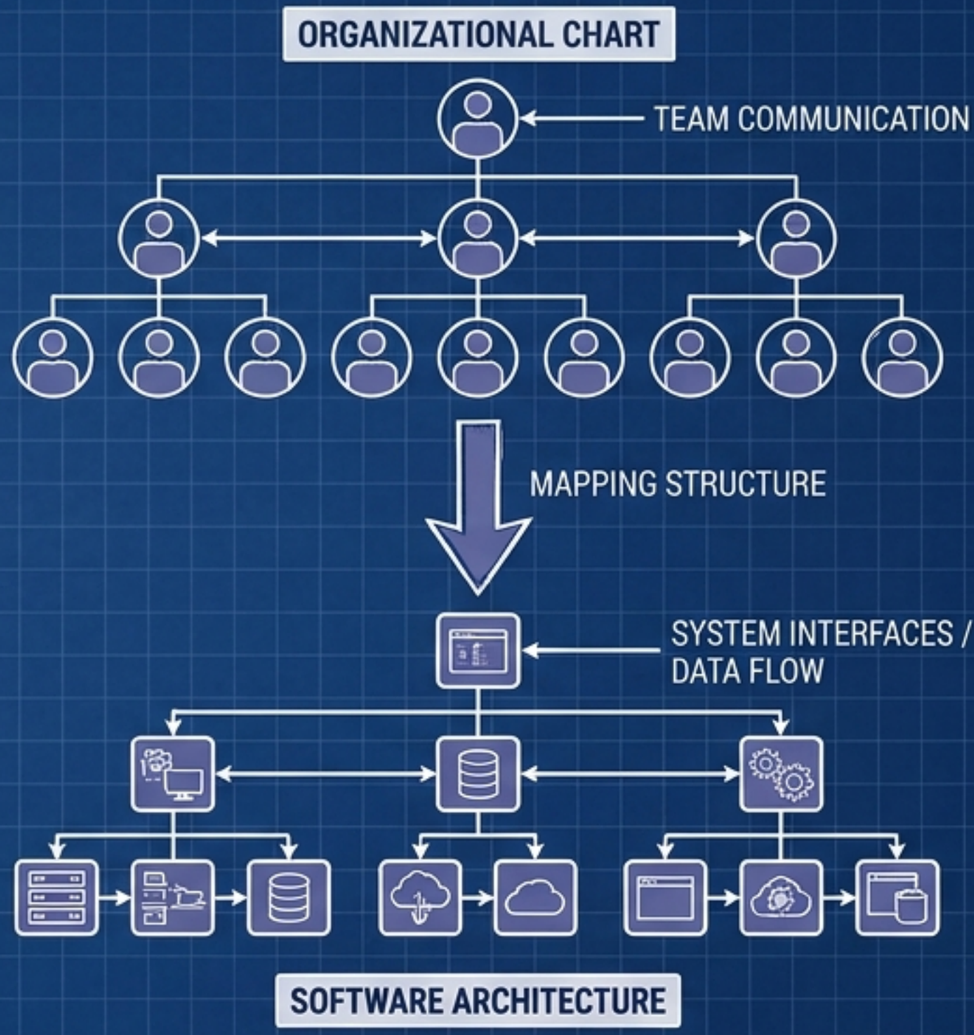
[TARGET_AUDIENCE: DEVELOPERS | TECH LEADS | PRODUCT MANAGERS]

Gemini Notebook

BROOKS' LAW: ESSENTIAL vs. ACCIDENTAL COMPLEXITY



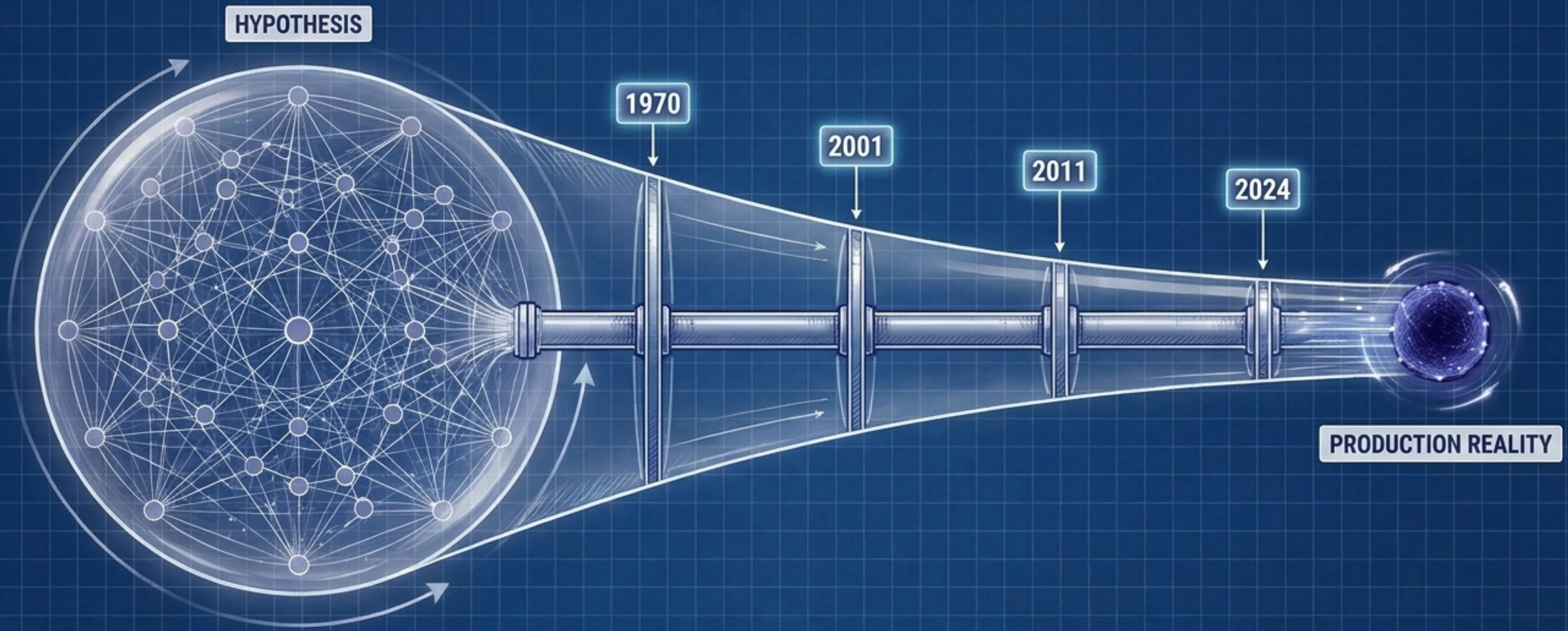
CONWAY'S LAW: ORGANIZATIONAL STRUCTURE MIRRORS SYSTEM DESIGN



Software delivery is governed by two constraints: We can only solve so much complexity at once (Brooks), and our solutions will inevitably mirror our communication structures (Conway).

