Our Journey to Agentification

A Case Study

John Rauser | Senior Director, Software Engineering | Cisco Systems

The Challenge

Excitement and energy are **not evenly distributed**.

People and teams are struggling to find their footing with AI.

The Problem Isn’t Technical

- GitHub Copilot ✓
- Windsurf ✓
- Cursor ✓
- Codex ✓
- Claude ✓

It is psychological and cultural.

We had the tools and training.

We needed a structured approach.

Leaders Build First

Leader Experience

1

Leader Passion

2

Team Adoption

3

Leader Experience

01

Activating potential through hands-on work

1 It Started With a Challenge

Ship something using AI.

Every leader. A real feature. Into the product.

1 Ship Anything

UI Feature

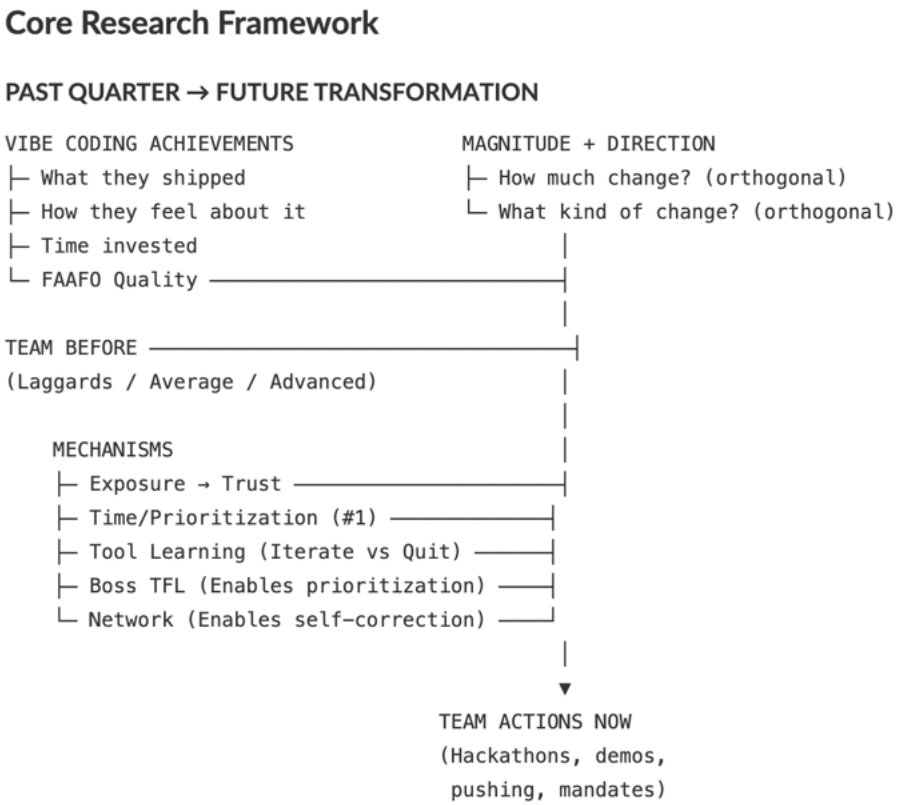
Backend

Automation

Ops Improvement

Be as ambitious (or not) as you like. Just build.

When John Told Me About This…



The Experiment

A Cisco SVP mandated ~100 engineering leaders — people who manage people, not write code — to **build and deploy something to production using AI coding tools in one quarter**.

These leaders span four levels of the management hierarchy:

```
                        CLOSEST TO CODE
                              |
  Manager (first-line)        |  Directly manages ICs
  n=6                         |  e.g., L3-Mgr, L9-Mgr
                              |
  Senior Manager (second-line)|  Manages Managers + some ICs
  n=9                         |  e.g., L2-SrMgr, L7-SrMgr
                              |
  Director (third-line)       |  Manages Senior Managers
  n=1                         |  L6-Dir (team of 66)
                              |
  Senior Director (fourth-line)|  Manages Directors
  n=1                         |  L1-SrDir (team of 55)
                              |
                        FURTHEST FROM CODE
```

