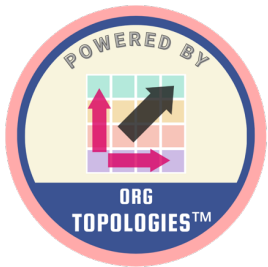


Org Topologies: Elevating Organizations with Human and Artificial Intelligence

with thoughtful org design choices



2024, Alexey Krivitsky



The *last* agile transformation.

The Agile Landscape v10

Developed by Christopher Webb



The Agile Industrial Complex

“Agile Industrial Complex” refers to the **commercialization** and **institutionalization** of Agile methodologies, where the focus shifts from Agile’s core principles to a more process-driven, **profit-oriented approach**.

This phenomenon involves the proliferation of proprietary **frameworks, certifications**, and **consulting** services that may prioritize **revenue** over genuine Agile practices.

martinfowler.com/articles/agile-aus-2018.html

How big are we talking?..

In 2022, this market was valued at approximately **\$27 billion**

with projections indicating growth to around **\$142 billion by 2032**, reflecting a compound annual growth rate of 18%.

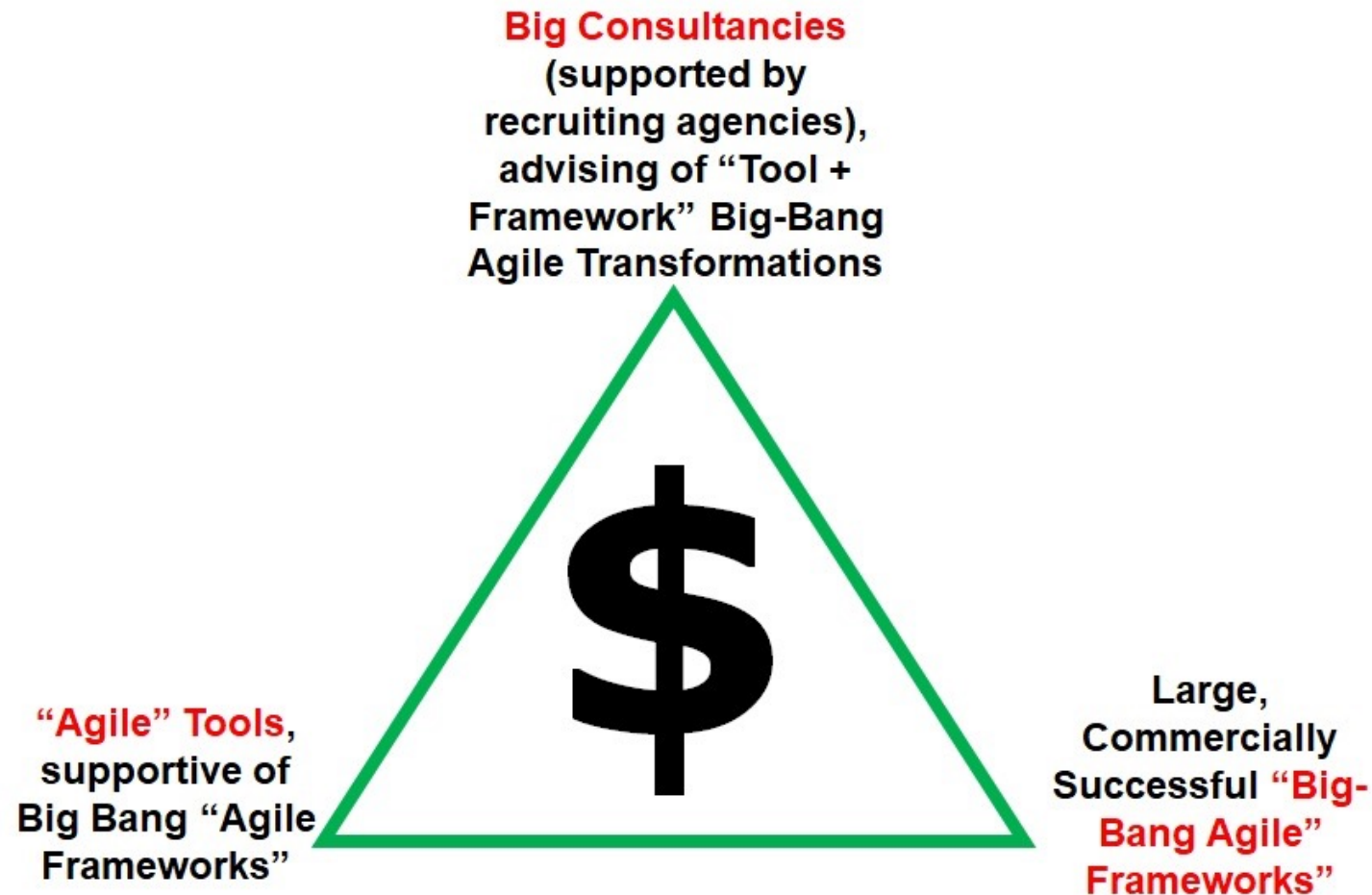
The Enterprise Agile Transformation Services Market is expected to reach **USD 31 billion in 2024** and is projected to grow at a CAGR of 17%.

It is comparable to the 2023 **GDPs** of countries like **Iceland** and **Malta**.

In short, that places Agile transformation **on par with established, globally recognized industries**.

Imagine companies spending
that money on actually
improving their products!

“Triple Taxation” on Client Companies



Gene Gendel

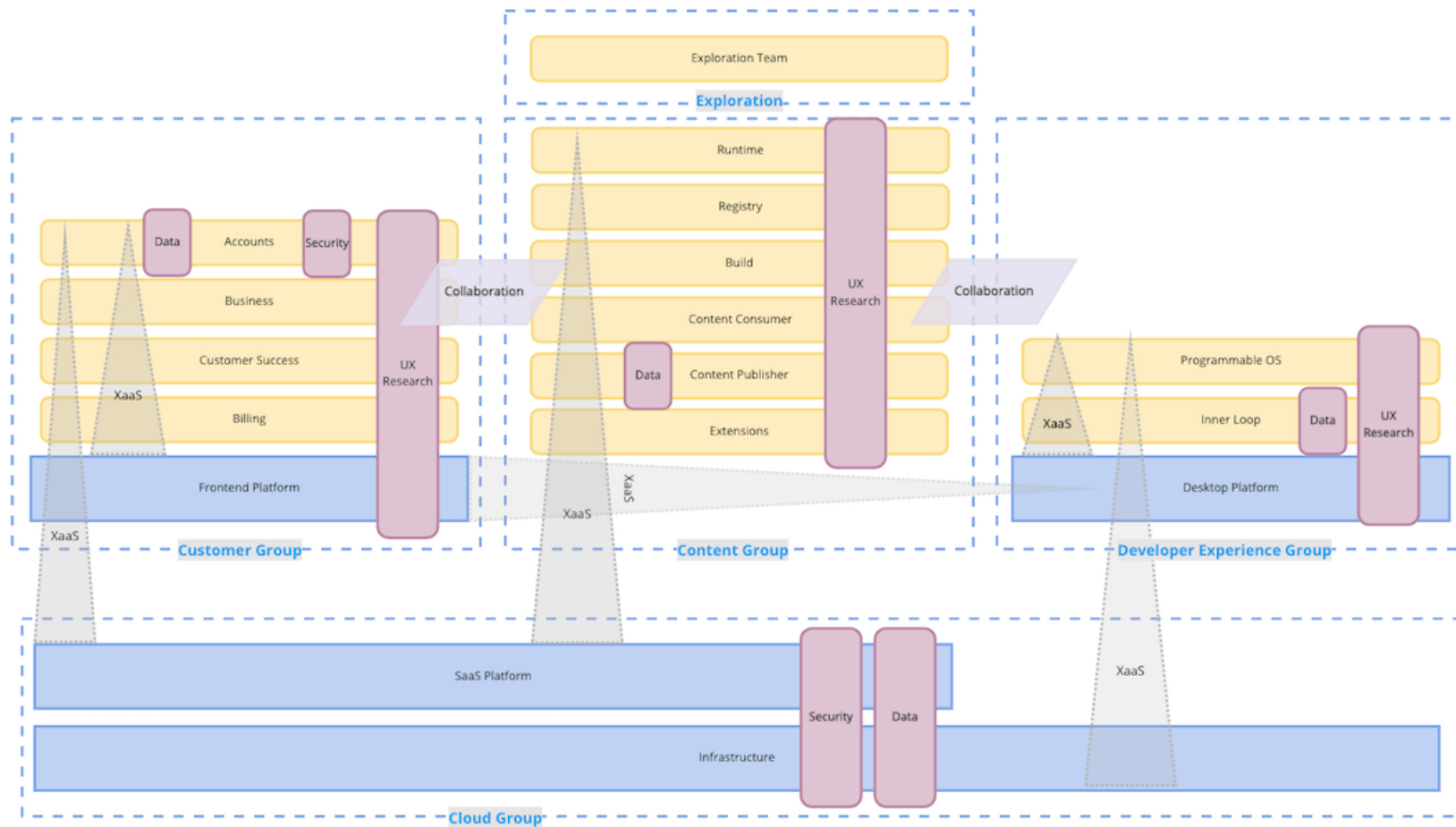


Brought to you by www.keystepstosuccess.com

Hopefully, once transformed,
the orgs will *finally* be agile,
right?

It depends ...

"Scaling by Splitting Stream-Aligned Teams and Forming Platform Teams"



www.docker.com/blog/building-stronger-happier-engineering-teams-with-team-topologies/

"Scaling by Splitting Stream-Aligned Teams and Forming Platform Teams"

Our new Stream-Aligned Teams branched mainly from **preexisting teams**.

For example, we split the Accounts & Growth Team into **Accounts**, **Business**, **Billing**, and **Customer Enablement** Teams.

"Scaling by Splitting Stream-Aligned Teams and Forming Platform Teams"

Our new Stream-Aligned Teams branched mainly from **preexisting teams**.

For example, we split the Accounts & Growth Team into **Accounts, Business, Billing**, and **Customer Enablement** Teams.

Scaling introduces complexity and challenges that make **team interactions and needs unpredictable**.

”Fighting the Conway’s Law”.

“Conway’s Law”

Organizations which design systems (in the broad sense used here) are constrained to produce **designs which are copies of the communication structures** of these organizations.

