

Field Notes of Flow

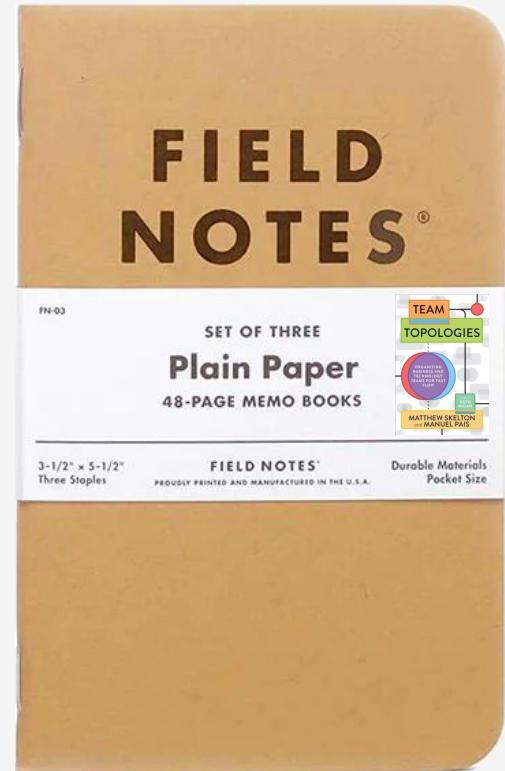
Being accidental practitioners of Team Topologies

Toli Apostolidis & Andy Norton

@apostolis09

@andyjnorton

(We know, we really should stop using Twitter...)



The screenshot shows the official website for the Scaled Agile Framework (SAFe). At the top, there's a banner for the "SAFe SUMMIT" with the text "We are back in Europe and hope you join us!" and a "LEARN MORE" button. Below the banner is the SAFe logo with the tagline "PROVIDED BY SCALED AGILE". A navigation bar includes links for "Home", "About", "Blog", "Read More", "Resources", "SAFe Studio", "Glossary", "Implementing", "Customer Stories", and "SAFe Training". There's also a "What's New in SAFe 6.0" section and a "Sign up for Scaled Agile news" button. A search icon is located on the right side of the navigation bar. The main content area features a quote from Matthew Skelton & Manuel Pais: "The restriction to these four team types acts as a powerful template for effective organization design." Below the quote is a small image of the "Extended SAFe Guidance" document. The main heading on the page is "Organizing Agile Teams and ARTs: Team Topologies at Scale". A detailed text block follows, discussing SAFe Principle #10: "Organize around value", which guides enterprises to organize people and teams around one goal: continuously delivering value to the customer. It notes that while this has been accomplished traditionally through organizing around features, components, sources of funding, geography, etc., the new approach involves organizing by feature and component. A footer note states: "Organizing by Feature and component has been the standard approach for teams and trains within SAFe and".

New in SAFe



New in SAFe



Team Topologies

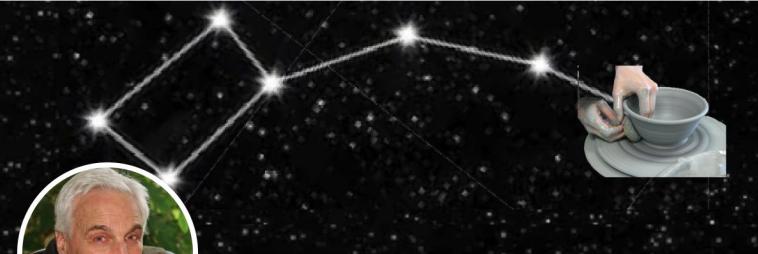
Animal-sacrifice driven-development

Elon Musk chaos-months™

Agile conference bingo card

Conway's Law				
Conway's Law				
Conway's Law				
Conway's Law				
Conway's Law				

← **Mel Conway**
14.6K Tweets



Mel Conway
@conways_law

The choice of communication pathways used within a community constrains the set of inventions that community can put into practice.

④ Beverly, Massachusetts, USA Joined September 2012

350 Following 7,835 Followers

...



...



Mel Conway @conways_law

Darwinian evolution and Conway's Law share a concept; let's call it the "opportunity space". (Note that there is no positive value being put on the word "opportunity".)
1/9

17:08 · 13/02/2020

...



Mel Conway @conways_law · 01/07/2021

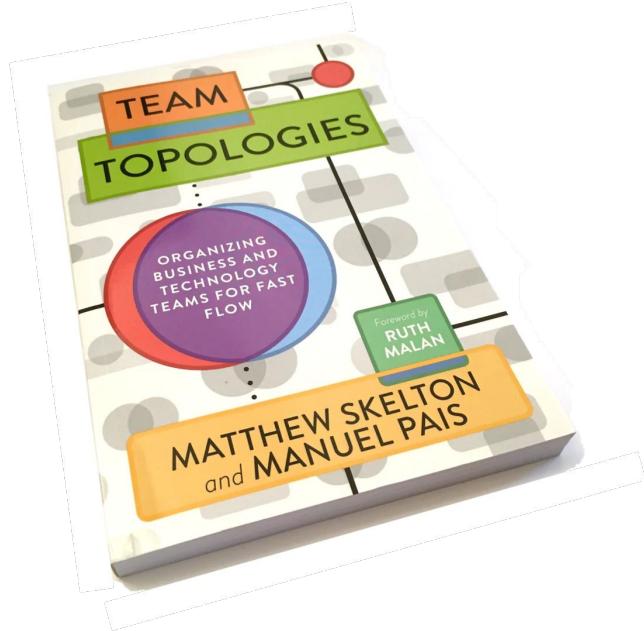
If we are to believe all the available evidence about the consequences of intensive human development, including climate change and resource exhaustion, we are facing a *scale barrier*, on the other side of which we have no governance model.

14/31

1 1 5 ↑



2019

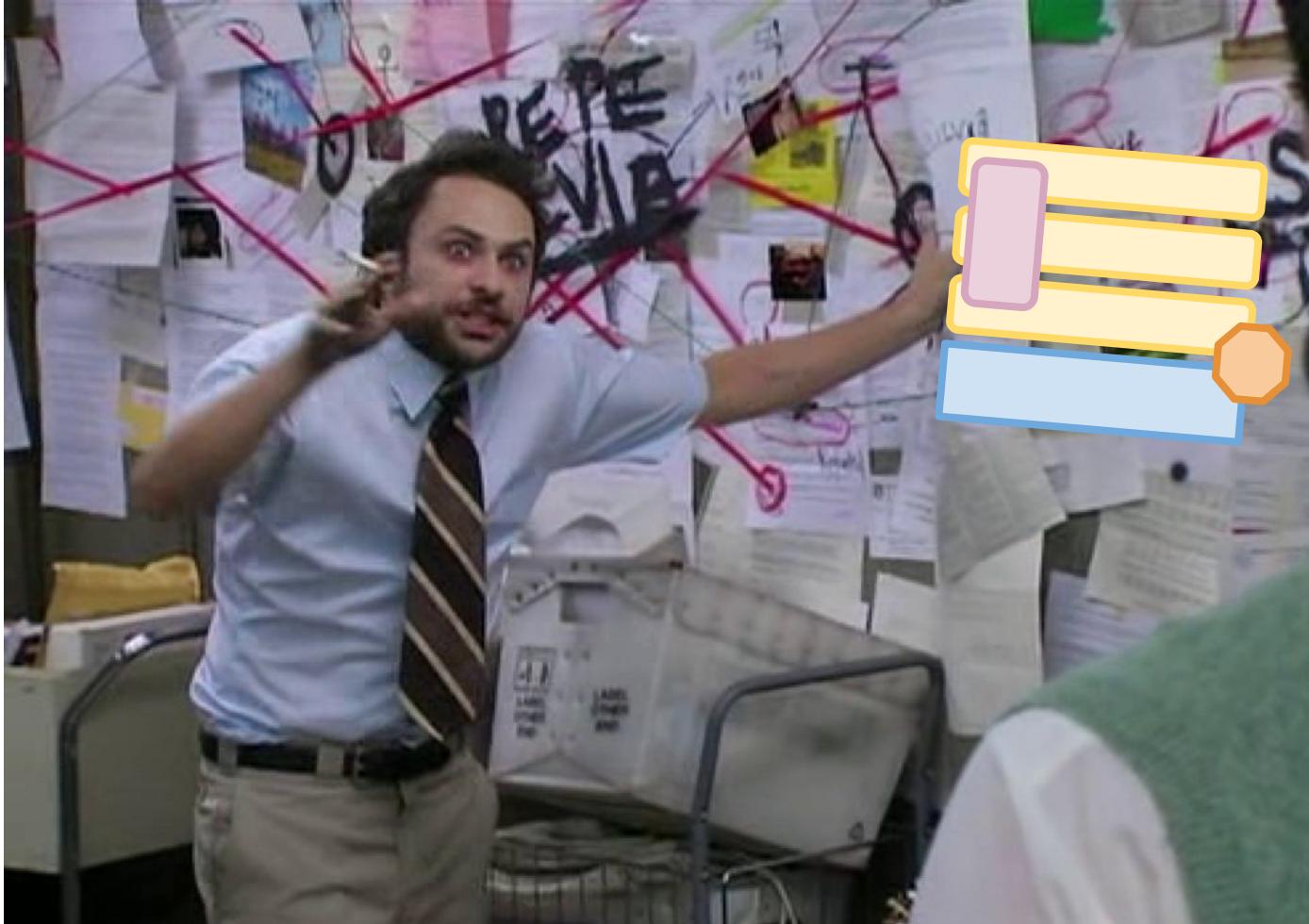


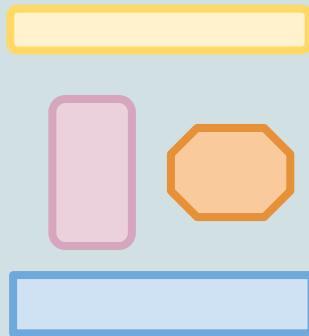
When someone new mentions 'stream-aligned teams'





DID WE JUST BECOME BEST FRIENDS?





+



cinch

=

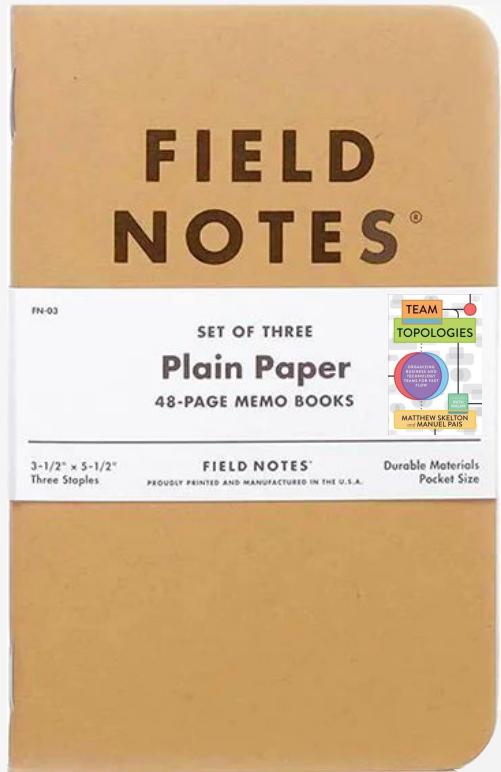




Andy Norton
Senior Engineering
Manager,
Prolific



Toli Apostolidis
(on a career break)



What did we find?

