Agile Expat

Org Design Summit 2026 · Amsterdam

Fluency is an org property.

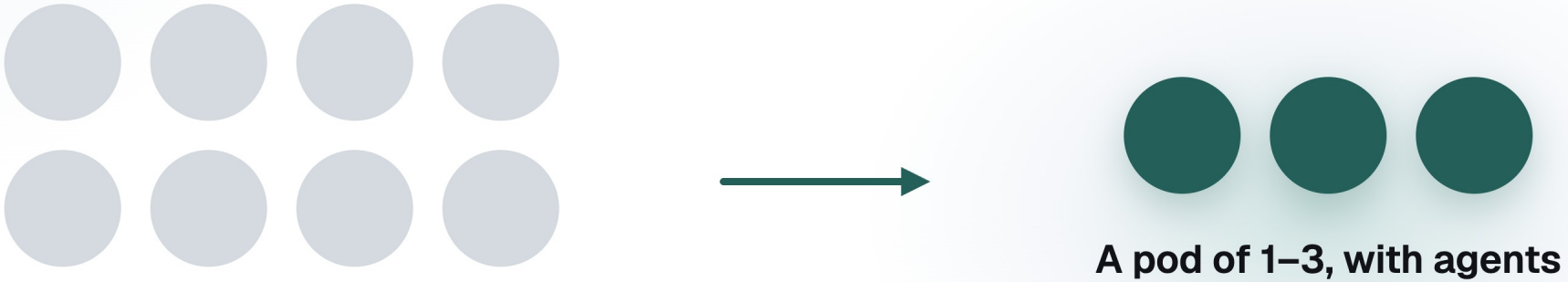


Organizational readiness for agentic development

Denis Salnikov · agileexpat.com

Spring 2026: the bet

The goal: deliver much more, much faster. The assumption: the organization has to change before the agents pay off.



A team of 6–8

A pod of 1–3, with agents

Every engineer DRI-ready. Not so everyone works alone, so that no pod depends on one person.

Results

Product work got faster

We measured output, not outcome. On purpose, for now.

Typical product epic

34 → 16

days, median

Slowest sixth of product epics

55 → 37

days, 85th percentile

Inside our 30-day target

43 → 78%

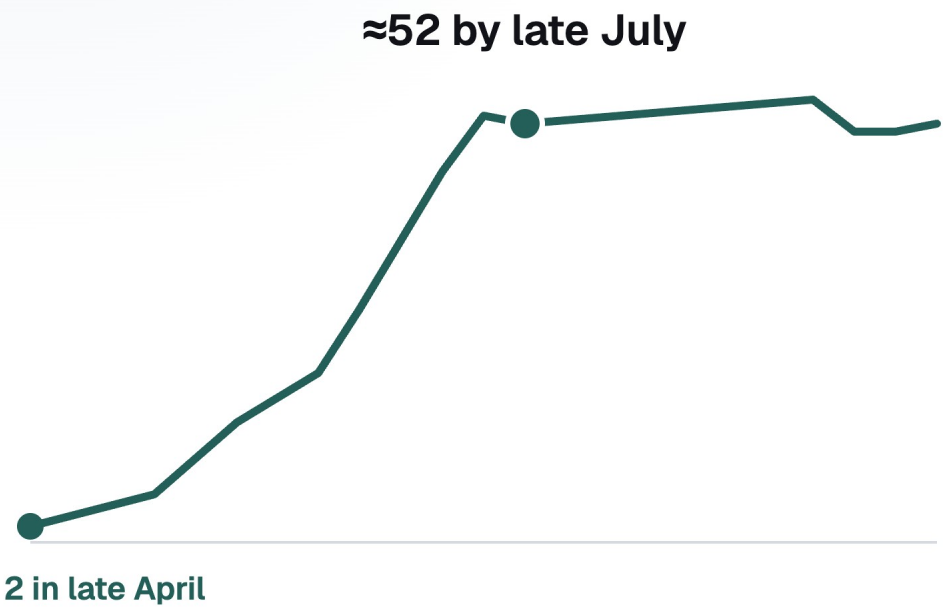
of product epics

And quality held. Bug fixes went from **28%** to **22%** of merged changes, while the number of changes doubled.

Half the org went AI-native.
Few teams orchestrate.

