{codemotion} CONFERENCE MILANO 2026

The product engineering team

Own the problem, collab on specs, let agents code

Mattia Minotti · TeamSystem

CONFERENCE
MILANO 2026



Mattia Minotti

Tech & AI Acceleration · TeamSystem

PRODUCT — Accounting — Invoicing

PLATFORM — Customer Journey

AI — AI Innovation — AI Adoption

Your engineers have changed.
Has your sprint?

Discovery | Requirements | Planning | Assigned tickets | PR review | Release

UNCHANGED

The code changed. The process didn't.

+98%
Pull requests

+91%
Review time

=
Delivered value

Source: Faros AI, The AI Productivity Paradox Report 2025 (July 2025) · 10,000+ developers, 1,255 teams · faros.ai/blog/ai-software-engineering

Direction beats speed.

Yet classic teams were always built the other way around:

1 product manager → **many engineers executing**

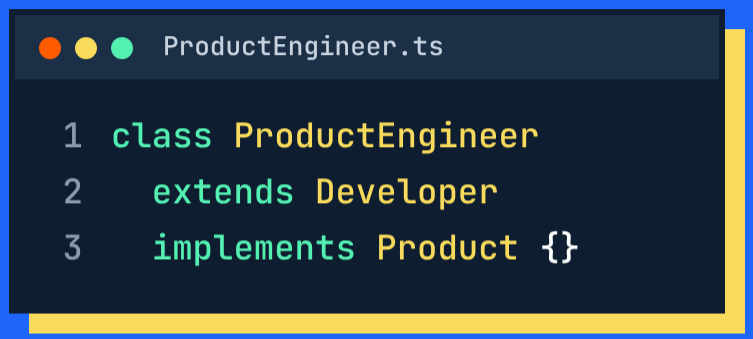
If, in the age of AI, code is no longer the bottleneck...

...for a new way of building products, we must redesign the software engineering team.

01

Own the problem

What is a Product Engineer?



CONFERENCE
MILANO 2026

A software engineer who:

01 is customer obsessed

02 starts from the user's problem, not the ticket

03 takes extreme ownership of ideas and ships them end-to-end

04 genuinely loves building

Source: PostHog, Product Engineer Handbook · posthog.com/product-engineer

SKILLS

CONFERENCE MILANO 2026

What a product engineer is made of



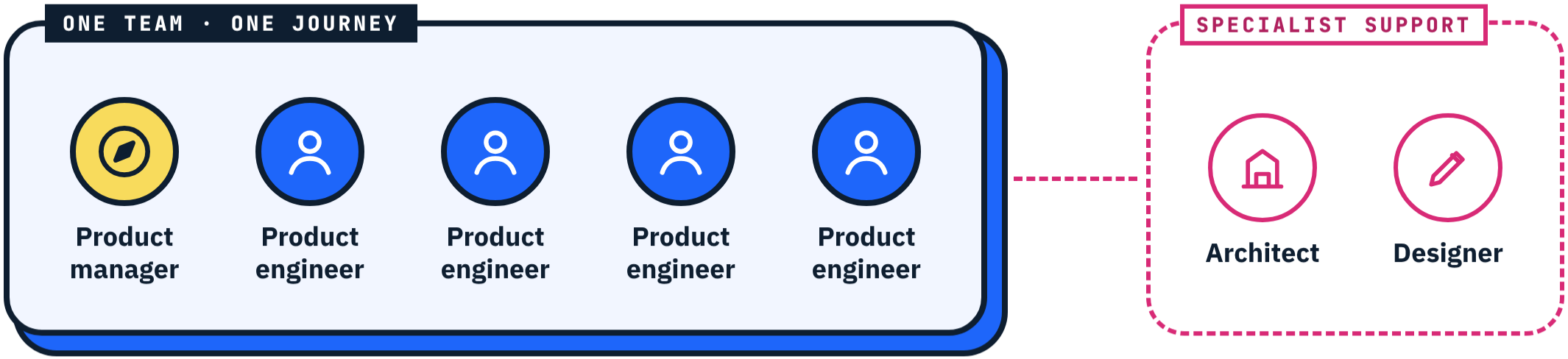
AI

- Customer & business knowledge
- UX & design fundamentals
- Technical depth

CONFERENCE
MILANO 2026

Our team: TouchX @ TeamSystem

Responsible for building the TeamSystem customer journey.