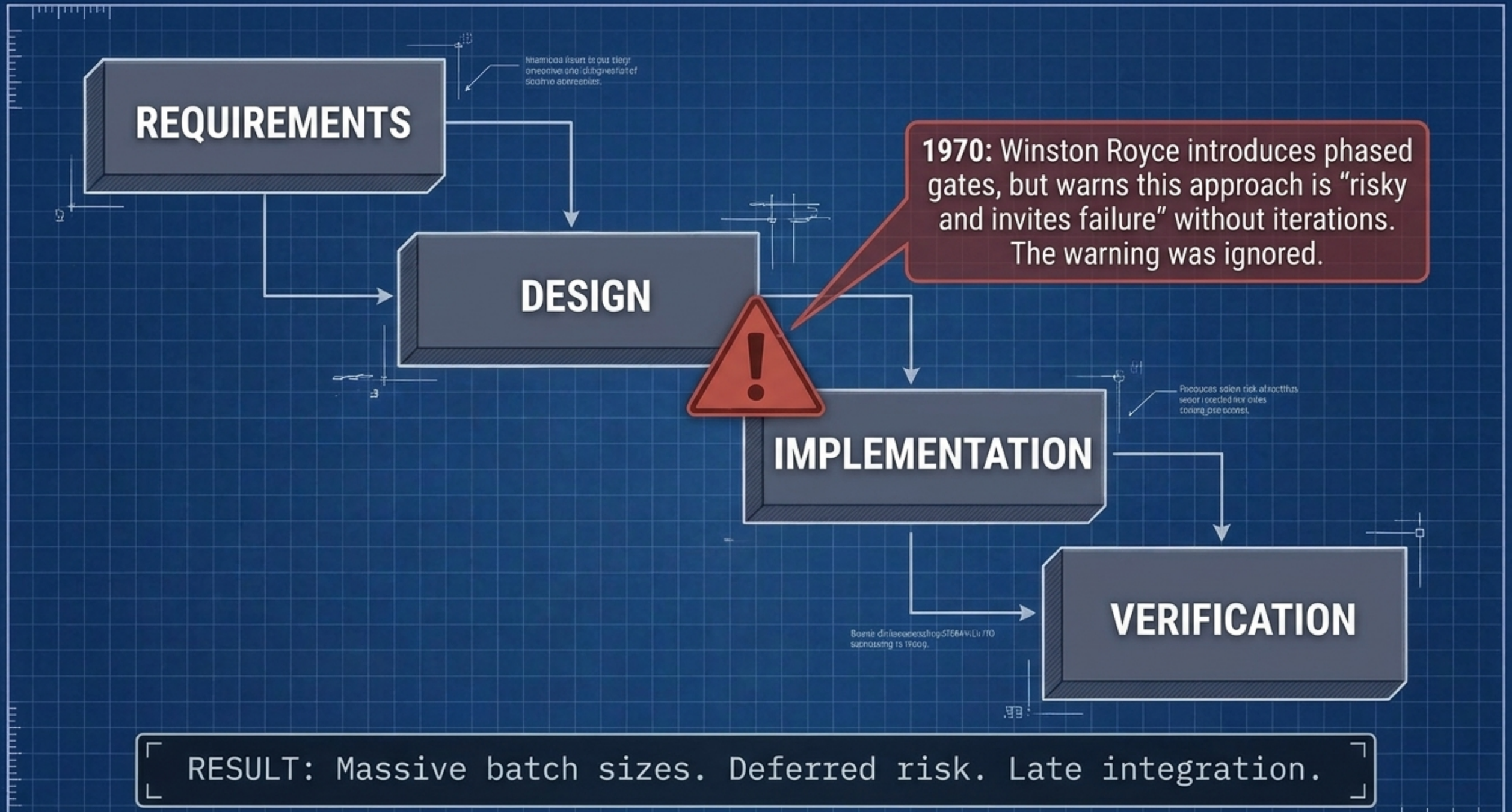
Gemini Notebook

The 55-Year Unifying Thread

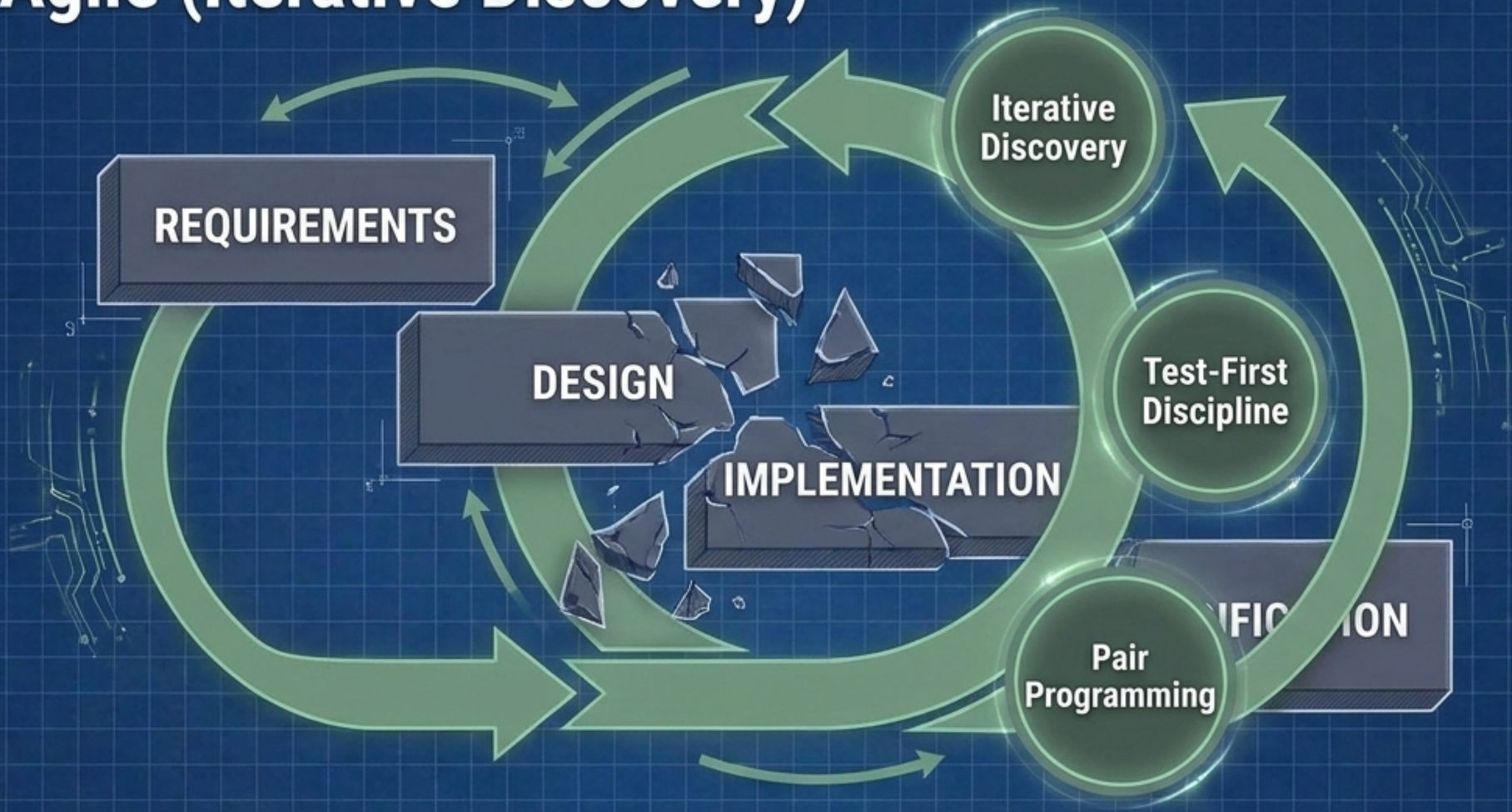


Every methodology shift is an attempt to solve one physics problem: Shrinking the feedback loop.
Reducing the time and batch size between having an idea and verifying it with a real user.

Gemini Notebook


REQUIREMENTS
DESIGN
IMPLEMENTATION
VERIFICATION
1970: Winston Royce introduces phased gates, but warns this approach is “risky and invites failure” without iterations. The warning was ignored.
RESULT: Massive batch sizes. Deferred risk. Late integration.

