Title

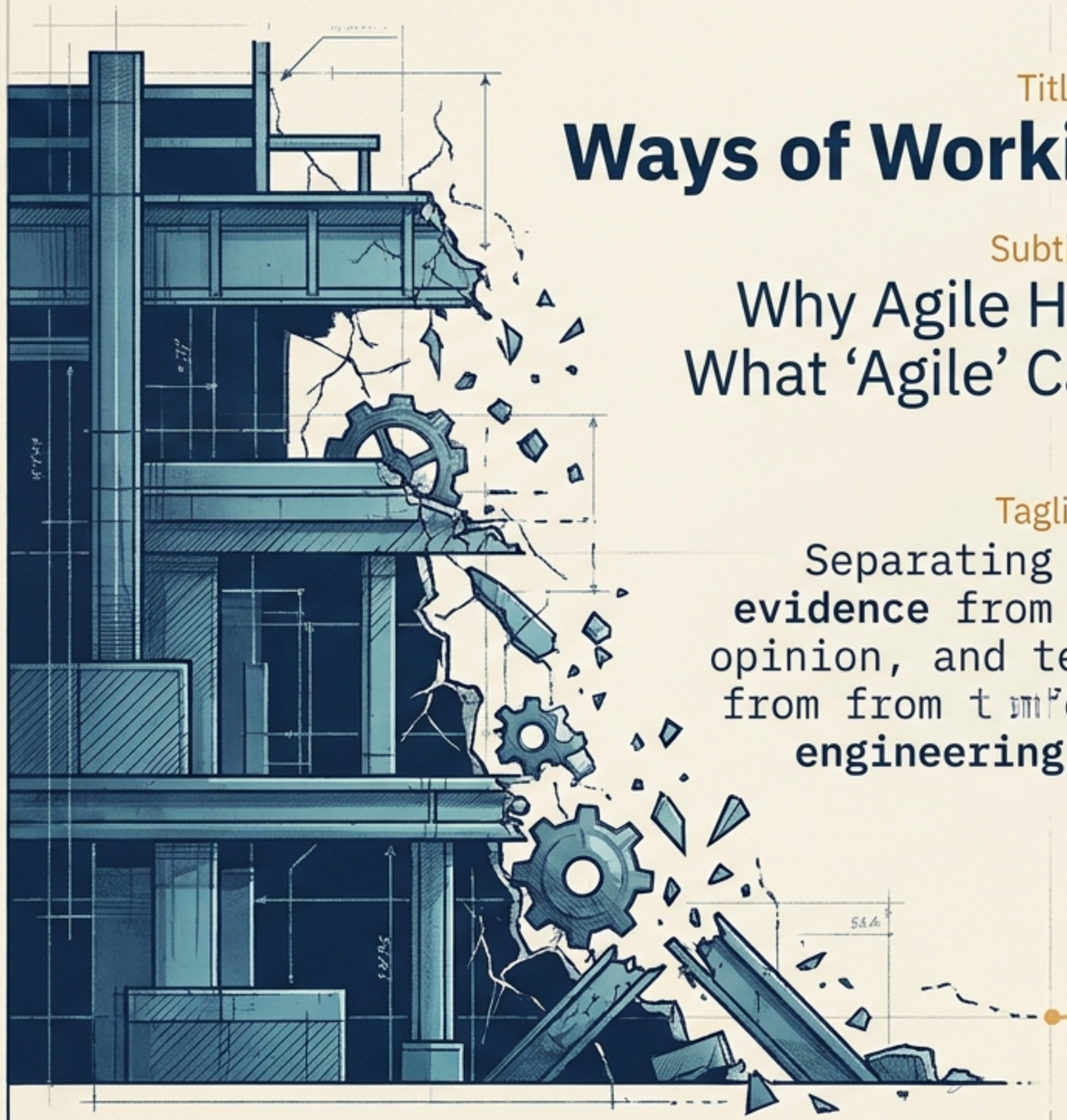
Ways of Working — S07E01

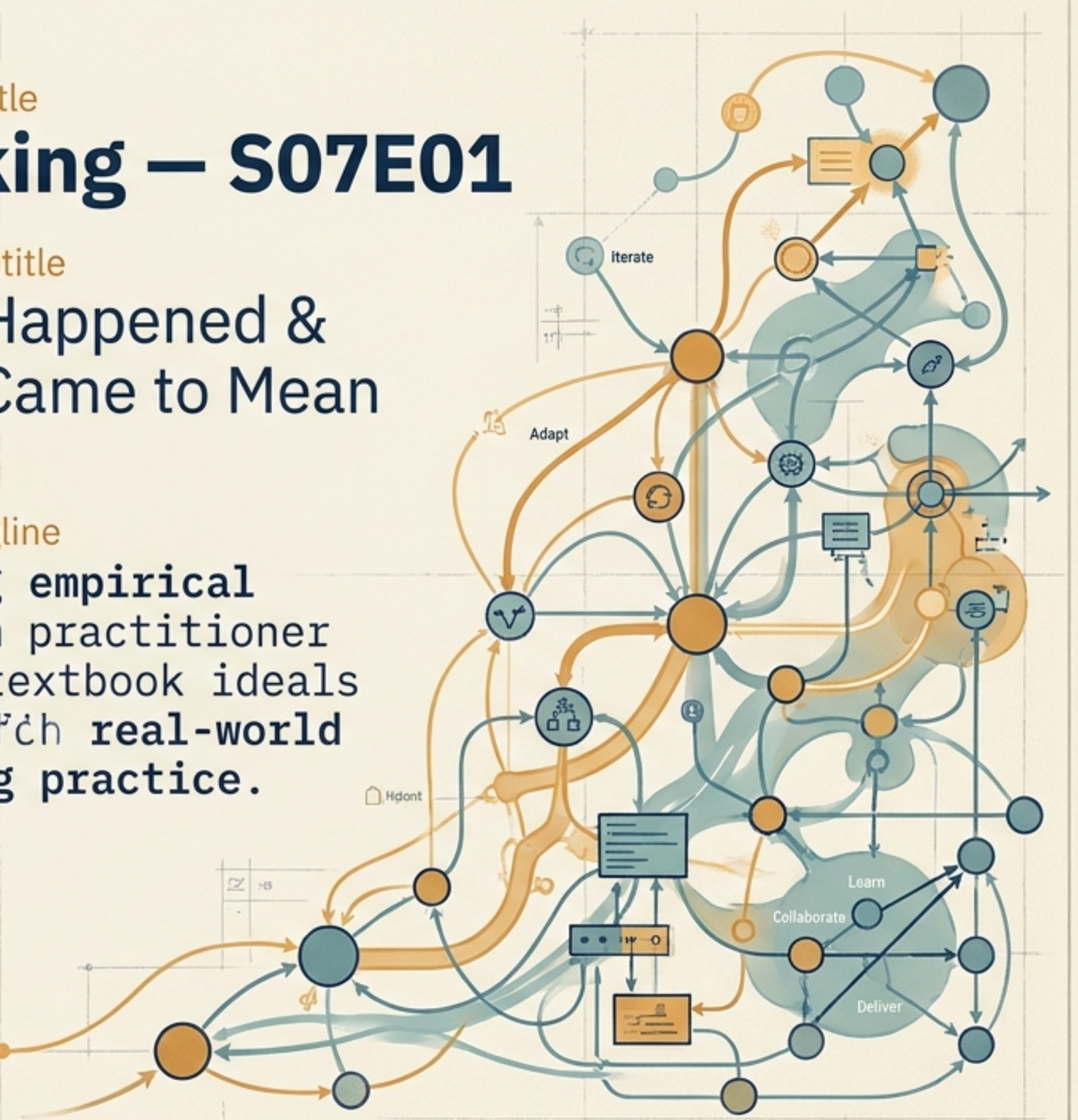
Subtitle

Why Agile Happened & What ‘Agile’ Came to Mean

Tagline

Separating **empirical evidence** from practitioner opinion, and textbook ideals from from t ɯ ʃ ch **real-world engineering practice.**