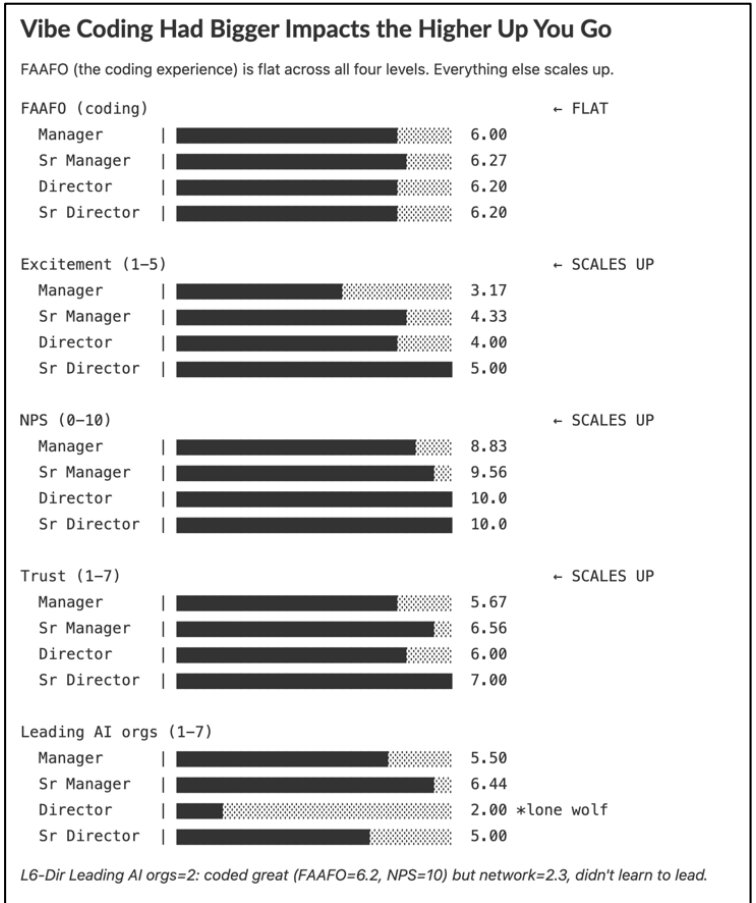
Response Rate by Level

Based on Scenario B org structure estimate (~100 mandated leaders):

Level	Est. Population	Responded (N)	Response Rate	
Senior Director	~4	1	25%	
Director	~12	1	8%	
Senior Manager	~30	9	30%	← highest
Manager	~54	6	11%	
TOTAL	~100	17	17%	

Senior Managers responded at 3x the rate of Managers and Directors. This is consistent with Finding 4 (seniority amplifies transformation) — the most transformed group was the most eager to tell us about it.

Finding 1: Higher Up, More Fun



Finding 1: 85% Changed How They Lead

Finding 1: Vibe Coding Created or Amplified Four Leadership Behaviors

The mandate doesn't just create coders. It creates mandate-issuers.

88% of leaders say their teams are ahead of them on AI. 71% said teams were already "Advanced" at the start. The teams didn't need the mandate. **The leaders did.** It gave them a personal FAAFO experience that unlocked four leadership behaviors only they can provide — and that their teams were waiting for:

1. Recalibrated Vision — they experienced the new speed and now set expectations around it

L4-SrMgr: *"I spend less time on first drafts and more time reviewing outcomes, validating designs, and coaching."* L2-SrMgr: *"My expectations on when work will be completed has moved from 2-week long sprints to results in days."*

2. Raised the Bar — they made AI the default, not the exception

L2-SrMgr: *"If AI is not going to be used then I will need a definitive answer why not."* L8-Mgr changed hiring criteria and rewrote performance goals. L3-Mgr: *"AI is no longer optional and must be a core part of our engineering culture."*

3. Shifted from Problem-Solver to Direction-Setter — they stopped doing the work and started shaping the work

L5-SrMgr: *"My thinking about leadership has shifted from being the primary problem-solver to setting clear direction, constraints, and quality bars, while letting AI accelerate exploration."* L4-SrMgr: *"I focus more on defining direction up front and less on micromanaging implementation."*

