

Beyond the Spotify model using Team Topologies for organisation dynamics with software delivery

Matthew Skelton, Conflux
co-author of *Team Topologies* - @matthewpskelton



IPEXPO Mancs, 03 April 2019, Manchester



The Spotify Model



Limitations



Team Topologies

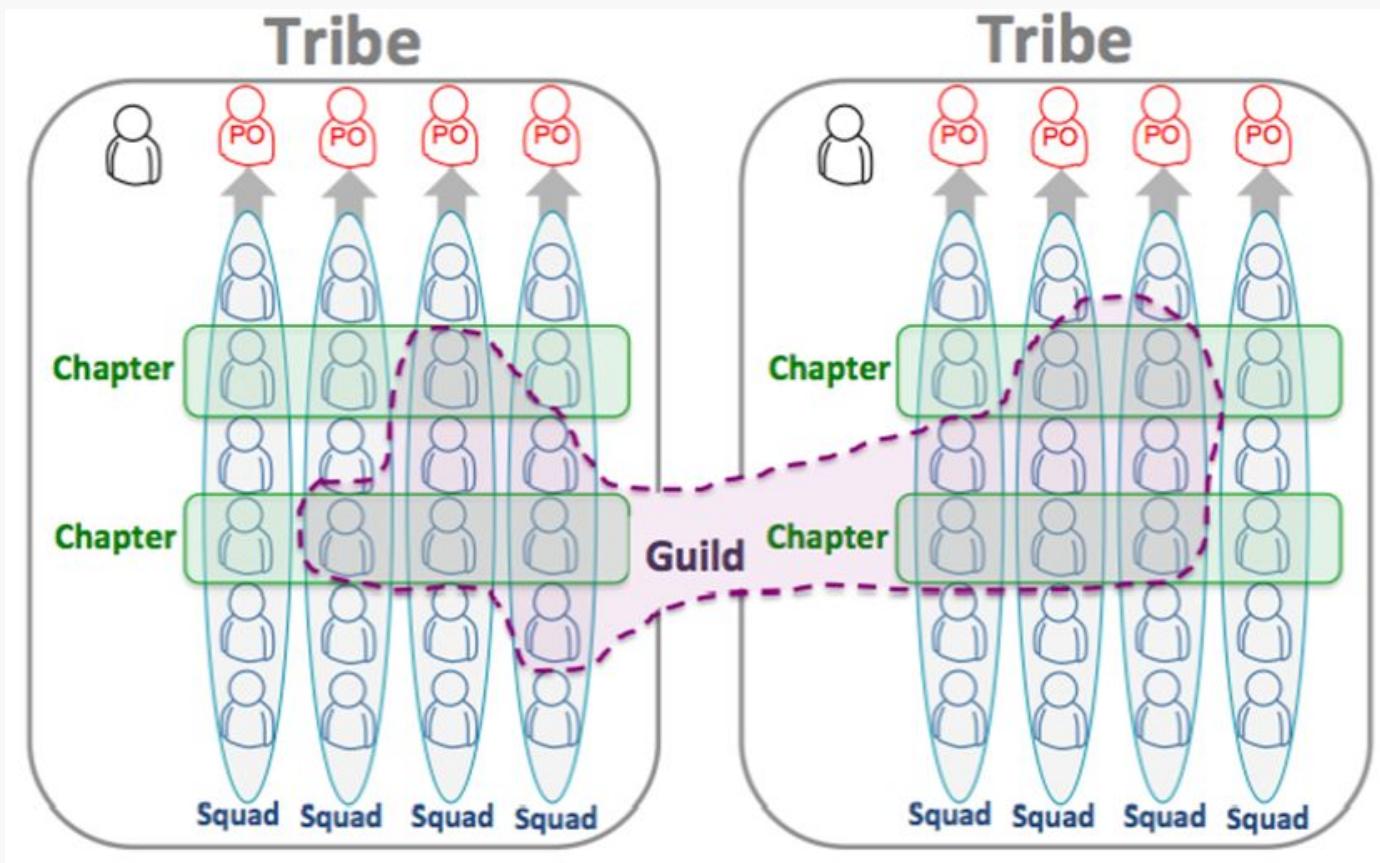


Getting started





The Spotify model of team design for software delivery



Henrik Kniberg & Anders Ivarsson, 2012

<https://blog.crisp.se/wp-content/uploads/2012/11/SpotifyScaling.pdf>

The Spotify model

Squad: semi-autonomous delivery team

Tribe: family of Squads - related work

Chapter: line management within a Tribe

Guild: cross-Tribe interest/specialist group



The Spotify model helps to...