How much · AI-native engineers

95%+ of merged work AI-assisted, out of ~100



How well · where we are on the ladder

Eight rungs of agentic engineering · our estimate

T8 · Agents in production	A few individuals
T7 · Fleet operations	A few individuals
T6 · Orchestration	A few teams
T5 · Checkpointing and hardening	A few teams
T4 · Loop until done	**Most of us**
T3 · Context management	Everyone
T2 · Shaping and slicing	Everyone
T1 · Professional prompting	Everyone

Agile Expat

Apr → Sep 2026 · weekly AI-adoption measurement from merged changes · ladder placement is our estimate

What else the agents unlocked

Beyond a narrow hiring funnel

Strong engineers from any stack. Agents carry the language learning curve.

Multi-specialist engineers

One engineering role across the stack. Agents fill the gaps.

Quality shifts left

Engineers own testing. AI reviews specs and changes.

Engineers in discovery

Engineers join discovery much earlier, not only delivery. Every engineer is capable to do so.

Rapid prototyping

Clickable prototypes inside the real product, before moving the idea to delivery is decided.

Cross-domain contribution

Another domain's code is easy to work in. Approval is now the wait.

Where we are today



Today · B3, missions in C

Pods own a business area end to end.

The company's top bets borrow them for a while.

The tools made it possible. Organizing around them made it pay off.