Conway's Conclusions

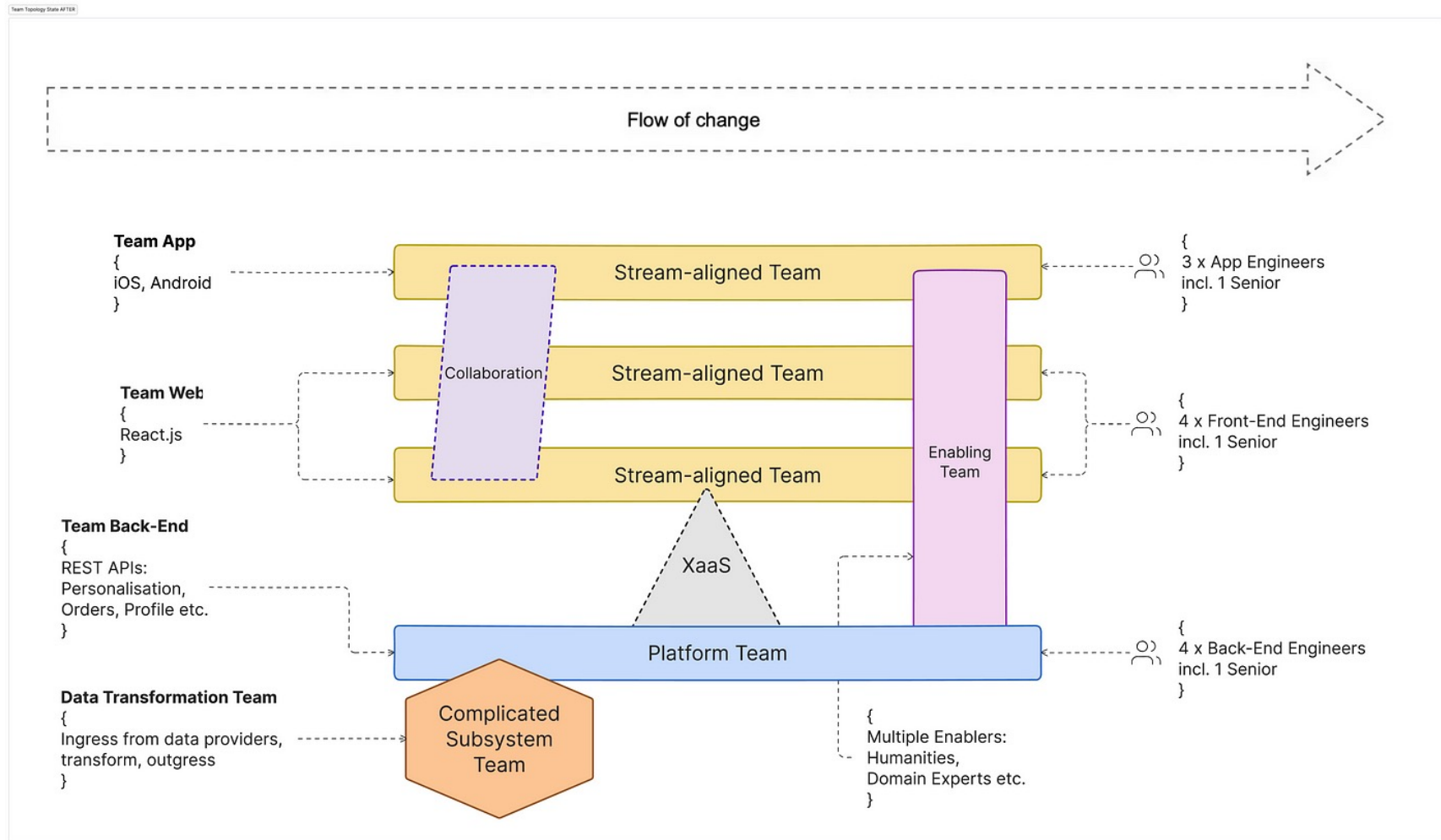
Because the design which occurs first is almost never the best possible, the prevailing system concept may need to change.

Therefore, **flexibility of organization is important to effective design.**

(Watch Craig's keynote at LeSS Madrid 2024, once it is live)

https://www.melconway.com/Home/Committees_Paper.html

“Case Study: Efficient Software Delivery with Team Topologies”



<https://medium.com/@alanhaarhoff/case-study-efficient-software-delivery-with-team-topologies-bfa9995c5d91>

“Case Study: Efficient Software Delivery with Team Topologies”

In fact, by fixing teams to components and “value streams” such OD reinforces the Conway’s Law.



<https://medium.com/@alanhaarhoff/case-study-efficient-software-delivery-with-team-topologies-bfa9995c5d91>



**YOU WERE SUPPOSED TO FIGHT EVIL,
NOT JOIN IT**

makeameme.org

That seems like a rather inner-focused view of the transformation.

OD is baked into Current Architecture



Can this by any chance
be the *last* transformation?

OK, maybe the techies didn't get it?

Let's talk to the product people!

"Optimizing teams: A dual perspective on Team Topologies"

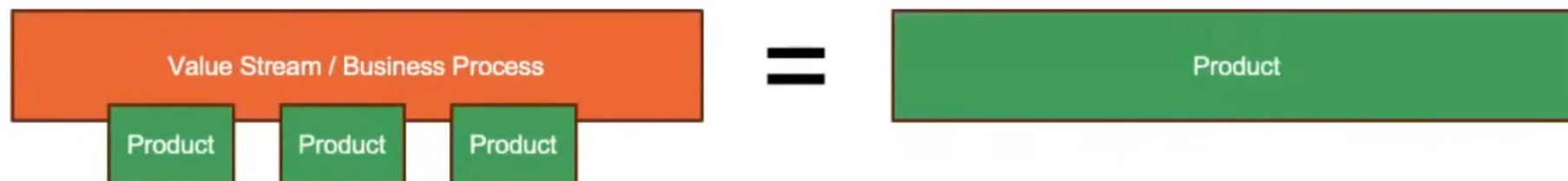
1. Well-defined scope
2. Disambiguate cross-team boundaries
3. Distinct value contribution
4. Strong product leadership
5. Clear user focus
6. End-to-end responsibility
7. Minimize dependencies
8. Effective goal setting and alignment
9. ...



<https://www.mindtheproduct.com/optimizing-teams-a-dual-perspective-on-team-topologies/>



A Simple Approach To Product Definition



- Each customer value stream becomes a product
- Internal user value streams become products
- Shared services become products
- Over time optimize
 - Reducing dependencies
 - Core vs context

internal user value stream becomes a product.

So, once all the teams have a well-defined scope and know their *boundaries* with end-to-end *responsibilities* on clear *user focus* ...

Once *everything is a product* ...

... then, perhaps it is the *last* one?

DISCUSSION TIME – CONTINUE THIS SENTENCE:

GIVEN

the teams are fixed to own narrow product parts

WHEN

the product strategy changes

THEN

...

GIVEN

the teams are fixed to own narrow product parts

WHEN

the product strategy changes

THEN

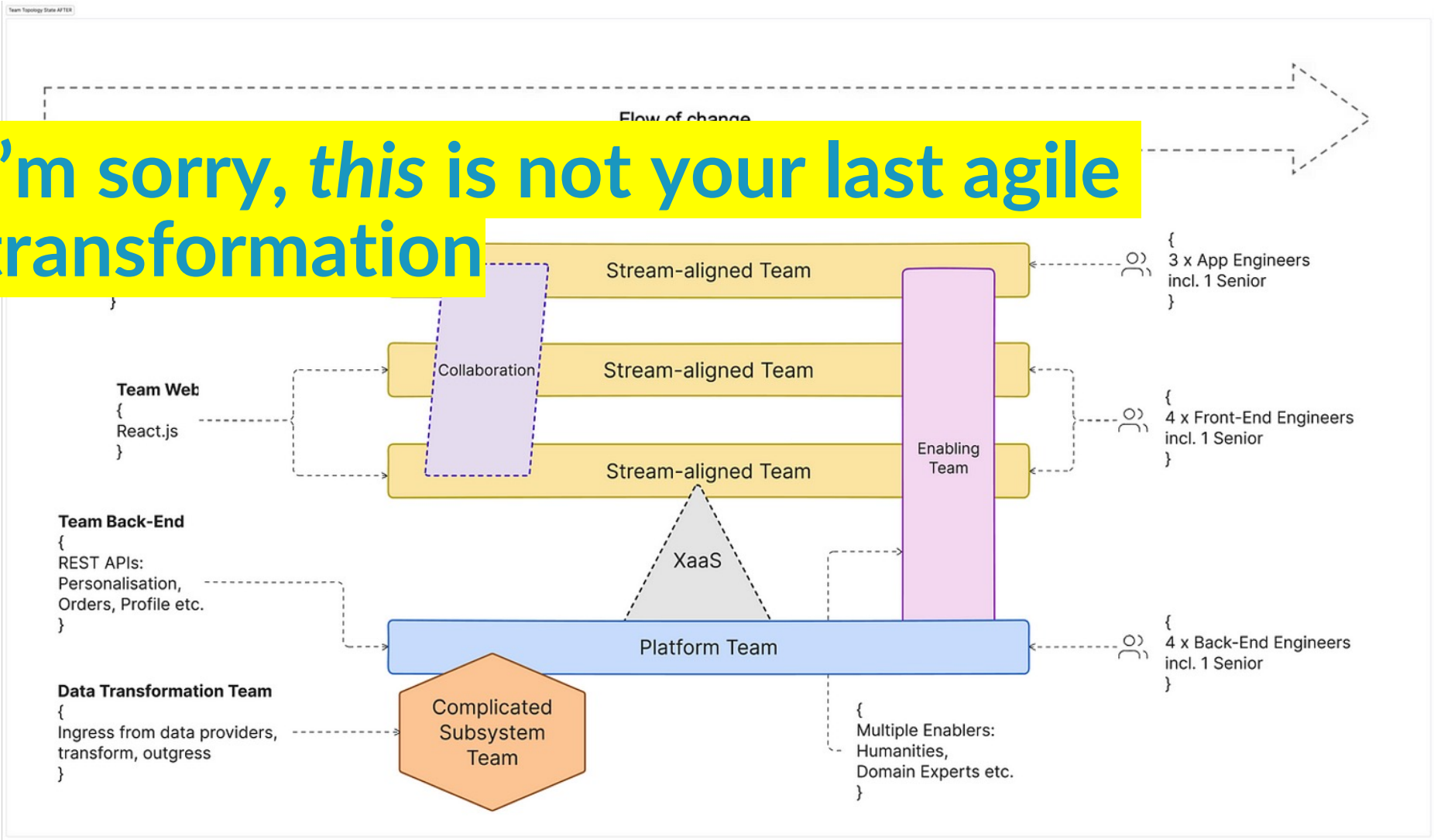
we need to re-adjust the teams



The Agile Industrial Complex

...