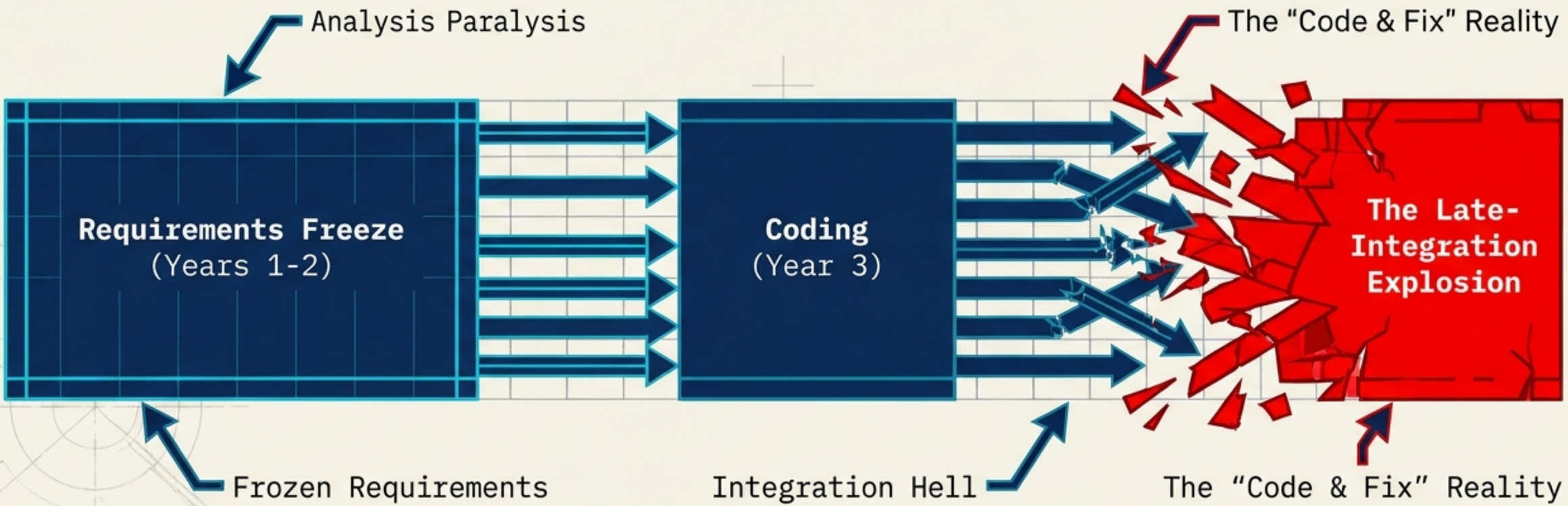




Gemini Notebook

The Pre-2001 Delivery Crisis: Planning to Fail

Recap (S07E00): 1990s software development was defined by massive upfront planning and multi-year requirements freezes, inevitably leading to "code and fix" chaos when rigid plans collided with reality.

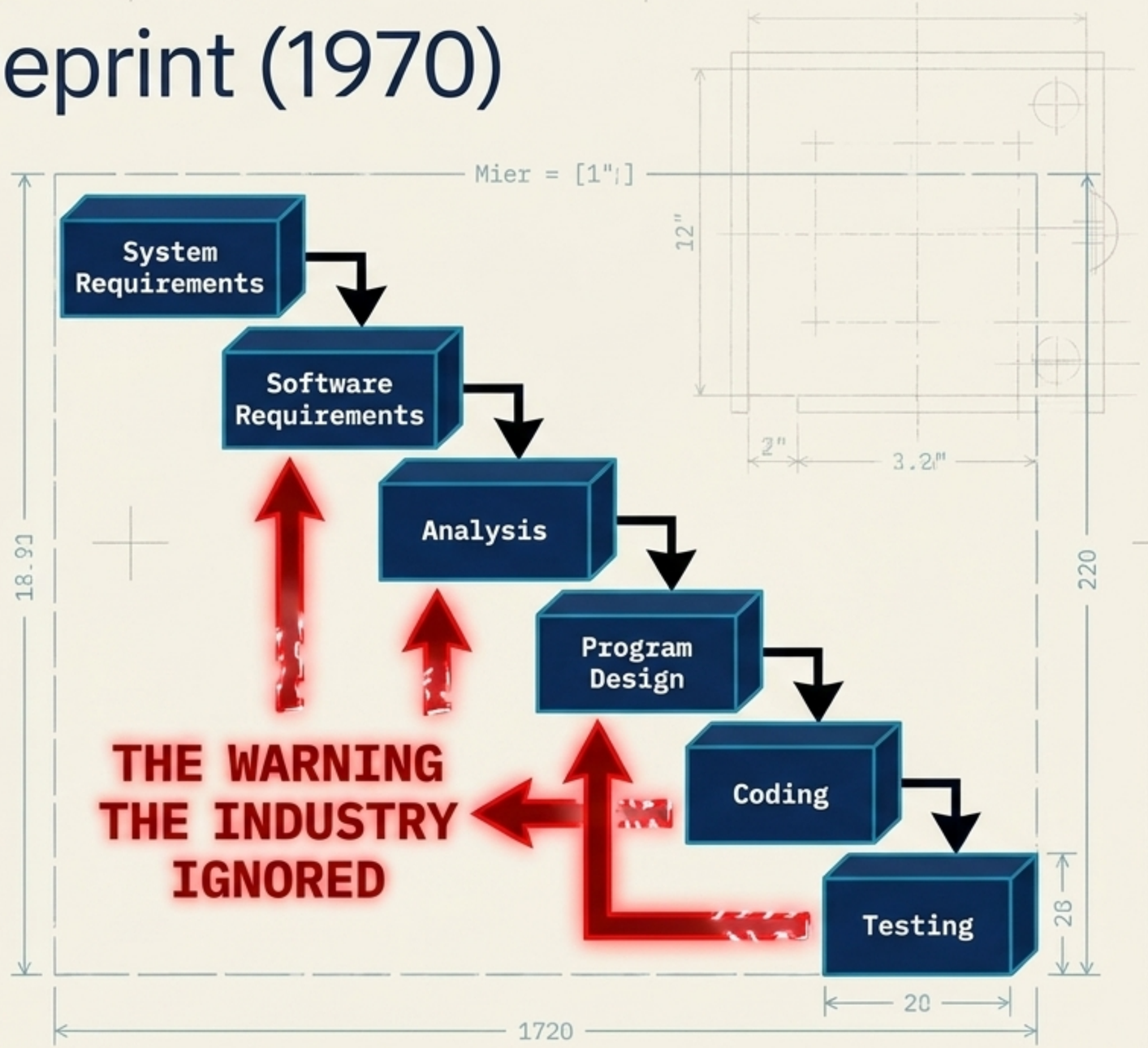


Gemini Notebook

Royce’s Misread Blueprint (1970)

The term "Waterfall" was never used in Royce’s paper. While he introduced the activity-phased cascade, he explicitly warned that long backtracks (e.g., from Testing to Design) were fatal.

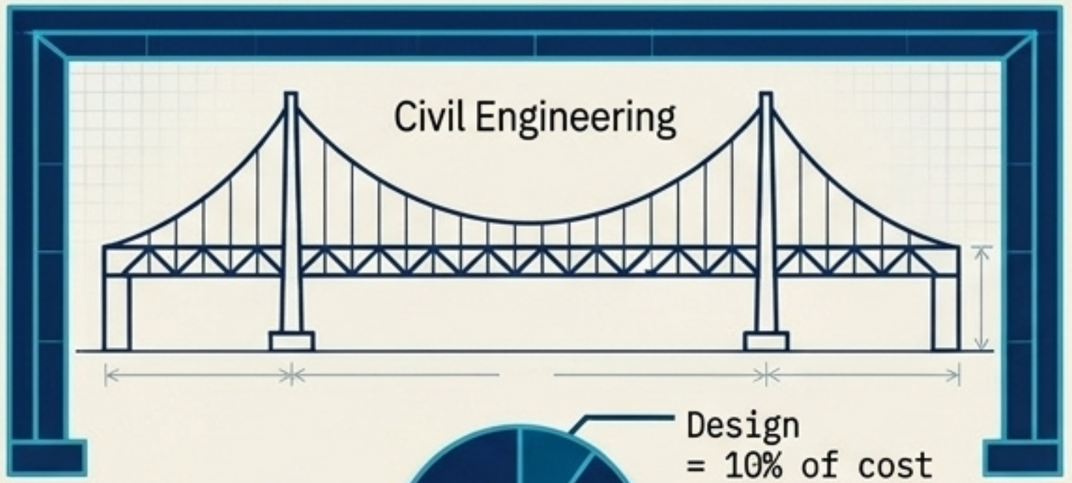
The industry adopted the rigid shape of his diagram but entirely ignored his caveats about iterative feedback and the danger of **strictly separating design from construction.**