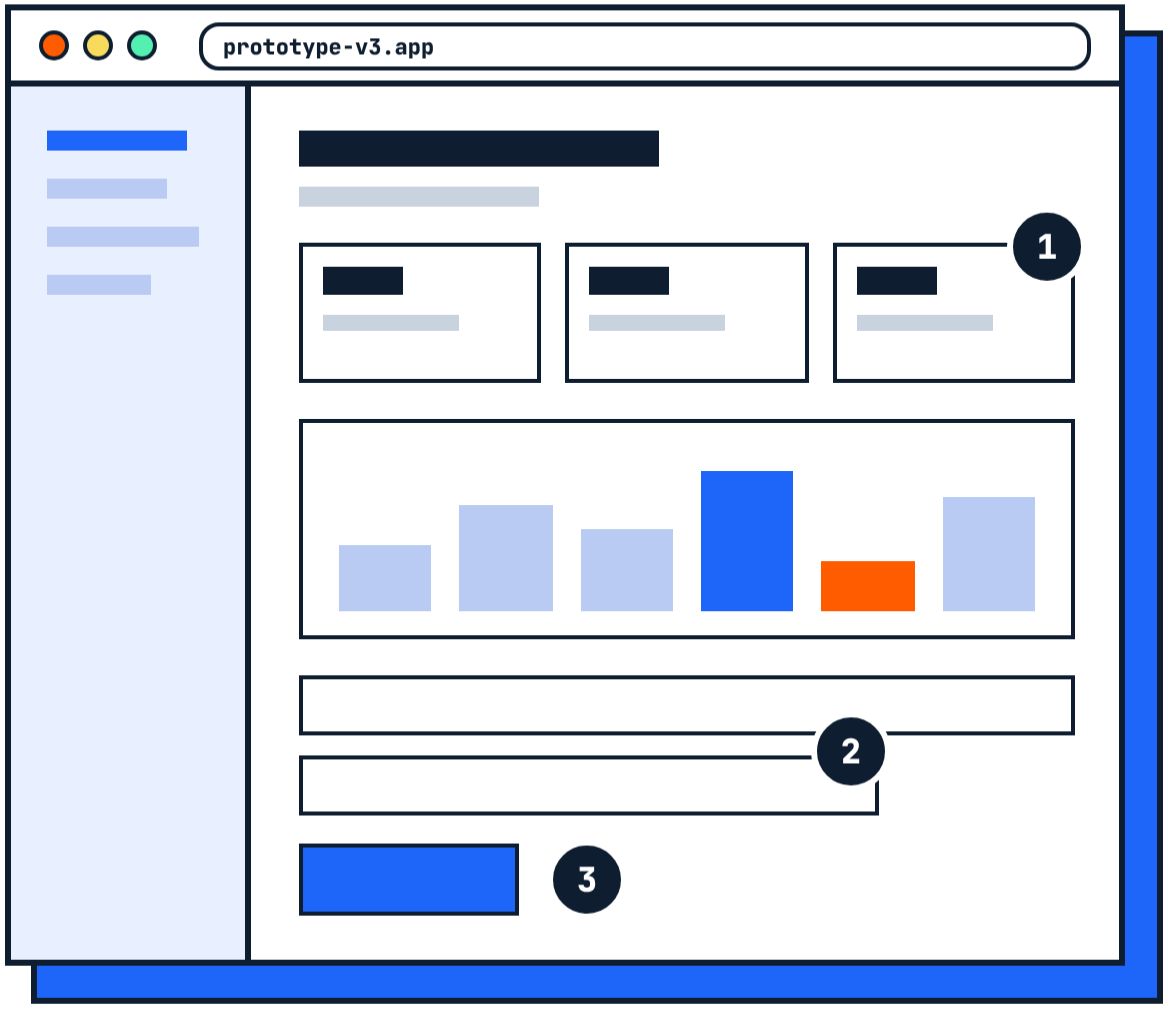
LIVE EXPERIMENT

We're experimenting with all of this, right now.