I'm sorry, *this* is not your last agile transformation



I'm sorry, *this* is not your last agile transformation

And this is not the last time you're being taxed!

Team Back-End

{
REST APIs:
Personalisation,
Orders, Profile etc.
}

Data Transformation Team

{
Ingress from data providers,
transform, outgress
}

Stream-aligned Team

{
3 x App Engineers
incl. 1 Senior
}

Stream-aligned Team

Enabling Team

{
4 x Back-End Engineers
incl. 1 Senior
}

"Triple Taxation" on Client Companies

Big Consultancies
(supported by
recruiting agencies),
advising of "Tool +
Framework" Big-Bang
Agile Transformations

"Agile" Tools,
supportive of
Big Bang "Agile
Frameworks"

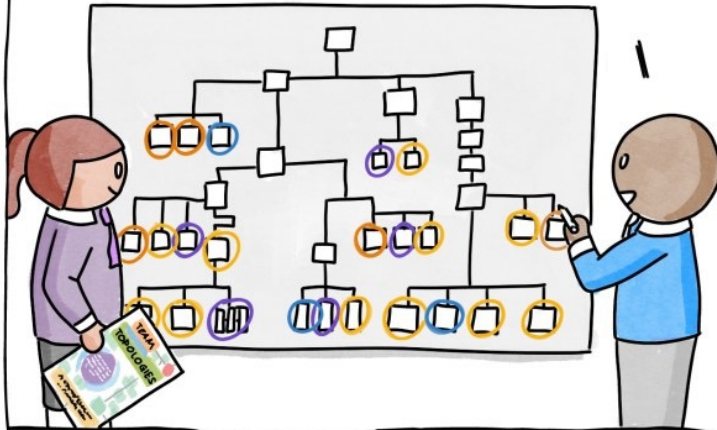
Large,
Commercially
Successful "Big-
Bang Agile"
Frameworks"



Brought to you by www.livestrongtosuccess.com

Comic Agilé

There. Our teams are now aligned with the four Team Topologies team types.



Later

I like the team types and all, but calling ourselves a Complicated Subsystem Team hasn't helped reduce all the other teams' dependencies to us.



None of the other teams build up any of our knowledge or skills. It doesn't help us nor them become a Stream-Aligned Team.

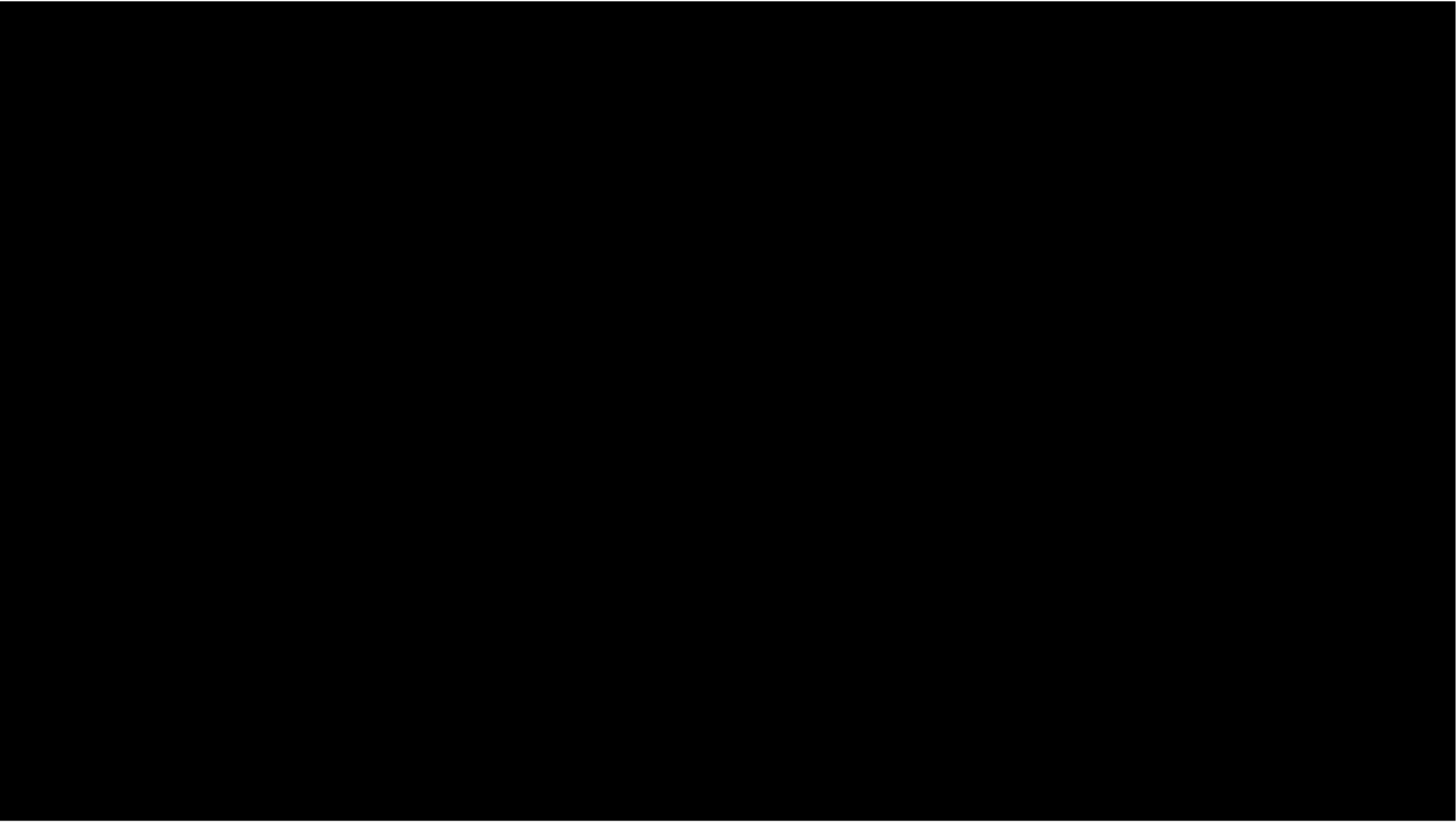
So, what do you suggest?



That we stop categorizing our teams and share competencies, instead.

No! Do you know how expensive those books were?!





We're going to see this pendulum swing because **there's so much hype around products right now ... no, not everything is a product.**

I'm kind of excited by the idea of these 30 to 50 person pods, if that would happen, because I think that that maybe is the next level of doing this.

I mean, like **with 50-people pods, you can do something pretty meaningful.**



John Cutler

30K Followers

Multiple hat-wearer. Prod dev nut. I love wrangling complex problems and answering the why with qual/quant data. @johncutlefish on Twitter.

<https://findingourway.design/2024/10/08/49-unraveling-complexity-in-product-development-ft-john-cutler/>

What a fresh idea!



Here's what "pods"
literally mean in English

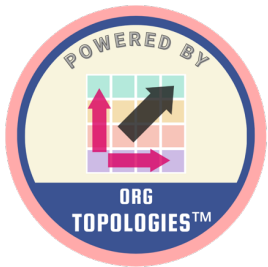
Very fresh!

We need more fresh ideas!