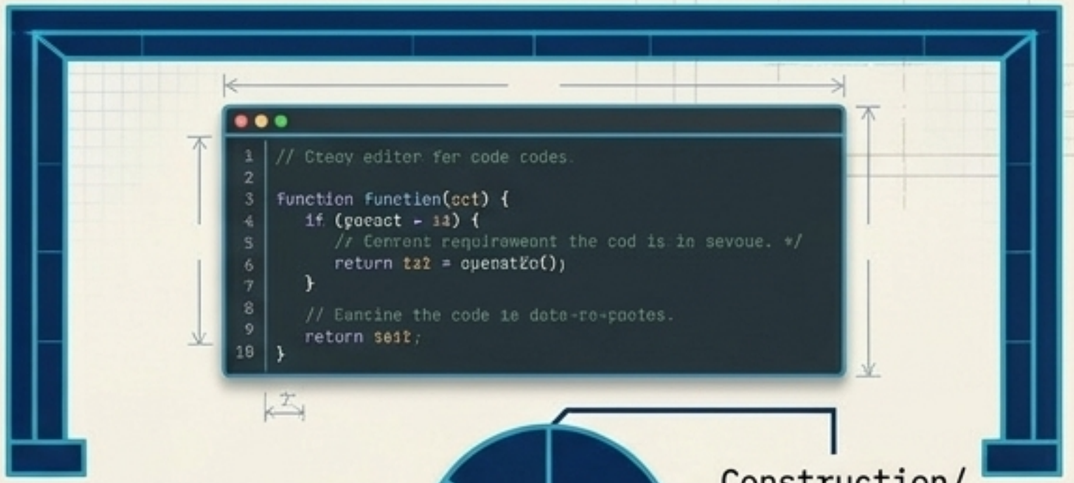


Gemini Notebook

The Flawed Civil Engineering Metaphor

Martin Fowler's The New Methodology: Predictive planning fails because it assumes coding is "construction." In software, compiling is free construction; writing code is the design process.





The source code is the design document.

Takeaway: Creative design processes are fundamentally unpredictable. We cannot manage software using a metaphor built for pouring concrete.

Gemini Notebook

Precursors to Snowbird: The Iterative Underground



Barry Boehm’s Spiral Model introduced the concept of validating risky assumptions early through iterative cycles rather than assuming the plan is flawless.

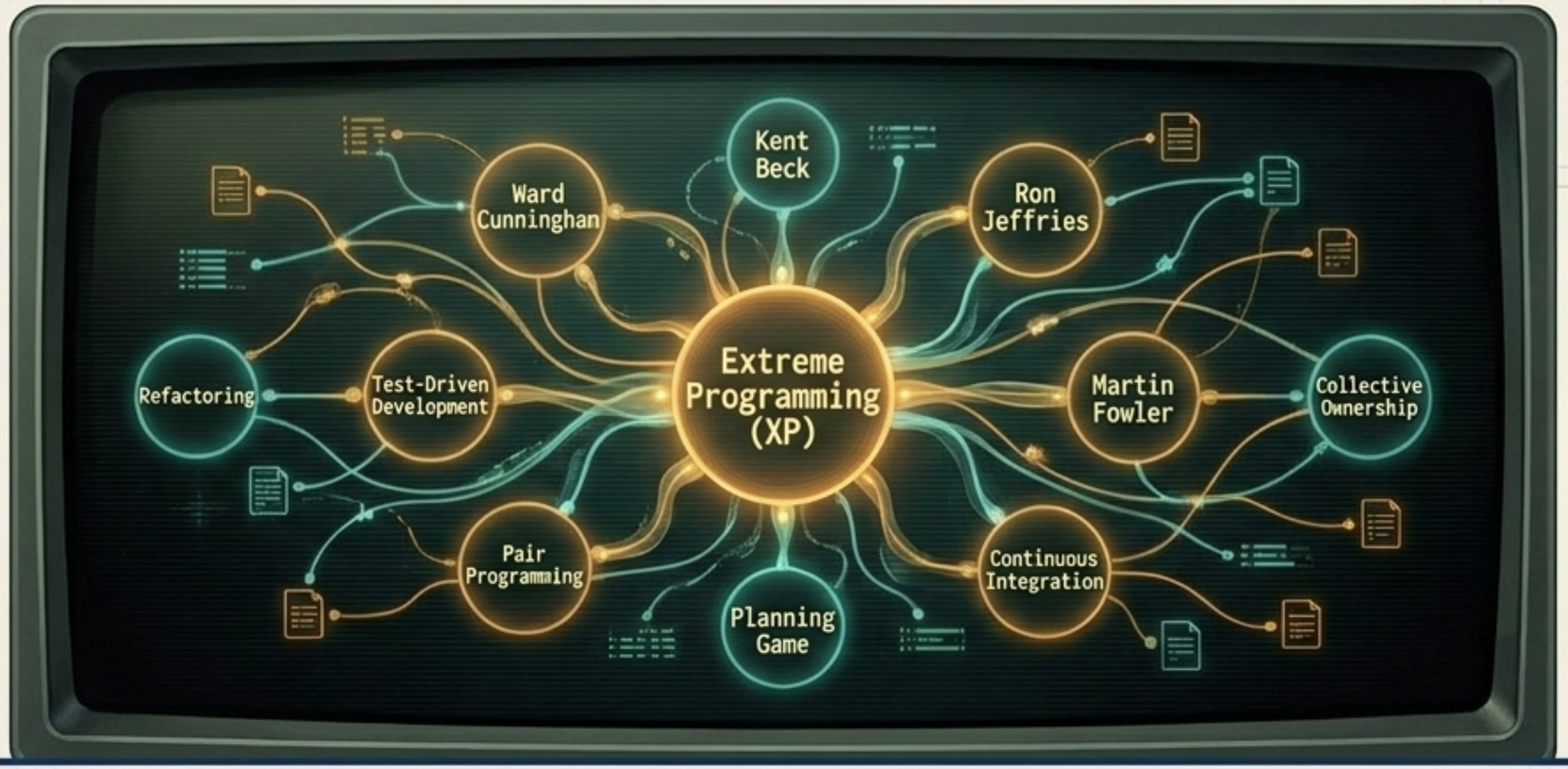
Fred Brooks argued software must be grown organically rather than built predictively, adapting to its environment.



Gemini Notebook

The C2 Wiki Era & The Birth of XP (Late 1990s)