A large, dark, corrugated metal pipe lies horizontally across a grassy field. The pipe is dark brown or black with visible vertical ridges and some rust or wear. It stretches from the left foreground towards the right background. The ground is covered in green grass and some taller, yellowish-brown plants in the foreground.

Encourage flow
of change

A photograph of a construction or roadwork site. In the foreground, several yellow traffic cones are stacked in piles. Some cones have a blue and red 'No Waiting' symbol on them. Behind the cones, there are stacks of large, dark grey concrete plates. In the background, a worker wearing a high-visibility vest and hard hat is visible near a white van. A car is parked on the right side of the image. The ground is paved and has fallen autumn leaves scattered across it.

Establish and
clarify team
responsibilities



Promote good
kinds of team
collaboration

A photograph of a wooden boardwalk path winding through a dense green forest. In the foreground, a shallow stream flows over several large, smooth stones. The water is clear, reflecting the surrounding greenery. The overall scene is peaceful and suggests a natural, outdoor setting.

Plan and budget
for cross-team
enablers

The Spotify model helps to

- Encourage flow of change

- Establish and clarify team responsibilities

- Promote good kinds of team collaboration

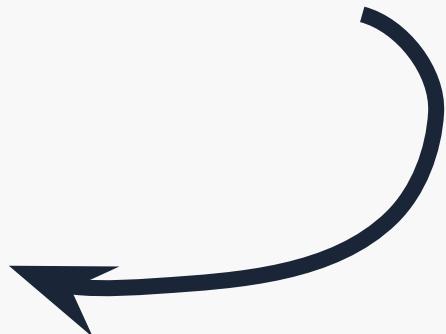
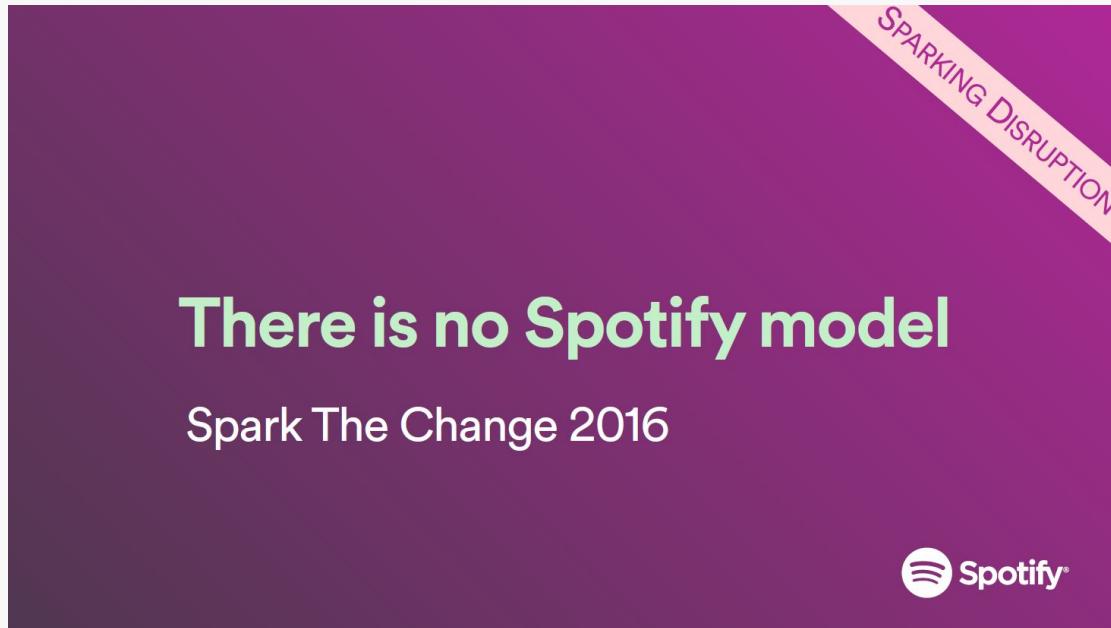
- Plan and budget for cross-team enablers

Limitations of the Spotify model

“This article is only a snapshot of our current way of working - a journey in progress, not a journey completed. By the time you read this, things have already changed.”