Let me bring one tool to put
everything in perspective

Org Topologies: Elevating Organizations with Human and Artificial Intelligence

with thoughtful org design choices



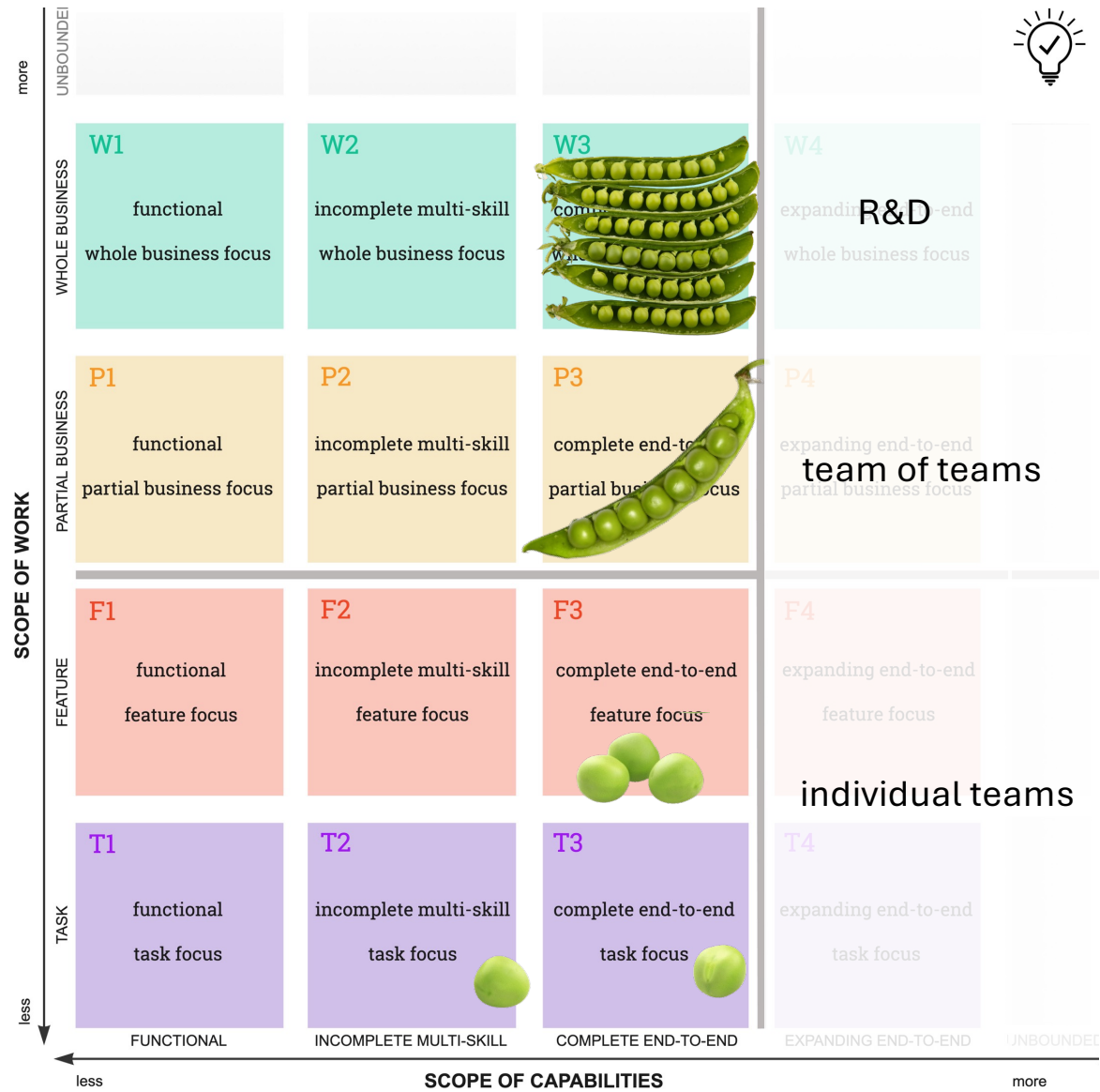
2024, Alexey Krivitsky



ARCHETYPES OF ORG TOPOLOGIES



MAPPING PODS



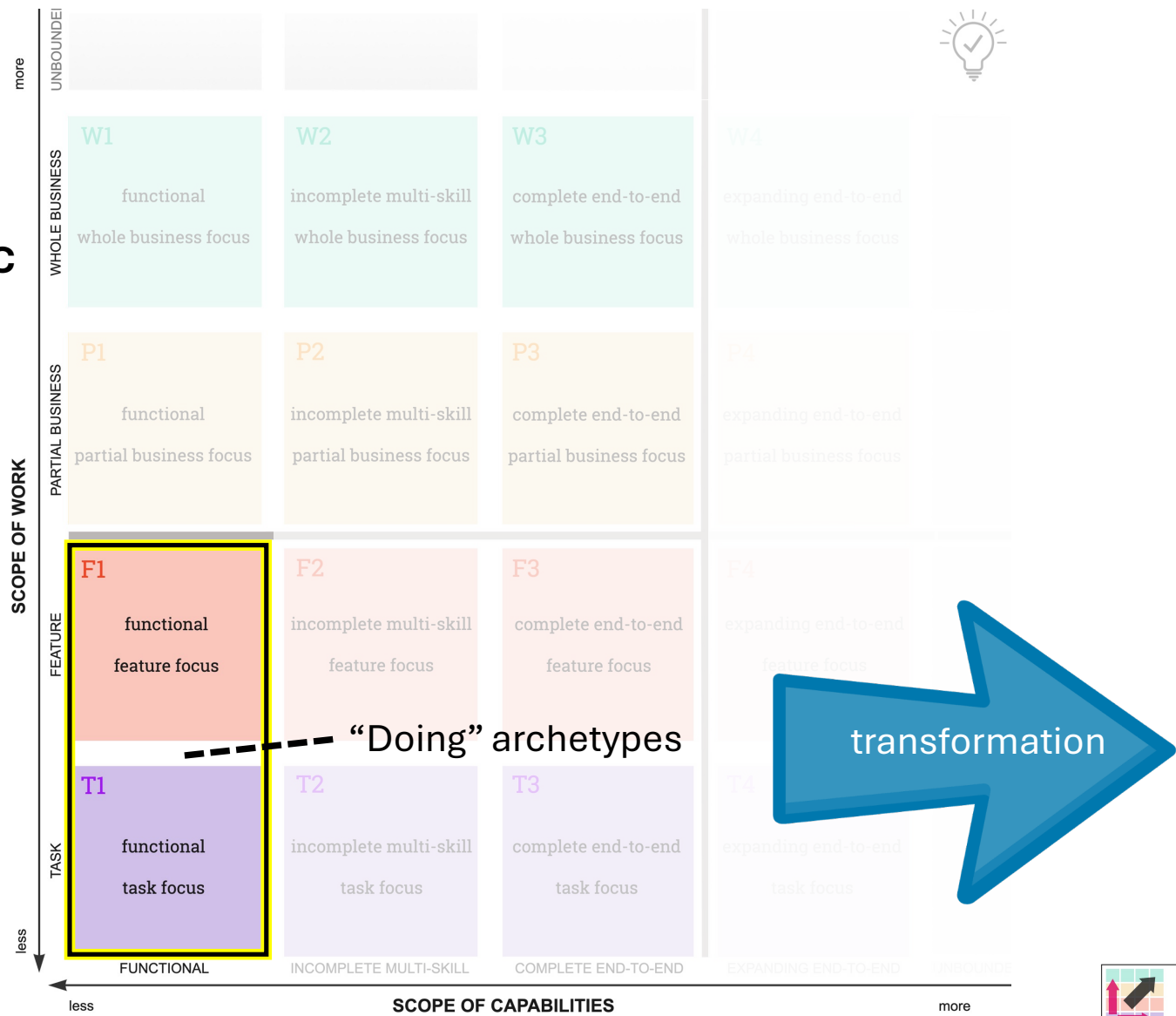
A TOOL TO MAP
ANY ORG DESIGN

AND DIRECTION
OF ORG CHANGE



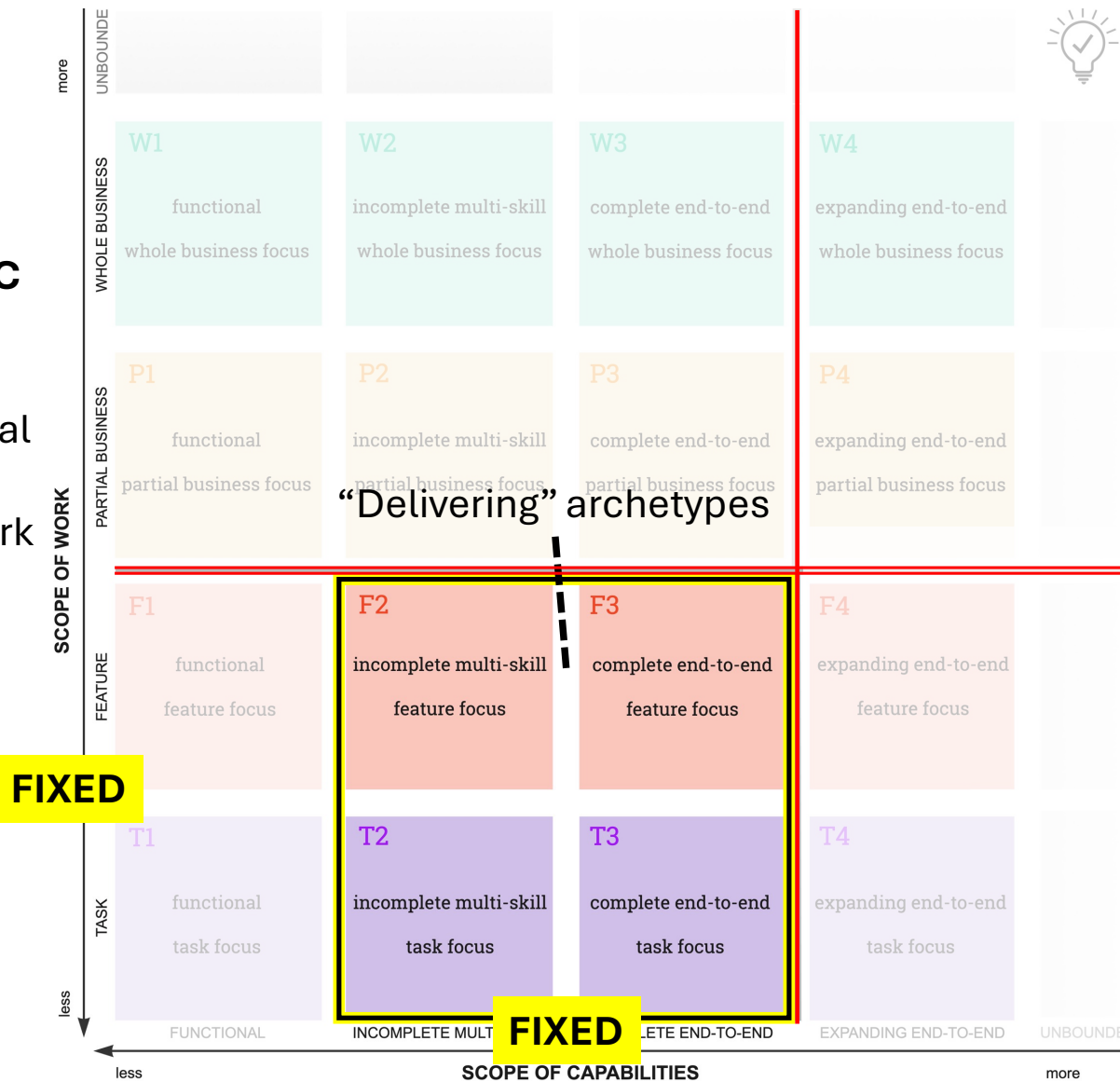
RESOURCE-CENTRIC
ORG TOPOLOGY

with people treated
as resources

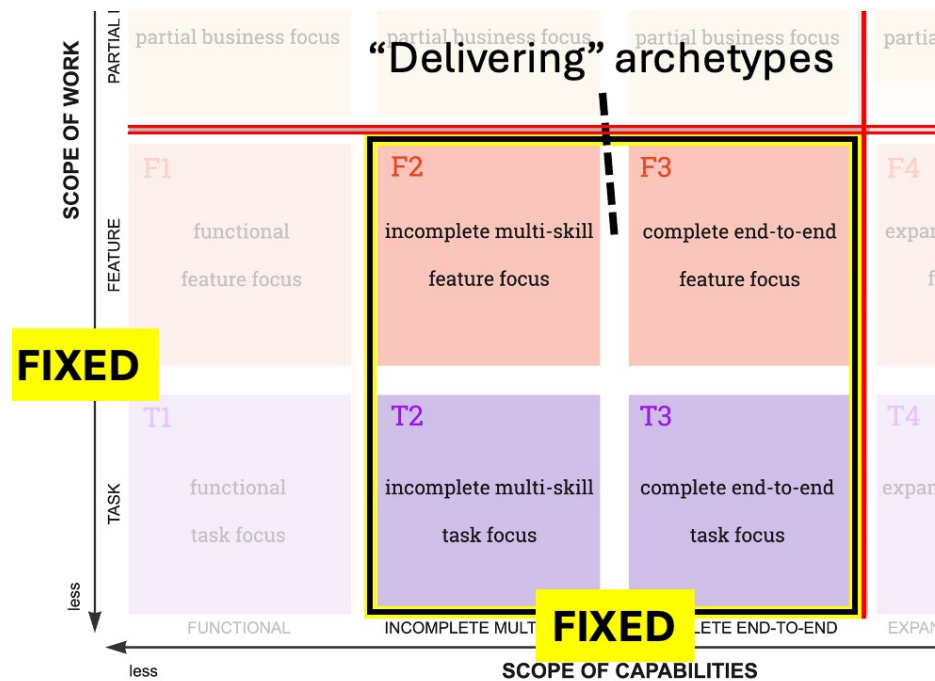


DELIVERY-CENTRIC ORG TOPOLOGY

with cross-functional
teams with fixed &
narrow scope of work



Suffering in product development



We need fresh ideas!

