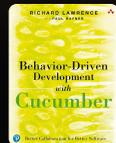


# Design in the Loop

SENSE IT. SHAPE IT. CHECK IT. HONE IT.

**Paul Rayner**

CEO, Virtual Genius



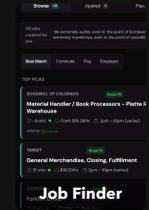
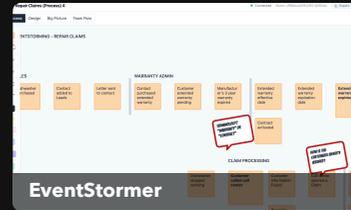
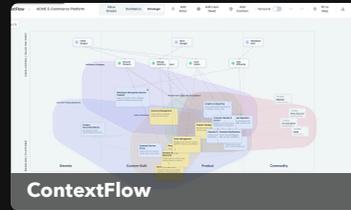
# When AI generates most of the code...

The work that matters changes.

Design skills become **more** valuable, not less.

The rigor doesn't disappear. It **expands**.

# 40 projects. 10 months. One person.



## Products

- CodeCohesion – code visualization (OSS)
- ContextFlow – context mapping (OSS)
- EventStormer – digital canvas (OSS)
- Binah – Hebrew grammar drills
- Flowline – Monte Carlo forecasting
- Claude Conversation Extractor – CLI tool
- YT Summary – video summarizer
- Story Brief – story planning
- VG Website – company site
- EDDD Website – conference site
- Pixel Office – pixel art agent monitor
- The Hub – AI business operating system
- Job Finder – job search app

## Operational Tools

- Tito API scripts
- Tito dashboard + actuals
- Harvest invoicing
- EDDD dashboard
- VG newsletter
- Newsletter performance
- Revenue scoreboard
- Funnel staleness
- Claude session stats
- Session classifier
- Relationship scoring
- Pipeline contacts
- Feedback analyzer

## Glue

- MD-to-email
- Google Calendar CLI
- iMessage CLI
- Telegram CLI
- Discord CLI
- Signal CLI
- YT Summarize
- RSS research scanner
- Weekly review prep
- Weekly pattern detect
- What-now prep
- Sessionize tools
- Date utils

When code is cheap, you build things you'd never have planned.

# DORA 2025: AI is an Amplifier

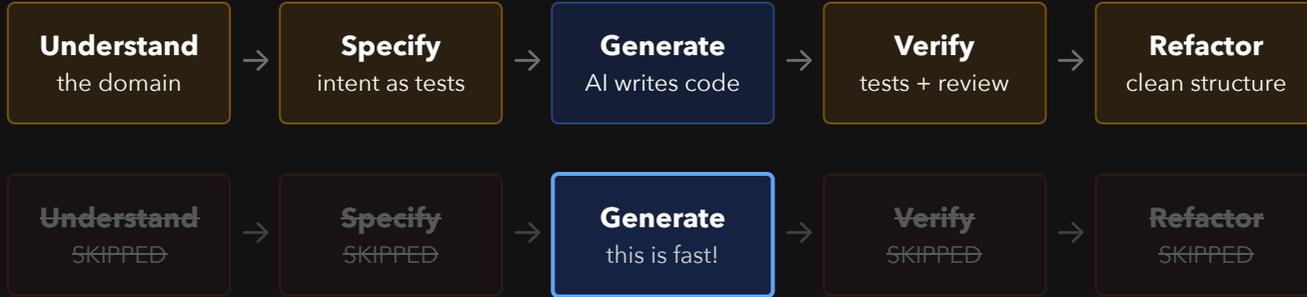
Google surveyed ~5,000 professionals and found:

- AI **magnifies** existing organizational strengths AND dysfunctions
- High-performing teams with strong practices see AI accelerate their work
- Struggling teams see their dysfunctions **amplified**

**The perception gap** (METR study, cited in DORA): developers slowed 19% by AI on familiar codebases still *believed* they were 20% faster.

If you can't tell you're slower, you can't tell your discipline is broken.