PROTOTYPING

Prototype to design.

1. discuss the requirements
2. define the solution
3. align stakeholders



Context is the first deliverable

01

Team LLM wiki

Meetings, decisions and documents go in as sources; an agent keeps the knowledge base current

02

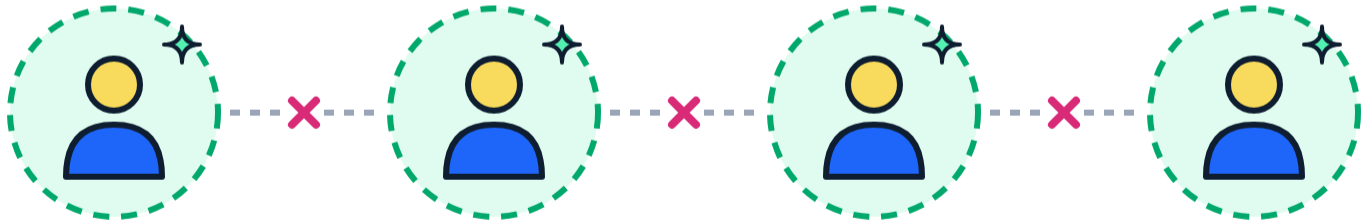
Solid harness

Agent instructions, shared skills, MCP tools, hooks, subagents and automated checks

03

Spec-driven development

Specs, plans and tasks written first and versioned with the code: one source of truth for people and agents



AI pushes developers into **isolation**.

If everyone can do everything,

what is the team for?