A fresh idea? **Scrum!**

Moving the Scrum Downfield

From interviews with organization members from the CEO to young engineers, we learned that leading companies show six characteristics in managing their new product development processes:

1. Built-in instability
2. Self-organizing project teams
3. Overlapping development phases
4. “Multilearning”
5. Subtle control
6. Organizational transfer of learning

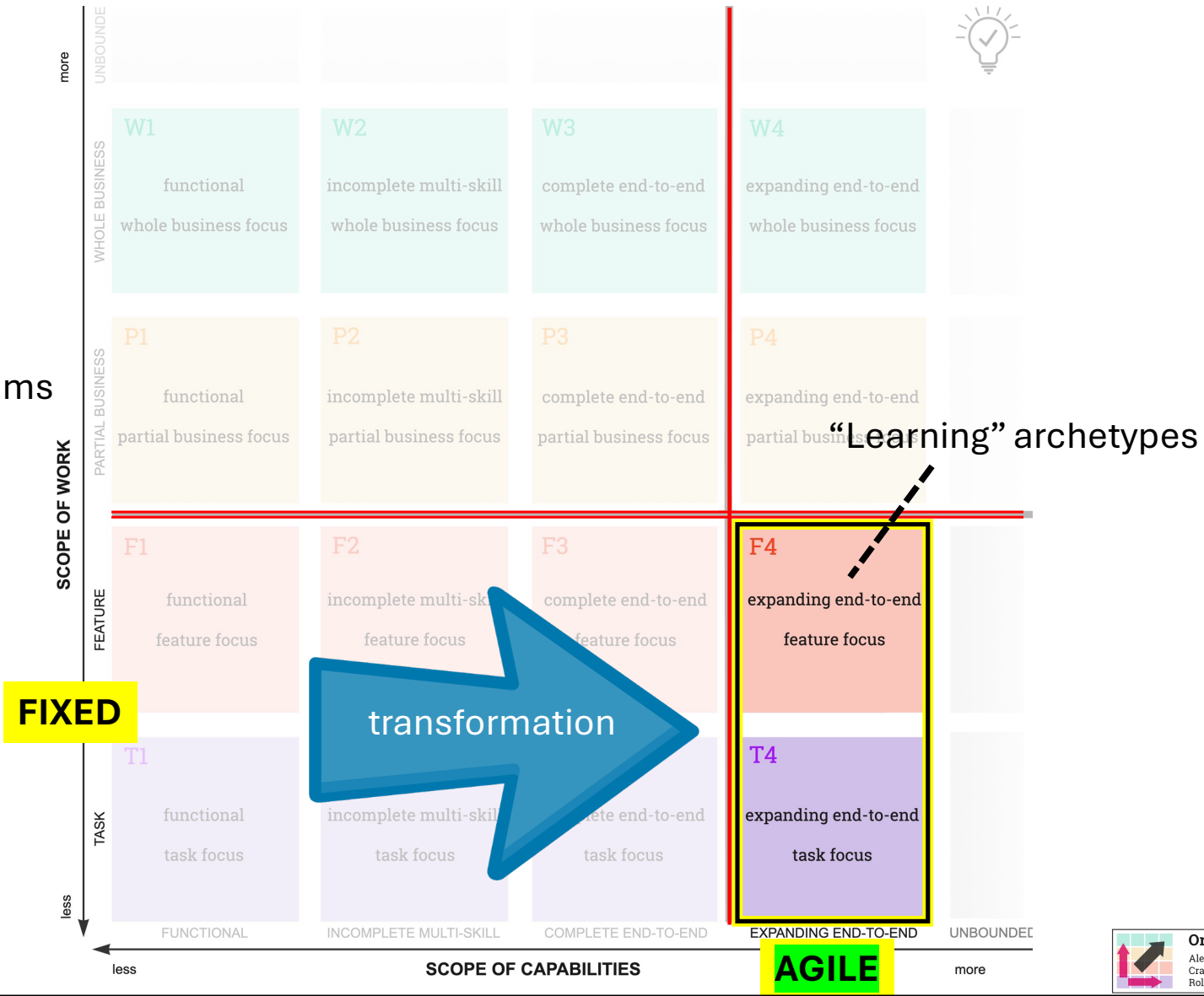
Multilearning

Because members of the project team stay in close touch with outside sources of information, they can respond quickly to changing market conditions. Team members engage in a continual process of trial and error to narrow down the number of alternatives that they must consider. They also acquire broad knowledge and diverse skills, which help them create a versatile team capable of solving an array of problems fast.

Such learning by doing manifests itself along two dimensions: across multiple levels (individual, group, and corporate) and across multiple functions. We refer to these two dimensions of learning as “multilearning.”

AGILE
ORG TOPOLOGY

with learning in teams



novelty of knowledge work

← routine expertise → learning expertise

has/will be automated by AIs

human value-add (for now)

LeSS Conference Singapore

We want to thank our wonderful sponsors!