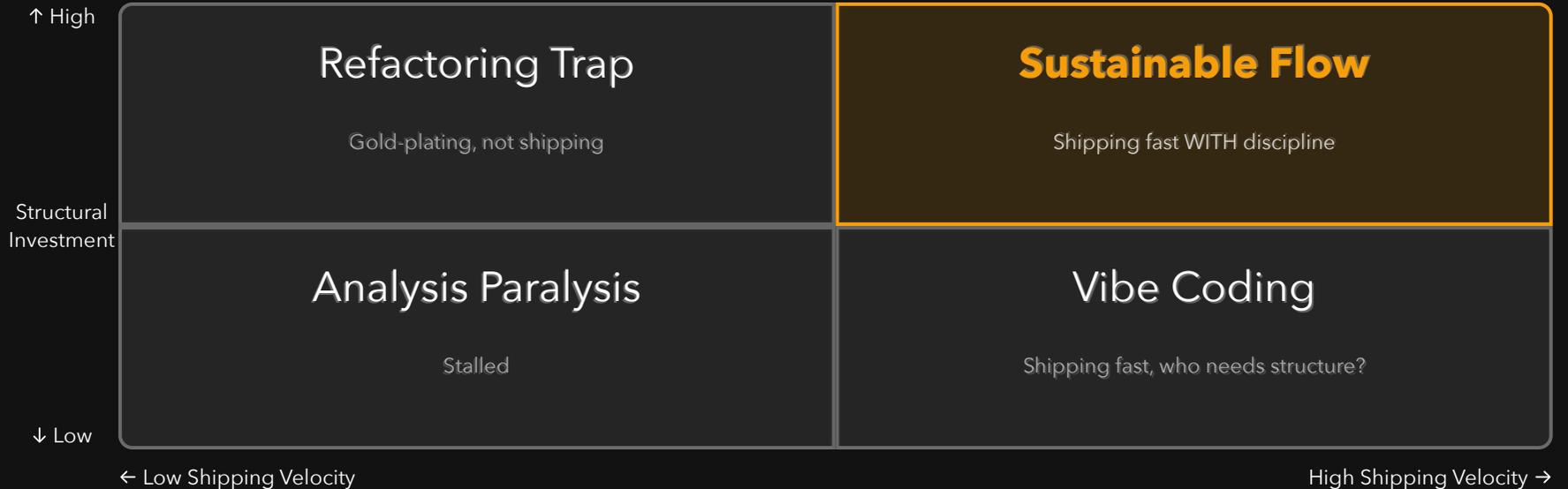
# The Vibe Coding Problem



Vibe coding: 4 of 5 steps skipped.

DORA 2025: AI now improves throughput. Speed is solved.  
But instability still climbs at the same rate. Speed isn't the bottleneck. The bottleneck is what constrains **predictability**.

# Where Are You?



The goal: **ship fast WITH discipline.**

# Every Major Shift Looks Like Removing Discipline



Each shift *looked* like removing discipline. The rigor just expanded.

**Sense it.**

# Sense-Making Is the Work That Remains

*"A little bit of mist in your mind is going to be an impenetrable fog for the AI."*

AI has no sense-making capability. It has pattern-matching from training data.

It can't synthesize meaning from **contradictory** domain expert perspectives.

It can't recognize that two stakeholders mean **different things** by the same word.

The phase AI accelerates most (code generation) is the least important.

The phase that matters most (understanding the domain) still requires **you**.

# EventStorming: Group Sense-Making

- Diverse people surface **conflicting mental models** through sticky notes
- Hot spots (pink stickies) mark where understanding has **broken down**
- The output is shared understanding, not documentation
- The person who best understands the **customer's workflow** may be the most valuable person on the team, even if they never touch the code

## Warehouse robotics company

EventStorming surfaced the complex handoffs between warehouse robots and human operators. Hot spots revealed where process ambiguity was costing them. Months later, their lead engineer is building AI agents on those exact pain points.

They didn't start with "where should we use AI?" They started with "where is our process broken?"

# Context Boundaries Scope What AI Can Work With

- Smaller, well-bounded contexts = **better AI output**
- When the AI loses track, the context is too big.  
**Split it.**
- Project instructions = durable context that **survives sessions**

Annegret Junker's experiment

AI-generated code from domain stories:

**3**

schemas without  
EventStorming



**9**

schemas with  
EventStorming

Correctly encoded business rules, properly separated  
bounded contexts.

Sense-making directly improved AI output. The domain understanding wasn't optional decoration; it was the input that made the AI useful.