02

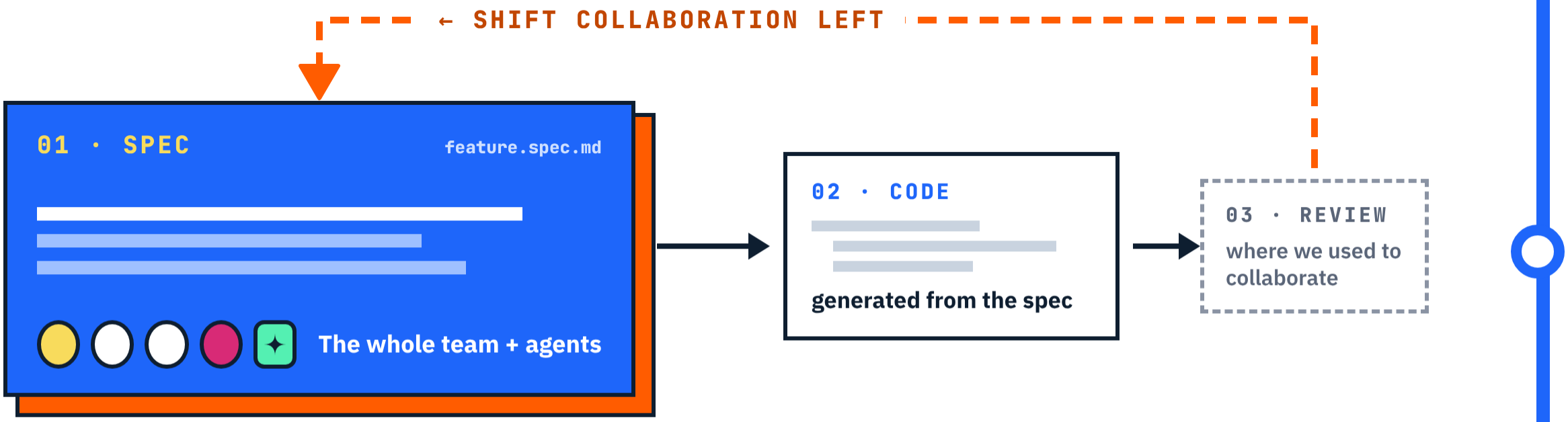
Collab on specs

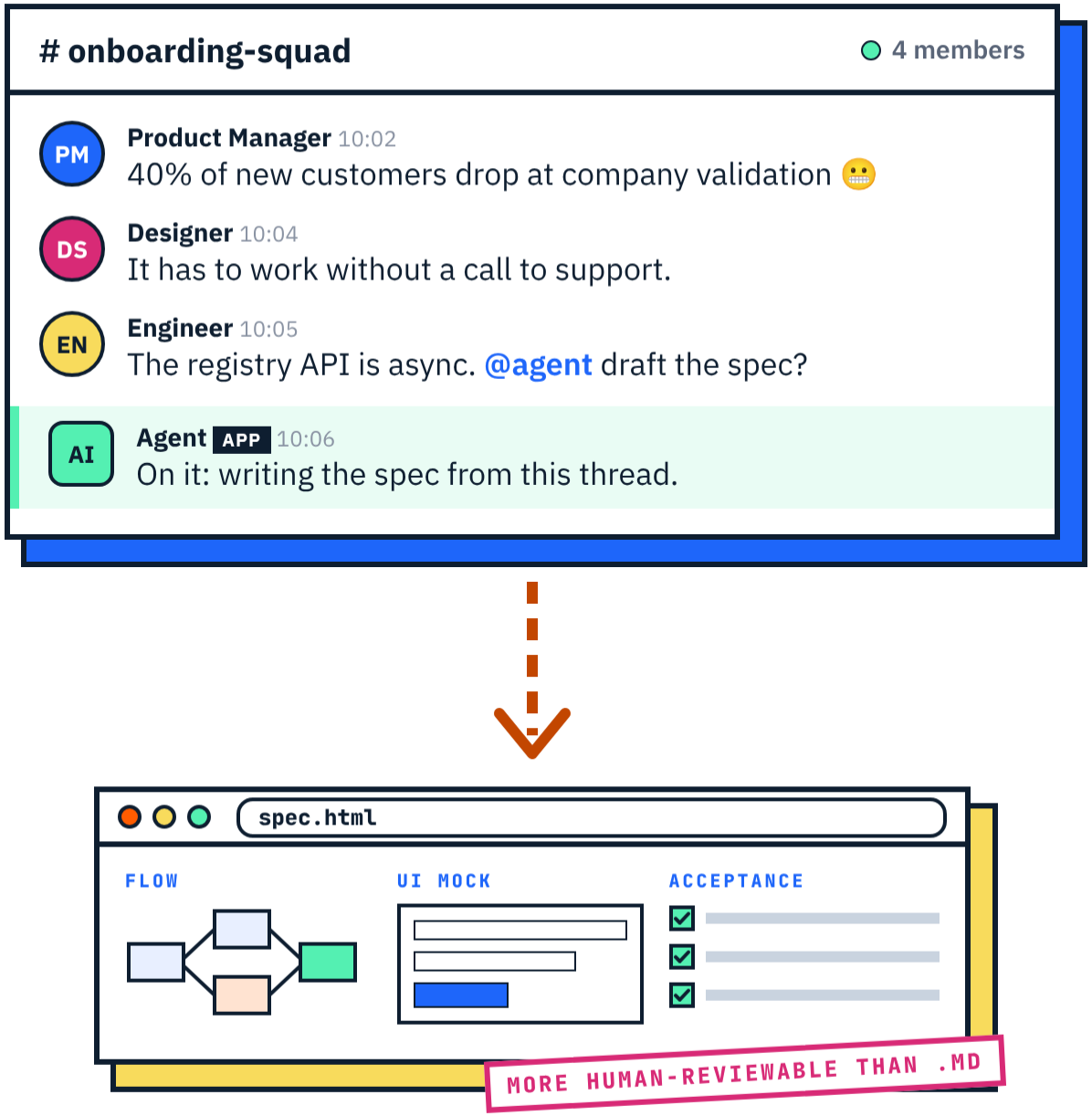
CONFERENCE
MILANO 2026

SHIFT-LEFT COLLABORATION

Specs are the new code.

So we write them together, shifting collaboration left.