Gold Sponsors

TITANISOFT

Jago

Silver Sponsors

Agile Consulting

Red Hat

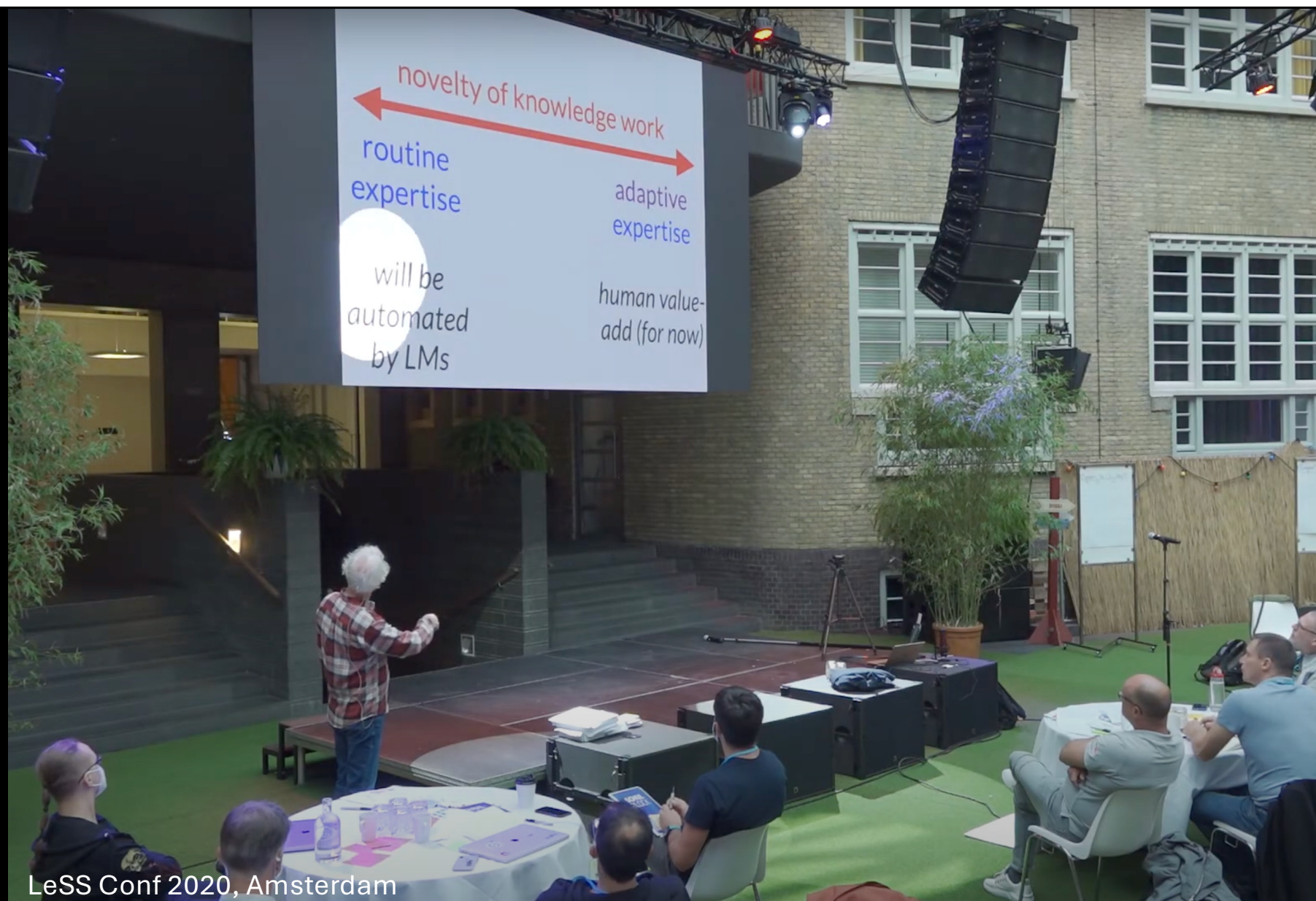
Community Partners

Agile Singapore

WOMEN DEVS

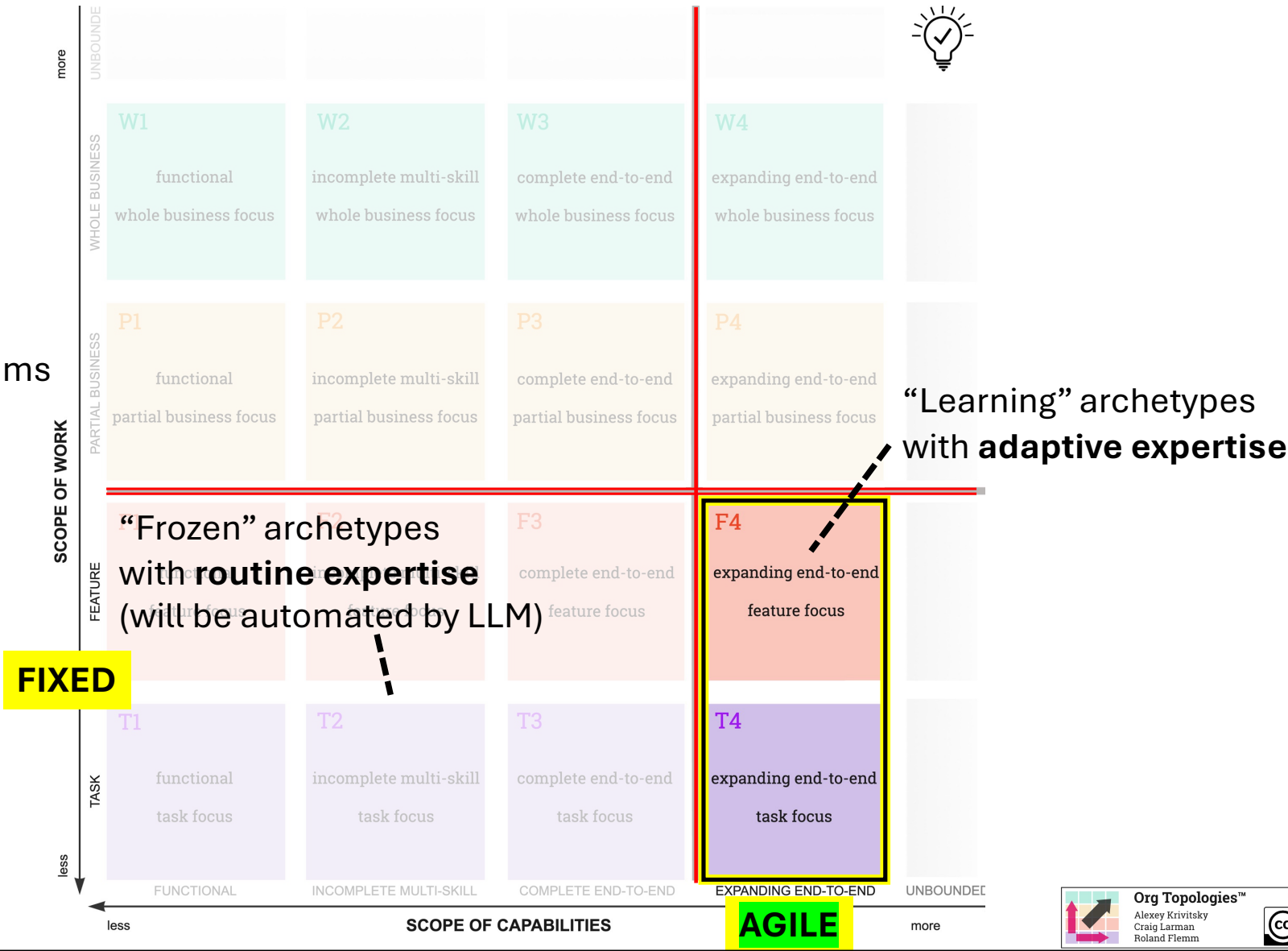
Event Co-ordinator

DBA+

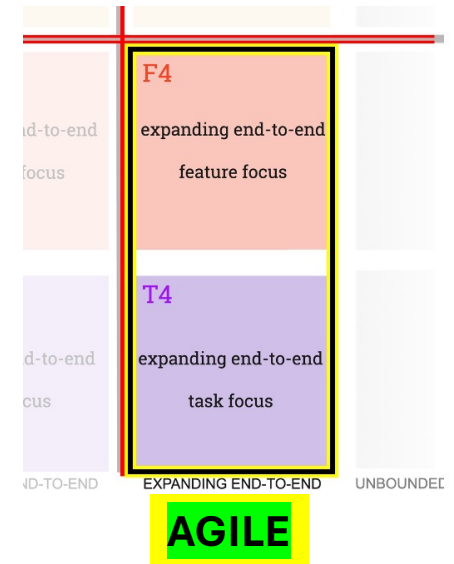
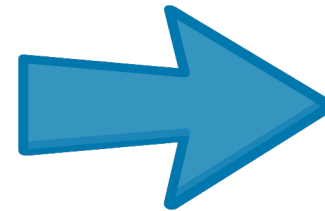


AGILE
ORG TOPOLOGY

with learning in teams



DISCUSSION TIME



From what you've heard today:

What do you want to try
in your org to move right?

We need more ideas!



Alistair Cockburn • 1st

Co-author of the Agile Manifesto, created Heart of Agile. Creatively sh...

3w •



Q&A: "Hi Alistair,

Thanks again for the Meetup last week in Utrecht! Can you explain the difference between 'agile in projects', 'adaptive around projects'? I cannot understand it still after a week :)."

Great question. To me (repeat: to /me/...) "agile" has to do with tracking rapidly changing requirements within a project - That matches the sense of the word agile, to me.

"Adaptive", to me, has to do with tracking cultural changes in the team/org over time, a slower process, once that involves changing the team's structure, norms, habits.

So, for example, Spotify is known for being agile: They put out lots of features rapidly in response to market shifts and change their internal architecture as needed. But people who read and believed Henrik Kniberg's "Spotify Model" writeup overlooked the the fact that the real Spotify Model is constantly changing the organizational structure, experimenting and changing to suit the problems they encountered with growing # employees and customer base. The true Spotify Model is not locking onto a single org model, but adapting the org to the times. Agile + Adaptive. I think you need both.

You and 100 others

6 comments • 5 reposts

https://www.linkedin.com/posts/alistaircockburn_qa-hi-alistair-thanks-again-for-the-meetup-activity-7250484785851117569-h8Ha



The point Alistair was making (1)

- "**Agile**" has to do with tracking **rapidly changing requirements** within a project
- "**Adaptive**", to me, has to do with **tracking cultural changes** in the team/org over time, a slower process, once that involves **changing the team's structure**, norms, habits.



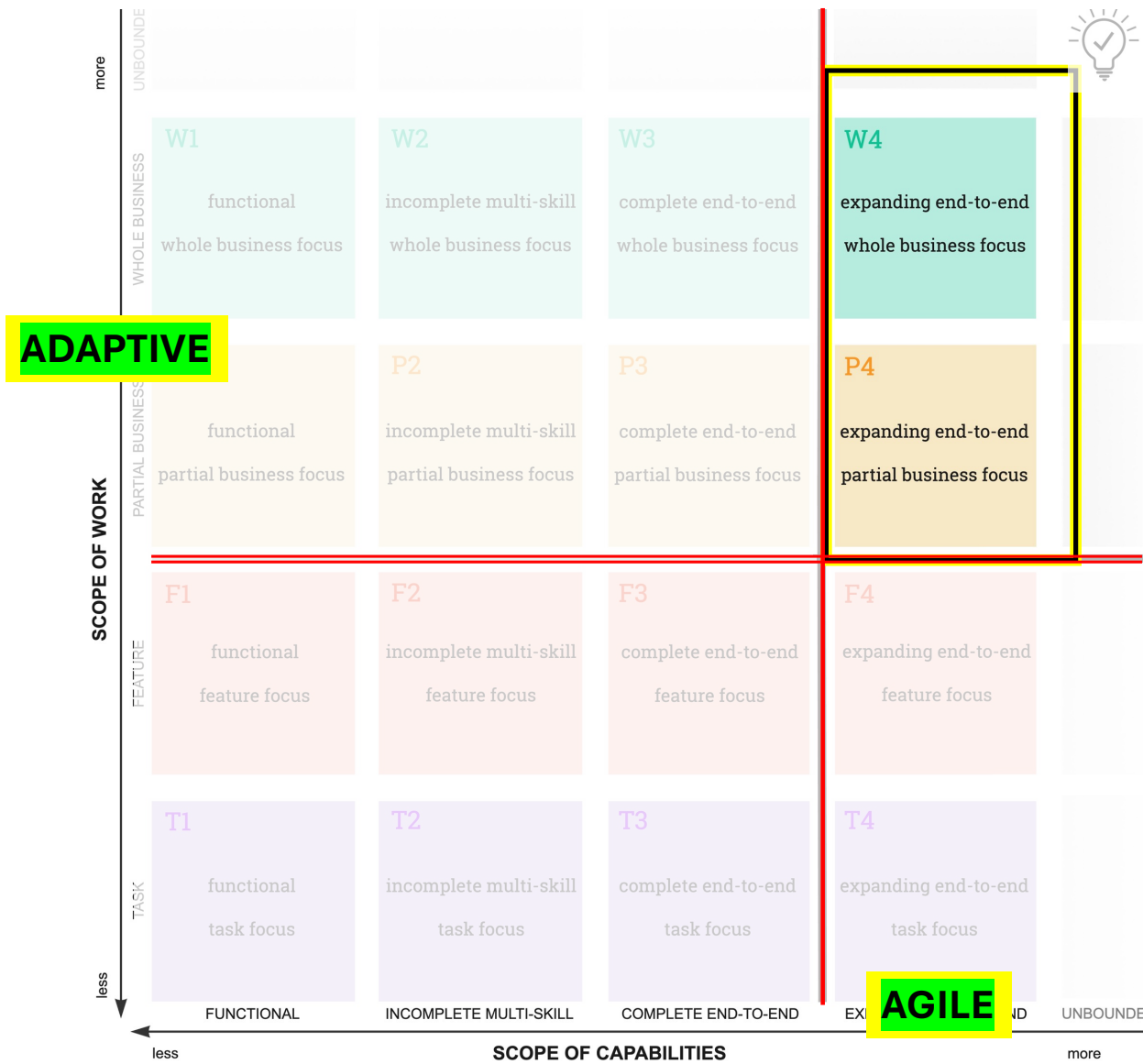
The point Alistair was making (2)

- **Spotify** is known for being **agile**: they put out lots of features rapidly in response to market shifts
- **Spotify Model is changing the org structure** constantly, experimenting and changing to suit the problems they encountered with growing # employees and customer base
- Not locking onto a single org model, but **adapting the org** to the times



The point Alistair was making (3)

- **Agile + Adaptive.** I think you need both.



