

MORE CODE, MORE PROBLEMS

Why AI Gains Disappear at Scale

Stefan Grothkopp · CTO, aicx

Claude Code Meetup Munich · March 23, 2026

NOTORIOUS BUILDER

“Stay far from timid, only make moves when your heart’s in it.”

— The Notorious B.I.G., Cowards

Stefan Grothkopp

- **aicx** — CTO. Building an AI agent platform for SMBs to Enterprise.
- **Altruja** — Co-founded. Online fundraising, 200M+ in donations.
- **MatchApply** — AI-powered job matching based on soft factors.
- **RevoluSun** — Energy communities and renewable sharing models.
- **SolarJob** — Job platform for the solar industry in DACH.
- **ManniWorld** — Financial education for kids.



THE PROMISE

Experiment by Google in 2024:

Researchers tested whether AI tools actually make developers faster.

- 96 professional software engineers at Google
- Complex enterprise task in a large C++ codebase
- Randomized: half used AI tools, half didn't

AI-assisted developers were

+21%

faster

THE PREDICTION

A different team ran a second experiment in 2025:

METR asked experienced open-source developers to predict how much faster AI would make them.

- 16 developers, each with 5+ years on their own repos
- 246 real-world tasks from their own issue trackers

Their prediction:

+24%

faster

WHAT ACTUALLY HAPPENED

Actual result with AI tools:

-19%

slower

They were slower. Not faster. Slower.

And the kicker: after seeing their own stopwatch data, they still believed AI had made them **20% faster**.

THE COPILOT PARADOX

GOOGLE STUDY

- Isolated coding task
- Controlled environment
- One codebase
- No coordination overhead

METR STUDY

- Real-world tasks, real projects
- Context switching
- Review & integration
- Coordination

The coordination and communication part eats the individual gains.

**“The Copilot Paradox”:
AI makes the individual faster, but the team slower.**

WHERE DOES THE TIME GO?

Developers spend ~40% of their time on non-coding work.

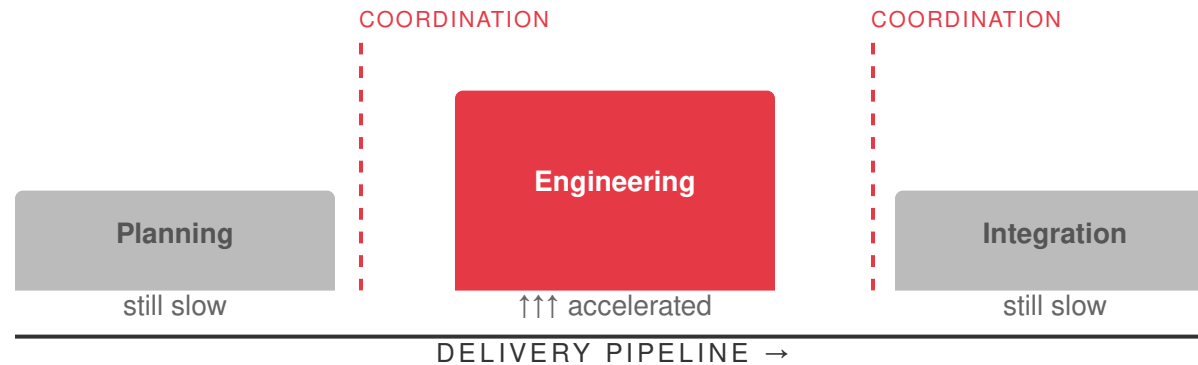
- 33% — Maintenance and technical debt
- 15–20% — Meetings and sync communication
- 5–10% — Code review and collaboration
- 3–10% — Deployment and operations

AI accelerates the 60%. The 40% stays. Or grows.

THE CONSTRAINT MOVED

“Improving efficiency at a non-bottleneck is an illusion.”

— Eliyahu Goldratt, The Goal (1984)



Before AI: Coding was the bottleneck.

After AI: Review, integration, coordination are the bottleneck.

More code, more problems.

(DORA 2025: 98% more PRs, 7.2% less delivery stability)

THE AMPLIFIER EFFECT

“AI functions as an amplifier, magnifying both organizational strengths and weaknesses.”

— DORA State of AI in Software Development, 2025

Good processes + AI = great results

Bad processes + AI = faster chaos

“You can’t AI your way out of bad process.”

This is true for teams. And it’s even more true for organizations.

FROM TEAM TO ORGANIZATION

This is not just a team problem.

The same coordination overhead that eats individual gains at the team level scales up to the entire organization:

MICRO: TEAMS

AI generates more code, but review and integration can't keep up.

MACRO: ORGANIZATIONS

Teams ship faster in isolation, but alignment, architecture, and process break down.

**The pattern is the same at every level:
Speed up one part, and the bottleneck moves to coordination.**

TWO IDEAS

The problem lives on two levels.

