

An outsider assessment of Spotify Engineering Culture by an insider

Jason Yip

Agile Coach, Monetization Tribe, [Spotify NYC](#)

jiy@spotify.com

[@jchyip](#)

<https://jchyip.blogspot.com>

<https://medium.com/@jchyip>

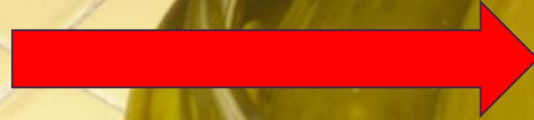


“Cucumbers get
more pickled
than brine gets
cucumbered.”

Jerry Weinberg



Pickle = Jason



Jar of brine
= Spotify



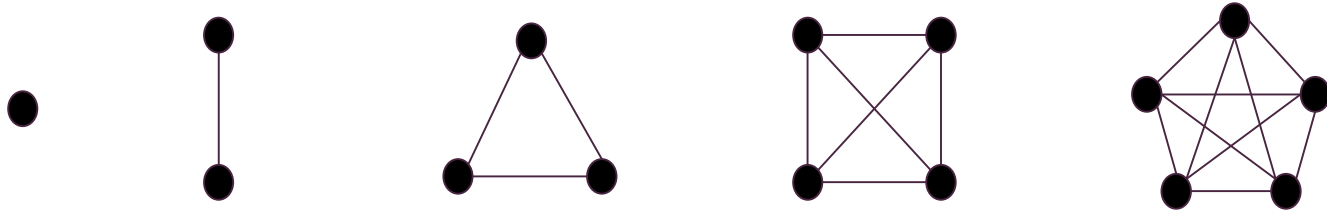
**How would I assess
Spotify if I were a
cucumber?**



**What problem is
Spotify trying to solve?**



Everything comes down to “How do you deal with rapid growth?”



$$\# \text{ of communication channels} = n(n - 1) / 2$$

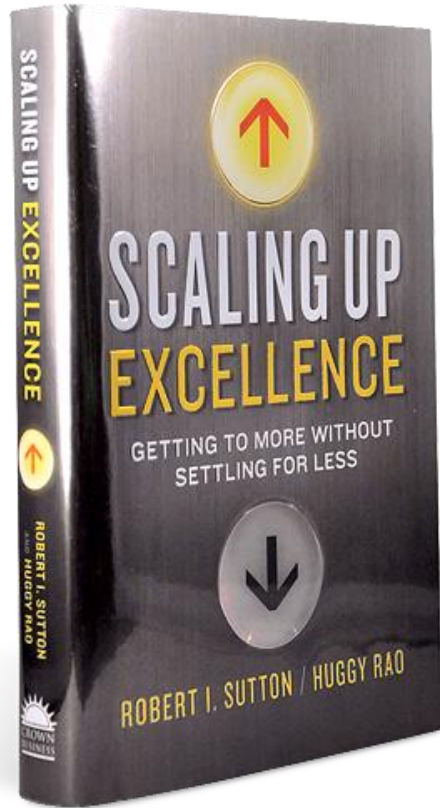
**How might you ~~maintain~~
level up speed and
quality even if you double
in size?**

Proposed target condition

The experience of working in Spotify Engineering is one of being **free of friction**, where **everything makes sense**, where **you're not surprised by strange decisions**, where **you can trust everything works as expected**, and where there is a **strong sense of safety** that allows people to produce **high quality outcomes, quickly, and efficiently.**

The air war vs the ground war





SCALING UP EXCELLENCE



ROBERT I. SUTTON
HUGGY RAO



SCALING UP EXCELLENCE

GETTING TO MORE WITHOUT
SETTLING FOR LESS

ROBERT I. SUTTON / HUGGY RAO

“...those who master what venture capitalist Ben Horowitz calls ‘the black art of scaling a human organization’ act as if they are fighting a ground war, not just an air war.”

“Air war”

- ▶ Leadership buy-in
- ▶ Messaging
- ▶ Values
- ▶ Etc.



LIFE

“Ground war”

- ▶ Individual buy-in
- ▶ Day-to-day conversations
- ▶ Day-to-day behaviour
- ▶ Day-to-day routines
- ▶ Policy
- ▶ Etc.