Teams need to be information radiators

That gaps will emerge that need fixing

TT thinking influenced things that we didn't expect

The approach needs to be team first

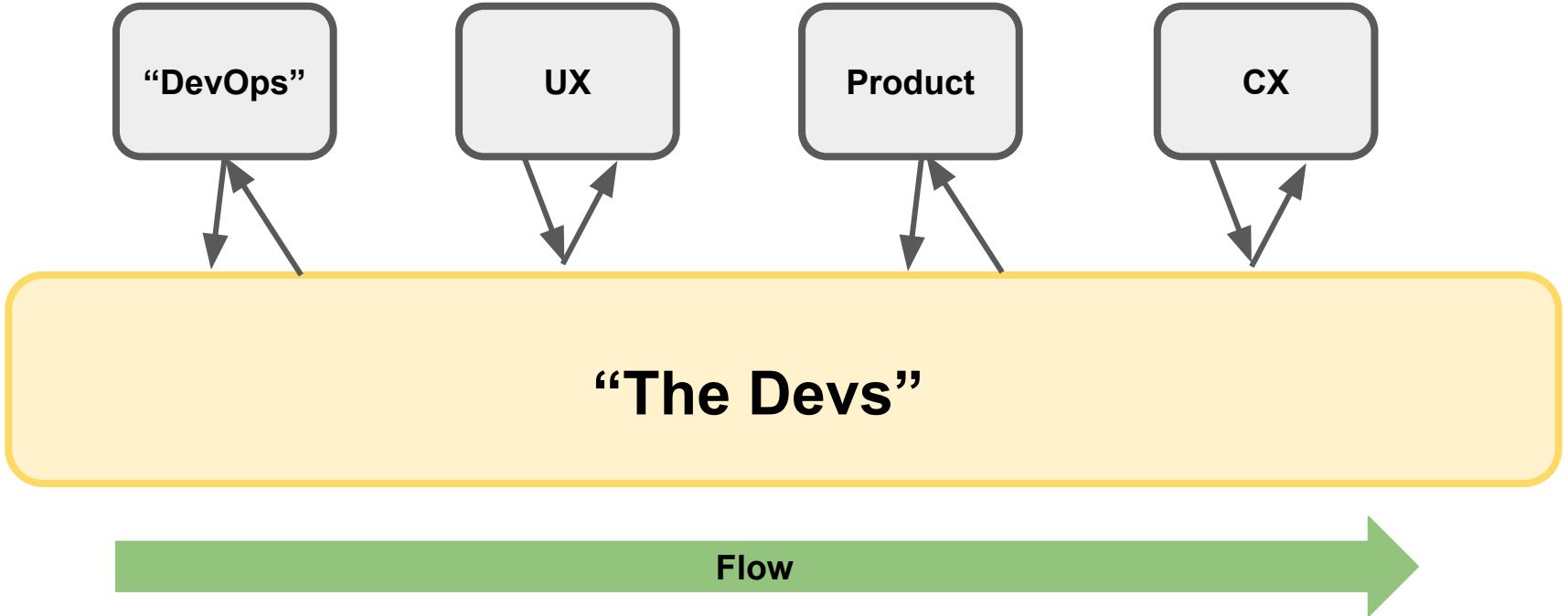
and needs to be aligned to the value stream

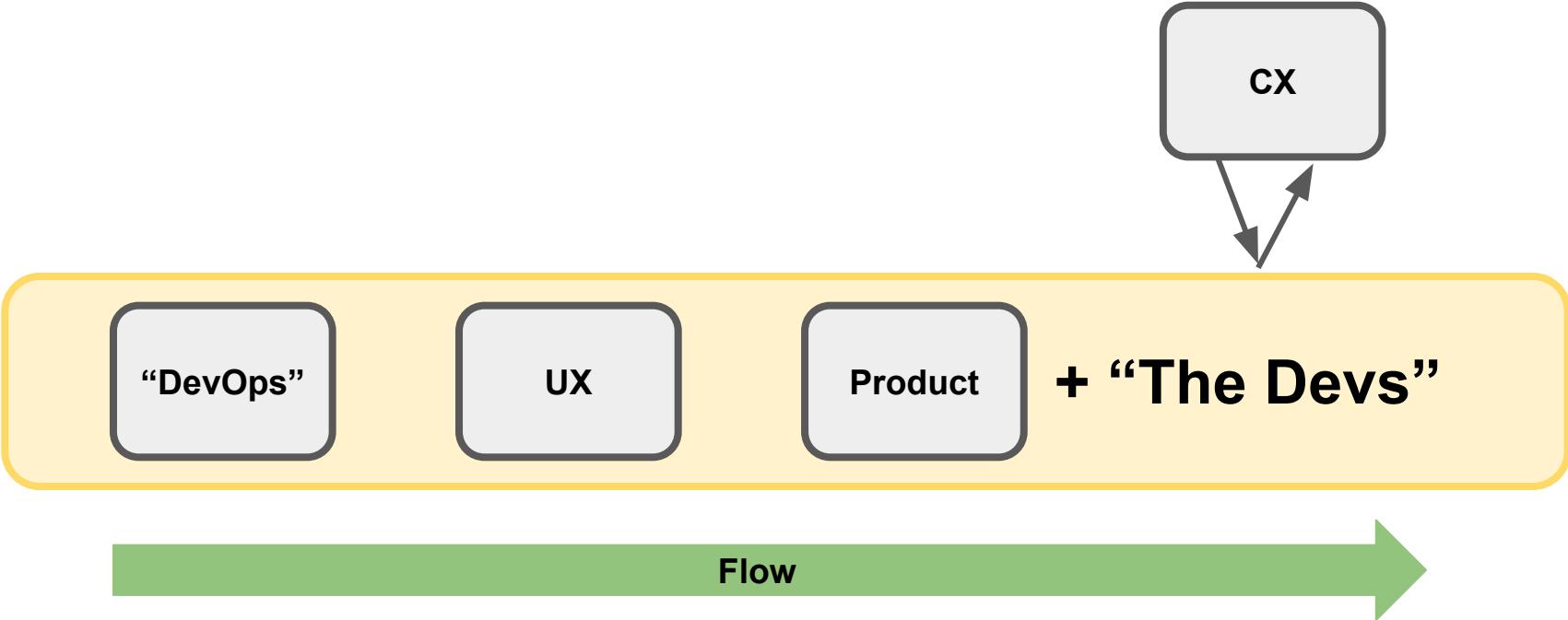
and the team needs to have bounded autonomy

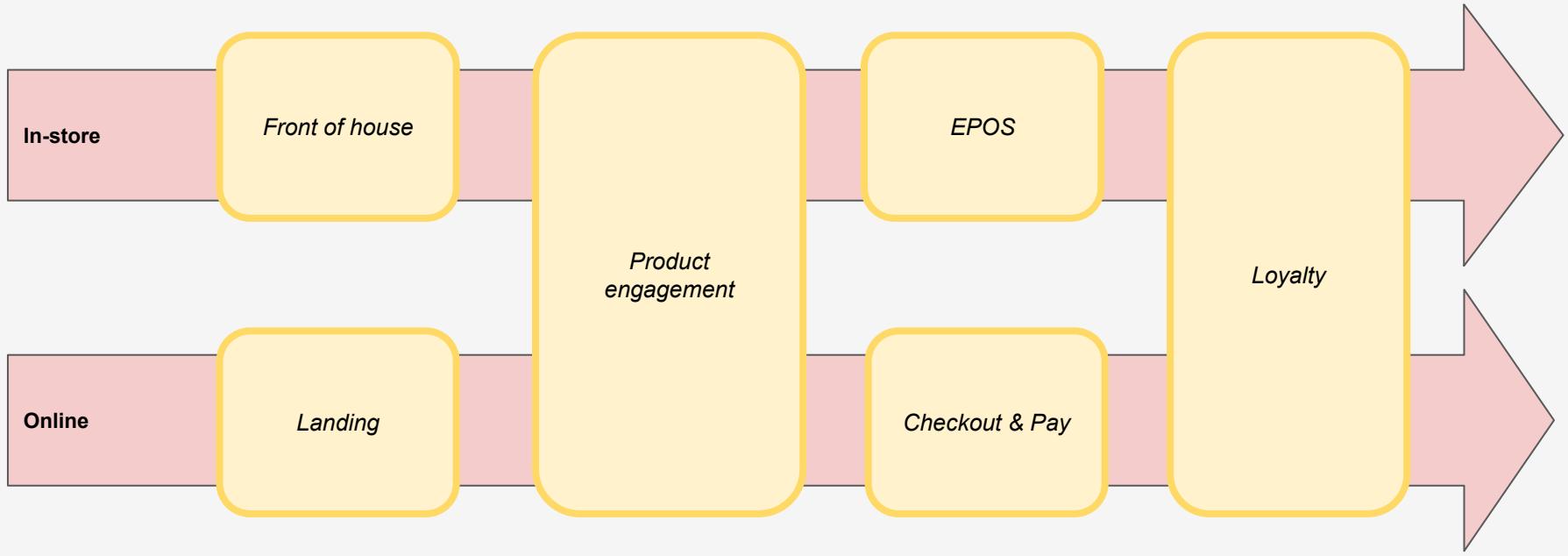
It has to be a team-first culture

**THE UNIT OF
DELIVERY
IS THE TEAM**

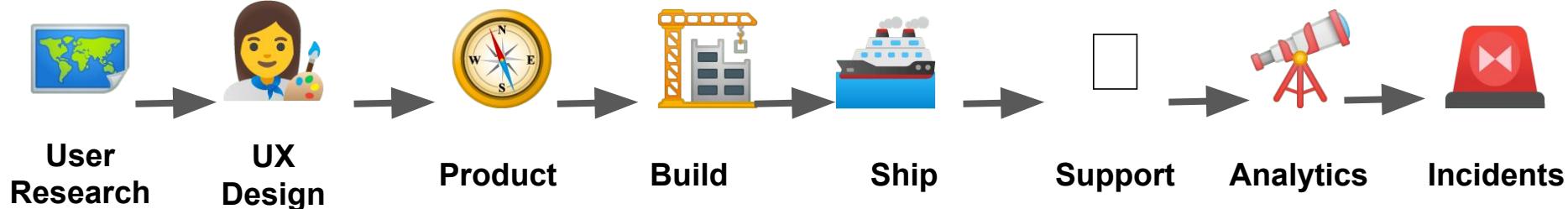




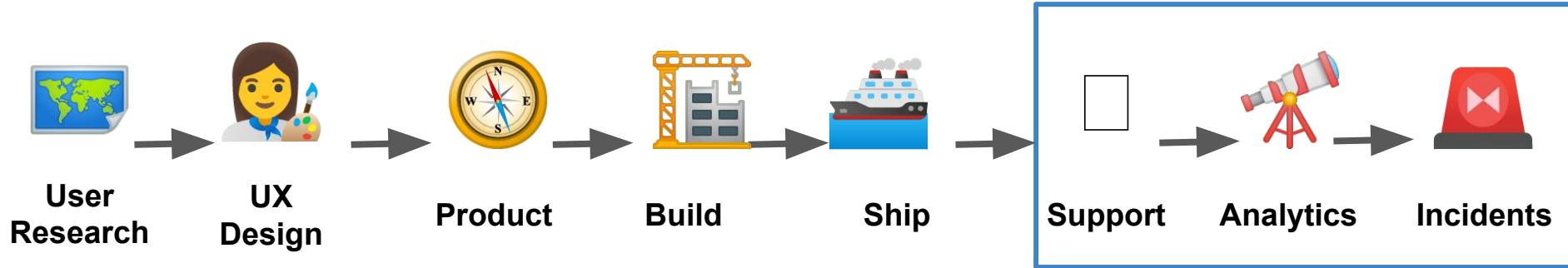




**Aligning to a value stream is
aligning to a problem space**



The development value stream
enables building the solution
space



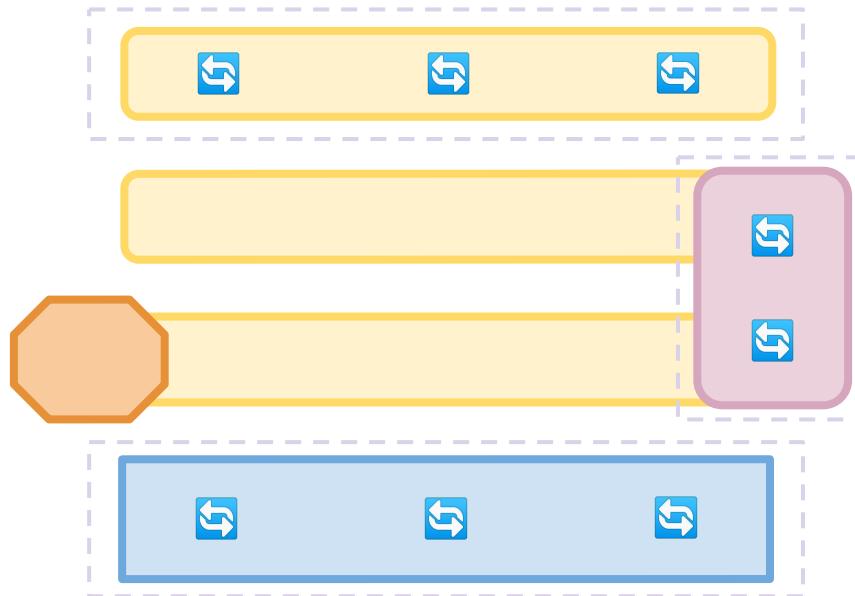
The development value stream
enables building the solution
space
(and supporting it)