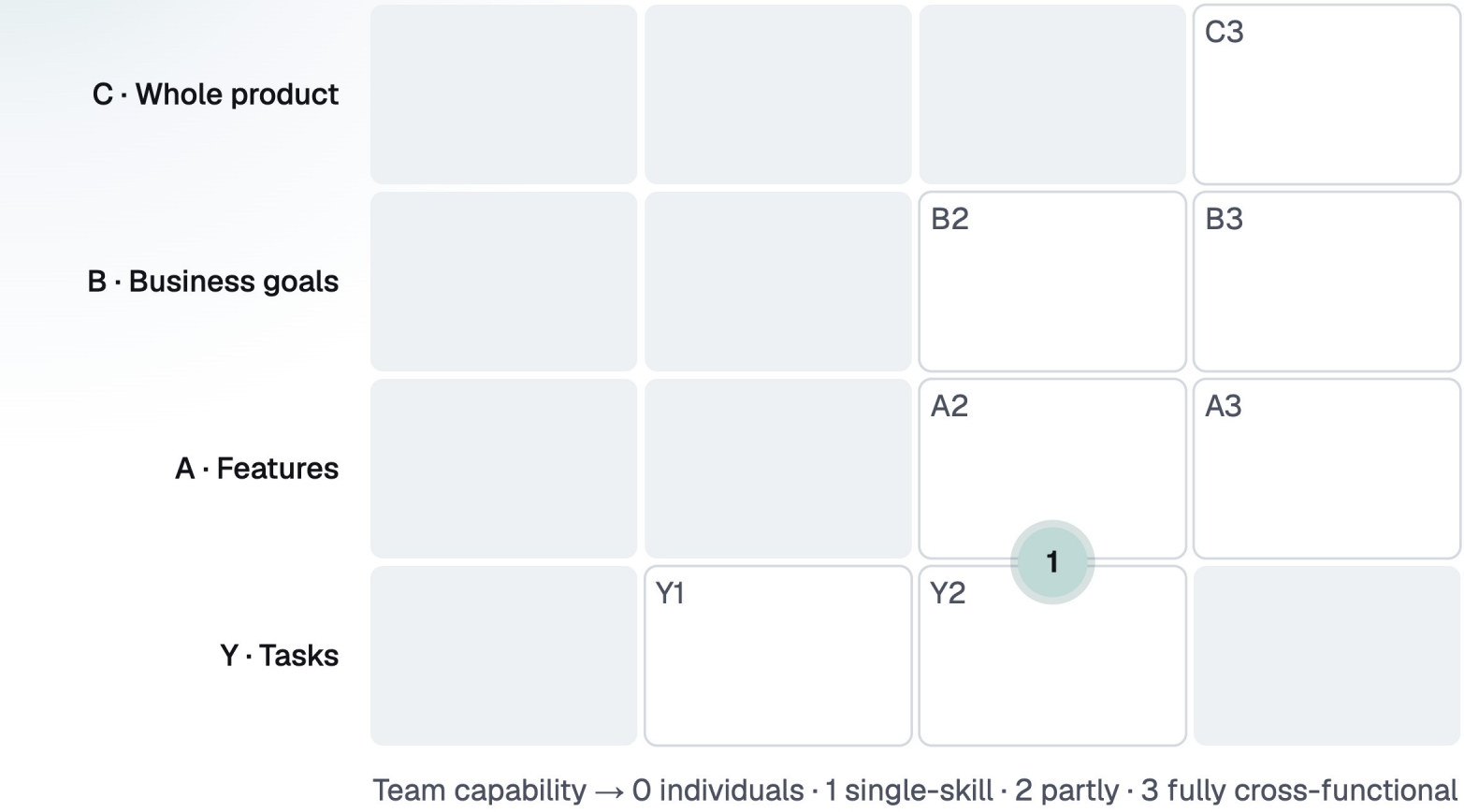
Enablers

Agents, spec-driven development, per-domain context, a modular codebase

Org design decisions

Who owns what, how big the unit is, which roles exist, who decides, how bets get staffed

Our path on the Org Topologies map



Mid 2024 · Y2 / A2

Teams own components.

Work waits: at every hand-off between teams.

Our path on the Org Topologies map



Mid 2024 · Y2 / A2

Teams own components.

Work waits: at every hand-off between teams.

Move 1 · Ownership → A2 / B2

A cluster owns a business process end to end.

Work waits: between roles inside the cluster: FE, BE, QA.

Move 1 · Ownership · 2025

What we did

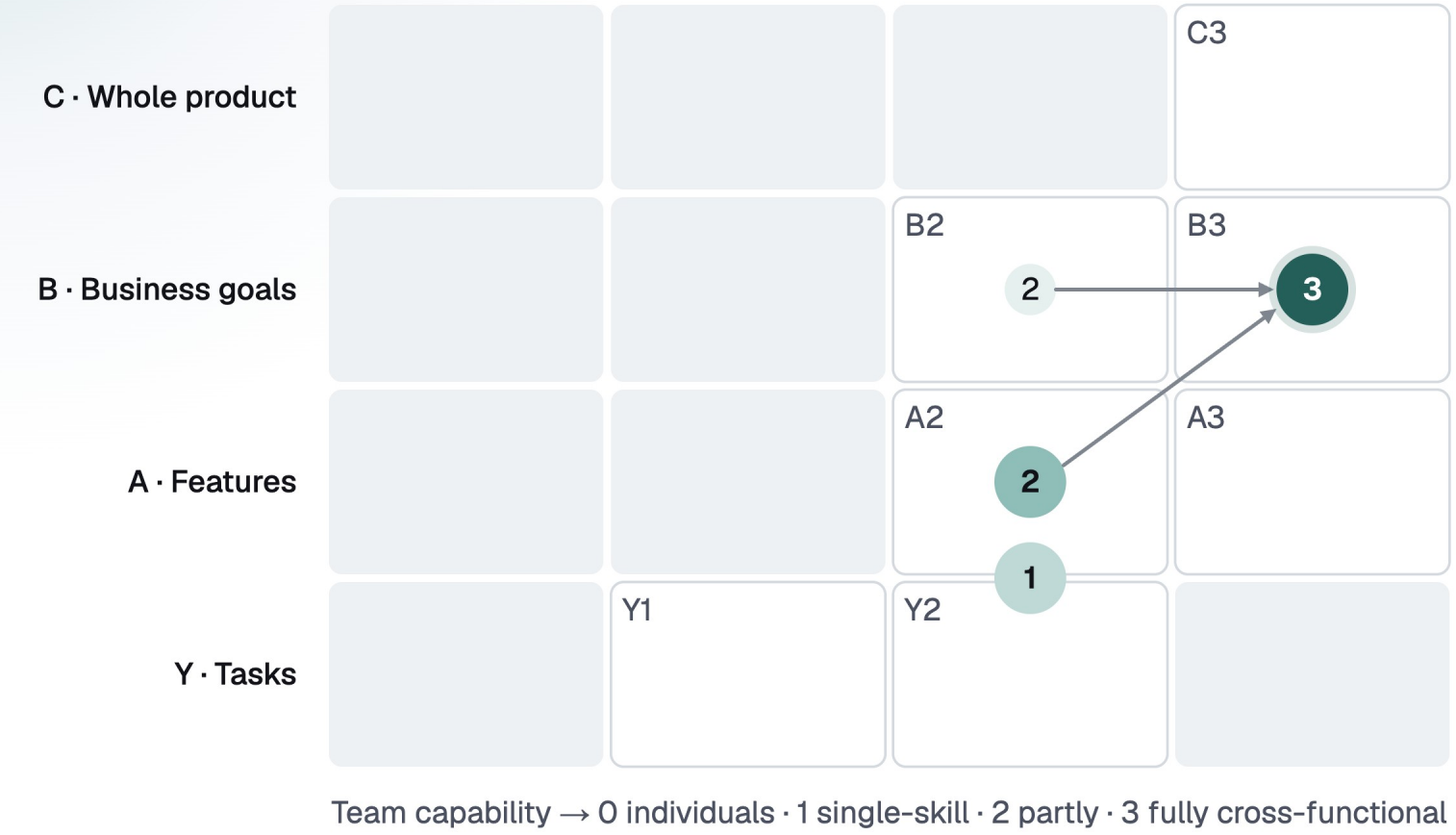
A pilot cluster **owns a whole business process**, and is renamed after it