FOR TEAMS

colog

Better collaboration infrastructure
so AI gains don't get lost
in coordination overhead.

FOR ORGANIZATIONS

The Book

A new operating model
so you don't just amplify
broken processes faster.

One tool for teams. One book for organizations.

COLOG — LET YOUR AGENT HANDLE COLLABORATION

Open-source framework. One idea:

Let your agent keep track of decisions, tasks, and changes.

- Every event is a git commit — decisions, tasks, changes, ideas

Git for the rest of us: the agent handles the git part

- Full history: who decided what, when, and why
- Ask the agent: `/colog:ask` — “When did we decide X?”
- No Jira. No Notion. No context switching.

I’m testing this across several side projects right now.

Works with Claude Code, Codex, Cursor, Gemini CLI, and more.

github.com/grothkopp/colog

We just decided to go with PostgreSQL for the user data. MongoDB was the other option but relational queries won out.

c ✓ Logged as a decision.

`a3f8c21 decision(db): use PostgreSQL for user data @SG`
Considered MongoDB – PostgreSQL chosen for better relational query support.

HOW COLOG WORKS

`/colog:sync`

The orchestrator. Runs the full cycle:

- Pull latest changes from remote
- Detect events from conversations → log them
- Commit file changes with semantic messages
- Two-way task sync (tasks.md ↔ git)
- Push and notify

`/colog:status`

Project overview with time range:

- `status` — open tasks + last 24h
- `status open` — uncommitted changes
- `status last week` — sprint review
- Tasks, activity by type, per user

`/colog:log`

Create a semantic commit

`/colog:save`

Stage + commit file changes

`/colog:ask`

“When did we decide X?”

`/colog:setup`

One-time project setup

Schedule `/colog:sync` as a heartbeat → everything happens automatically.

Add `/colog:status` as a morning briefing → daily summary delivered to the team.

THE AGENTIC OPERATING MODEL

Why AI Breaks Your Org Chart — And What to Build Instead

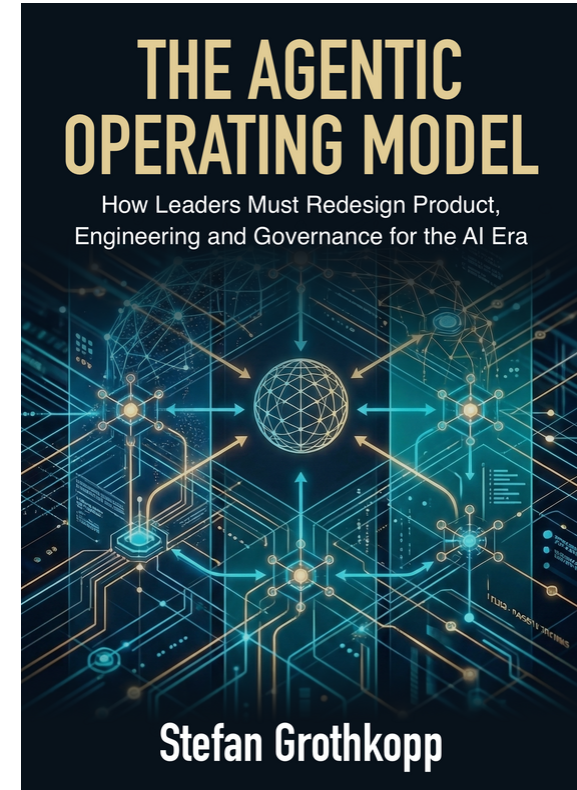
AI doesn't just change how we code.

It changes how we organize, decide, and deliver.

- Why individual AI gains disappear at scale
- What “AI-ready” actually means for an organization
- New roles, new team structures, new leadership models
- A practical framework for the transition

18 chapters. Based on real experience building AI-native teams at aicx and in my own projects.

Get the first two chapters for free:
book.grothkopp.com



THE TAKEAWAY

“We can’t change the world unless we change ourselves.”

— The Notorious B.I.G., Juicy

1. Individual AI speed gains are real (+21%).
2. They disappear at team level (−19%) because the bottleneck moves to coordination.
3. The same pattern repeats at org level — faster teams don’t mean faster delivery.
4. The fix is not faster tools. It’s better collaboration infrastructure and operating models.
5. Agents can own team-level coordination. Organizations need to rethink how they work.

**Don’t just give your developers AI.
Give your teams an agent — and your organization a new playbook.**

STEFAN GROTHKOPP

MORE CODE, MORE PROBLEMS · CLAUDE CODE MEETUP

stefan@grothkopp.com

Happy to connect by mail or on LinkedIn.

book.grothkopp.com

“The Agentic Operating Model” — free chapters

github.com/grothkopp/colog

Open-source agent skills for team collaboration



linkedin.com/in/grothkopp