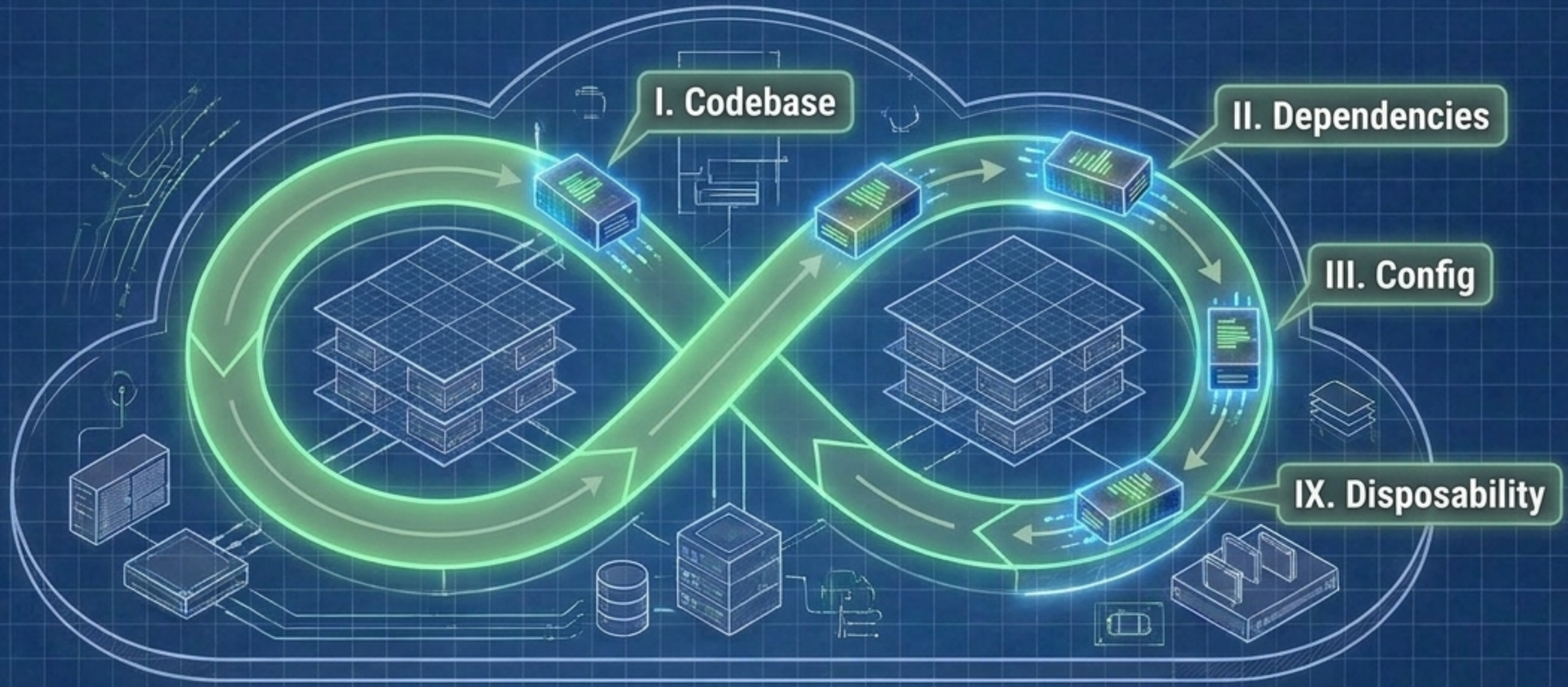
Era 2: Agile (Iterative Discovery)



The 2001 Agile Manifesto prioritized responding to change over following a plan. Extreme Programming (XP) introduced test-driven discipline to handle the accidental complexity of continuous change.

Gemini Notebook

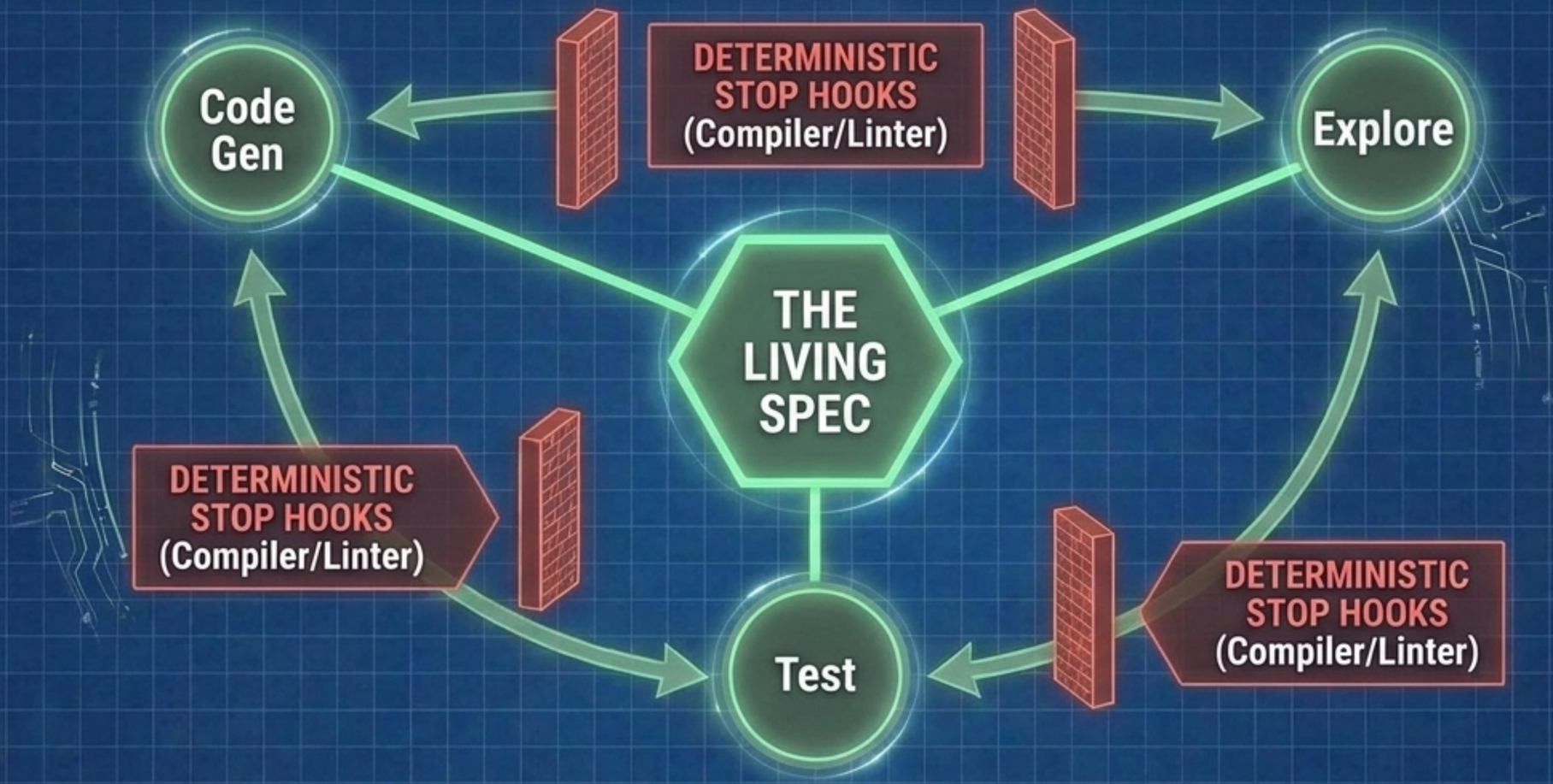
Era 3: DevOps & Continuous Delivery



Shrinking the loop required isolating dependencies (12-Factor), separating build/run stages, and automating the pipeline. Delivery became a continuous stream of stateless processes.