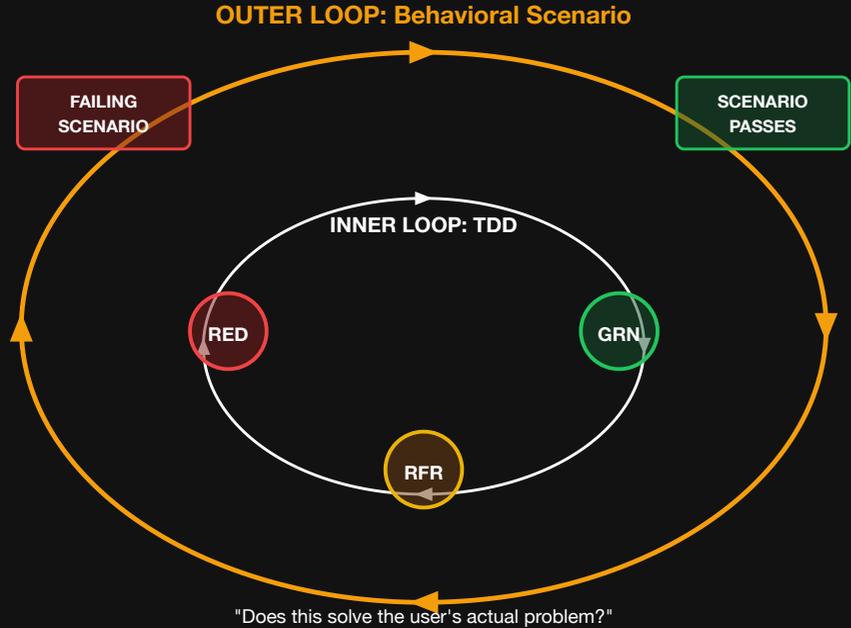
**Shape it.**

# My Actual Workflow



Clear context between every stage. Stay in the bottom of the context window.

# Double-Loop BDD: Implement



## Who does what?

Work with AI to define the scenarios.

**AI** drives the red-green-refactor loop.

The tests are your executable intent. They survive code regeneration.

**Check it.**

# Verification: Trust but Check

- **Pre-push checks** - automated skill that audits scope, tests, and quality before code leaves your machine
- **Git hooks** - linters, type checks, and formatting enforced on every commit
- **Test suite as gate** - nothing merges without green tests; AI-generated code gets the same scrutiny as human code.
- **Manual review** - read the diff; AI can explain its changes but you decide if they belong

Generation is probabilistic. Verification must be deterministic.

**Hone it.**

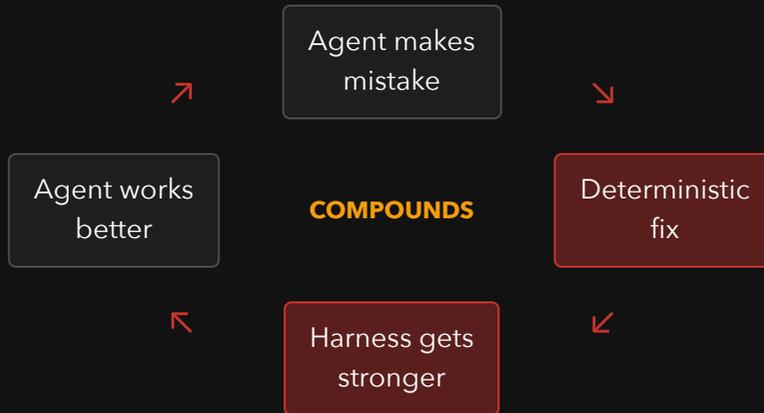
'Probabilistic inside, deterministic at the edges.'

Chad Fowler

"Probabilistic inside, deterministic at the edges,  
**judgment all around.**"

# Where the Rigor Landed: The Harness

- Every agent mistake becomes a **deterministic fix** that compounds over time
- Tests, type checks, linters, project instructions, architectural fitness tests
- The harness IS the expanded rigor



"The same stuff that's good for humans is good for agents."

# Honing in Practice: Extract, Encode, Compound



**#1 friction pattern:** "Claude skips TDD" (60 occurrences in 5 weeks).

**Fix:** Encoded Double-Loop BDD/TDD process directly into project instructions.

**Result:** Every session now follows the process automatically.

Daily reflections feed tomorrow's priorities. Memory files accumulate lessons. The harness only grows.

# Rigor expanded to two places.

## **Sense-Making**

EventStorming, example mapping, context boundaries, domain modeling.  
Understanding the problem before solving it.

## **The Harness**

Tests, linters, type checks, project instructions. Deterministic at the edges.  
Compounds over time.

Probabilistic inside, deterministic at the edges, judgment all around.

~~Ship Fast~~  
**Ship with Tempo.**

Sustainable rhythm with strong orientation beats raw speed  
every time.

# Design in the Loop

Monday morning: pick one thing you're about to build.  
Write the test before you prompt. That's the loop.



**virtualgenius**  
leading by design



Paul Rayner · @thepaulrayner (Bluesky) · [linkedin.com/in/thepaulrayner](https://www.linkedin.com/in/thepaulrayner)