One working standard: WIP limits, vertical slices, review turnaround, 3 Amigos

Teams assess themselves against the list of standard practices agreed with leadership

Feature Lead experiment: one engineer owns a feature end to end

Our path on the Org Topologies map



Mid 2024 · Y2 / A2

Teams own components.

Work waits: at every hand-off between teams.

Move 1 · Ownership → A2 / B2

A cluster owns a business process end to end.

Work waits: between roles inside the cluster: FE, BE, QA.

Move 2 · Unit, roles, DRIs → B3

Pods of 1–3, one engineering role, DRIs instead of team leads.

Work waits: outside the pod: cross-domain design, release gate.

Agile Expat

Move 2 · Unit, roles, DRIs · 2026

What we did

Org design

- Teams of 6–8 → **pods of 1–3**
- Frontend + backend → **one engineering role**
- Team leads → **DRIs**; every engineer DRI-ready
- The DRI **joins every idea** once its validation starts

Enablers

- Codebase **split into domains**
- **Agent context** written for each domain
- **Two months** of capacity on foundations
- Hiring open to engineers **from any stack**

What got us here won't get us further.

Everything we changed made the pod faster. The next gains sit outside the pod.

Work waits where it crosses a line

Requirement review

A technical design takes more than twice as long to agree. On purpose: our safety net while code volume doubles.