4. Modeling and Multiplying — they lead by doing, then create structures for others to follow

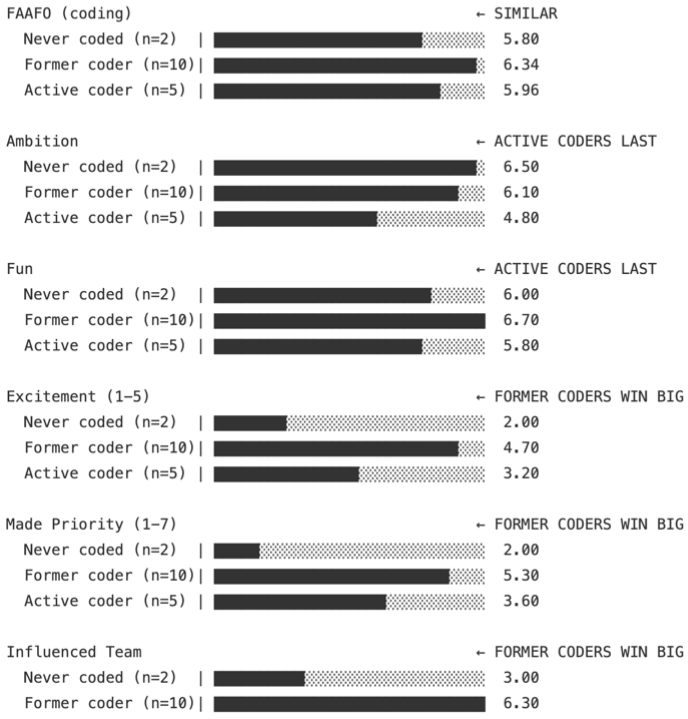
L4-SrMgr: *"A Security SLA tracker I initially built was picked up by a team member and transformed into a QuickSight dashboard."* L3-Mgr: *"I want to introduce weekly demos where team members can showcase how they are using AI."*

Finding 2: Former Coders Outperformed

Finding 2: Former Coders Aimed Bigger, Had More Fun, Influenced More People

The most statistically significant finding in the entire dataset — and the most counterintuitive.

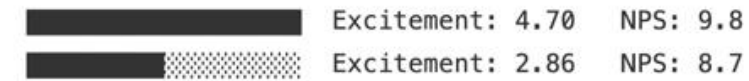
Three populations: leaders who **never coded** (n=2), **former coders** who'd stepped away (n=10), and **active coders** who still code daily/weekly (n=5). Former coders dominated. Active coders came last on nearly everything.



Finding 3: Fun Matters

Finding 3: Fun Is the Mechanism, Not Productivity

Fun had the strongest correlation with excitement about AI's future — stronger than Faster, Ambition, Autonomy, or Options.



Leaders who had fun **made their own time** despite organizational priority scores of only 4.4/7. Nobody told them to prioritize it. They wanted to.

The chain: Fun → Excitement → Made Time → Higher FAAFO → Changed Leadership → Issuing own mandates.

Stats:

- Fun ↔ Excitement: **rho=0.833** (strongest correlation in dataset)
- Excitement ↔ Made Time: **rho=0.829**
- Fun=7 → NPS 9.8 vs Fun<7 → NPS 8.7
- Priority (D5) mean only 4.44/7 — they did it anyway

The design implication: We've been designing AI adoption programs around ROI and efficiency. The data says design them around *fun*. If the experience isn't fun, nothing downstream happens.