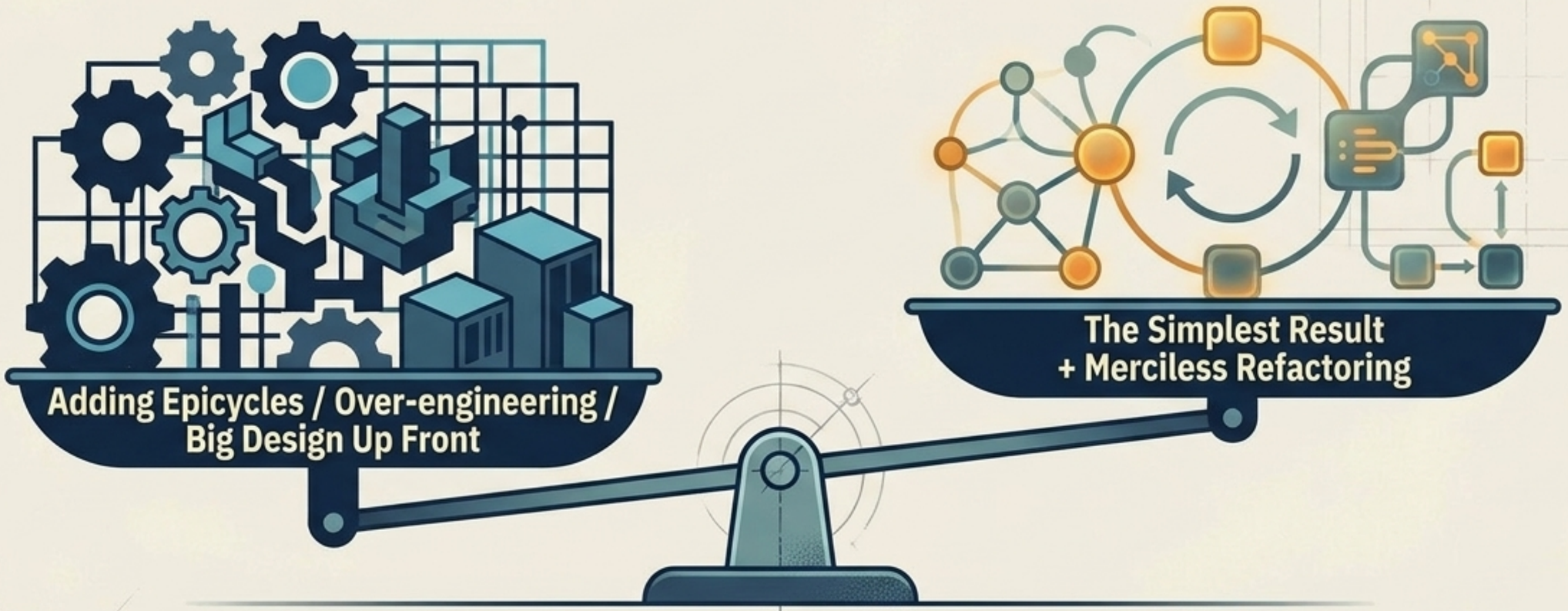
Before “Agile” became a corporate buzzword, a tight-knit community of object-oriented pioneers (Kent Beck, Ward Cunningham) used the first wiki to codify practices that openly defied Taylorist management.



“Alistair Cockburn observed that software’s most crucial factor is its people, who must be treated as **‘non-linear, first-order components,’** not **‘plug-compatible programming units.’**”

Gemini Notebook

DTSTTCPW & YAGNI: Discipline, Not Hacking



Do The Simplest Thing That Could Possibly Work. Kent Beck's approach to getting code on the screen for immediate evaluation. It demands creating the simplest result today, backed by the safety net of merciless refactoring.

You Aren't Gonna Need It. Rejecting presumptive features that bloat software. Do not build for a **hypothetical future;** keep the cost of change low so you can adapt when the future actually arrives.

Gemini Notebook

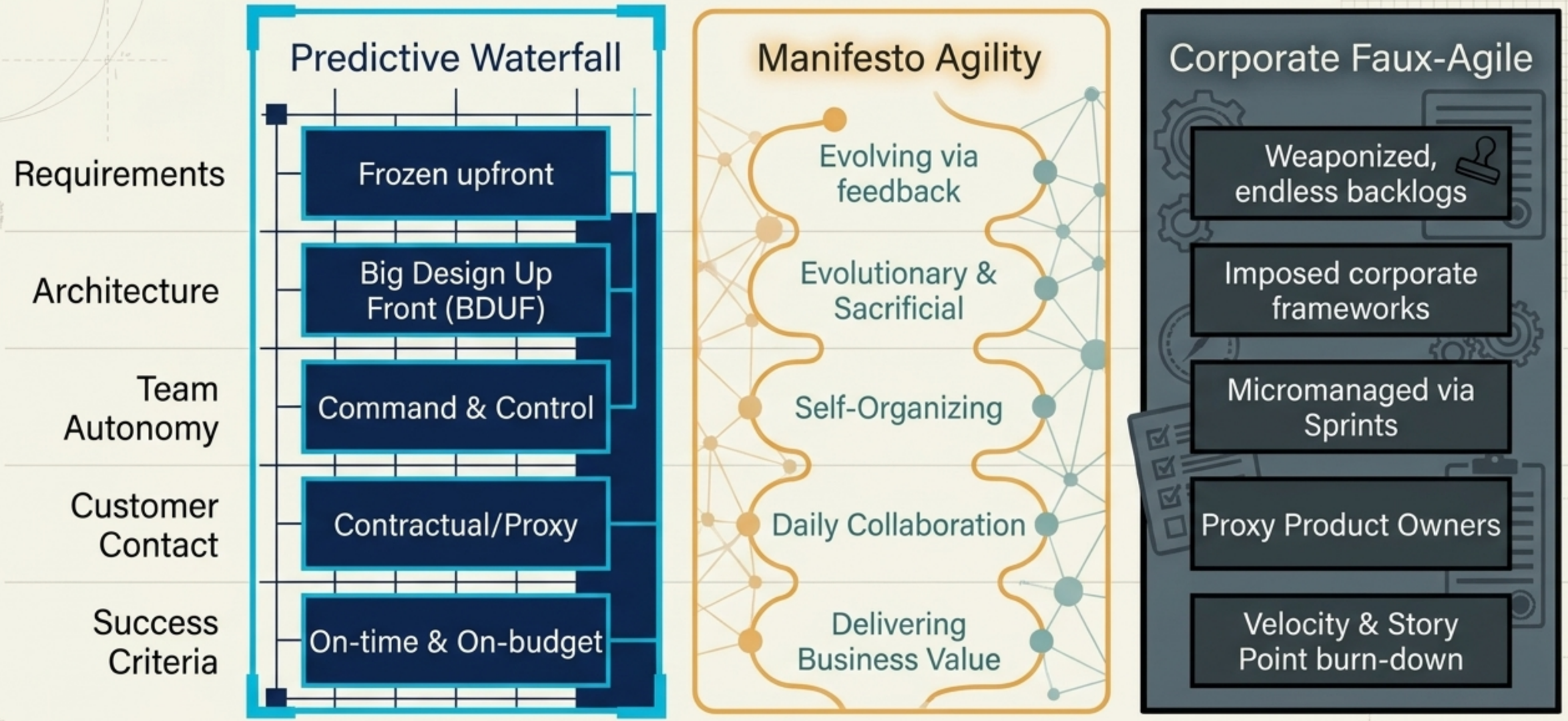
The 2001 Snowbird Consensus

17 organizational anarchists—representing Scrum, XP, DSDM, and Crystal—found common ground against Dilbertesque corporate make-work.

Individuals and Interactions	**OVER**	Processes and Tools
Working Software	**OVER**	Comprehensive Documentation
Customer Collaboration	**OVER**	Contract Negotiation
Responding to Change	**OVER**	Following a Plan