The release gate

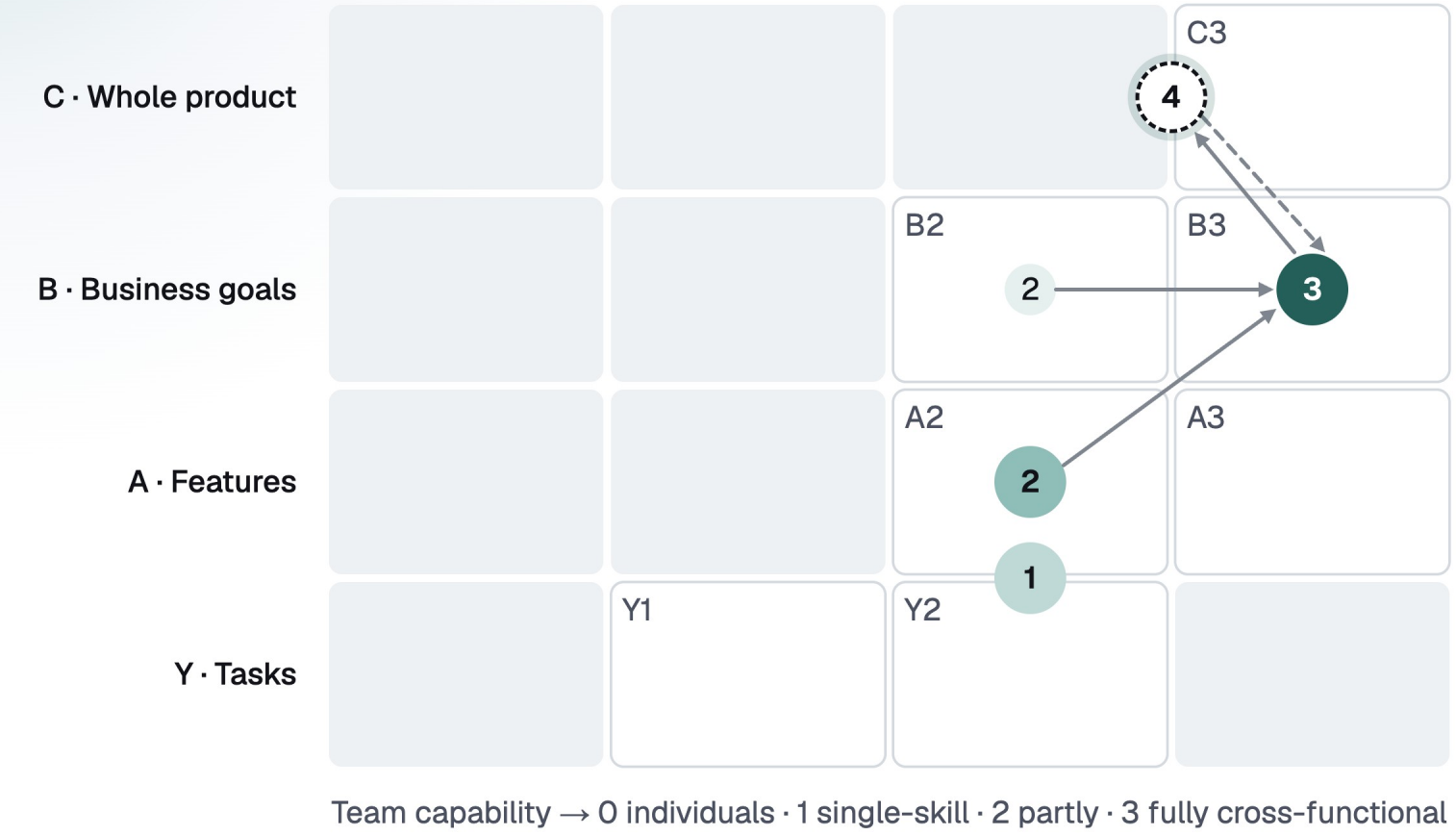
Finished work waits longer to ship than it did before agents.

Work no pod owns

The slowest work is the foundation we keep moving down the list.

Not in code review: time per change stayed flat while the number of changes almost doubled.

Our path on the Org Topologies map



Mid 2024 · Y2 / A2

Teams own components.

Work waits: at every hand-off between teams.

Move 1 · Ownership → A2 / B2

A cluster owns a business process end to end.

Work waits: between roles inside the cluster: FE, BE, QA.

Move 2 · Unit, roles, DRIs → B3

Pods of 1–3, one engineering role, DRIs instead of team leads.

Work waits: outside the pod: cross-domain design, release gate.

Move 3 · Bet staffing → C

One company list of bets, staffed first. The top ones pull pods from anywhere.

Work waits: less, for that bet. Open question: the return date.

Agile Expat

Who owns each rung?

T8 · Agents in production	Whole org (C)
T7 · Fleet operations	Whole org (C)
T6 · Orchestration	Domain: pods + cluster
T5 · Checkpointing and hardening	Domain: pods + cluster
T4 · Loop until done	Pod
T3 · Context management	Domain: pods + cluster
T2 · Shaping and slicing	Pod
T1 · Professional prompting	One person

Pods can climb six rungs. The top two require company-wide platforms, guardrails and adaptation.

Change the org, not the tools

Readiness

Grow owners, not just AI users

Product thinking is everyone's job, not only the former team leads'. Who takes the role is the pod's call.

Gates

Choose your gates, and revisit them

Every mandatory gate is a choice. Give it an owner, a visible turnaround and a regular "still worth it?"

Foundation

Check the ground is agent-ready

Domains with clear boundaries, context the agents can read. Assess it before you scale the agents.

Three questions to take home

Who in your org is ready to **own product work end to end**, and is it only your leads?

Which of your gates did you choose, and when did you last ask **if they're still worth it**?

Are your **architecture and repos ready for agents**, or only your people?

Agile Expat

The agents are ready.
Is your organization?



Thank you · Denis Salnikov

agileexpat.com

denis@agileexpat.com

linkedin.com/in/dsalnikov23