Finding 5: Active Coders Did Not Change

Finding 5: Active Coders Got the Least Out of It

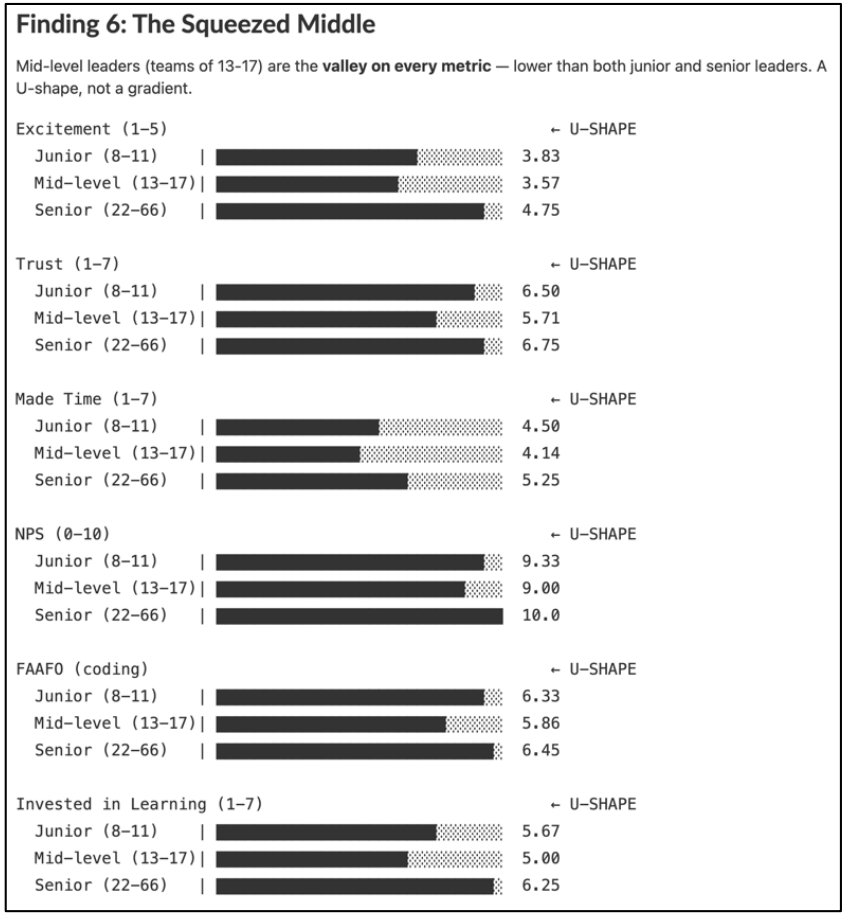
Leaders who still code daily or weekly had the least fun, dreamed the smallest, and described **nearly zero concrete leadership actions**.

```
Coded leadership actions from open-ended D8 responses:
  Rusty coders (n=10):  mean = 2.90 actions described
  Never coded (n=2):    mean = 2.00
  Active coders (n=5):  mean = 0.40 ← almost nothing
```

4 of 5 active coders described zero concrete leadership changes when asked. The one exception: L6-Dir (network=2.3, action score=2) who issued mandates despite being the most isolated person in the sample.

```
Active coders — individual detail:
  L6-Dir                  net=2.3  actions=2  MANDATE, STRUCTURE
  L11-Mgr                 net=7.0  actions=0  (none)
  L10-SrMgr               net=6.8  actions=0  (none)
  L12-SrMgr               net=4.8  actions=0  (none)
  L13-SrMgr              net=6.5  actions=0  (none)
```

Finding 6: Sr Managers: Toughest Job?



We Want To Publish The Data Set

1 Why This Works

Skin in the Game
Accountability through doing

Real Perspective
Grounded in actual work

Common Language
Shared experience to evangelize

Leader Passion

02

Enriching the leadership experience with agentic tools and techniques.

2 AMOS

The Agentic Management Operating System

A suite of capabilities and processes designed to help leaders manage their teams and organizations with AI.

Rich Knowledge Base → Common Management Behaviors

2 We are on to something …

Andrej Karpathy on Agentic Knowledge Bases

“A large fraction of my recent token throughput is going less into manipulating code, and more into manipulating knowledge”

x.com/karpathy/status/2039805659525644595

2 The Continuous Feedback Loop

First up? The dreaded weekly report.