Gemini Notebook

Era 4: Spec-Driven & AI-Native

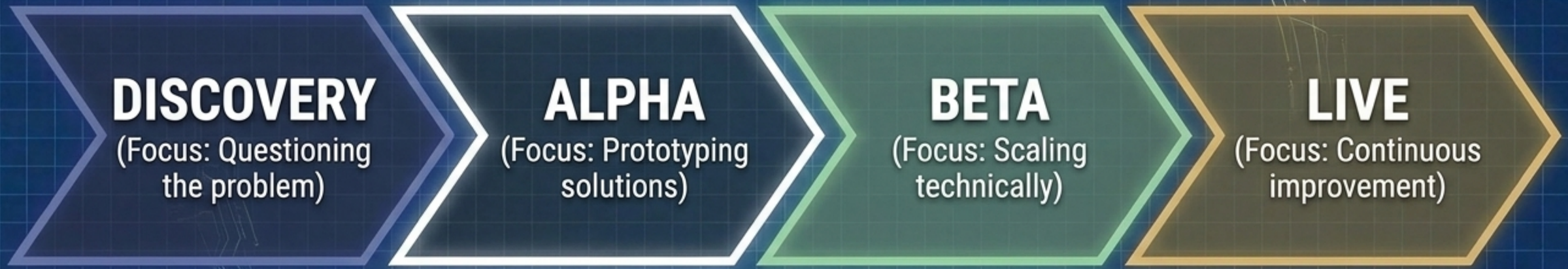


Code generation is cheap. The new bottleneck is intent. Development shifts to maintaining “living specifications” (SDD) while agents navigate the codebase and iterate against deterministic test harnesses.

Gemini Notebook

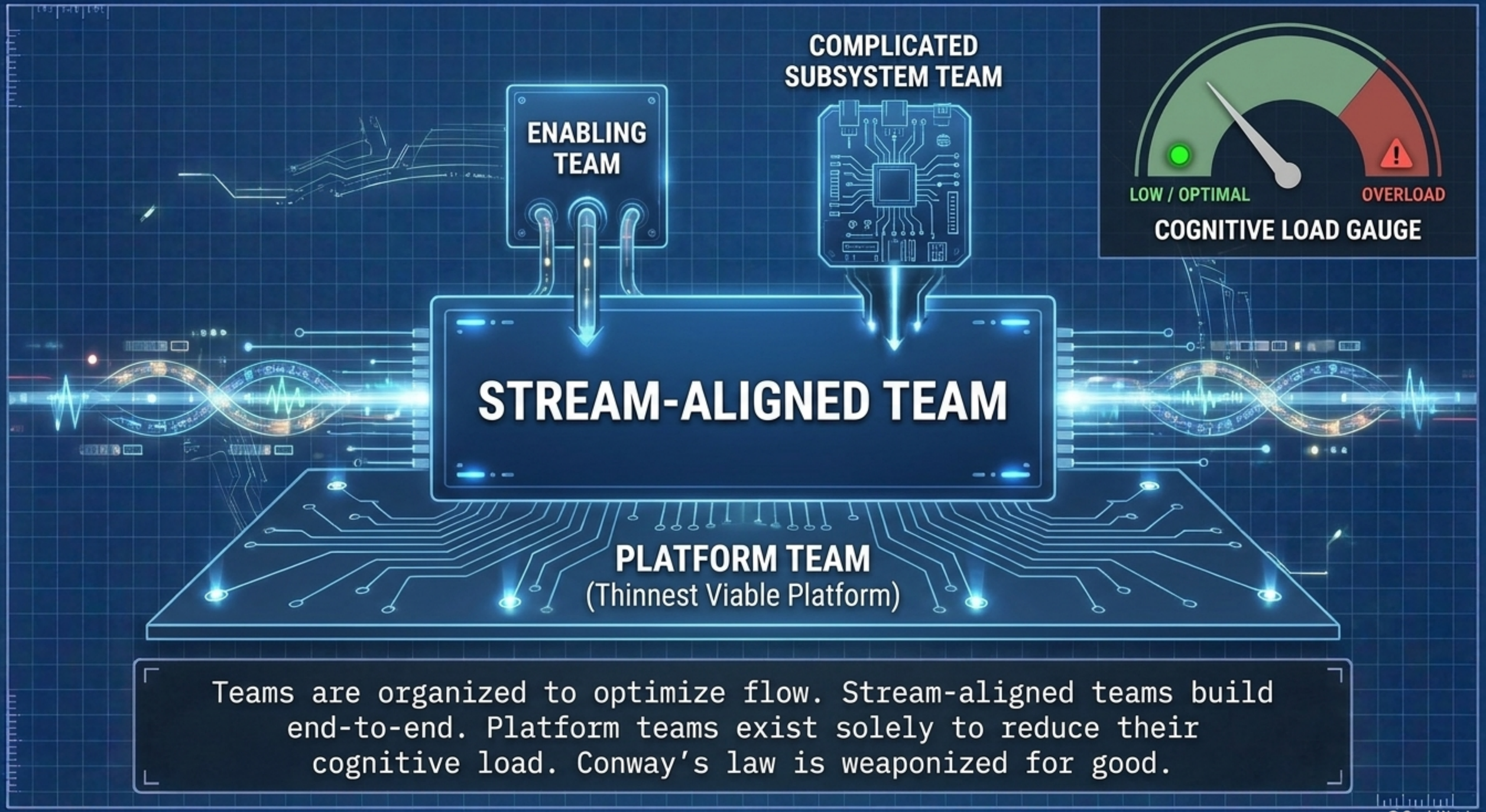
Dimensions	Era 1: Waterfall	Era 2: Agile	Era 3: DevOps/CD	Era 4: AI-Native/SDD
Primary Bottleneck	Requirements gathering	Developer typing speed	Pipeline automation	Human verification / cognitive load
Unit of Batch Size	The Project	The User Story	The Git Commit	The Prompt / Spec
Source of Truth	The PRD	Working Software	Version Control	The Living Spec (Memory Bank)
Verification Mechanism	QA Phase	Unit Tests (TDD)	CI/CD Pipelines	Agentic automated harnesses
Classic Failure Mode	Building the wrong thing	Flaccid Scrum	Deployment fatigue	Unreviewed AI code dumps

Gemini Notebook



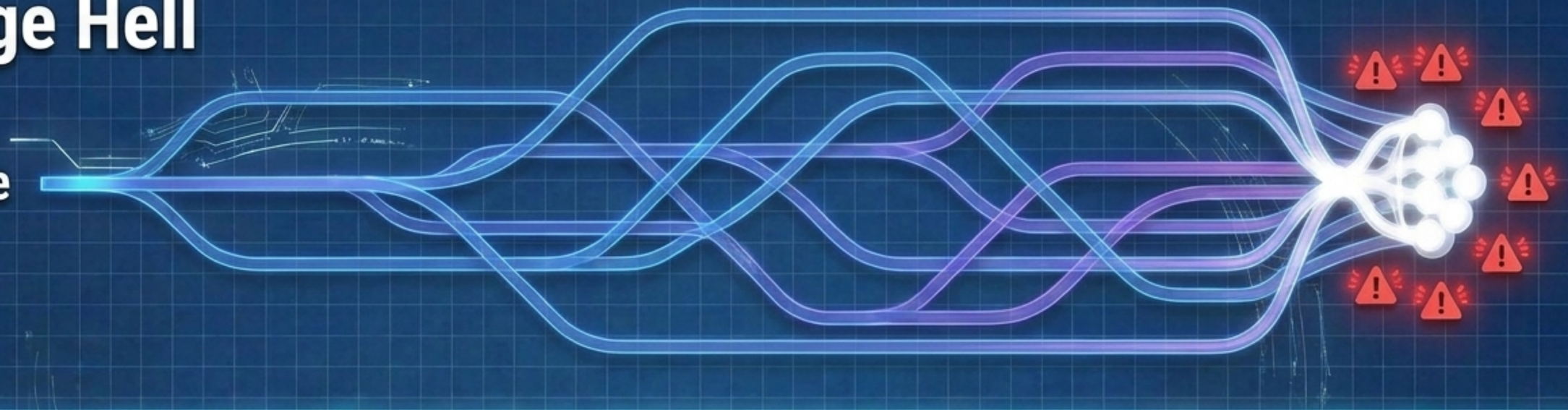
Synthesis: High-performing teams do not start coding immediately. Problem discovery must be separated from technical scaling, ensuring the right thing is built before optimizing how it runs.