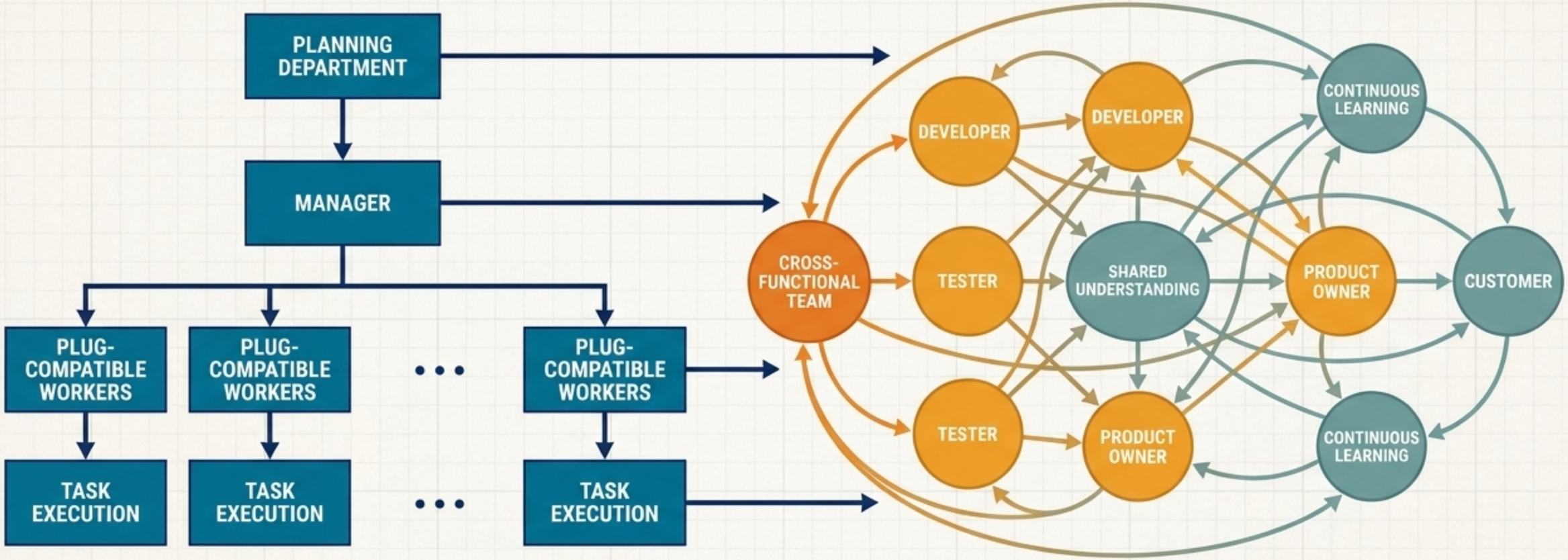
Gemini Notebook

Ways of Working: The Diagnostic Matrix



Gemini Notebook

People as Non-Linear Components

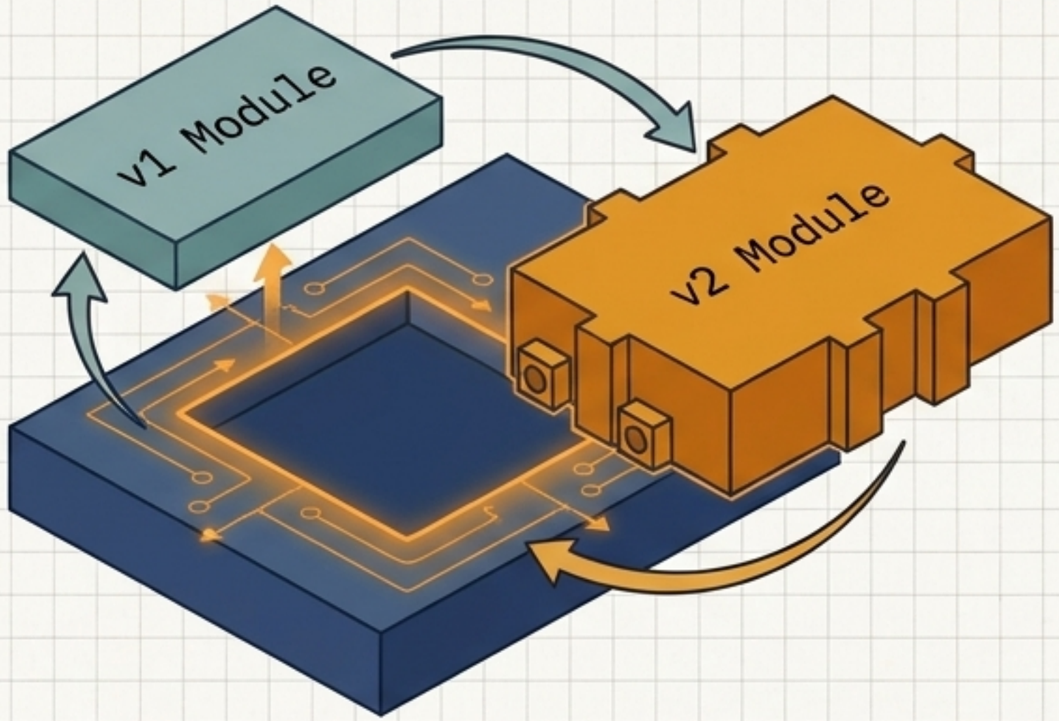


Taylorist management (Scientific Management) assumes a separate planning department can dictate tasks to "plug-compatible" workers.

Agile methods assert that no static process can make up for the skill of the team. The process must adapt to support the team, not force the team to serve the process. Imposing a rigid Agile framework from the outside fundamentally violates this principle.

Gemini Notebook

Sacrificial Architecture (Martin Fowler)



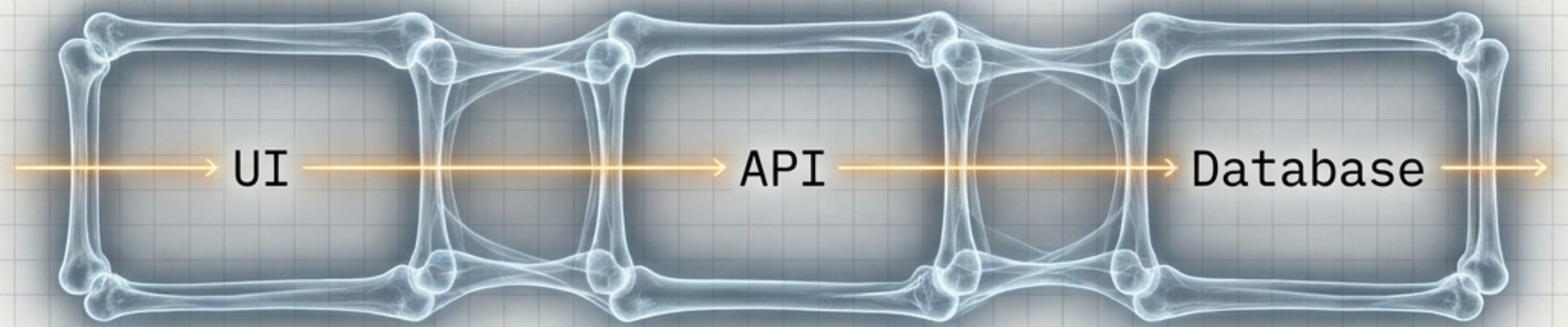
Good engineering means accepting that early code will be thrown away.

“Design for ~10X growth, but plan to rewrite before ~100X.”

Caveat: This is not an excuse for low internal quality. High modularity and clean boundaries are strictly required so you can safely sacrifice and replace modules as the system scales without breaking the entire application.

Gemini Notebook

Validating Risk: The Walking Skeleton



Concept (Alistair Cockburn):

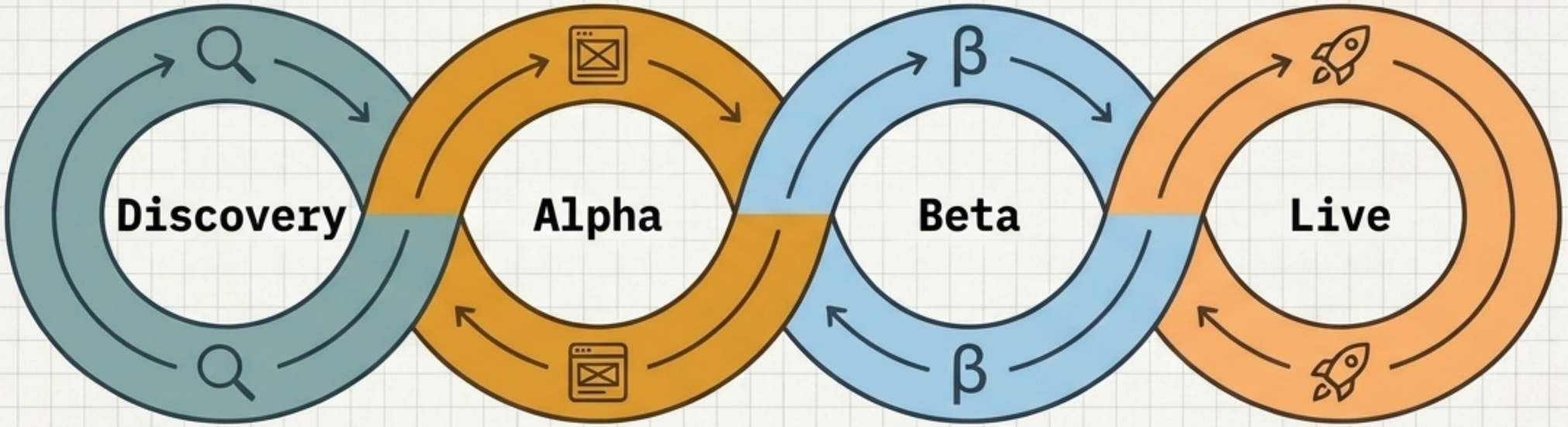
A tiny implementation of the system that performs a small, end-to-end function.