- Kniberg & Ivarsson

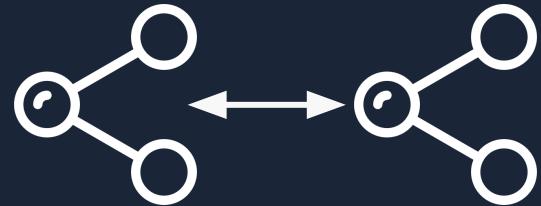
There is No Spotify Model



Software sizing and cognitive load



Heuristics for Conway's Law



Patterns for team interactions



Triggers for change and evolution



We also need to address

Software sizing and cognitive load

Heuristics for Conway's Law

Patterns for team interactions

Triggers for change and evolution

Team Topologies

topology

the way in which constituent parts are interrelated or arranged

Greek: *τοπολογία* (*τόπος* == 'place')

Team Topologies

Research over 5 years across multiple industry sectors

Informed by 50+ peer-reviewed journal articles

30+ client organizations - consulting and training since 2013 with orgs in CN, EU, IN, US, UK, +

Book: 12+ case studies from well-known organizations

Origins - DevOps Topologies

Anti-Type B: DevOps Team Silo

The DevOps Team Silo (Anti-Type B) typically results from a manager or exec deciding that they 'need a bit of this DevOps thing' and starting a 'DevOps team' (probably full of people known as 'a DevOp'). The members of the DevOps team quickly form another silo, keeping Dev and Ops further apart than ever as they defend their corner, skills, and toolset from the 'dunceless Devs' and 'dinosaur Ops' people.

The only situation where a separate DevOps silo really makes sense is when the team is temporary, lasting less than (say) 12 or 18 months, with the express purpose of bringing Dev and Ops closer together, and with a clear mandate to make the DevOps team superfluous after that time; this becomes what I have called a **Type 5 DevOps Topology**.

Anti-Type C: Dev Don't Need Ops

This topology is borne of a combination of naivety and arrogance from developers and development managers, particularly when starting on new projects or systems. Assuming that Ops is now a thing of the past ('we have the Cloud now, right?'), the developers wildly underestimate the complexity and importance of operational skills and activities, and believe that they can do without them, or just cover them in spare hours.

Such an Anti-Type C DevOps topology will probably end up needing either a Type 3 (*Ops as Infrastructure-as-a-Service*) or a Type 4 (*DevOps as a Service*) topology over time.

Type 2: Fully Shared Ops Responsibilities

Where operations people have been integrated in product development teams, we see a Type 2 topology. There is so little separation between Dev and Ops that all people are highly focused on a shared purpose; this is arguably a form of Type 1 (Dev and Ops Collaboration), but it has some special features.

Organisations such as Netflix and Facebook with effectively a single web-based product have achieved this Type 2 topology, but I think it's probably not hugely applicable outside a narrow product focus, because the budgetary constraints and context switching typically present in an organisation with multiple product streams will probably force Dev and Ops further apart (say, back to a Type 1 model). This topology might also be called 'NoOps', as there is no distinct or visible Operations team (although the Netflix NoOps might also be Type 3 (Ops as IaaS)).

Type 3: Ops as Infrastructure-as-a-Service (Platform)

For organisations with a fairly traditional IT Operations department which cannot or will not change rapidly enough, and for organisations who run all their applications in the public cloud (Amazon EC2, Rackspace, Azure, etc.), it probably helps to treat Operations as a team who simply provides the elastic Infrastructure on which applications are deployed and run; the Internal Ops team is thus directly equivalent to Amazon EC2, or Infrastructure-as-a-Service.

CC BY-SA devopstopologies.com



Philip Fisher-Ogden
@philip_pfo

Following

@manupaisable @matthewpskelton thanks for your insightful articulations of devops topologies. They inspired many discussions, and helped us think about what model Netflix teams could be/are using. Here's a recent post on that topic:

[Full Cycle Developers](https://fullcycledevelopers.com/)

Full Cycle Developers at Netflix – Netflix TechBlog – Medium
Operating what you build, empowered with amazing developer tools
medium.com

12:49 AM - 23 May 2018

8 Retweets 15 Likes

Pleasanton, CA
Joined January 2014

https://twitter.com/philip_pfo/status/999074792123740160

Philip Fisher-Ogden, Director of Engineering at Netflix:

"thanks for your insightful articulations of devops topologies. They inspired many discussions and helped us to think about what model Netflix teams could be/are using."

Kind of meta to attend the insightful talk by @cfhirschorn @QCon #qconlondon and have your upcoming book on @TeamTopologies (with @matthewpskelton) mentioned by the speaker! Nice to see how Conde Nast used

1 1 6

Manuel Pais @manupaisable · Mar 6
No worries! @matthewpskelton was who created the topologies in the first place. We've developed the thinking together through working with (and observing!) clients. We'd love to run a Q&A/interview with you at some point, amazing stuff in your talk today!