Gemini Notebook



Gemini Notebook

Merge Hell

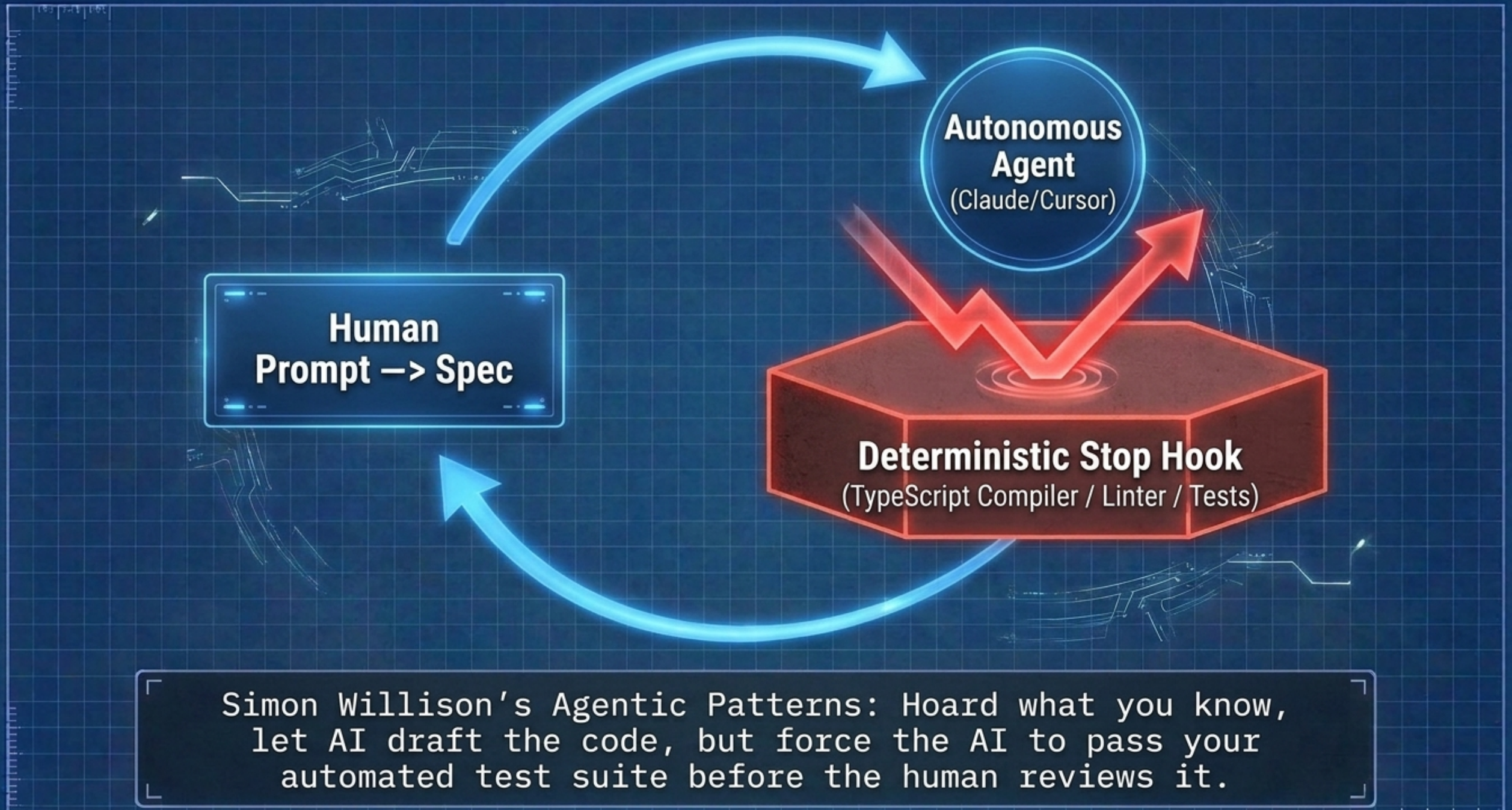


Trunk-Based



Eliminating the integration bottleneck. Developers collaborate on a single branch (trunk). CI/CD verifies that no one breaks the build. Changes are hidden via feature flags, not isolated in branches.

Gemini Notebook



Simon Willison’s Agentic Patterns: Hoard what you know, let AI draft the code, but force the AI to pass your automated test suite before the human reviews it.

Gemini Notebook



The Modern Operating Model Synthesized: Build the right thing (GDS) with the right people (Team Topologies), integrate continuously (Trunk), and generate code instantly but verify deterministically (AI-Native).