Why it works:

It validates that the architecture, deployment pipeline, and integration points actually function in reality before feature bloat sets in, eliminating the risk of late-integration explosions.

Gemini Notebook

Public-Sector User-Research Loops

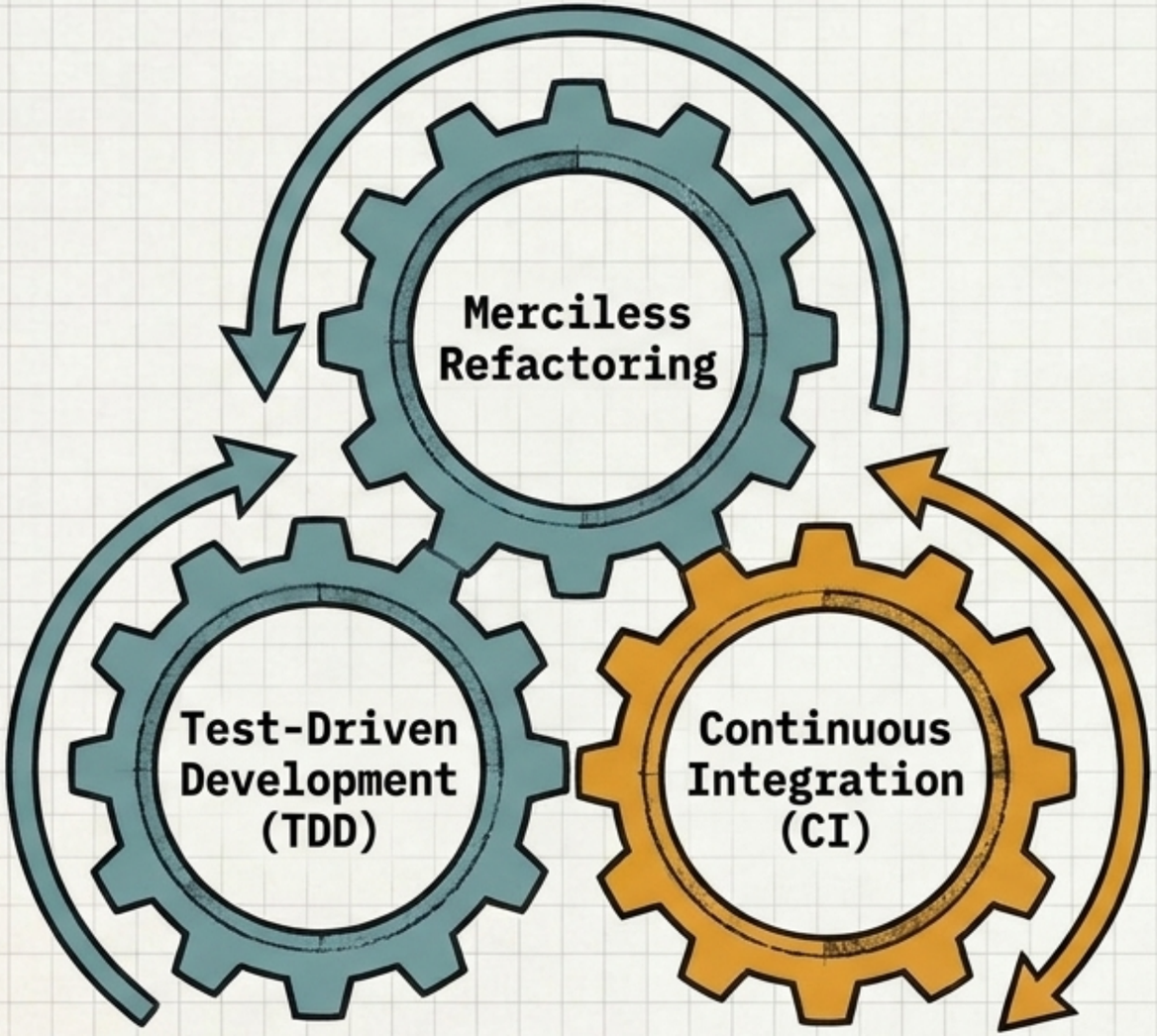


High-stakes government tech has abandoned multi-year, predictive procurement cycles.

The Playbook:
- Default to open.
- Deploy walking skeletons early.
- Rely on continuous user validation and rapid research loops rather than upfront, fixed-bid contracts that guarantee failure.

Gemini Notebook

Technical Excellence as the Enabler



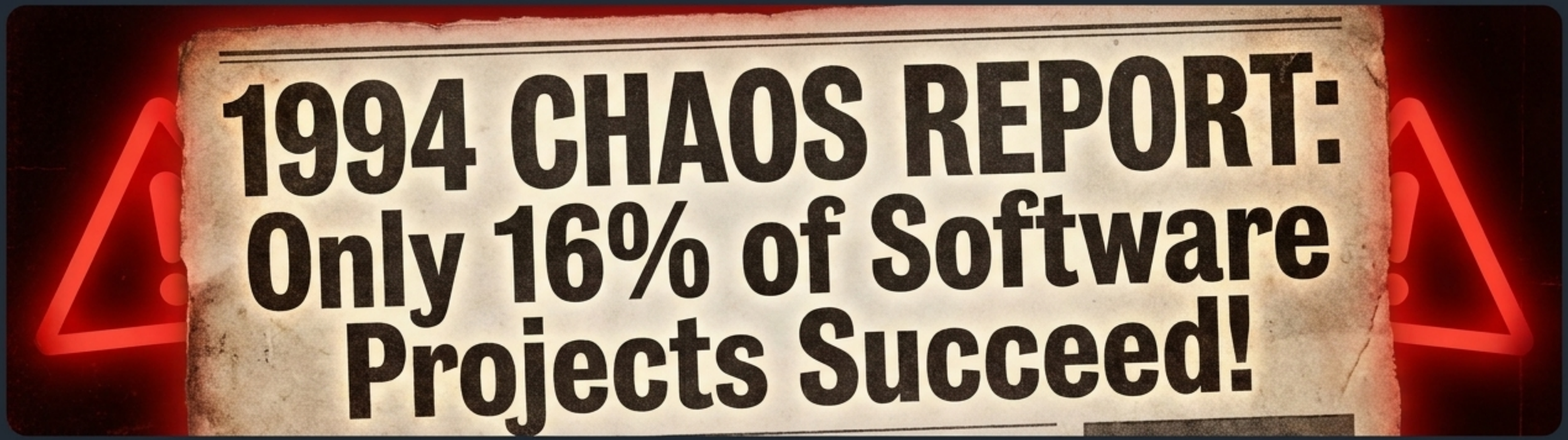
Agile project management without agile engineering practices is a recipe for disaster.

Synthesis (Accelerate data):

The ability to pivot and embrace change (Agility) depends entirely on technical practices that keep the cost of change low.
Releasing many times a day and maintaining low defect rates are not mutually exclusive—they reinforce each other.

Gemini Notebook

Fact-Checking Software Failure Lore



The Standish Legacy:

The original 1994 Standish Group figures (16% success, 53% challenged, 31% failed) spawned an industry of fear.

The Impact:

These figures are continuously cited to prove software is in a perpetual “crisis,” driving panicked organizations toward heavier, more bureaucratic management frameworks to regain “control.”