1 1 3

Crystal Hirschorn @cfhirschorn Following

Replies to @manupaisable @matthewpskelton

Absolutely, would love to discuss more about our ways of working and culture at Condé Nast. Your topological models resonated extremely well on both the Dev and Ops side btw! I like the balanced arguments, e.g. different perspectives, for each pattern.

8:11 PM - 6 Mar 2019

3 Likes 1 1 3

100 Followers you know

<https://twitter.com/cfhirschorn/status/1103387659890819073>

Crystal Hirschorn, Director of Engineering at Condé Nast International

"Your topological models resonated extremely well on both the Dev and Ops side btw! I like the balanced arguments, e.g. different perspectives, for each pattern."

Team Topologies

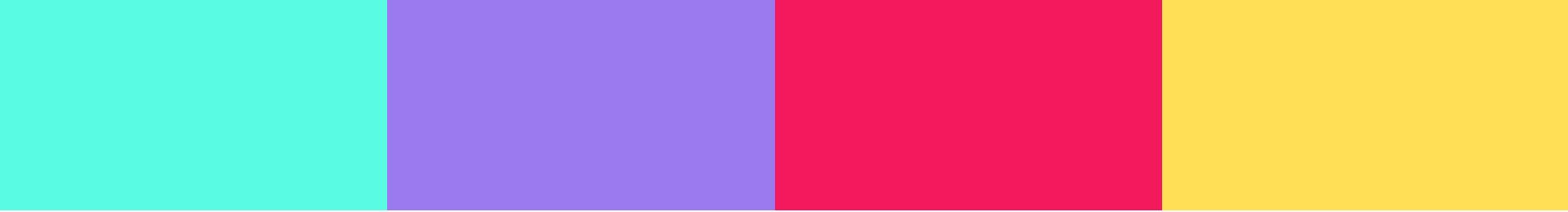
*Organizing business and
technology teams for fast flow*

**Matthew Skelton and
Manuel Pais**

Publication date: Sept 2019
IT Revolution Press

Pre-order from Amazon.com:
<https://teamtopologies.com/book>





“innovative tools and concepts for structuring the next generation digital operating model”

Charles T. Betz, Principal Analyst, Forrester Research

Team Topologies for fast flow

Team-first
Thinking

Conway's Law

Team
Interactions

Sensing for
Evolution

Software sizing and cognitive load



Team-first Thinking



Team-first Thinking



The team is the means of delivery

Team-first Thinking



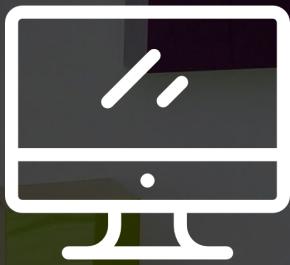
Design for team cognitive load

Team-first Thinking



Choose boundaries for team ownership

Team-first Thinking



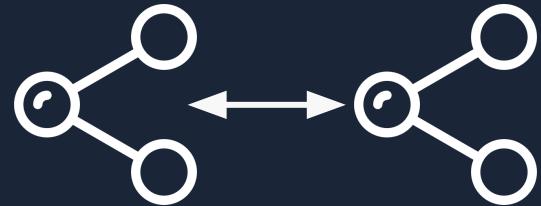
Physical and digital workspace

Team-first Thinking



**The team is the means of delivery
Design for team cognitive load
Choose boundaries for team ownership
Physical and digital workspace**