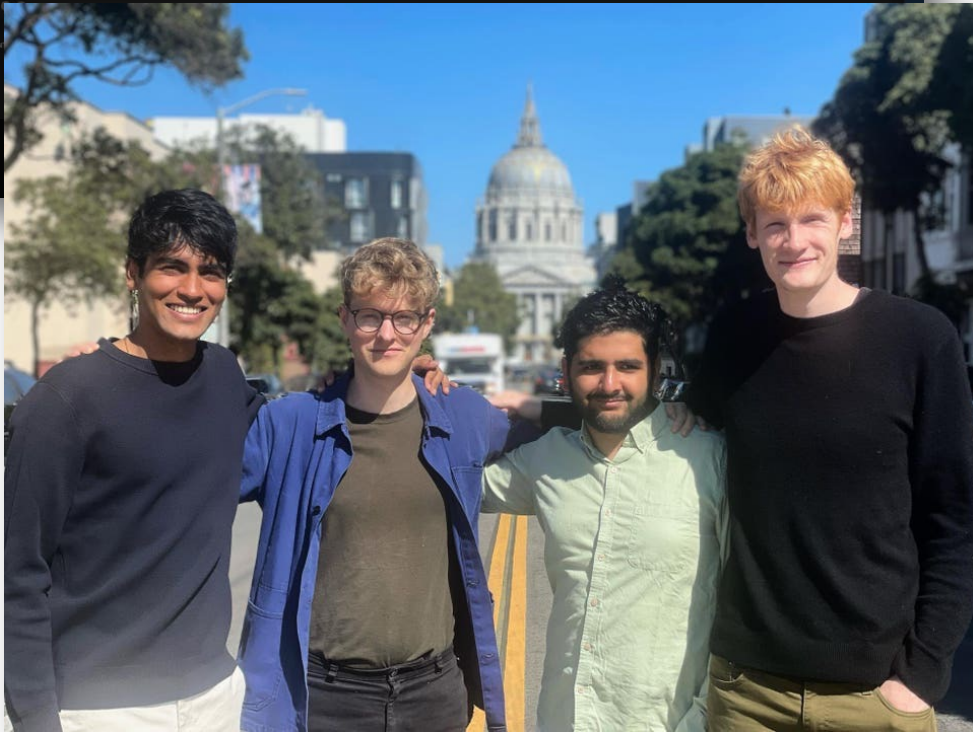
DESIRED STATE
FOR PRODUCT DEV

The AI-first Code Editor

Build software faster in an editor designed for pair-programming with AI

Download for Mac

Other



“We sort of doing it all in one...

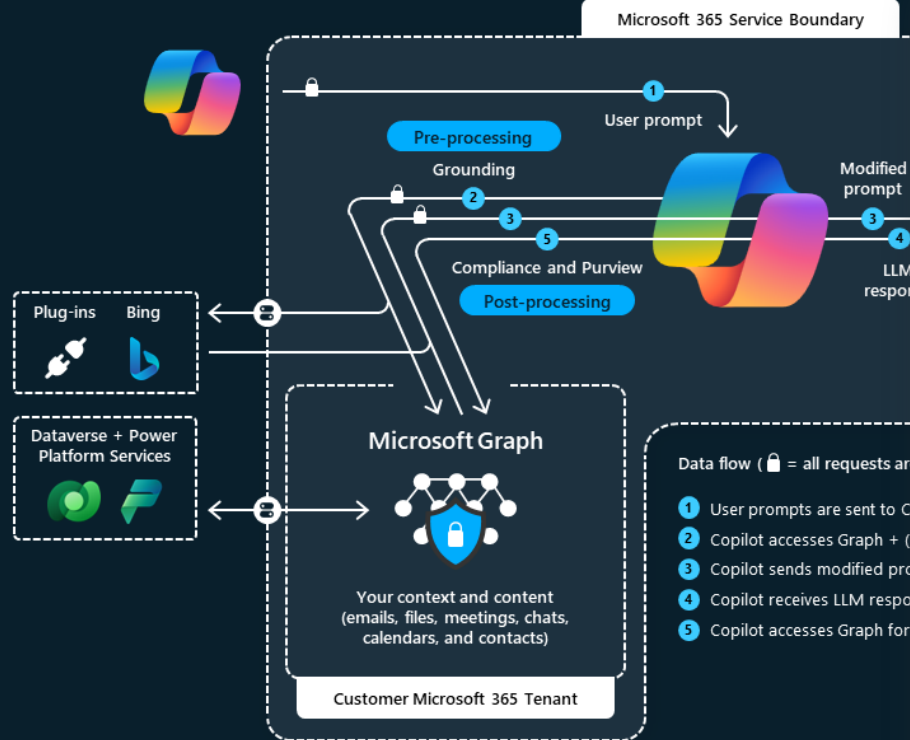
We're developing the UX and the way you interact with the model...

At the same time, we're working on making the model better: the promoter, training the model...

The same people are doing all that.”

<https://www.youtube.com/watch?v=oFfVt3S51T4>

Microsoft Copilot for Microsoft 365 architecture



Microsoft | Careers Locations Professions Programs Life at Microsoft Hiring tips

Search: City, state, or country/region

Experience Work site Profession Discipline Role type Employment type

Showing 1-20 of 227 results

Turn job alert on Share result Sort by Most relevant

Senior Software Engineer- Copilot

4 days ago Beijing, China + 1 more Up to 50% work from home

We are inviting you to join the Microsoft AI(MAI) organization, to build a new era to "Inform, support and empower Microsoft customers and consumers with responsible AI.". Microsoft AI builds all-platform products with Windows, Web, Android and iOS with innova...

[See details](#)

Senior Product Manager - Copilot Measurement & Evaluation

3 days ago Mountain View, California, United States Microsoft on-site only

As Microsoft AI we are pushing the boundaries of technology. We are creating unique, beautiful and powerful products that will change lives. A small, friendly, fast-moving team, we support each other to do the best work of our lives, always looking to break ne...

[See details](#)

Senior Product Manager - Copilot Product Operations & Strategy

3 days ago Mountain View, California, United States Microsoft on-site only

As Microsoft AI we are pushing the boundaries of technology. We are creating unique, beautiful and powerful products that will change lives. A small, friendly, fast-moving team, we support each other to do the best work of our lives, always looking to break



Andrej Karpathy ✓

@karpathy

Programming is changing so fast... I'm trying VS Code Cursor + Sonnet 3.5 instead of GitHub Copilot again and I think it's now a net win. Just empirically, over the last few days most of my "programming" is now writing English (prompting and then reviewing and editing the generated diffs), and doing a bit of "half-coding" where you write the first chunk of the code you'd like, maybe comment it a bit so the LLM knows what the plan is, and then tab tab tab through completions. Sometimes you get a 100-line diff to your code that nails it, which could have taken 10+ minutes before.

<https://x.com/karpathy/status/1827143768459637073>

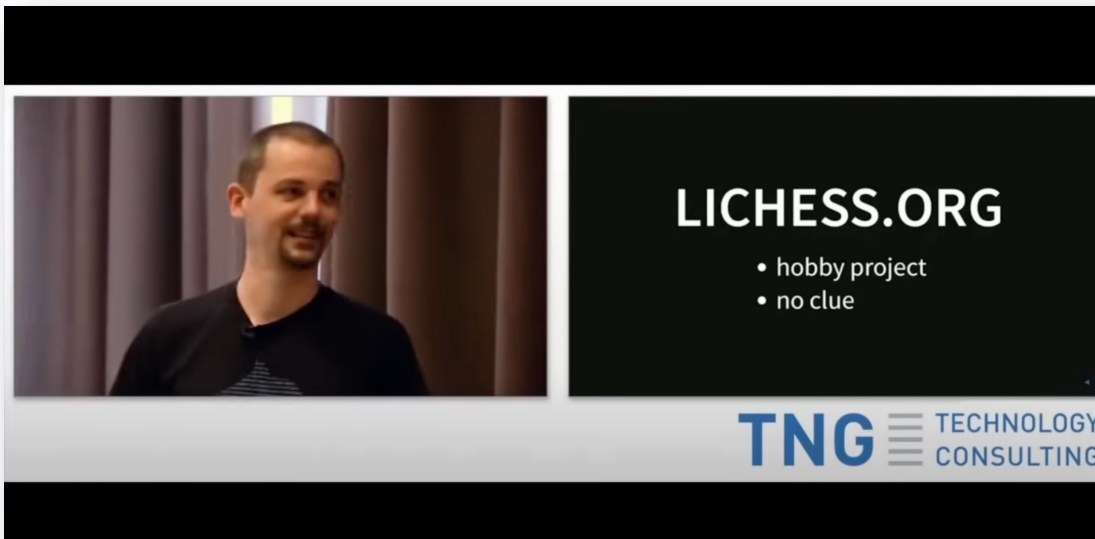
Andrej Karpathy ✓

@karpathy

Building @EurekaLabsAI. Previously Director of AI @ Tesla, founding team @ OpenAI, CS231n/PhD @ Stanford. I like to train large deep neural nets 🧠🤖💡

📍 Stanford 🔗 karpathy.ai 📅 Joined April 2009

939 Following 1M Followers



youtube.com/watch?v=7VSVfQcaxFY



Karel Smutný

- Chess.com is a for-profit company with over 700 employees, while **lichess.org is free, built, maintained, and operated by a single developer.**
- The server written in Scala, hosting **50m games/day**.
- Both sites chess.com and lichess.org are **comparable in features**

A Team of ML PDs, with AI Agents

multilearning GLAD dev
AI agents 1... 10

41



We want to thank our wonderful sponsors!