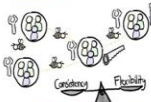


Spotify has air superiority

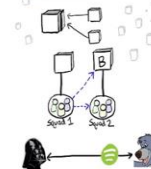
Spotify Engineering Culture Part 1 of 2

Mark Fishers Sep 2016

Cross-pollination > Standardization



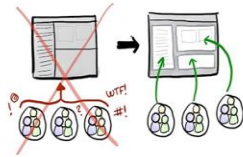
Internal Open-source model



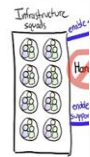
Small & frequent releases



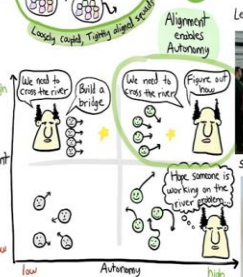
Decoupled releases



Self-service enable > sc



You are the culture



Spotify Engineering Culture Part 2 of 2

Mark Fishers Apr 2016

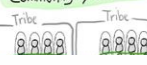
Continuous Improvement



Impact > Velocity



Community > Structure



Fail Fast > Learn Fast > Improve Fast



Lean Startup



Innovation > Predictability

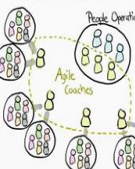
100% probability = 0% innovation



Value delivery > Plan fulfillment



Culture-focused Roles



Healthy Culture heals Broken Process



Limited Blast Radius



Hack Time = 10%



Spotify Hack Week



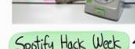
Visual Progress



Experiment-friendly Culture



Waste-repellent Culture



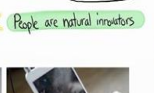
Weekly Demo



Daily Sync



Waste-repellent Culture



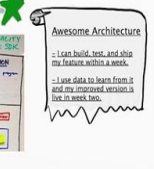
Minimize the need for Big Projects



Visual Progress



Definition of Awesome



Chaos > Bureaucracy



Improvement Boards



Improvement Boards



Improvement Boards



**The ground war is fought
Tribe to Tribe, Squad to
Squad, person to person**

**Here's a taste of the
ground war**



ify Culture

of 2

erg

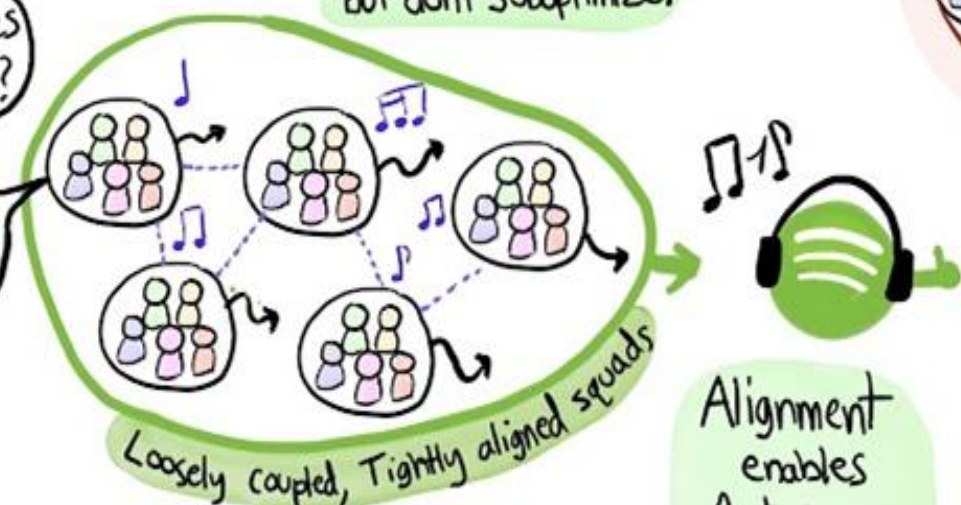
Standardization



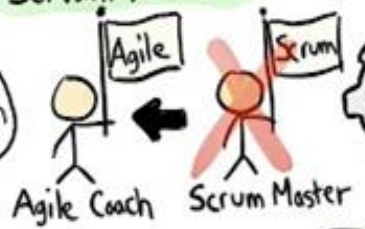
Flexibility

ce model

How does Spotify X?
Depends on which squad

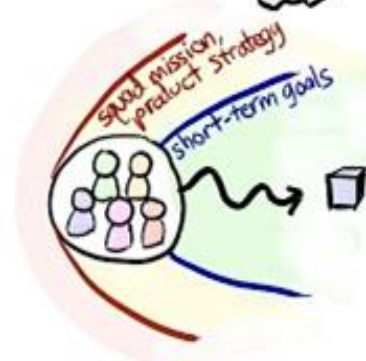


Agile > Scrum
Principles > Practices
Servant > Master



Autonomous Squads

Be autonomous, but don't suboptimize!