Gemini Notebook

Playbooks

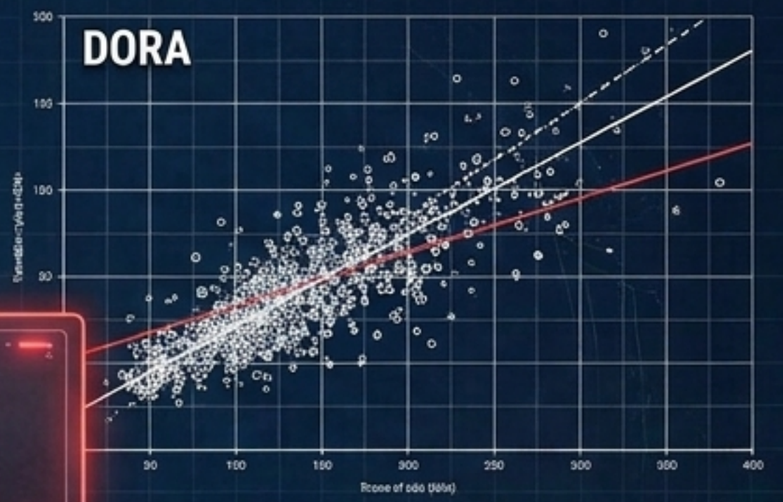
Scrum Guide

GitHub Spec-Kit

The Reality Check

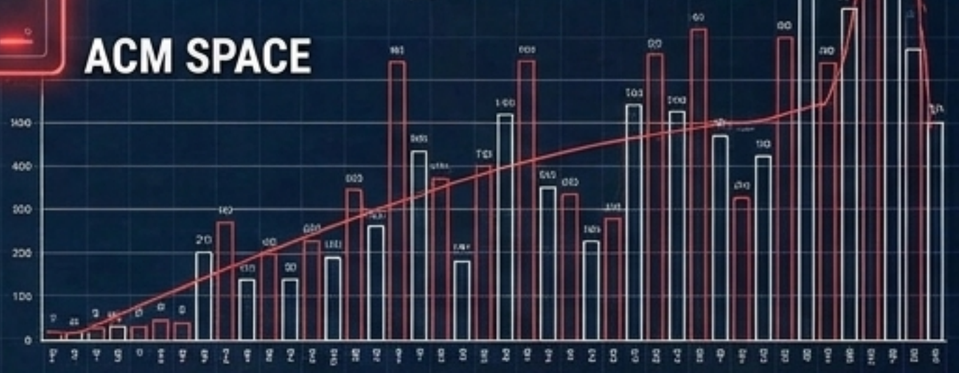
Empirical Data

DORA



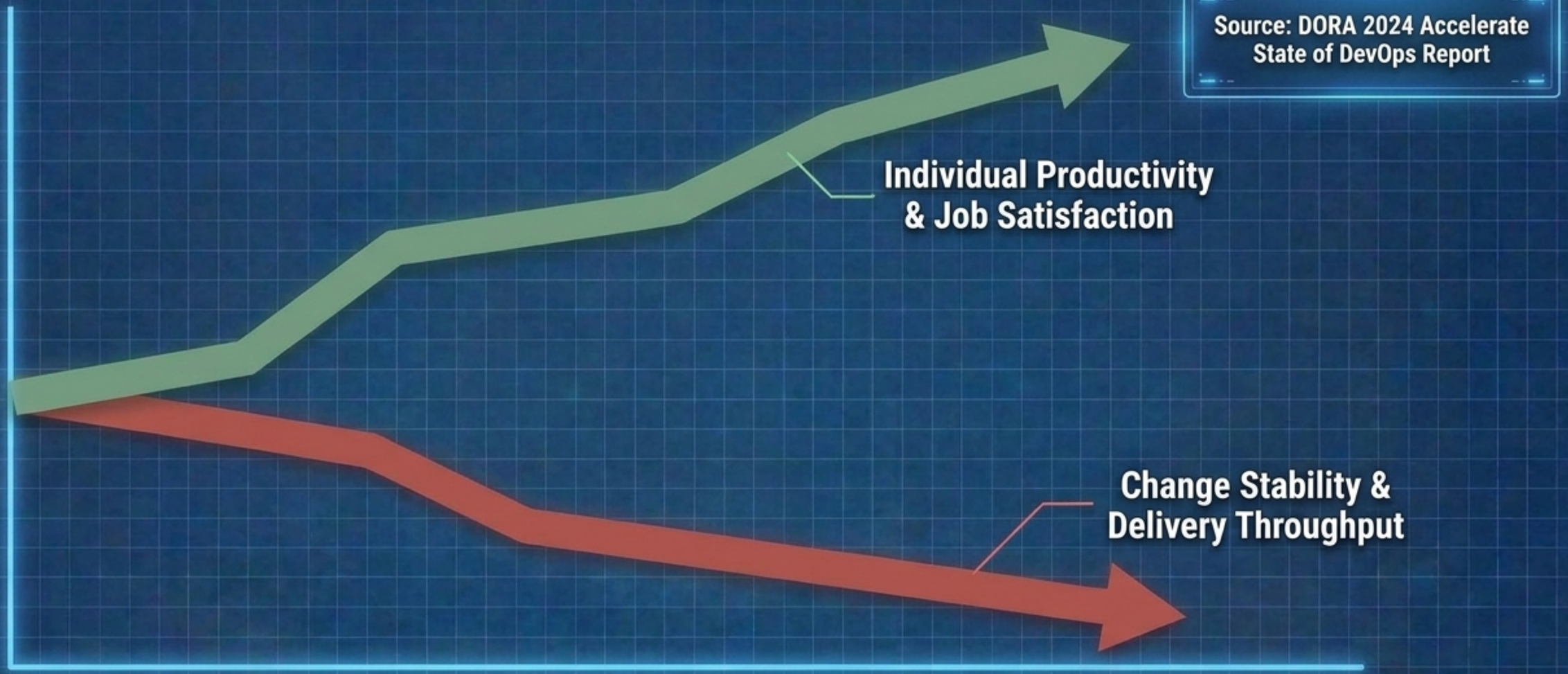
METR RCT

ACM SPACE



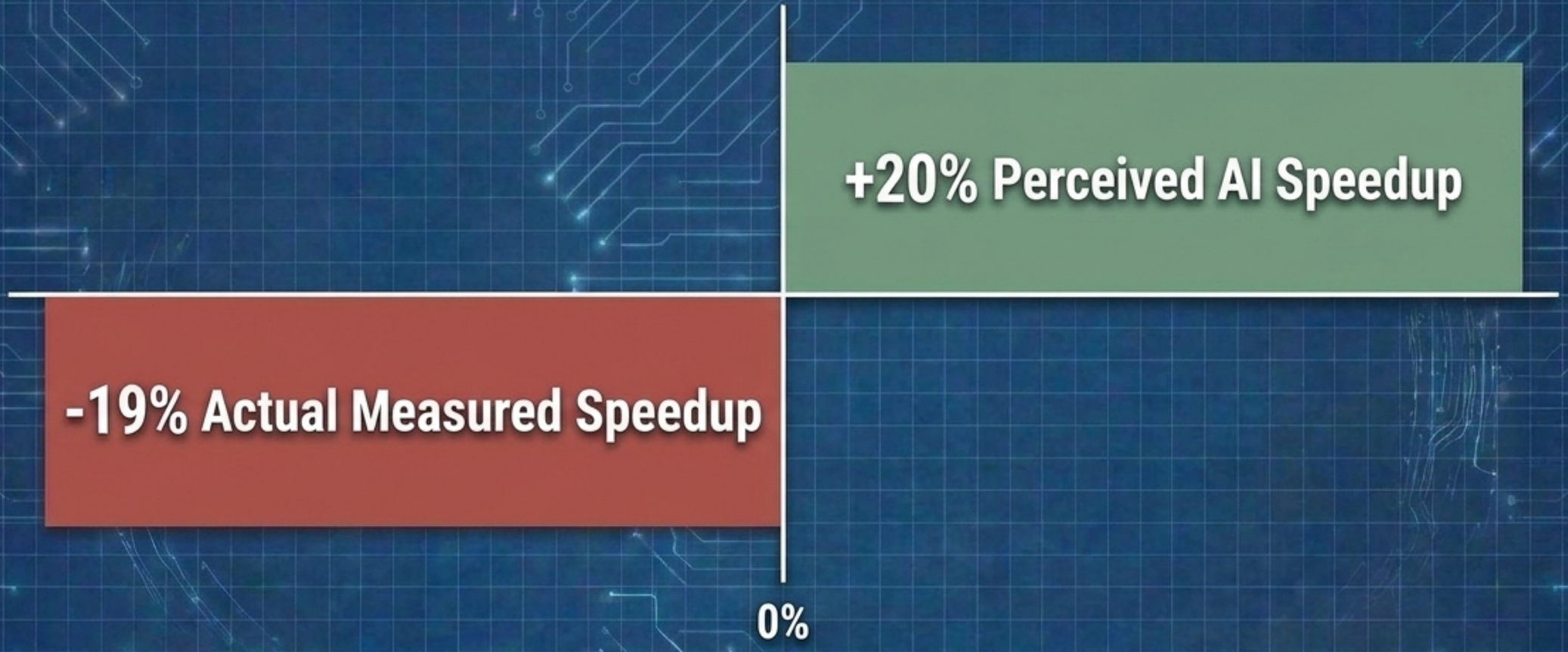
Frameworks tell us how things *should* work.
Empirical, peer-reviewed data tells us how things *actually* work.
We must balance opinionated toolkits with hard statistical realities.

Gemini Notebook



The AI Productivity Paradox: AI significantly increases individual flow, but is negatively impacting overall software delivery stability. The fundamentals—small batch sizes and test automation—are more critical than ever.

Gemini Notebook



Source: METR 2025 RCT (16 experienced OSS developers, 246 real-world issues).

A randomized controlled trial of experienced open-source developers showed that using early-2025 AI models actually made them 19% slower on complex tasks, despite believing they were 20% faster. AI excels at benchmarks but struggles with implicit repository context.

Gemini Notebook

Throughput

Lead Time for Changes

(Commit to Production)



Throughput

Deployment Frequency

(Time between deployments)



Stability

Change Fail Percentage

(Deployments requiring hotfixes)



Stability

Failed Deployment Recovery Time

(Time to restore service)



The Ultimate Barometer: Regardless of the era–Agile, DevOps, or AI–the DORA metrics remain the most reliable lagging indicators of performance. Speed and stability are not trade-offs; high performers achieve both.

Gemini Notebook

“Telephone Game”



Original Intent
(Agile Manifesto / XP)

Corporate
Implementation

Martin Fowler’s ‘Semantic Diffusion’: When a good methodology is adopted by the masses, its definition weakens until it means the exact opposite of its intent (e.g., ‘Agile’ becoming a rigid phase-gate process).

Gemini Notebook

⚠ Phase-Gate Theatre



Calling milestones ‘Sprints’ but still doing Waterfall handoffs.