CONFERENCE
MILANO 2026

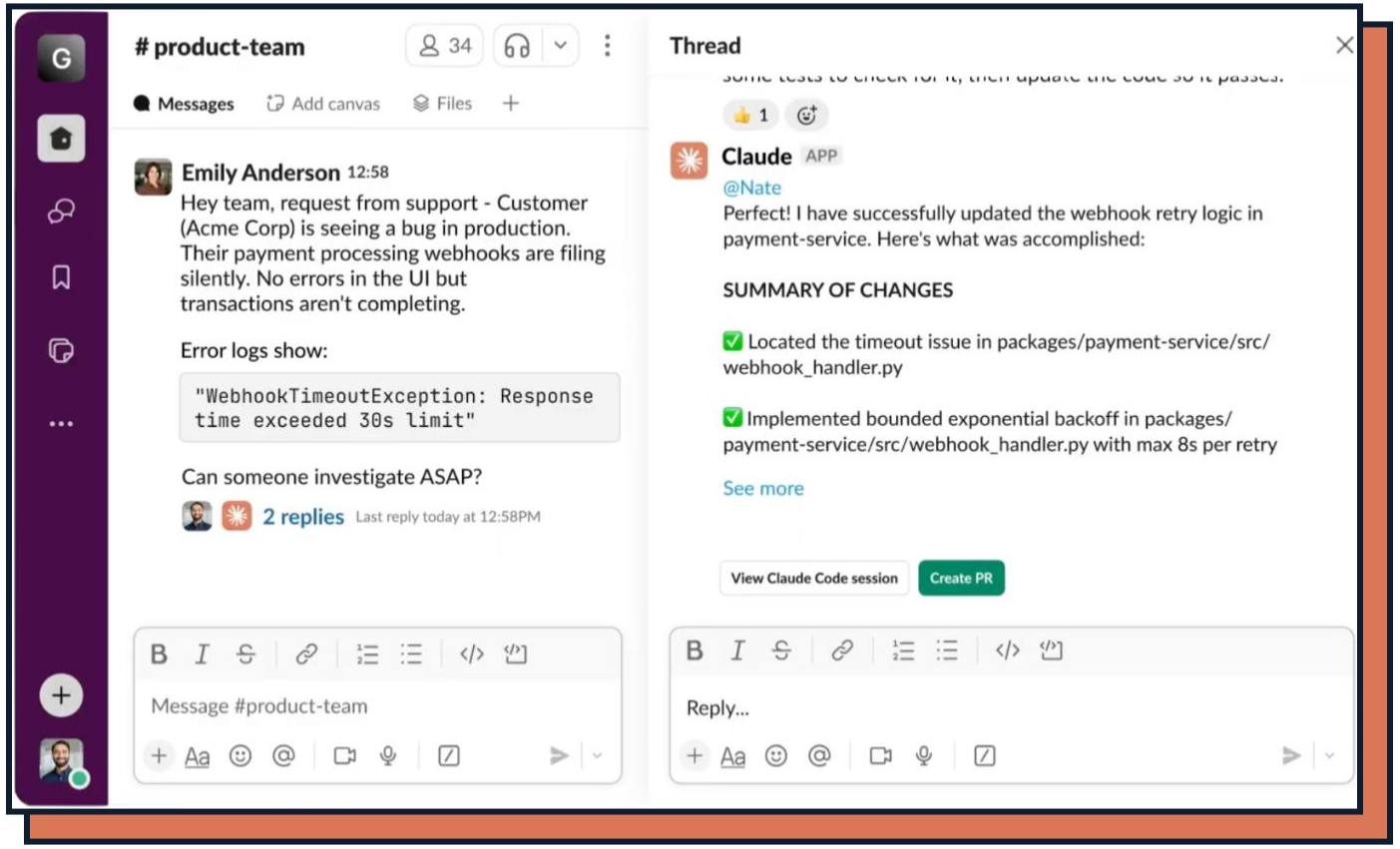
The spec is generated from a conversation between the team.

ANTHROPIC · CLAUDE TAG

“

Tag Claude into Slack and it works in channel with you. It's proactive, multiplayer, with its own identity and memory.

— Boris Cherny, creator of Claude Code



Don’t ask for discipline. Automate it.

In the TouchX team, a scheduled agent turns our conversations into a shared knowledge base.