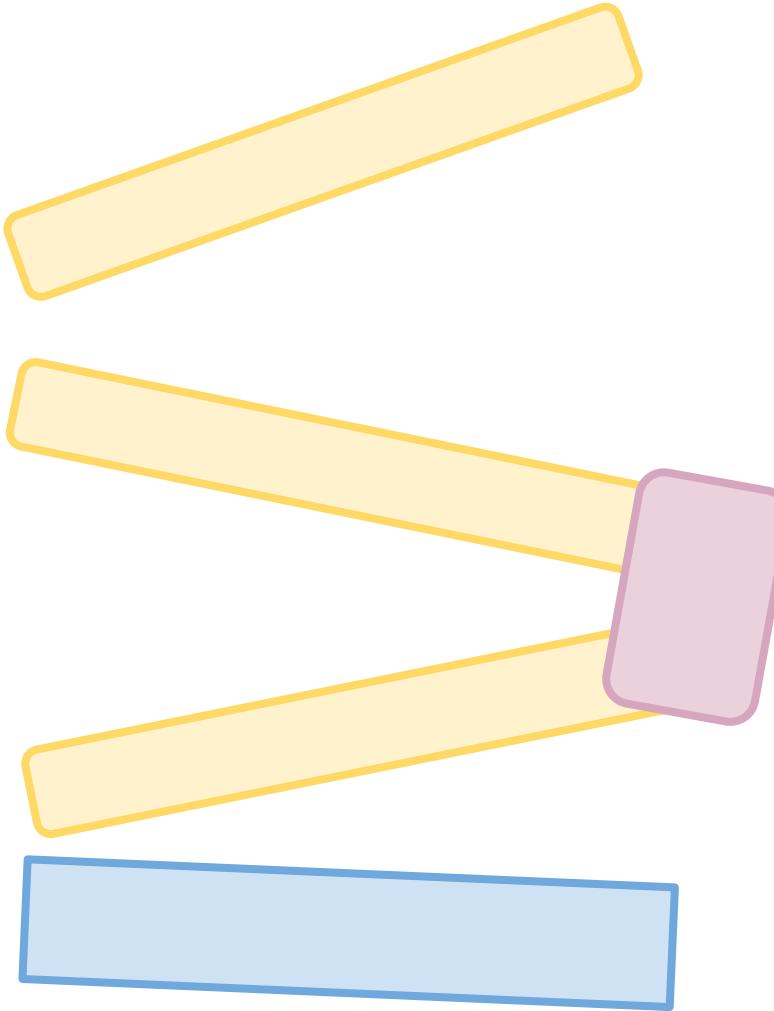
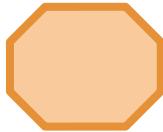
If it's value isn't visible...

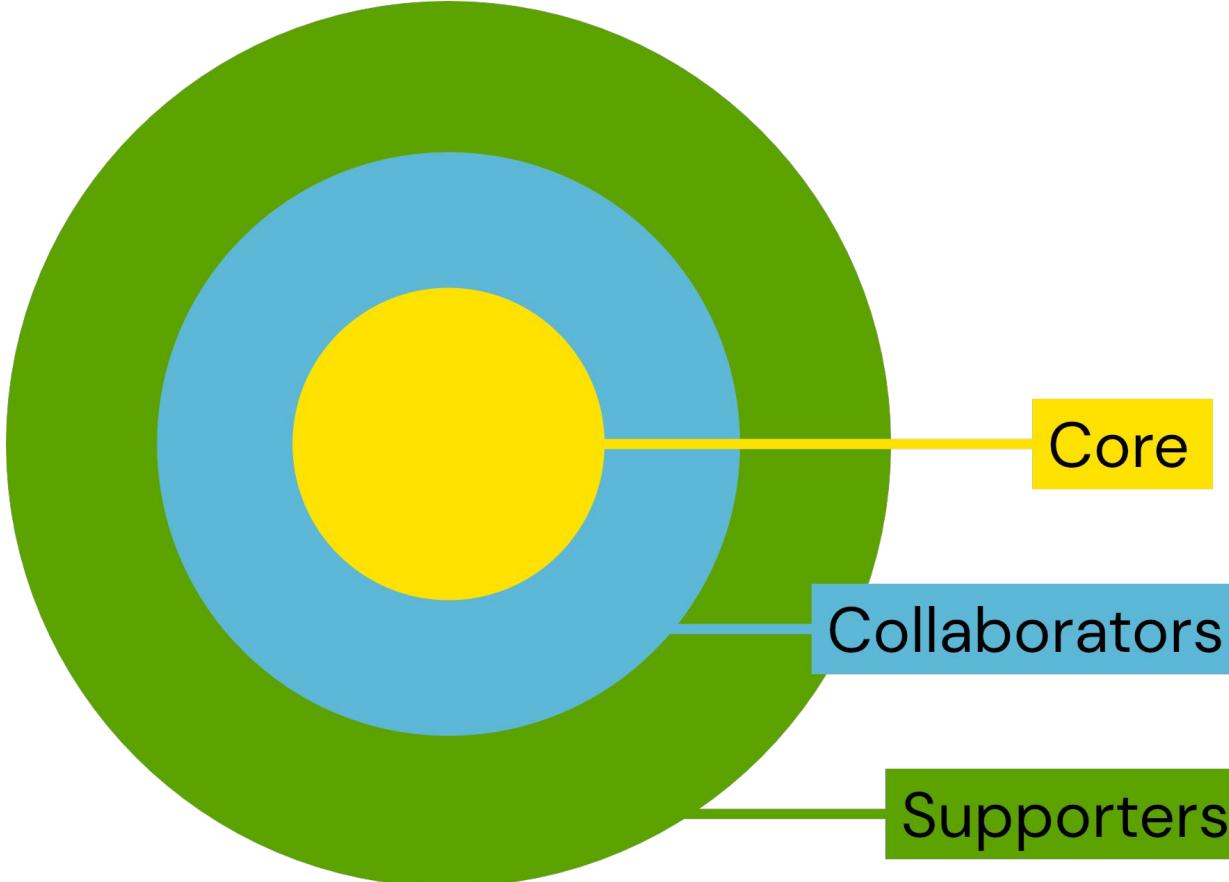












Work in the open

What are they up to?