⚠ Flaccid Scrum



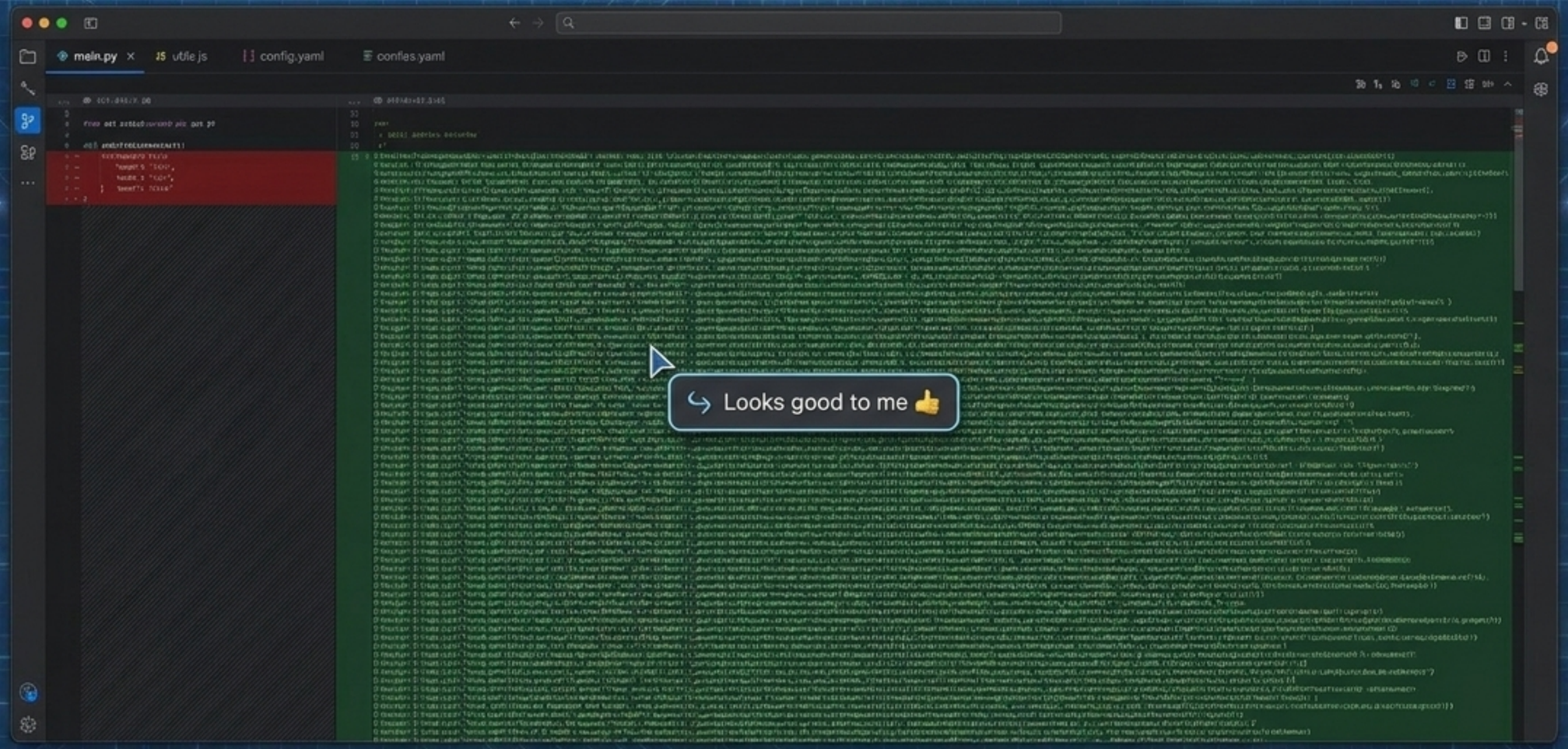
Adopting agile ceremonies (stand-ups, backlogs) without the underlying XP technical discipline (TDD, refactoring, CI).

⚠ Faux-Agile KPIs



Using story points to measure human productivity instead of managing flow.

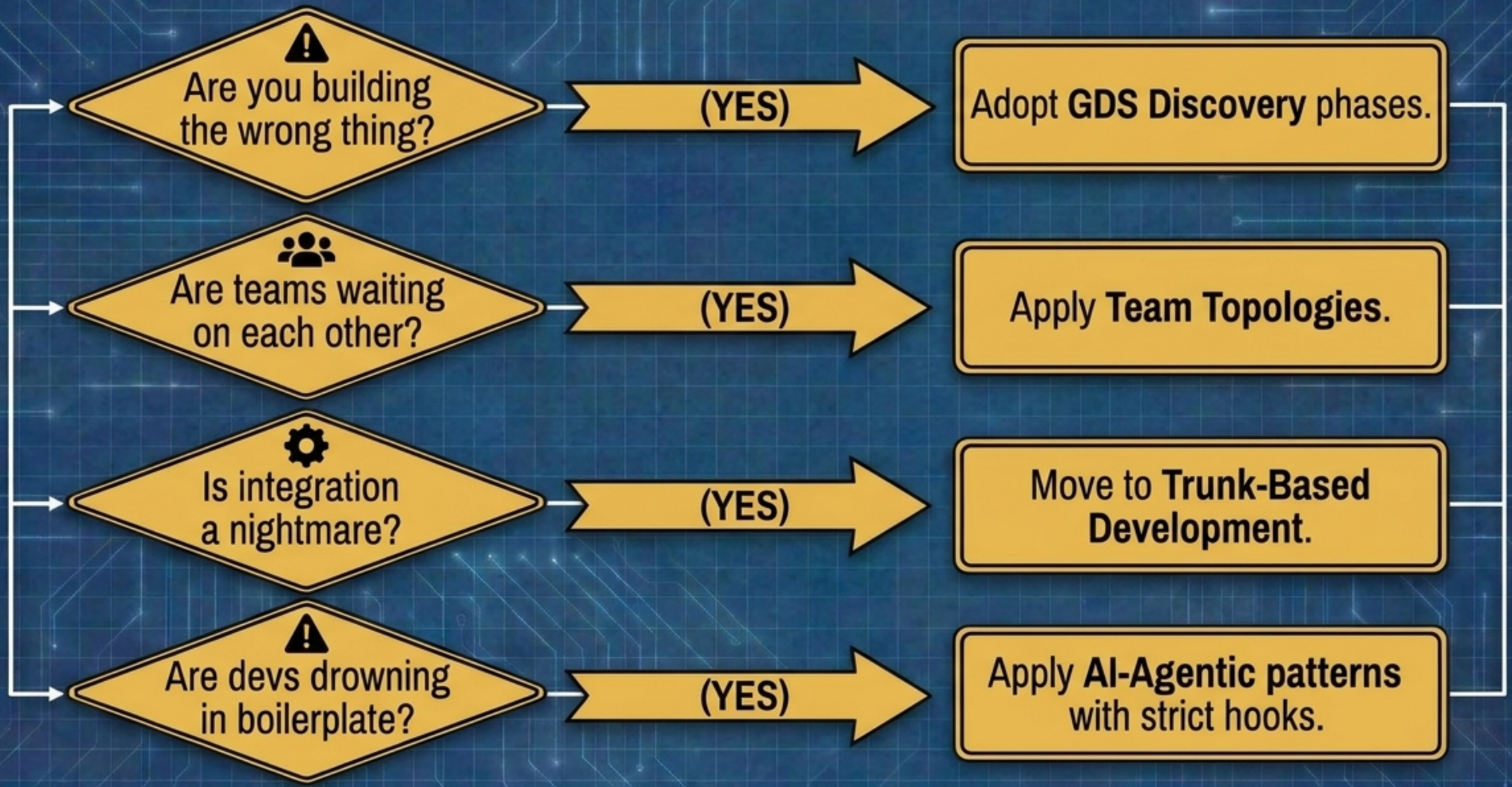
Gemini Notebook



The AI-Native Anti-Pattern: Inflicting unreviewed AI code dumps on collaborators. Because code generation is cheap, the cost of reviewing bloated code shifts to teammates. Solution: Agentic manual testing and strict spec-driven verification.