Gold Sponsors



TITANESOFT



Jago



DUXALIS

Silver Sponsors



Algam Consulting



Red Hat

Community Partners



Agile Singapore



WOMEN DEVS



Singapore DevOps

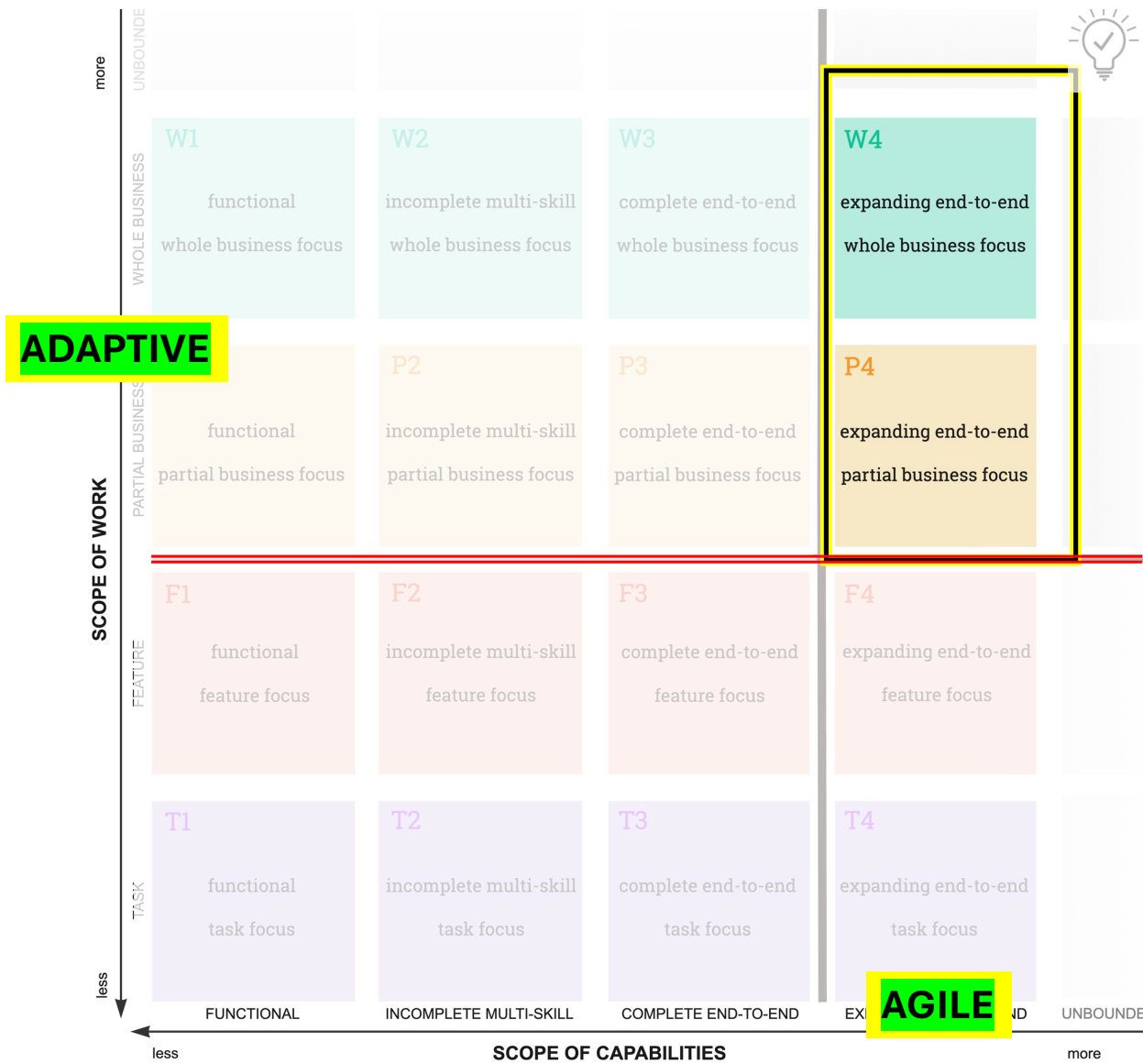


Singapore DevOps

Event Organiser



Ode



WHAT MAKES IT HARD TO CROSS THIS LINE?

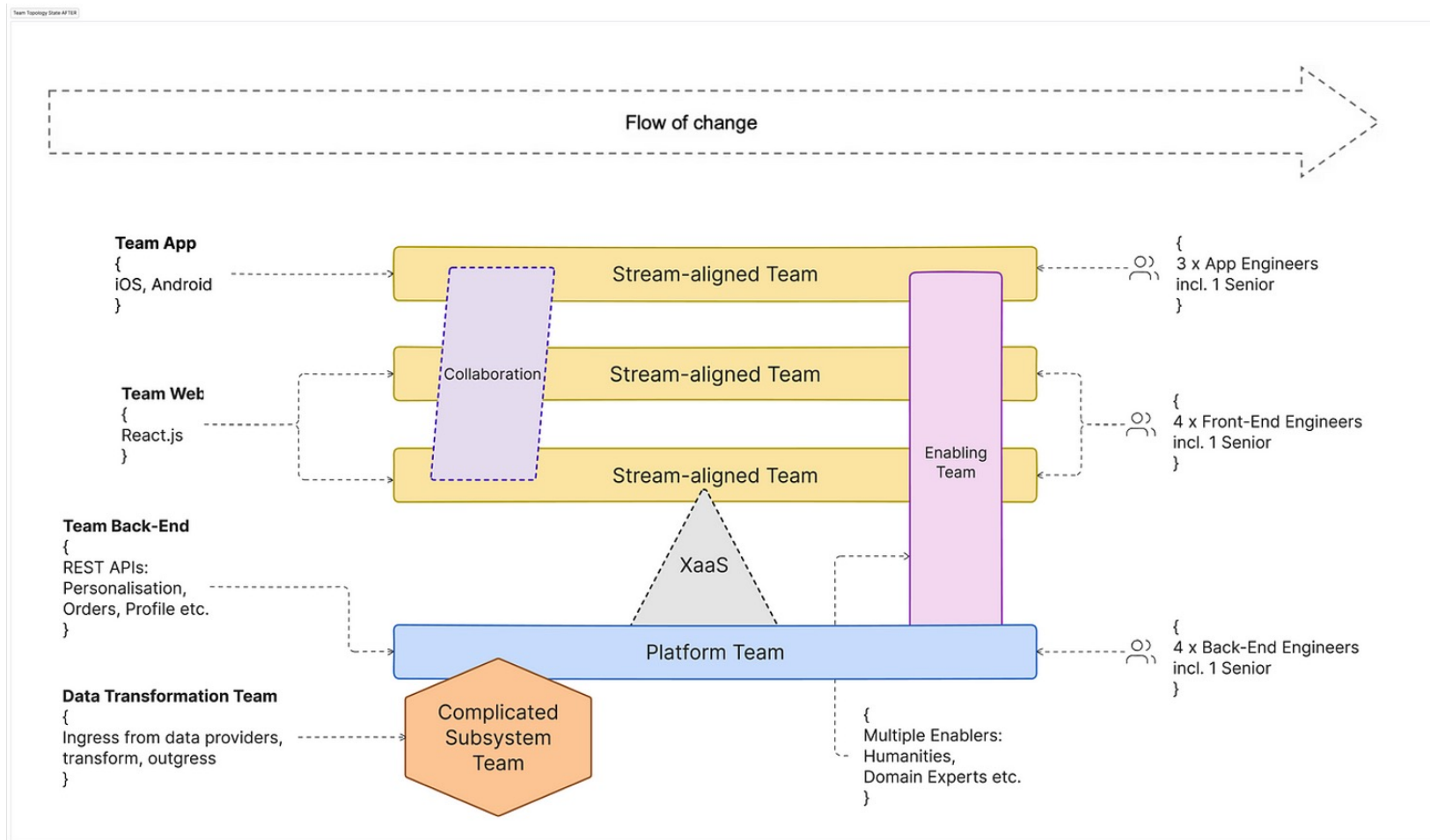
What makes it hard go up?

1. "Conway's Law"
2. Team's Cognitive (Over)load

Team's Cognitive (Over)Load



Solution?



Solutions!

Continuous Integration (CI) as a Practice

Just talk

Pairing / mobbing across the teams

Use modeling techniques

Remind of big picture

Avoid Jira

Connect teams and customers directly

Teams learn customer language

Avoid separating discovery and delivery

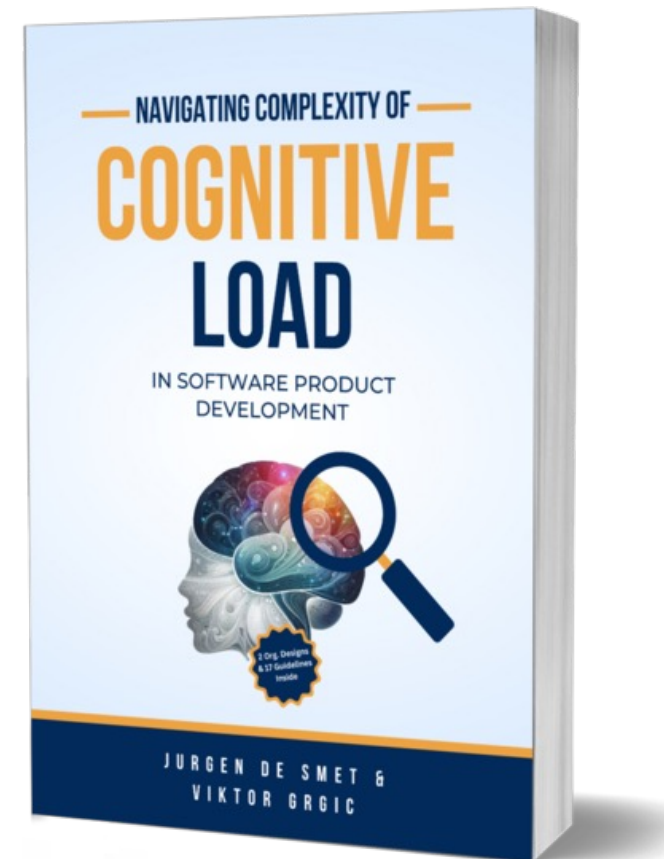
"Take a bite"

Embed business analysis into the teams

Reduce team's interruptions

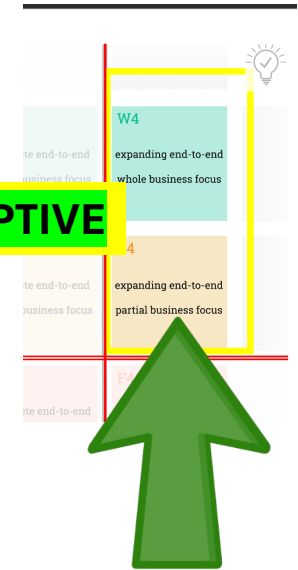
Simplify, automate and standardize

Use AI



DISCUSSION TIME

ADAPTIVE



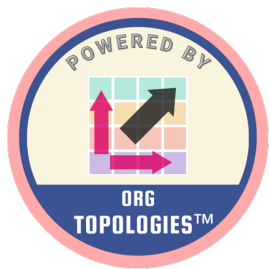
From what you've heard today:

What do you want to try
in your org to move up?

The *last* agile transformation?

... may it be at least a *lasting* one!

Thank you!



2024, Alexey Krivitsky