Heuristics for Conway's Law



Conway's Law ONES



Conway's Law

ONES



Heuristic for 'natural' expected design



Conway's Law

Mirroring in tech system + human system



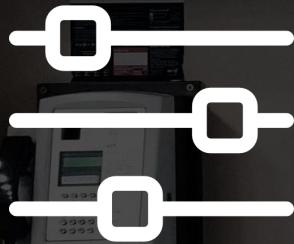
Conway's Law



Reverse Conway to mitigate worst effects

Conway's Law

ONES



Constraint on solution search space

Conway's Law



Heuristic for 'natural' expected design

Mirroring in tech system + human system

Reverse Conway to mitigate worst effects

Constraint on solution search space

Patterns for team interactions



Team Interactions



Team Interactions



3 defined Interaction Modes

Team Interactions



Collaboration: 2 teams working together



Team Interactions



X-as-a-Service: 1 provides, 1 consumes

Team Interactions



Facilitating: 1 team helps another

Team Interactions



3 defined Interaction Modes

Collaboration: 2 teams working together

X-as-a-Service: 1 provides, 1 consumes

Facilitating: 1 team helps another

4 fundamental topologies

 Stream-aligned team



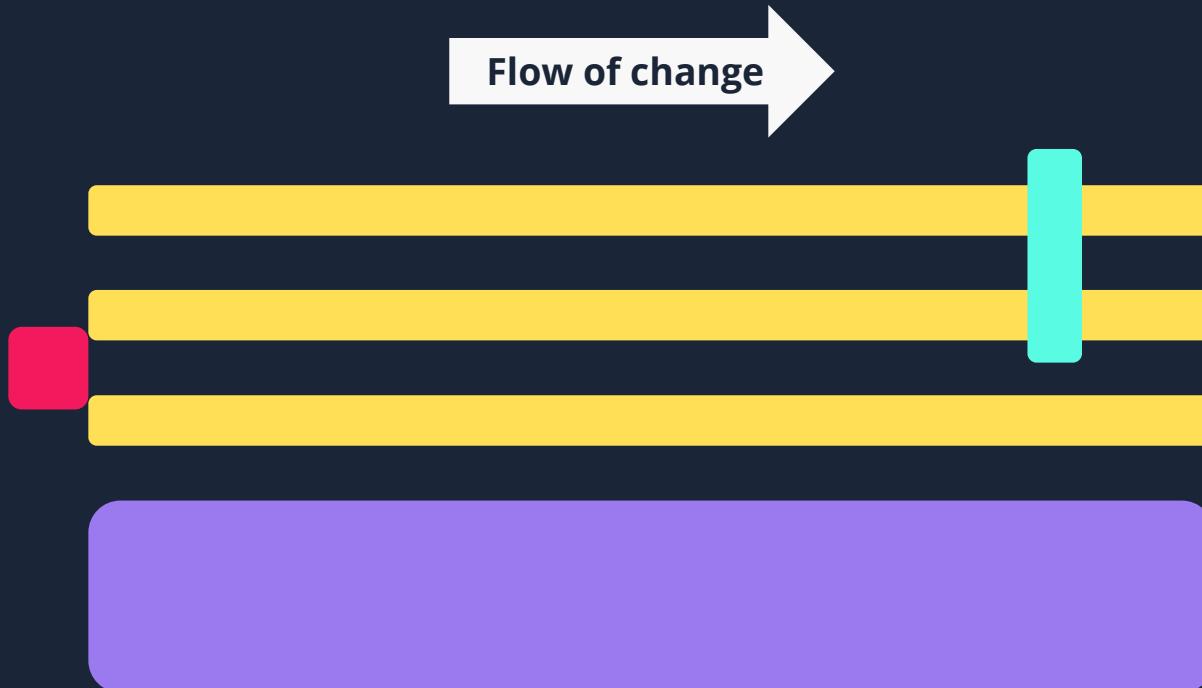
Enabling team



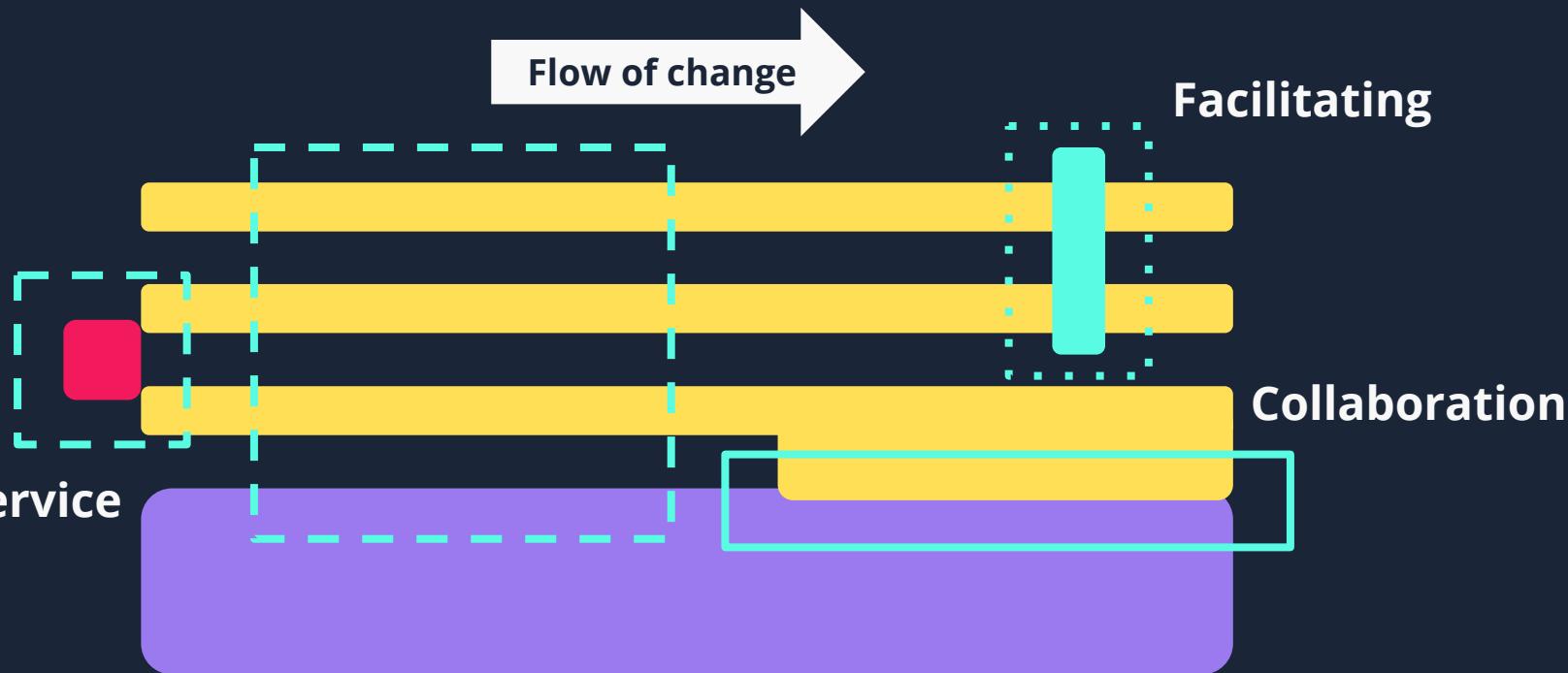
Complicated Subsystem team

 Platform team

4 fundamental topologies



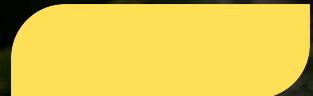
3 core interaction modes



Triggers for change and evolution



Sensing for Evolution



Sensing for Evolution



Not all teams in the org look the same



Sensing for Evolution



Discover, then push to Platform

Sensing for Evolution



Awkward team interactions are signals

Sensing for Evolution



Evolve the org with changing ecosystem

Sensing for Evolution



**Not all teams in the org look the same
Discover, then push to Platform
Awkward team interactions are signals
Evolve the org with changing ecosystem**

Getting started with the Team Topologies approach

Getting started



Explicit cognitive load