**Working
Groups**

**It's
complicated**

The teams are my customers



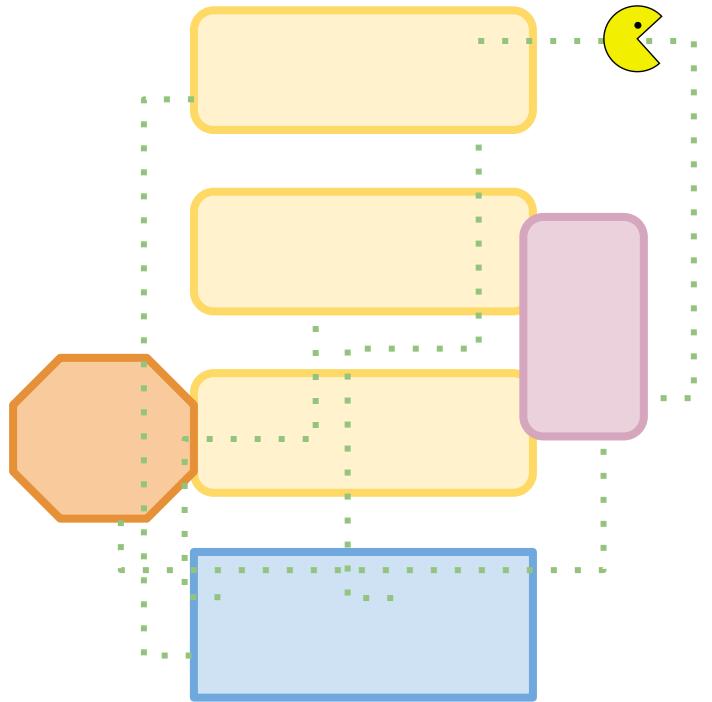
Transparency of information

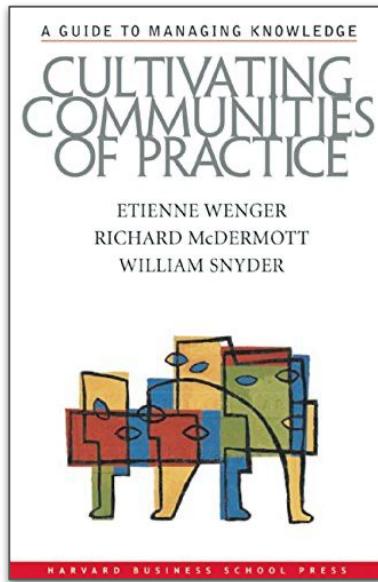


Continuous improvement



Shared understanding







Observability Community of Practice



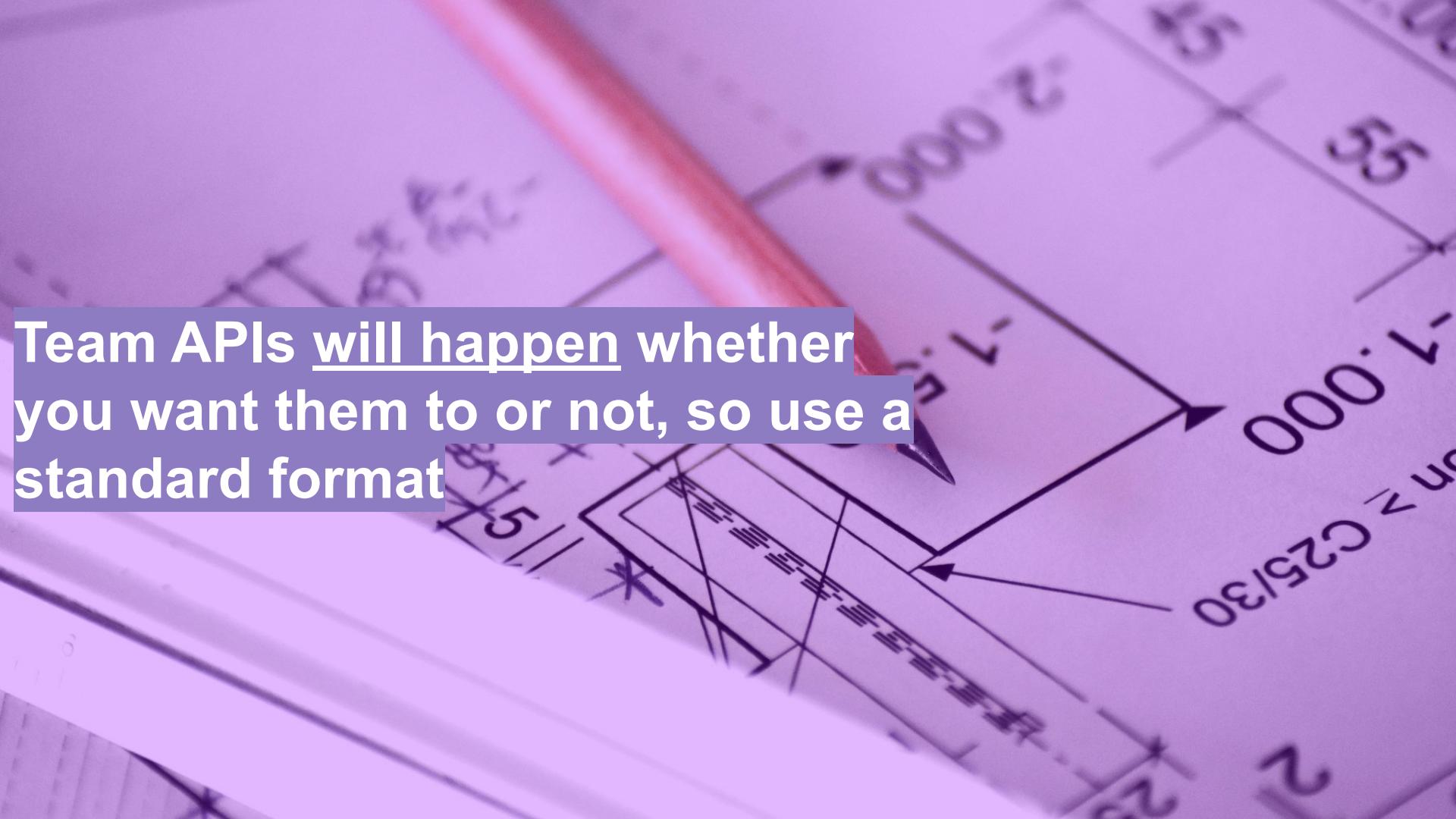
Agile Delivery Community of Practice



Engineering Community of Practice



Team APIs will happen whether
you want them to or not, so use a
standard format



github.com

README.md

Team API template

Overview

It can be useful to define a "Team API" for each team, to improve the team's clarity of purpose and help other groups understand how that team fits into the broader organization. Use this template to help teams think about how they would define their Team API.

Each team should answer the questions and fill in the details below. Remember that the answers and details will be a point-in-time snapshot of team relationships and team interactions.

Team API

Date:

- Team name and focus:
- Team type:
- Part of a Platform? (y/n) Details:
- Do we provide a service to other teams? (y/n) Details:
- What kind of Service Level Expectations do other teams have of us?
- Software owned and evolved by this team:
- Versioning approaches:
- Wiki search terms:
- Chat tool channels: #_____ #_____ #_____
- Time of daily sync meeting:

Team type: (Stream-Aligned, Enabling, Complicated Subsystem, Platform)

What we're currently working on

- Our services and systems:
- Ways of working:
- Wider cross-team or organisational improvements:

Teams we currently interact with

Team name/focus	Interaction Mode	Purpose	Duration
.			
.			
.			
.			

490 stars
38 watching
66 forks
Report repository

No releases published

No packages published

Contributors 5

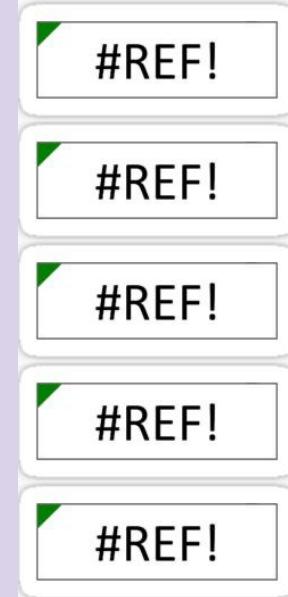


<https://github.com/TeamTopologies/Team-API-template>

**Because you know what
people are like.**



Someone will definitely end up making one in PowerPoint otherwise.



We even used Team APIs for our Communities of Practice

Community Name	Principal & Staff engineers
Community Type	PRACTICE
Community Maturity	TRANSFORMATION
Lead Members	
Core Members	
Confluence Space	
Miro Board	https://miro.com/app/board/ 
Kanban Board	
Labels	PRINCIPAL
Calendar	Lean Coffee - Tuesdays @ 11:15 Show & Tell - First Thursday of every month @ 11:30 Struggle of the week - Thursdays @ 11:30 CoP days - Ad-hoc every 2 months
Vision	
Internal Communication Channel	Teams Link
External Communication Channel	Show & Tell - First Thursday of every month @ 11:30 Teams Link

★ Orders Squad - Motherboard ▾

1h Past 1 Hour ▾

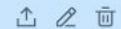


\$environment production ▾



Are orders looking healthy?

23 widgets



Is our part exchange service healthy?

12 widgets



What changes have been released?

6 widgets



Do we have any incidents?

1 widget



Are we seeing errors?

5 widgets



Are my APIs healthy?

12 widgets

<https://cinch.blogin.co>

cinch

Megan John
Software Engineer Mar 28, 2022 (08:19)

Meg's Mumbles - My Start At Cinch

In General Topics 0 min 12 46 10 0 Follow

I thought I'd add some insight to starting from Square 1 and my first week at cinch.

But first, an introduction to me.

I'm Megan. Or Meg. Or MJ. I don't really mind. I began my journey with cinch last week (21/03/2022) after graduating from a physics degree and then a first job which unfortunately was not a good fit - it happens. My specialities in my physics degree delved into astro and space physics. From a young age I have been really intrigued by the workings of outside of Earth and so decided a physics degree was the thing for me. And it was amazing. I fell further in love with the planets that spin alongside us and how things came to be. But I fell in love with something new too. Coding.



Nasa Image of Jupiter with its polar auroras visible

12 lab experiments and analysing the data received and moving into my dissertation on Jupiter's ionosphere (which I will happily go into depths of if anyone is interested), I found a passion for making things work through a computer and so came to a crossroads in my choices after completing my Bachelors degree. Continue being an astrophysicist or become a software engineer. I think you can predict which I

<https://cinch.blogin.co>

cinch

Apostolis Apostolidis
Principal Practice Engineer Mar 22, 2022 (10:18)

Autonomy of Schedule

0 min 4 34 Edit 9 0 Follow

ave what you would call in [Team-Topologies-speak](#), a number of stream-
s called *squads*. Squads are the unit of software delivery at cinch. As we have
e established other team types:

- Groups
- unities (of Practice or Interest)
- ubs ([The DevOps Handbook Book Club](#))

not why

about the purpose of all team types, but the 'how' of working outside
belong to a squad or a tribe, that is your core membership. You have made a
o the people in your squad or tribe, and they take priority over other

mean that you can't participate in things outside the squad. In fact, we
u to do so, but it is vital that you:

- ings visible
- onate with your core membership - your squad

[remove semantic meaning](#)

Style Sheets (CSS) is what's used to give websites life and to make them look
ever, to quote a phrase from a legendary Marvel movie:

Katie Raby
Software Engineer Apr 4, 2022 (12:56)

axe-con 2022 Digital Accessibility Conference

In Events 0 min 0 22 6 0 Follow

What is axe-con?

a free, open and inclusive digital accessibility (a11y for short) conference held
the conference focuses around building, testing, and maintaining accessible
ences. axe-con is suitable for developers, designers, business users, and
y professionals of all experience levels.

re live-streamed the axe-con sessions in the Manchester office (with lots of
cks - shout out to the sour cream & onion Pringles 😊). We connected with
ested in a11y at cinch, and had fun debating the topics we were watching in

Takeaways from a few talks we watched

ng

[remove semantic meaning](#)

Style Sheets (CSS) is what's used to give websites life and to make them look
ever, to quote a phrase from a legendary Marvel movie:

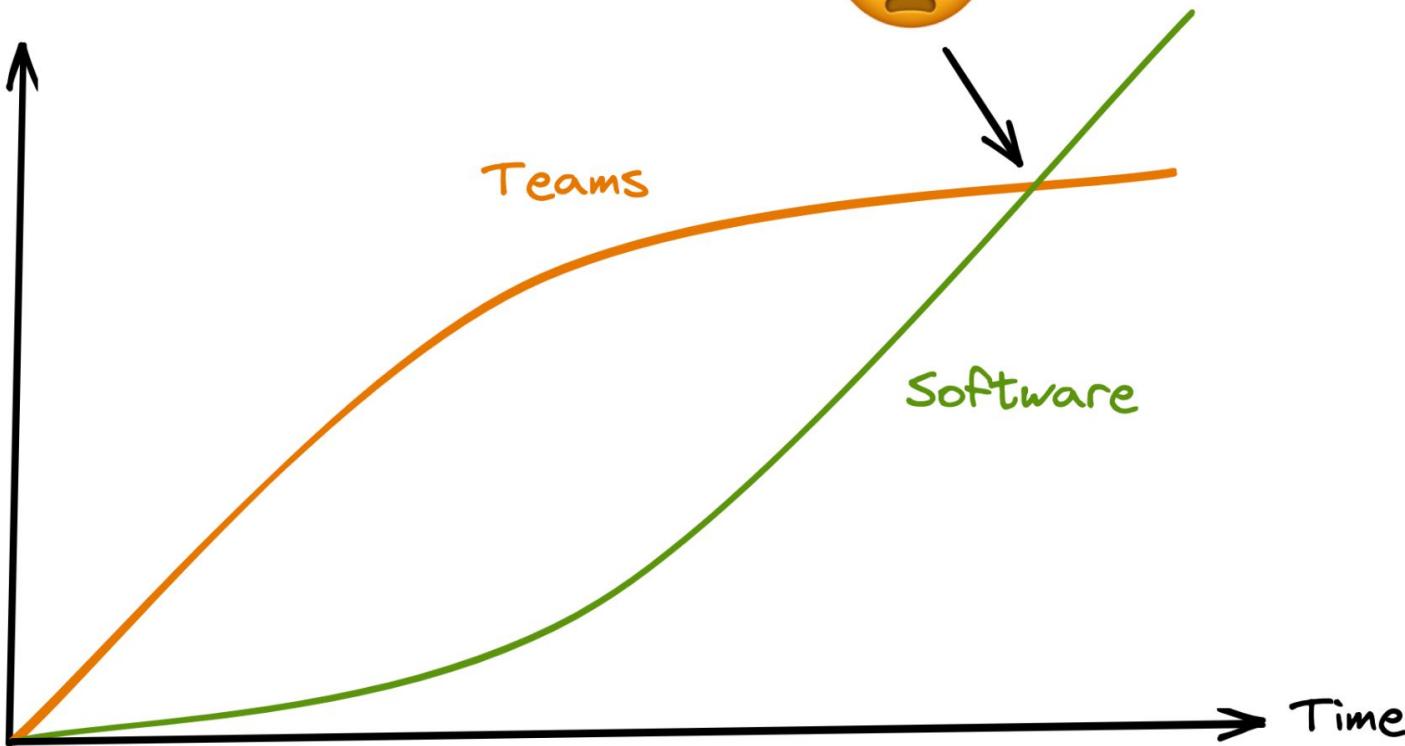
**6 MONTHS
LATER.**

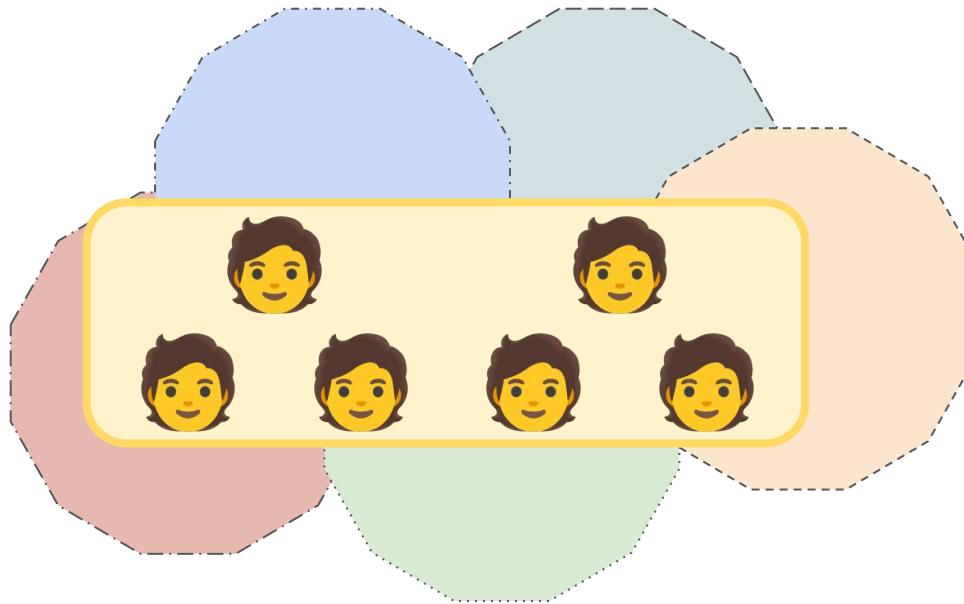
Teams need to be information radiators

That gaps will emerge that need fixing

TT thinking influenced things that we didn't expect

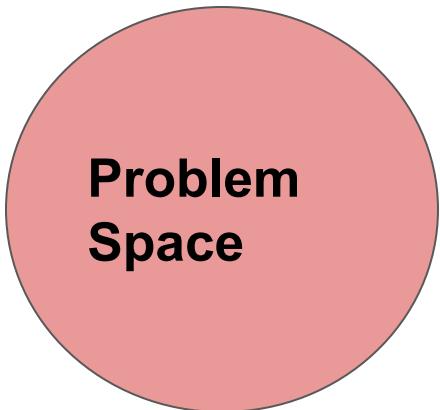
Amount of





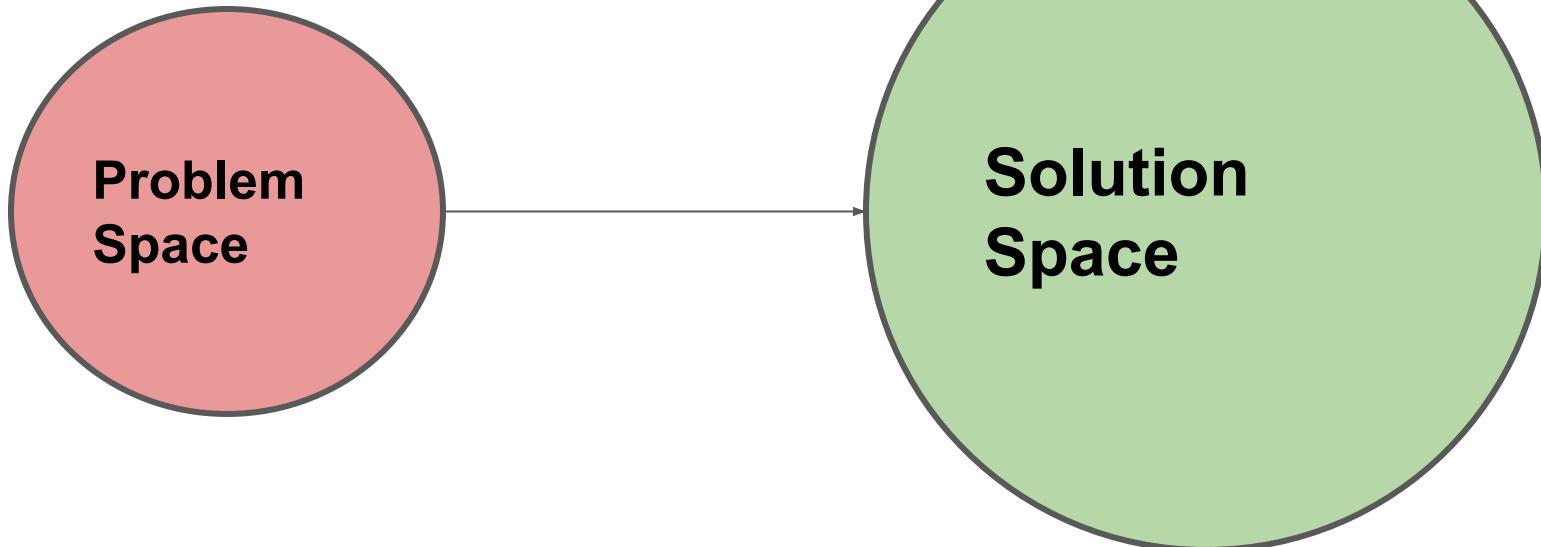


Are you committing to
problem spaces?



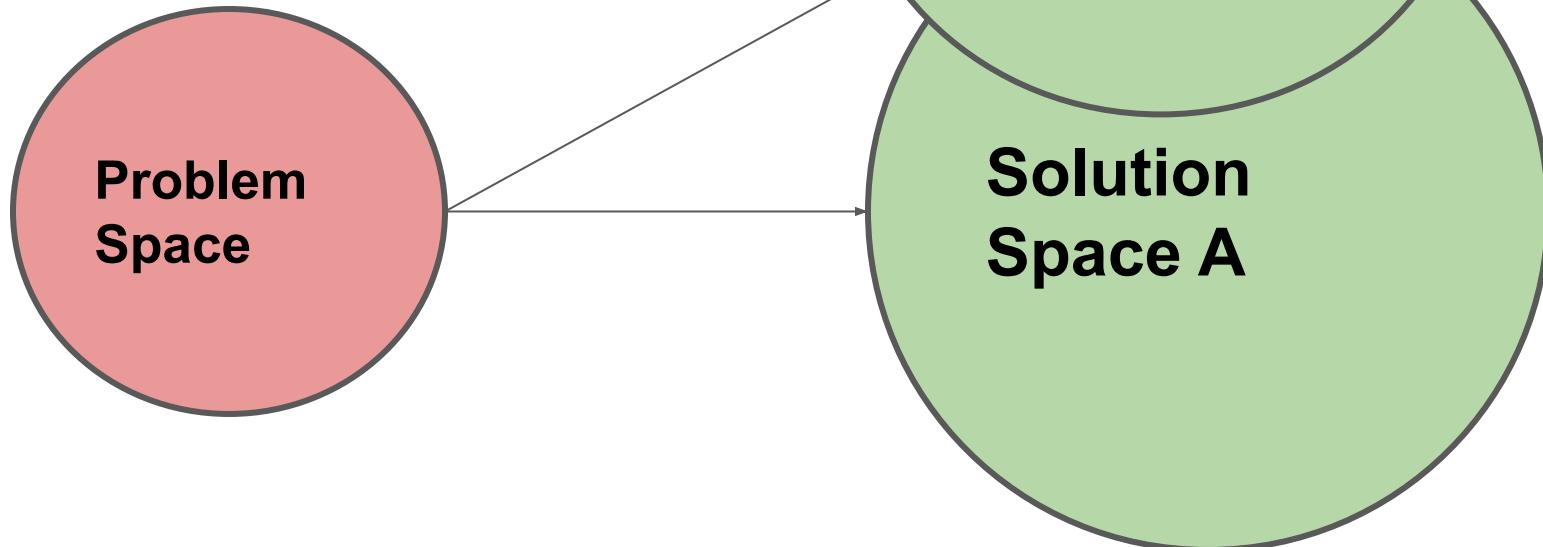
Are you committing to
problem spaces?

Or **solution spaces**?

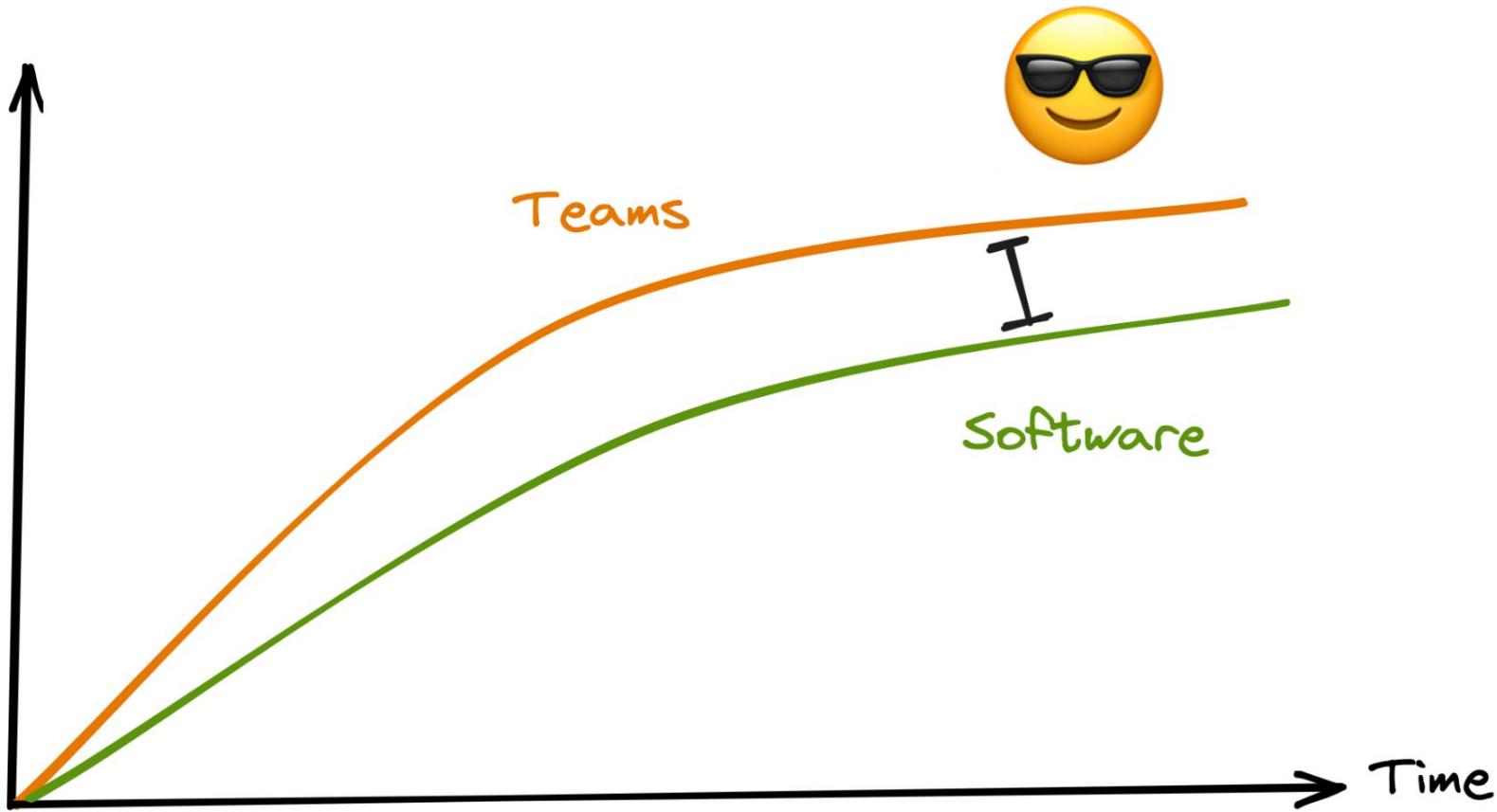


Are you committing to
problem spaces?