Conversations

- Slack threads
- Meetings
- PR comments

SCHEDULED · EVERY NIGHT

claude code

Agent

```
$ claude -p "/team-digest"
› collects the day
› summarizes decisions
› links code and specs
✓ wiki updated
```

Team wiki

- Daily digest
- Decisions log
- Open questions

CONFERENCE
MILANO 2026

CODE ↔ SPEC

Know which session every line came from.

A diff shows *what* changed. The spec shows **why**.

01 Review the intent, not only the diff

02 Give the next agent the context, not just the code

TOOLS

NOW PART OF OPENAI



Git AI
Tracks AI code to production

Entire

Entire
The Git forge for agents

CURSOR

Agent Trace
Open spec for AI code attribution

03

Manage the load

A PRODUCT ENGINEER HAS TO…

- Understand customers and the business
- Run discovery with users
- Prototype and design the UX
- Write the spec
- Build front end and back end
- Orchestrate and review agents
- Ship and run it in production
- Measure the impact

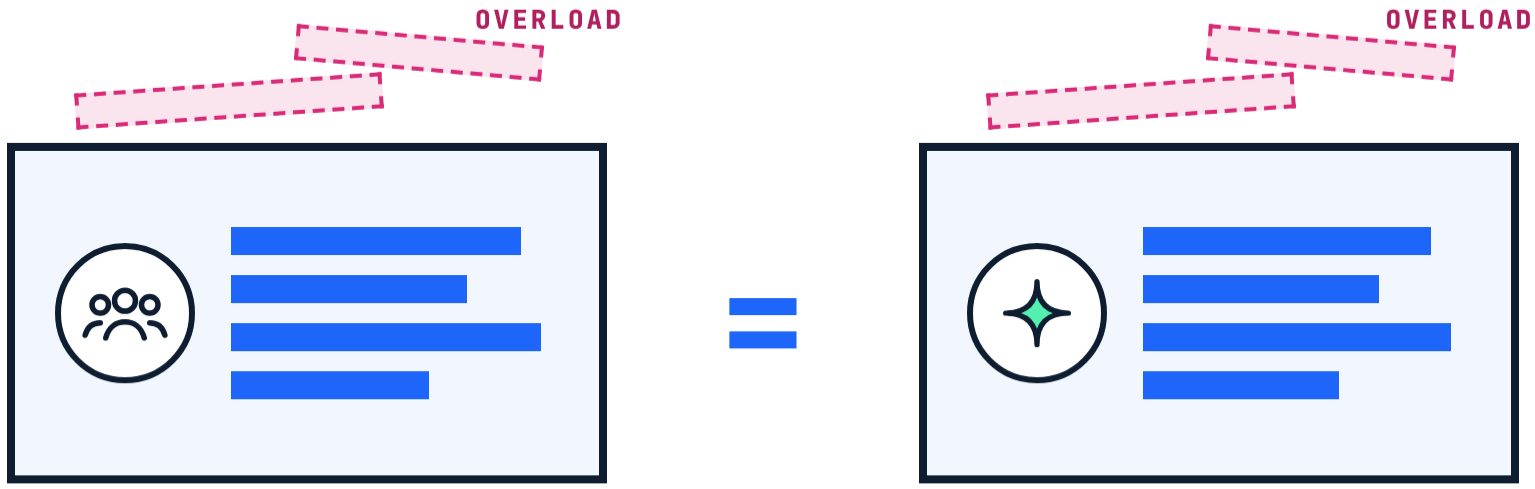
End-to-end ownership

=

more load.

How do we manage all this cognitive load ?

Same problem: a limited context window.



Team
COGNITIVE LOAD

=

Agent
CONTEXT WINDOW

TEAM TOPOLOGIES × AI

Team Topologies is the infrastructure for agency.

The boundaries that give human teams autonomy are the same ones that keep AI agents governable.