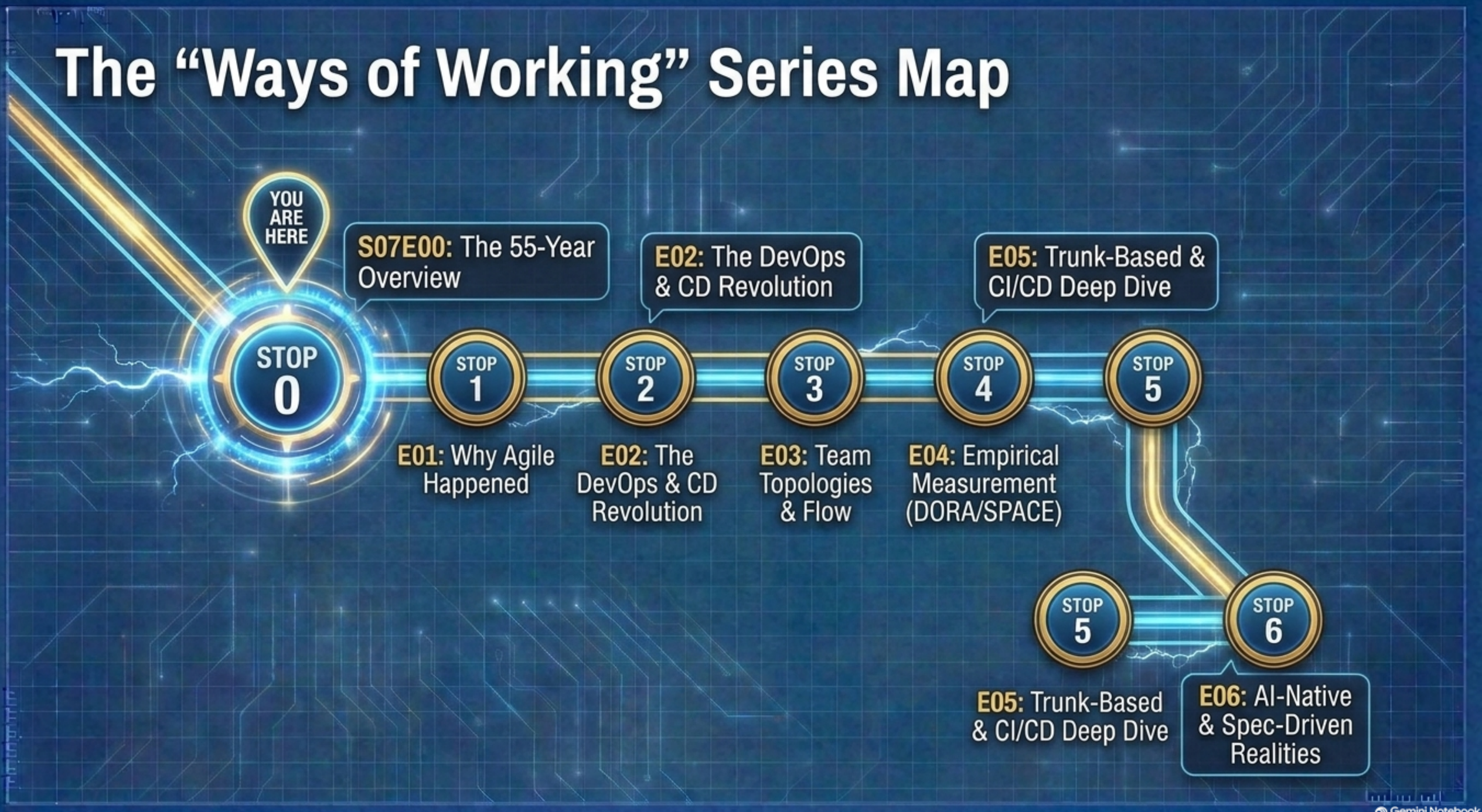
Gemini Notebook

Audit Your Team's Bottleneck Today

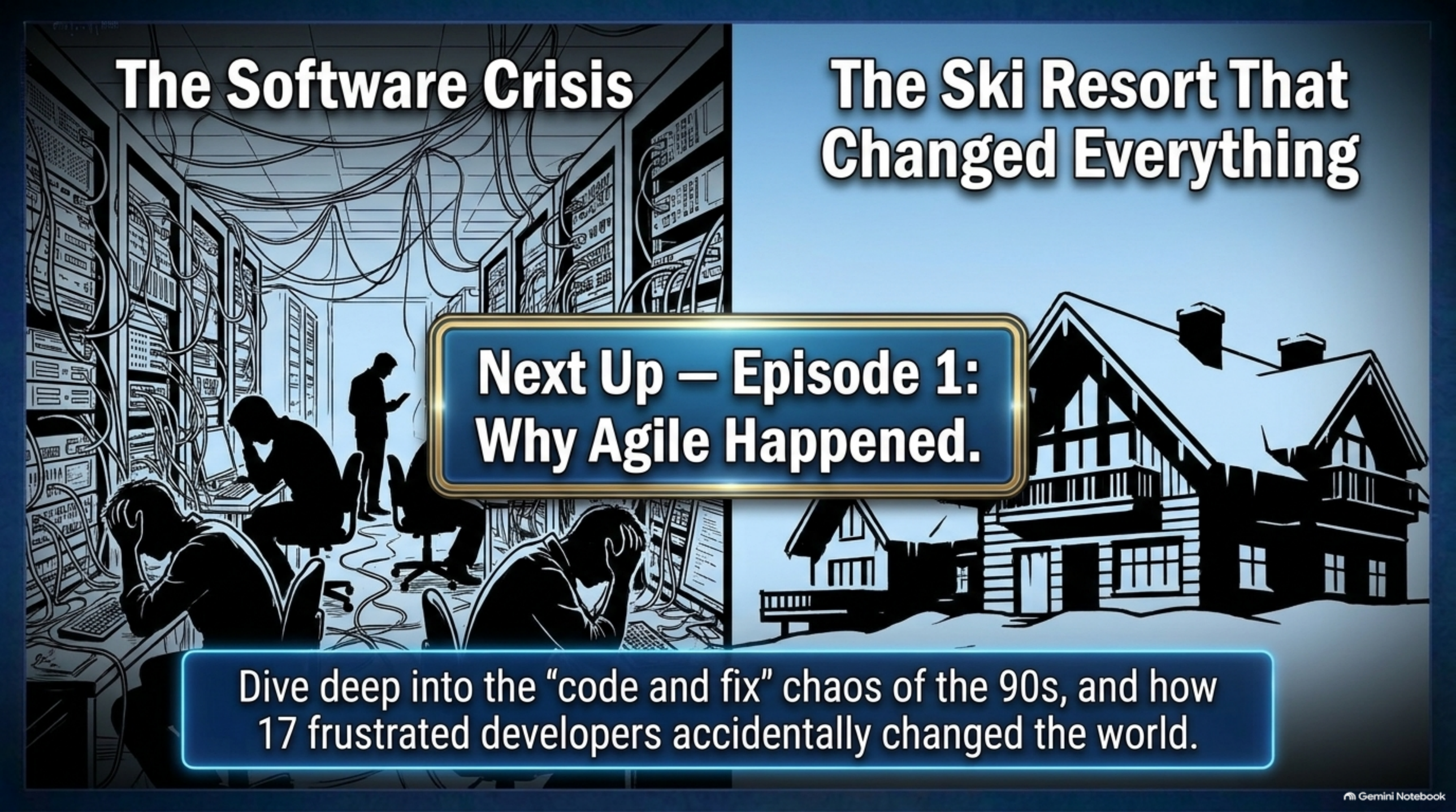


Gemini Notebook

The “Ways of Working” Series Map



Gemini Notebook



Gemini Notebook