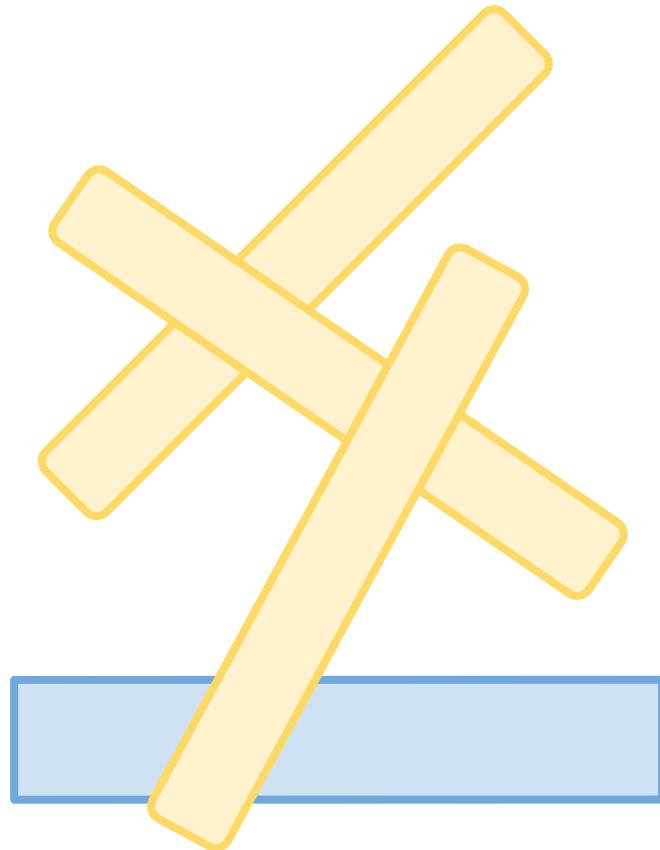
Or **solution spaces**?



Amount of



Optimising for flow can be
make it **harder** to pivot



Search Service

Inventory Service

Checkout Service

Infrastructure repo



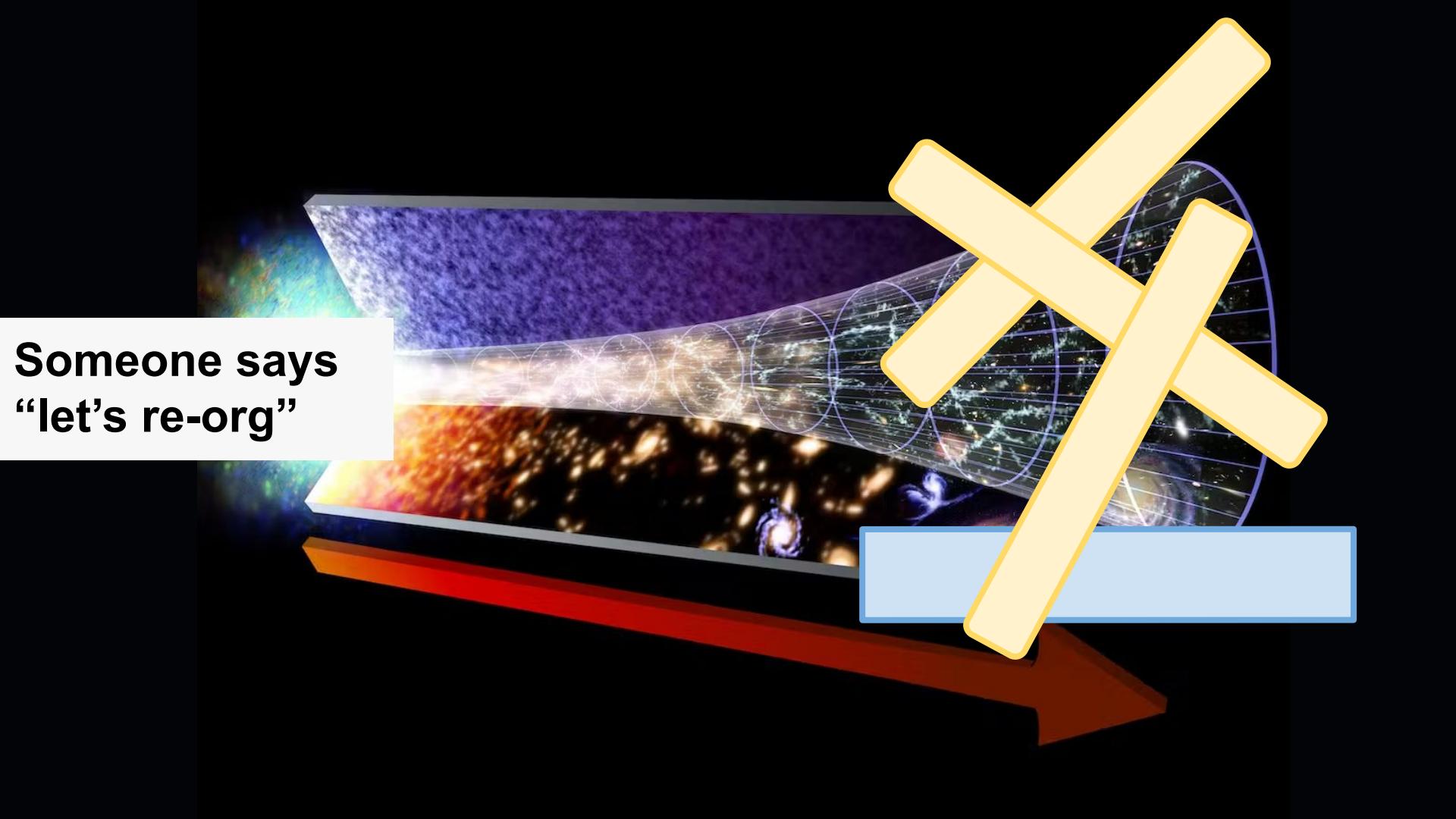
Search team

Inventory team

Checkout team

Infrastructure team





**Someone says
“let’s re-org”**

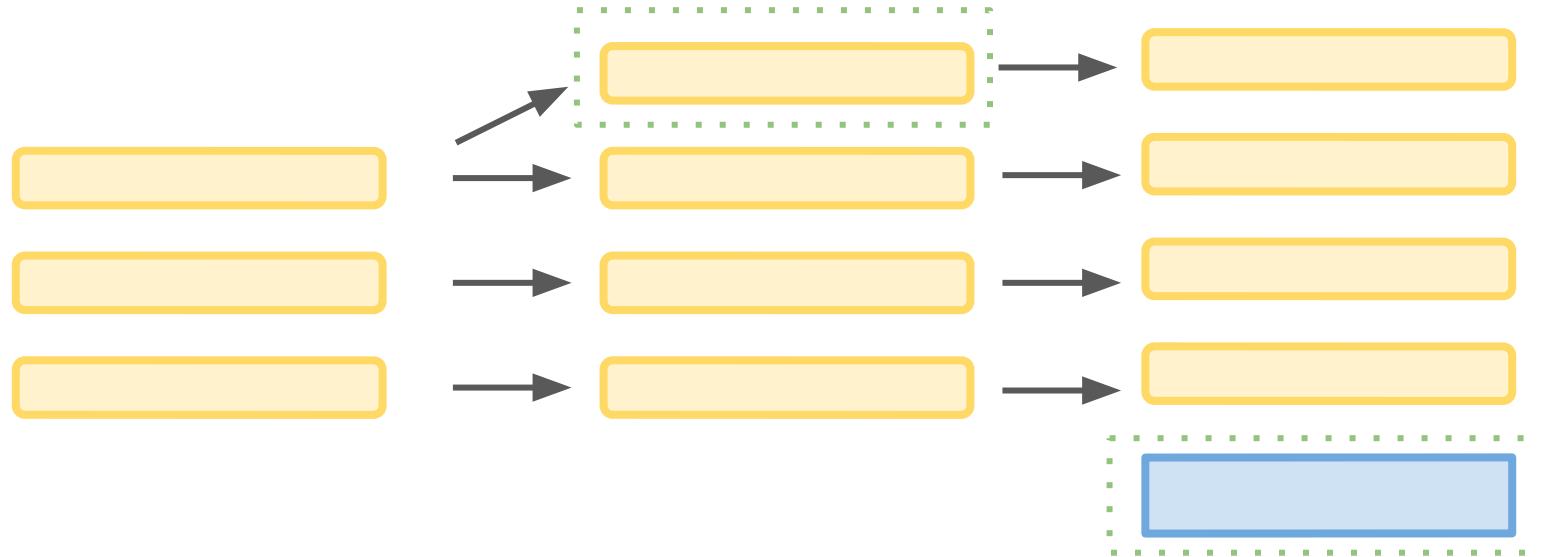
Domain bashing - what's the worst that could happen?



You need to be at ease with
uncertainty and complex
socio-technical systems

Adopt an organisational strangler pattern (don't strangle anyone though)

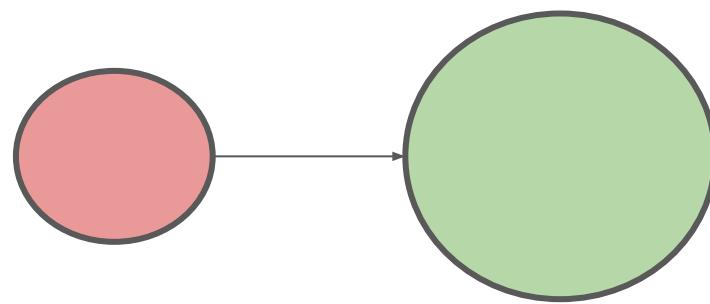
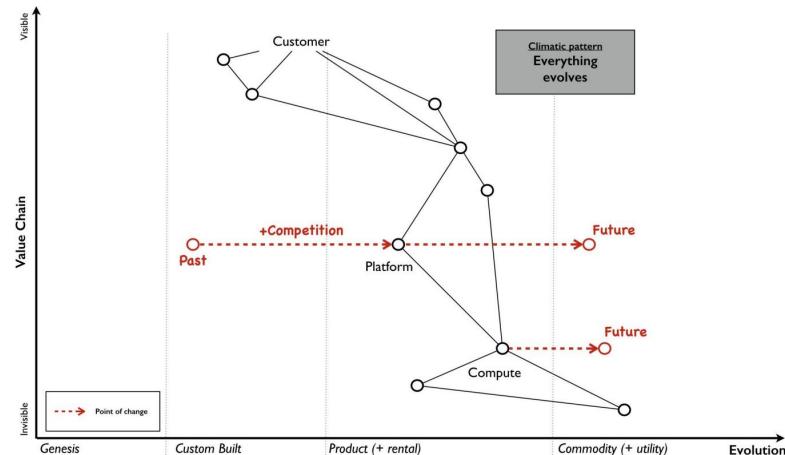
***(this isn't Twitter)**