Explicit cognitive load

How well can the team as a unit “grok” the systems they own and develop?

Push some things into a Platform?

Are skills or capabilities missing?

Getting started



Large Conway mismatches

Large Conway mismatches

Are there major mismatches between
the team interactions and the required
software / system architecture?

What could be easily adjusted?

Getting started



Team Interactions

Team Interactions

What would change if we adopted the
3 team interaction patterns?

Collaboration, X-as-a-Service, Facilitating

How would teams react & behave?

Getting started

Thinnest Viable Platform

Thinnest Viable Platform

How is your Platform defined?

What is the thinnest platform that could work?

What's needed to run and support it?

Getting started

-  Explicit cognitive load
-  Large Conway mismatches
-  Team Interactions
-  Thinnest Viable Platform

Review

The Spotify model helps to

Encourage flow of change

Establish and clarify team responsibilities

Promote good kinds of team collaboration

Plan and budget for cross-team enablers

We also need to address



Software sizing and **cognitive load**



Heuristics for **Conway's Law**



Patterns for **team interactions**



Triggers for **change and evolution**

Team Topologies for fast flow

Team-first
Thinking

Conway's Law

Team
Interactions

Sensing for
Evolution

Getting started

-  Explicit cognitive load
-  Large Conway mismatches
-  Team Interactions
-  Thinnest Viable Platform

Sign up for news and tips:

TeamTopologies.com

Thank you!

info@teamtopologies.com



Matthew Skelton, Conflux
@matthewpskelton



Manuel Pais, Independent
@manupaisable



Copyright © [Conflux Digital Ltd](#) 2018-2019. All rights reserved.
Registered in England and Wales, number 10890964

Icons made by [Freepick](#) from [www.flaticon.com](#) - used under license