Gemini Notebook

Unpacking the Standish Methodology



The Flaw (Eveleens & Verhoef, IEEE Software 2010):

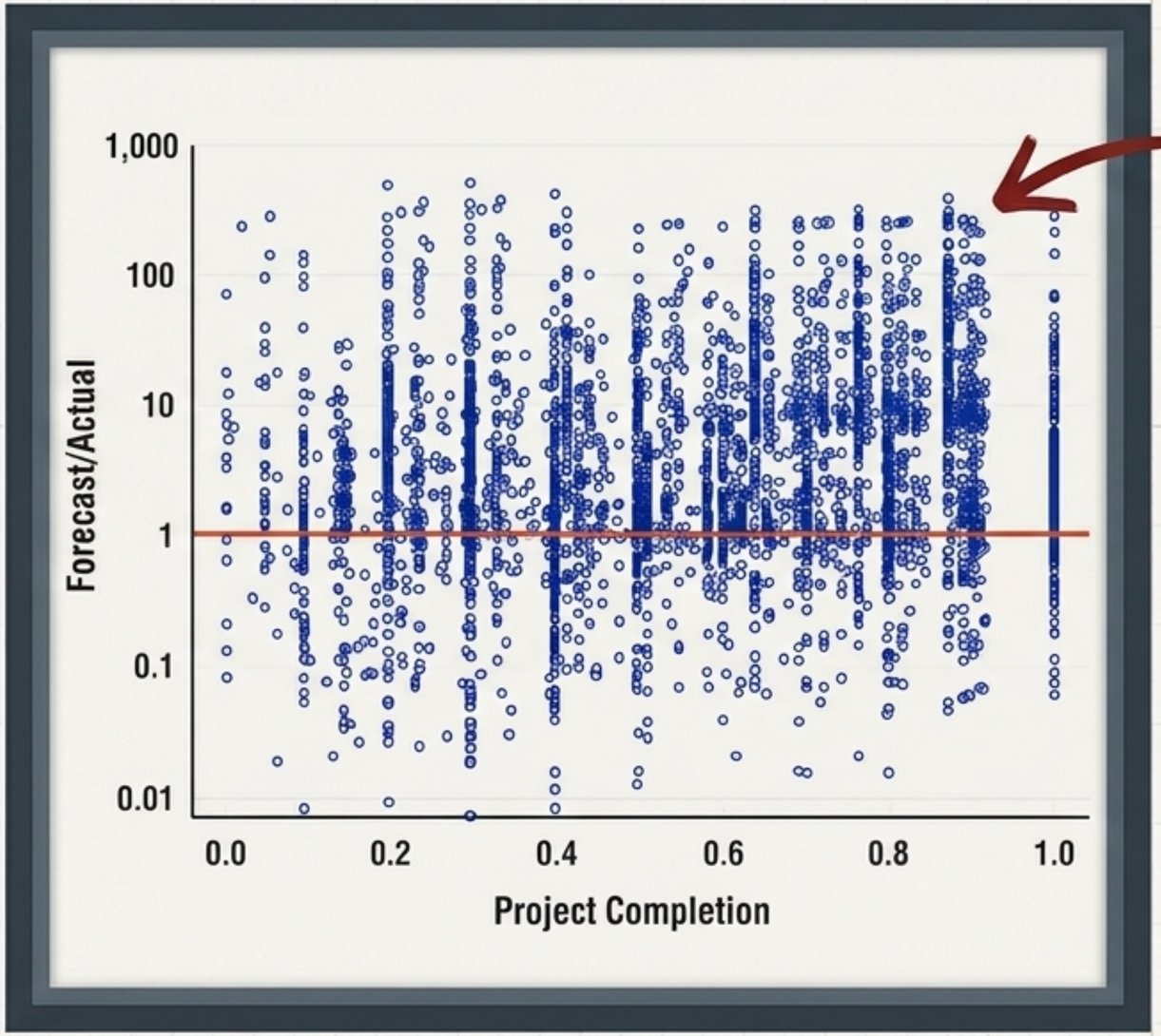
Standish's definition of "success" (Resolution Type 1) is based purely on initial estimation accuracy.

The Absurd Result:

A project delivered early, under budget, and generating massive revenue, but omitting 10% of features because users didn't actually need them, is officially classified as a "Challenged" or failed project by Standish.

Gemini Notebook

The Perversion of Estimation

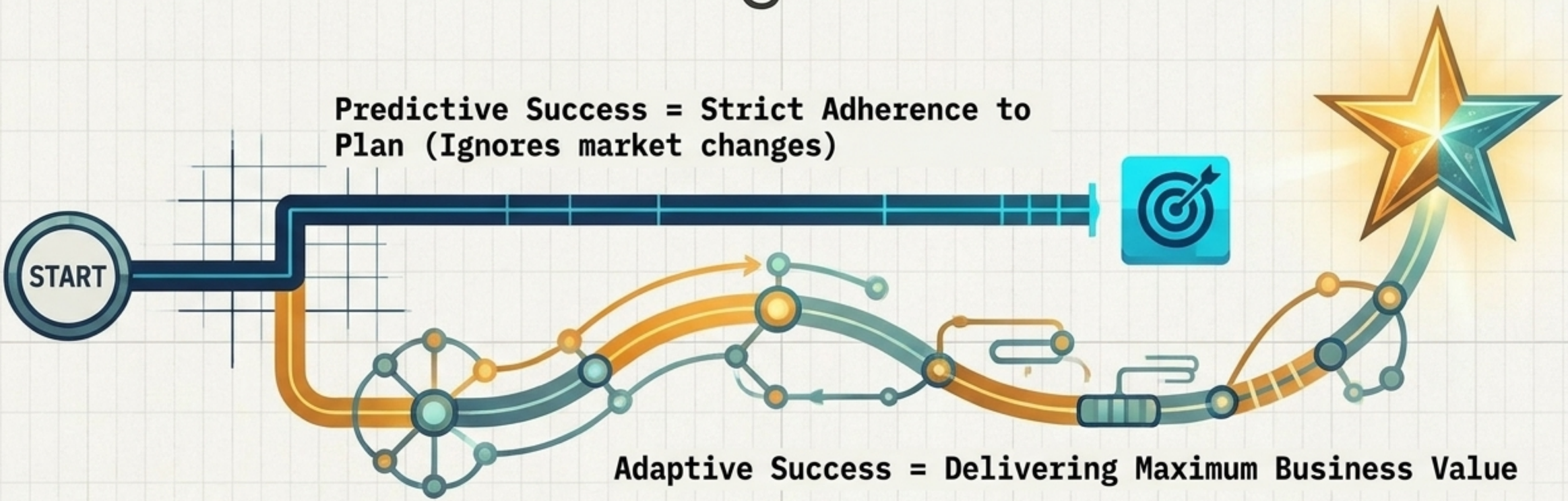


The Data: Eveleens & Verhoef audited 3,767 forecasts (Org X). The scatter plot reveals massive, systemic overestimation.

The Cause: When organizations measure success by Standish definitions (hitting the exact plan), they pervert the estimation practice. Project managers drastically inflate budgets and timelines (sandb ‘Successful’ label, resulting in bloated, inefficient projects and meaningless metrics.

Gemini Notebook

Rethinking “Success”



Synthesis:
The Chaos Report measures predictive accuracy, not software quality or user satisfaction.

The Fowler Principle: A successful predictive project hits its exact plan. A successful agile project builds something different and better than the original plan foresaw by adapting to user feedback.

Gemini Notebook

Semantic Diffusion & Inversion



Semantic Diffusion (Fowler): The process where a precise methodology loses its original meaning as it spreads, becoming a fashionable, empty buzzword.

Semantic Inversion: When a term degrades so much it means the exact opposite of its intent. (e.g., “DevOps” becoming the name of a siloed operations team, or “MVP” being used to describe a $12M, two-year first release).

Gemini Notebook

The Agile Industrial Complex



The Myth:
The **Certification-Competence Correlation**. Certifications became a lucrative industry functioning as corporate entry gates.

The Reality: They prioritize a 2-day classroom attendance over actual software engineering competence. This **industrial complex stripped teams of the self-determination** and **technical excellence** at the very heart of the Snowbird manifesto.

Gemini Notebook

Agile Imposition & “Dark Scrum”



The Problem (Ron Jeffries): When “Agile” is imposed from the top-down without engineering support or team autonomy, it degrades into “**Dark Scrum**.”

The Result: Management **weaponizes** agile rituals (sprints, standups, velocity) to micromanage developers, resulting in more interference, higher pressure, zero time for refactoring, and ultimately, **lower quality** software.