Version 1.0

Version 1.1

Version 1.2





Teams need to be information radiators

That gaps will emerge that need fixing

TT thinking influenced things that we didn't expect

**Team Topologies thinking influences architecture,
ways of working, engineering practices, and even
recruitment.**

1. Architecture

Search Service

Inventory Service

Checkout Service

Infrastructure repo

Search Service

Inventory Service

Checkout Service

Infrastructure repo



Search team

Inventory team

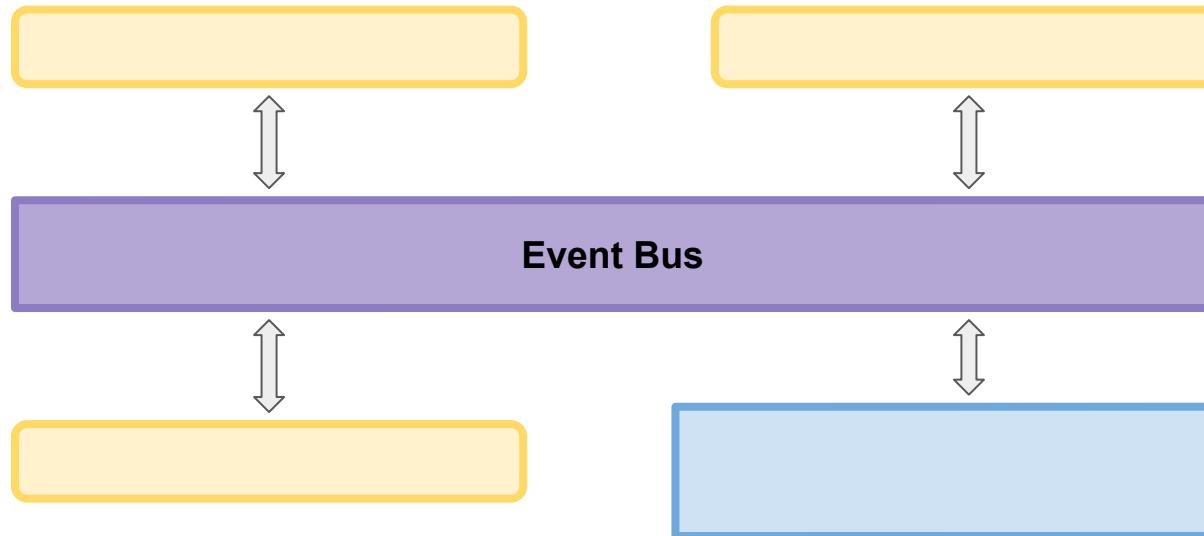
Checkout team

Infrastructure team

@conways_law
Mel Conway

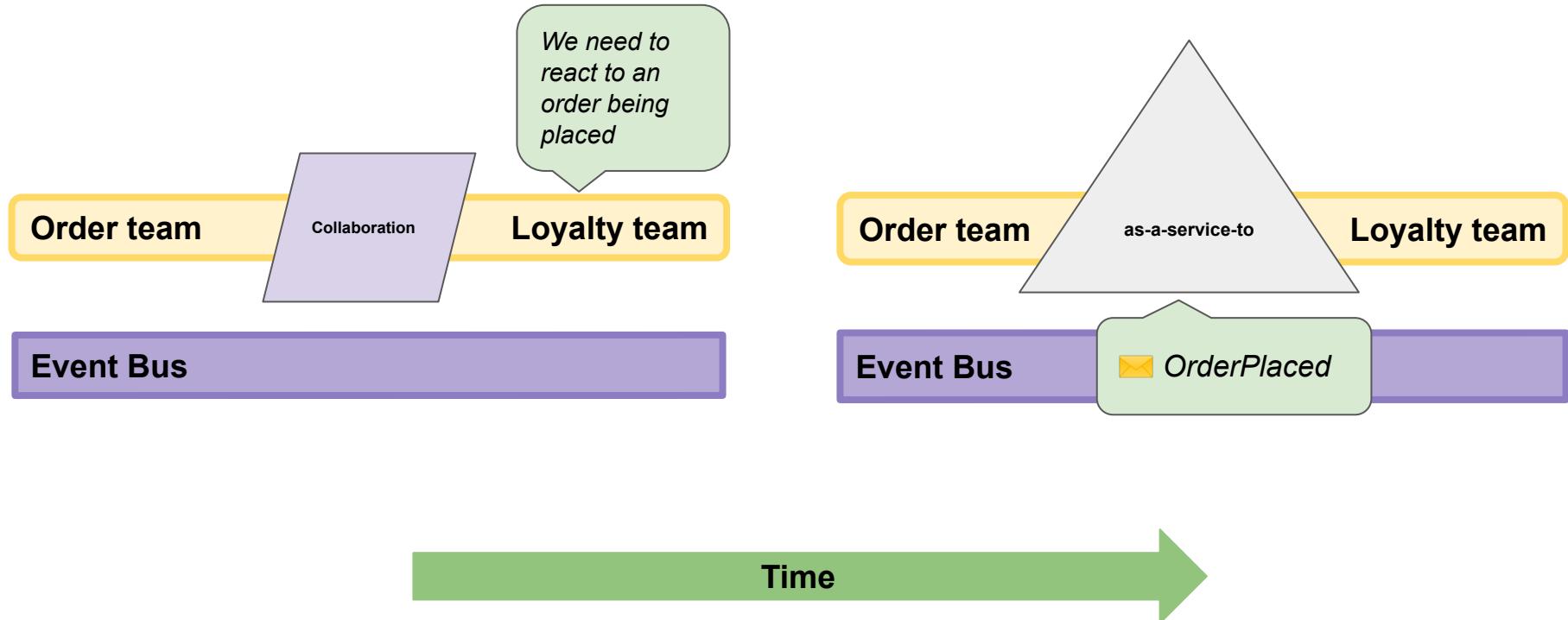


TT enables good practices around EDA and microservice-first approaches





**There's some offers of coupling
you can't refuse**



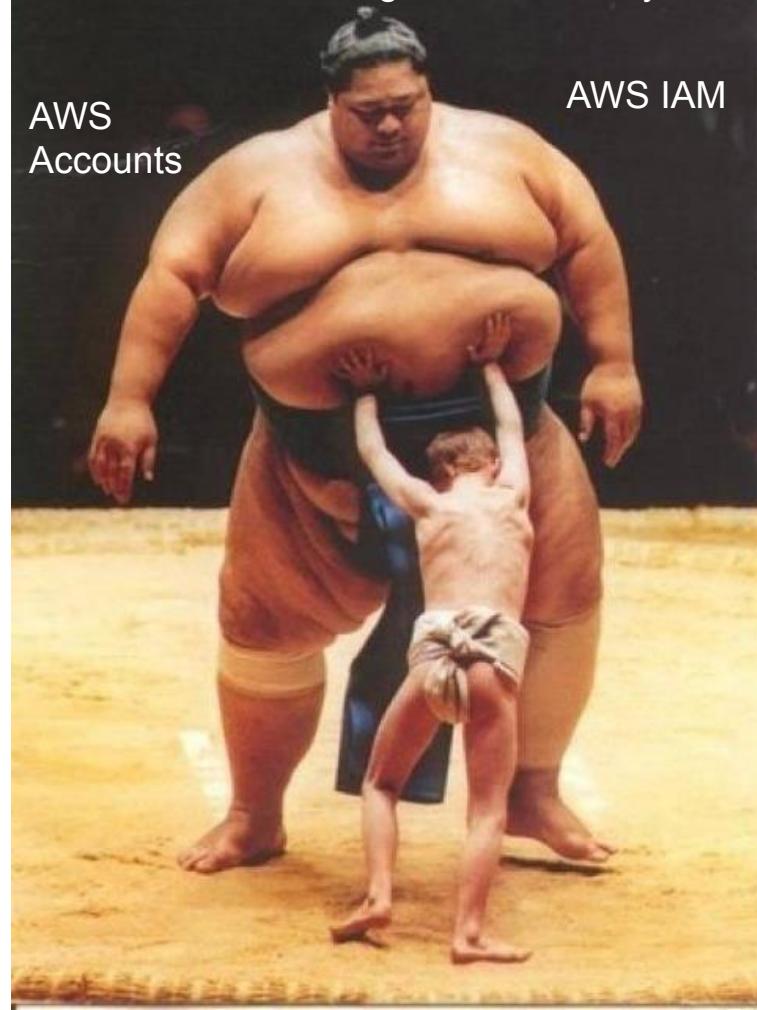
Platform teams in the land of serverless



GitHub Datadog Security

AWS Accounts

AWS IAM



2. Ways of working

Documenting depends on your team type

It's an afterthought

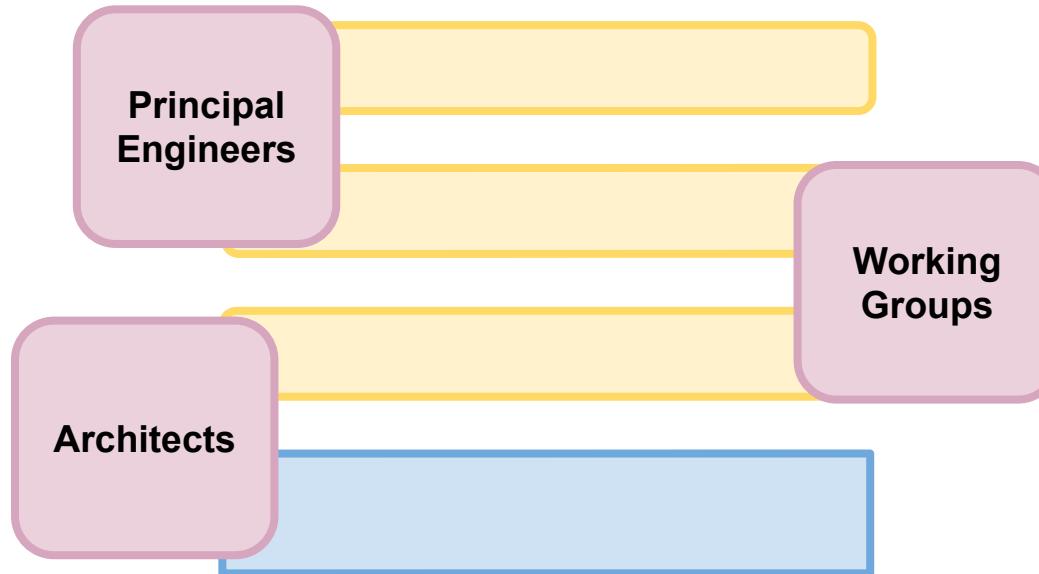
It's an afterthought

It's an afterthought

**We codify
knowledge**

The teams are our customers

Enabling teams happen more than you think





PoKéMoN

**Working Groups happen
in any company, you
know what they look
like, what they smell like**

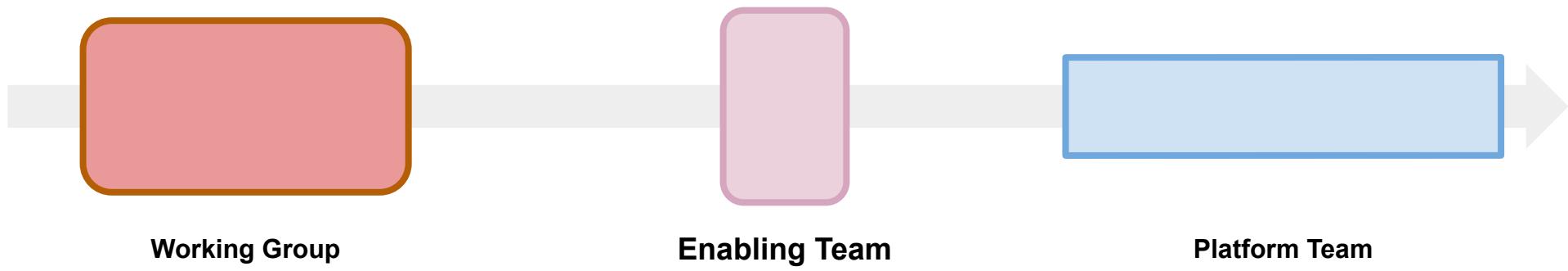


Short lived ‘virtual’ teams around a problem

The things we need to figure out and define

Once it's done, it's done (with a clear kill criteria)