01 Write down what each **team and agent can decide** alone.

02 Keep **scope small** enough for one team and one agent's window.

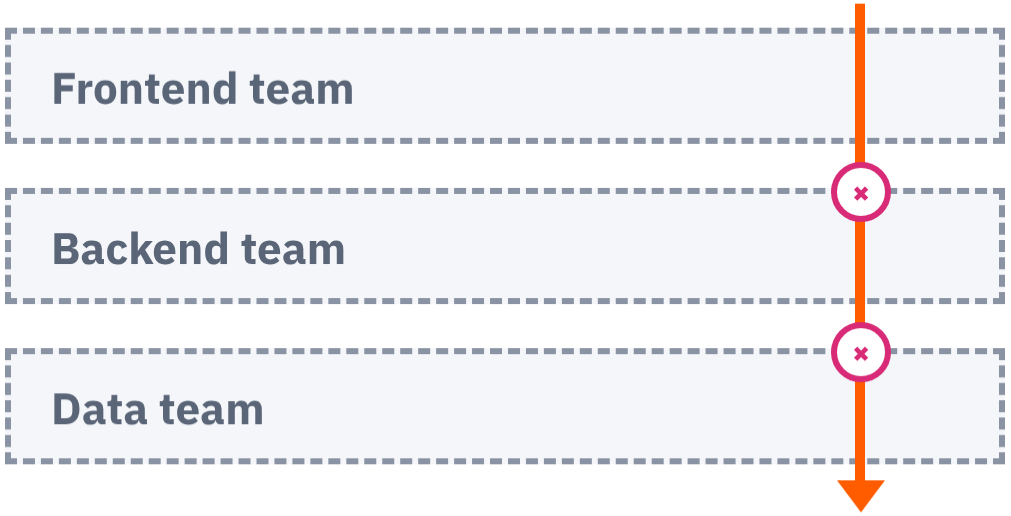
03 Give every service **one team** that owns its **context and outcomes**.

04 Give agents **only the access** their task needs.

Source: Matthew Skelton & Manuel Pais, "Team Topologies as the 'infrastructure for agency' with AI", keynote (QCon London, March 2026) · teamtopologies.com/keynote-talks

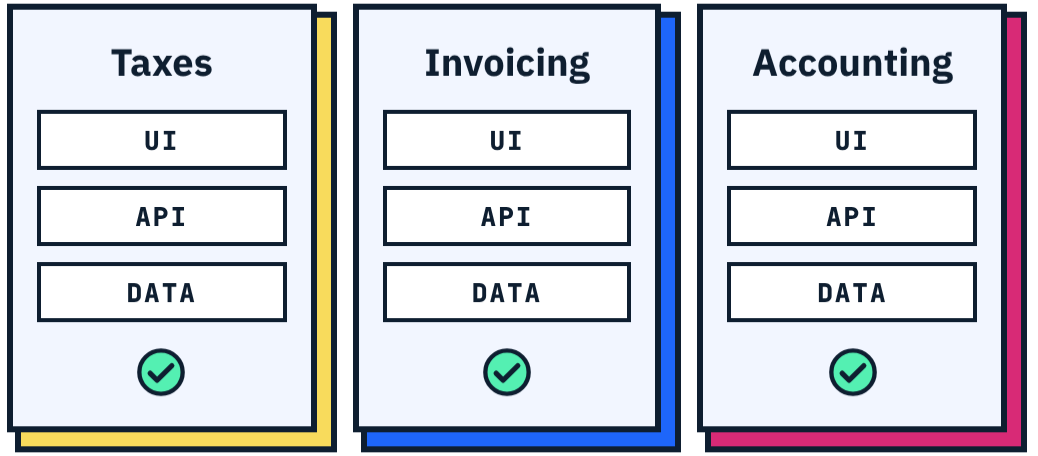
Split by domain, not by technology.

BY TECHNOLOGY



One feature, three teams, many handoffs.

BY DOMAIN



Starts from the problem, owns the outcome, **end to end**.

Less cognitive load + **Full ownership**

Monday morning

- Give product engineers problems, not tasks.
- Collab on specs as a team, let agents generate the code.
- Draw boundaries around heads, splitting by domain.

{codemotion} CONFERENCE MILANO 2026

Want to work in a team like this?

We're hiring.

TeamSystem®



Mattia Minotti

TeamSystem

Scan Me

linkedin.com/in/mattia-minotti-3b48b247