Gemini Notebook

Agile is Dead (Long Live Agility)

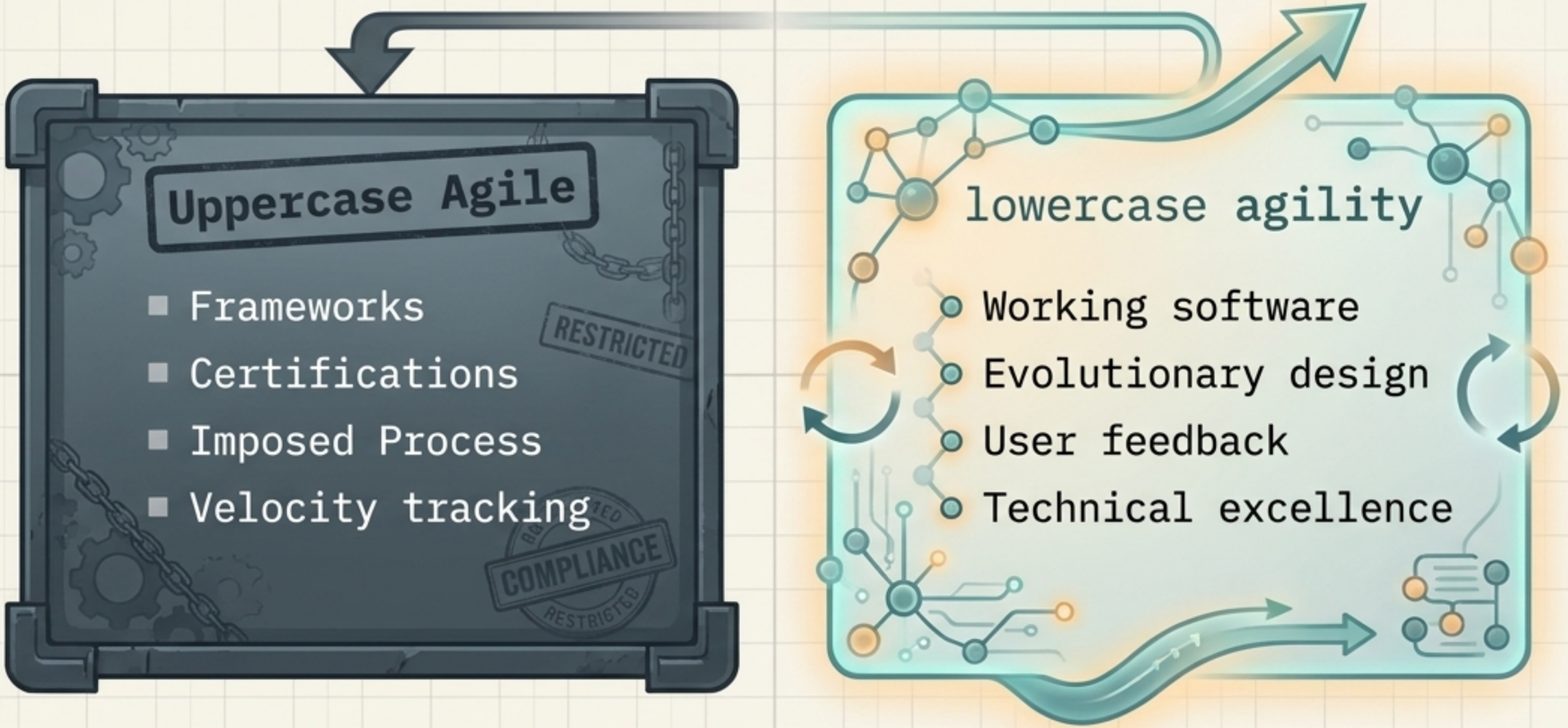


The Pushback (Dave Thomas): “Agile” as a noun has become a meaningless brand used to hawk consulting services and tools.

The Solution: Developers must abandon adherence to named frameworks. Return to the verbs and adjectives: **developing with agility**—tight feedback loops, continuously adjusting understanding, and making future change easier.

Gemini Notebook

Uppercase ‘Agile’ vs. lowercase ‘agility’



Ultimately, what we do trumps what we call it.
The **Core Loop**: Find out where you are -> Take a small step -> Adjust based on learning -> Repeat.
True agility is rooted in engineering practices (TDD, CI, Refactoring), not management rituals.

Gemini Notebook

Diagnostic Red Flags for Faux-Agile Teams

- [ ] The process is imposed from the outside; the team has no authority to alter how they work.



- [ ] Success is measured by “velocity” or strict plan-adherence, rather than delivering working software to users.



- [ ] Zero capacity is allocated for technical excellence (merciless refactoring, paying down tech debt).

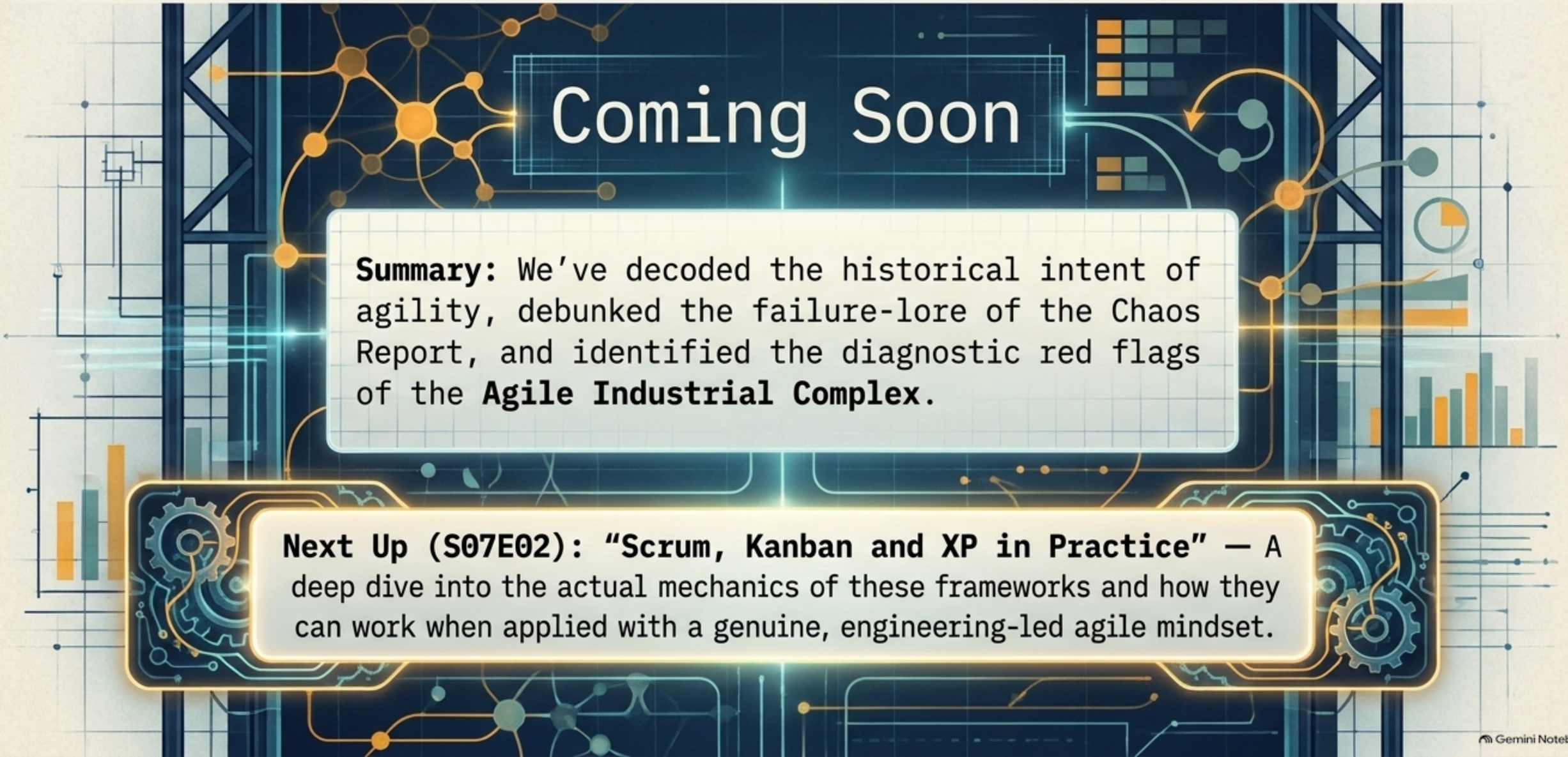


- [ ] “Product Owners” act as requirements dictators handing down comprehensive specifications, rather than collaborative partners exploring uncertainty.



Gemini Notebook

Conclusion & Next Up



Gemini Notebook