Swarm, and then evolve or disband



3. Engineering practices

**Trunk based
development**

IaC

Observability

SRE

CI/CD

Teams need to learn these things



You have to get *really* good at knowledge management



Team APIs



ADRs



Event documentation

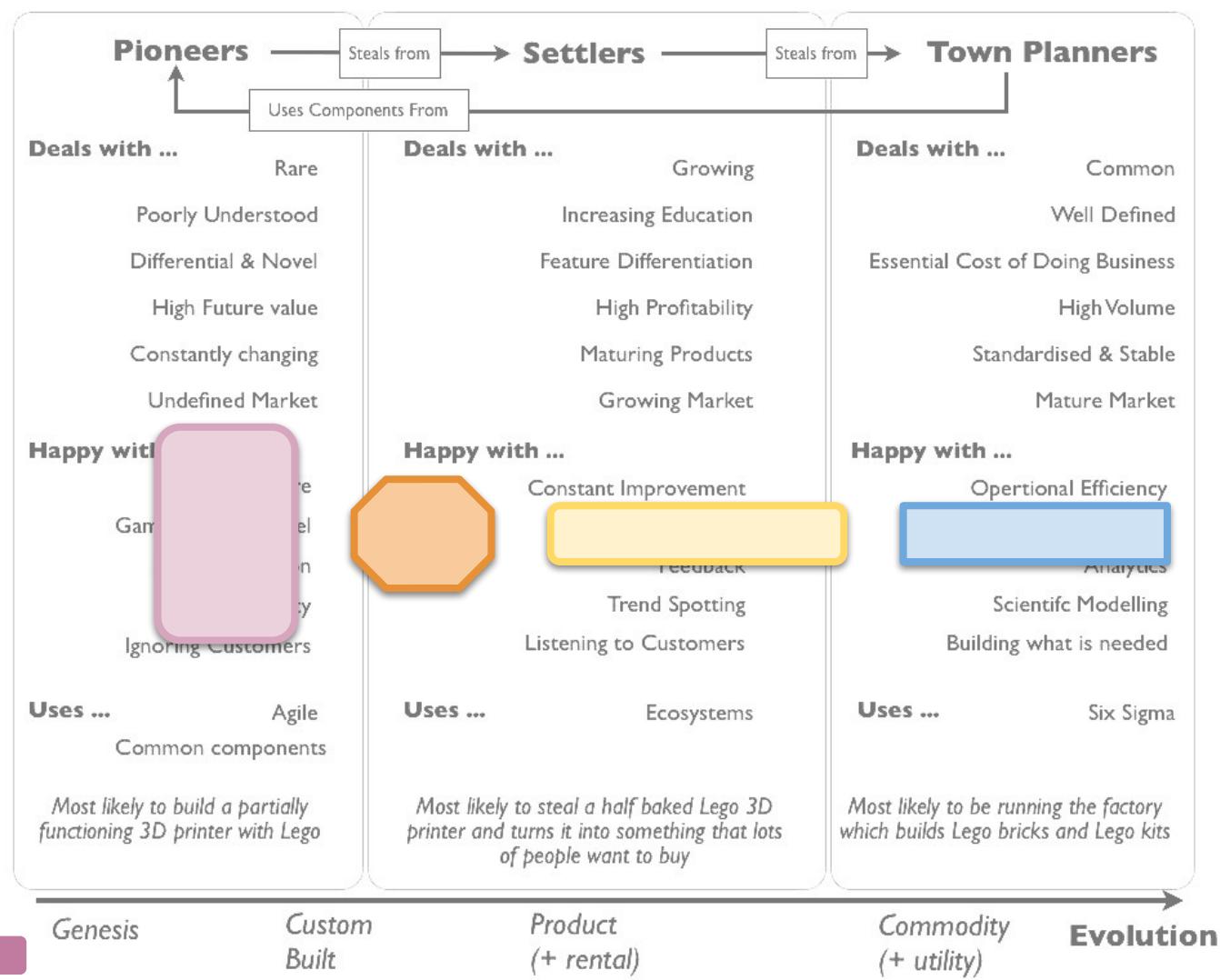
4. Recruitment

**Optimising for flow
using TT is part of the
recruitment process**



Recruit for team players





How you do the role

Tech Lead

Start with the customer need

Holds good domain knowledge, spending time to understand our customers, business and the wider industry.

Can take a 10,000 ft view of the domain, and feed into our product roadmap from a technical and strategic point of view.

Advocates for engineers spending time understanding the customer journey through visits to sites and time spent with service designers.

Solve problems at the roots

Leads on incident management and promotes a blameless culture across the Squad.

Looks for learning opportunities in failures, and outages, and how to put in place changes to reduce the likelihood of repeat issues by leading on post-incident review outcomes.

Promotes observability throughout the Squad's work, including instrumentation so that useful telemetry can be consumed.

Develop in the open

Promotes working in the open across the Squad, looking for opportunities to share knowledge asynchronously and keep other Squads in the Tribe informed of programme challenges and lessons learned.

Conducts code reviews for writing up, writing up to consumers design and breaking down complex solutions into inclusive, clear highlights.

Communicating out technical blockers both inside / outside the Squad.

Design for production

Sets the technical direction for the team by leading on architecture design, creating and sharing a technical vision for the team, and guiding Software Engineers on the implementation.

Conducts code reviews to get code into production to validate assumptions and technical spikes before stabilising.

Leads the team in owning and running their own services, without handoffs and dependencies across different Squads.

Make things others can build on

Sets the technical direction for the team by leading on architecture design, creating and sharing a technical vision for the team, and guiding Software Engineers on the implementation.

Helps support engineers to learn practices such as TDD, trunk-based development, pair and mob programming and codifies this processes for others.

Identifies opportunities for improvement in engineering practices and works with other Tech Leads across the Tribe in order to implement better ways of working and improvements.

Helps shape the circus Technical Blueprint based on what their Squad needs.

Be knowledge managers

Spots areas where there are single-points of failure or a low bus-factor - where there are individuals who are the only person that know about a particular process.

Looks to reduce tacit knowledge and encourage a knowledge-sharing culture.

Leads on documenting team ways of working using Team APIs, and ADRs to communicate the rationale of important decisions.

Feedback at every level

Has trust in peers and the Squad to get the job done, and provides the empowerment to help them do it, purposefully delegating to avoid becoming a single point of failure.

Take a leading role in building a happy, engaged and supported group of Engineers.

Make everyone heard

Encouraging the forming of strong relationships across the Squad, supporting distributed ownership, and addressing team conflict in order to create a psychologically safe working environment.

Become a force-multiplier for Software Engineers in the Squad, investing time into sharing different approaches to problem solving by not solving problems for them, but big coaching on how to use tools and resources to help troubleshoot more effectively.

What skills you need

Tech Lead

	Notice	Rand	Write	Tech	Mastery
	1	2	3	4	5
Tech stack	Has some contexts to what the team can touch looks like the team, from org to backlog and data storage layer, with some knowledge of the system and how to interact with it.	Has working knowledge of the team's tech stack, including the handing off of the team's builds in AWS, and is comfortable making changes to the system to support parts of the team's services.	Has good, working expertise of the team's tech stack, including the handing off of the team's builds in AWS, and is comfortable making changes to the system to support parts of the team's services.	Has an excellent and deep understanding of the team's tech stack, from the components to how they're built. Actively feeds into best practices and is able to make significant contributions to increase their knowledge of the system.	Is a subject matter expert in several, if not all, the main-wide tech stacks and leads changes both in terms of tackling complex problems, and also defining future changes.
Writing code	Has limited experience writing code. Can follow rules and guides to write code that is readable.	Has experience writing code. Experience in refactoring existing code or writing in multiple programming languages.	Can write well-structured code that follows the style of the team's codebase, and can solve common issues. Has exposure to complex codebases and can identify and suggest improvements.	Can design and implement solutions following the style of the team's codebase, and can give examples of 'what good looks like' when refactoring components, can coach on patterns.	Defines the standards at which when it comes to writing code. Has a deep understanding of the style of the team's codebase, and can coach on best practices, including domain-driven design, design patterns, and how to make sure code is following coding practices.
System design	Has an awareness of the high-level view of the Squad's components.	Can make changes that are aligned with the existing component architecture.	Can consistently design components that are aligned with the existing component architecture.	Defines the standards at which when it comes to system design. Has a deep understanding of the style of the team's system design, and can coach on best practices.	Anticipates future scale and needs at a system level, and can make sure that all is set up to look in 4-6 months time and define the system design.
Testing	Writes some simple tests to verify the outputs of a function. Learns towards manual checks and tests.	Implements unit tests to validate the code they're writing. Writes some higher-level tests with guidance from more senior engineers.	Understands the path to production and the testing required to get there.	Defines strategies for improving CI/CD pipelines. Has a good understanding of the CI/CD pipeline and can coach on best practices.	Advocates for quality and testing across Tribes and creates examples that others can follow on shifting left and building testing into the pipeline.
CI/CD	Has some awareness around the build and deployment of the Squad's work.	Understands the path to production and the testing required to get there.	Supports the ongoing learning and growth of the team's CI/CD pipeline, including identifying dependencies and creating regular unit tests.	Has a solid understanding of the team's build and deployment process and the practices to enable flow of new features to production.	Has CI/CDs and infrastructure as code in place, including domain-driven design, design patterns, and how to make sure code is following coding practices.
Mentorship	Is part of a mentorship process as a mentee. Roles hearing on the support of others.	Asks more senior engineers for help on solving specific problems that may have security implications.	Asks more senior engineers for help on solving specific problems that may have security implications.	Forming guides for people to help them find opportunities for others.	Actively mentors multiple individuals and/or groups, and coaches them to success. Often sought after by potential mentees for the ability to provide constructive long-term career planning.
Security	Understands the importance of security.	Approaches all engineering work through a security lens. Has a good understanding of security concepts and can coach on software engineering.	Approaches all engineering work through a security lens. Has a good understanding of security concepts and can coach on software engineering.	Protecting engineers with infrastructure and processes that make sure they're able to share knowledge and insights about work in progress and how to approach security in everyday processes.	Defines check-standards on security concepts and leads on security related initiatives.
Observability	Is aware of the Squad's monitoring practices.	Has an awareness of the operational data for the Squad's monitoring practices.	Has an awareness of the operational data for the Squad's monitoring practices.	Forming guides for people to help them find opportunities for others.	Defining the principles of observability culture, and encouraging teams to improve their observability data to improve stability and reliability.
Leadership	Shows vulnerability by being open about knowledge gaps.	Shows vulnerability by being open about knowledge gaps.	Shows vulnerability by being open about knowledge gaps.	Creates a culture of psychological safety and inclusion for others, and lead on initiatives.	Highly creative and innovative motivation for others, communicating client outcomes and leading on initiatives.
Agile ways of working	Has an awareness of why agile methodologies help teams to deliver things sooner.	Has working knowledge of agile practices that the Squad uses. Likes to use it as a tool to respond to change and adapt to delivery.	Has working knowledge of agile practices that the Squad uses. Likes to use it as a tool to respond to change and adapt to delivery.	Practices active listening and is an effective communicator.	Practices active listening and is an effective communicator.
Engineering practices	Has some awareness of practices such as Git, refactoring, TDD and pair programming.	Has some experience of pair programming or refactoring. TDD and pair programming.	Has some experience of pair programming or refactoring. TDD and pair programming.	Trains on active role in helping the team manage pairing, mobbing and refactoring skills. Leads on initiatives to encourage the use of different engineering practices to suit the products and environments.	Can coach others on how to improve their pairing, mobbing and refactoring skills. Leads on initiatives to encourage the use of different engineering practices.
Technical writing	Has an awareness of how the Squad documents its work and why it's important.	Is learning to follow the established ways of working in a digital team. Can write documentation.	Is learning to follow the established ways of working in a digital team. Can write documentation.	Actively writes technical documentation as part of working in a digital team. Can write documentation.	Advocates for clear, concise and appropriate documentation. Encourages others to follow the general principle of how Squads should document their work, and coaches others on how to write better docs.

Progression frameworks start to incentivise behaviours for team types, interaction modes

If you like to
skip to the end
of a book...

Teams need to be
information radiators

That gaps will emerge
that need fixing

TT thinking influenced
things that we didn't
expect

FIELD NOTES®
fieldnotesbrand.com

Proudly printed and manufactured in the U.S.A. — "A DDC/CP JOINT"

Teams need to be information radiators

That gaps will emerge that need fixing

TT thinking influenced things that we didn't expect

Thank you!

Toli Apostolidis & Andy Norton

@apostolis09

@andyjnorton