2 The AMOS Toolkit

Instructions

Memory

Skills

Commands

Subagents

Hooks

Triggers

Automations

Plugins

MCPs

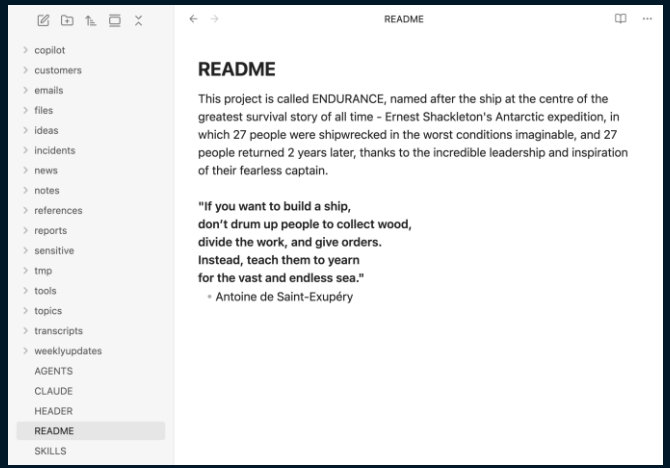
Extensions

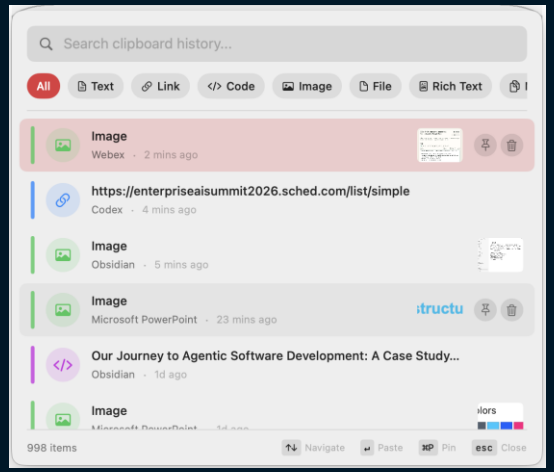
Apps

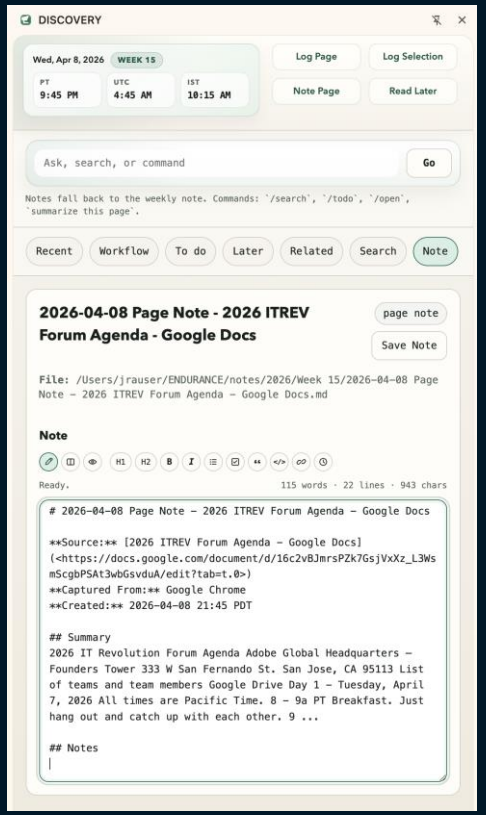
Dashboards

2 AMOS in Action

Leaders quickly evolve based on their needs, inventing new capabilities all the time.







2 Key Enablers

Ditch The Old Note Systems

Standardize on markdown. VSCode + Obsidian. Local Storage.

Generate docs on the fly using hooks, triggers and automations.

Build Shared Reference Maps

Capture entities & relationships.

Maintained by humans + agentic CI.

Share Ideas Across Leaders

Innovation spreads organically.

Predictive dashboards, early warnings, scheduling insights.

03

Team Adoption

Driving Agentic Systems into engineering at scale.

3 Moving Beyond Using Tools

The goal is not just using agentic coding tools.

It's true Agentic SDLC.

3 AI Squads

Teams = EC2

Persistent, always running

Squads = Lambda

Spin up, focus, deliver, dissolve

Nimble. Focused. Clear mission.

Short time horizon.

3 The First Squads – Finding Agentic

Find the bottlenecks. Attack every step.

Build agentic systems into entire delivery lifecycle.

Build

Grow

Evangelize

Show the org, by example

3

DECAF

DEClarative Agentic Framework

- Repeatable, plugin-driven “Agentification” capability
- Bootstraps AI infrastructure per repo
- Hierarchical agentic communication systems
- Activities and prompts for tribal knowledge

Feature Patterns • Operational Practices • Architecture • Domain Knowledge

3 Expanding the Learning

Documentation

Workshops

Demos

Best Practices

Centralize & standardize what works.

Our Journey

01

Leader Experience

Hands-on Experience

02

Leader Passion

The Agentic Management Operating System (“AMOS”)

03

Organizational Transformation

AI Squads & Agentic SDLC

3 Help I’m Looking For

The goal is not just using agentic coding tools.

It's true Agentic SDLC.

Thank You!

John Rauser

Senior Director of Software Engineering

Cisco Systems

Find me on LinkedIn