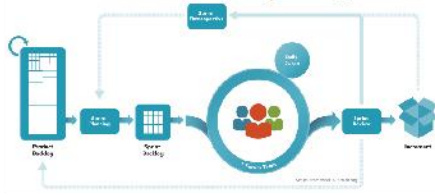


Alignment enables Autonomy

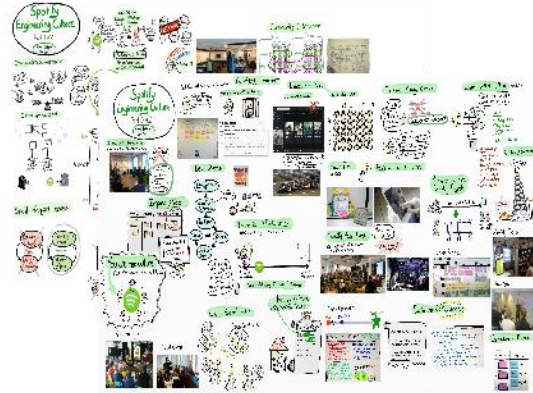
Leader's job: Communicate what needs to be solved.



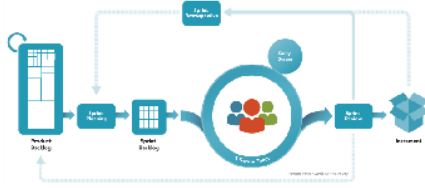
Scrum is legacy...



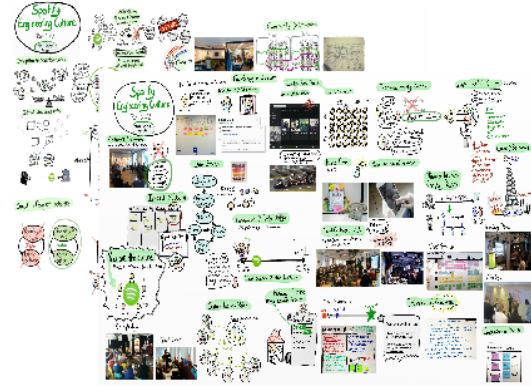
Agile @ Spotify



Scrum is legacy...



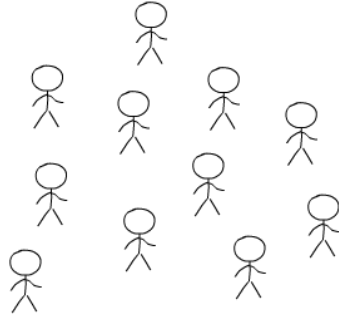
Agile @ Spotify



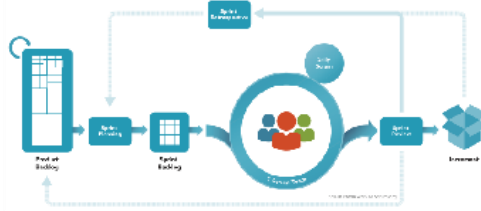
...except maybe not for all
the new people coming in...



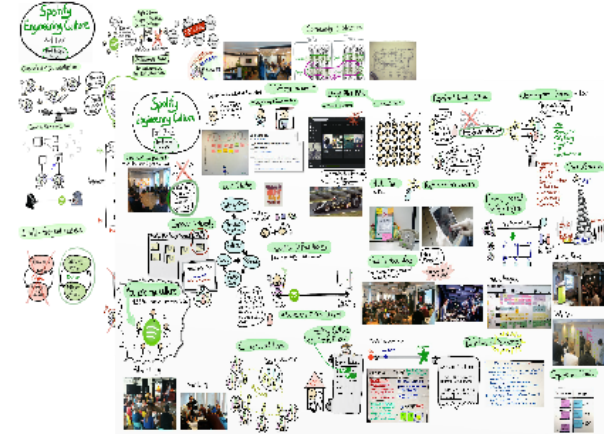
Gaps in basic concepts
and practices



Scrum is legacy...



Agile @ Spotify



...except maybe not for all the new people coming in...



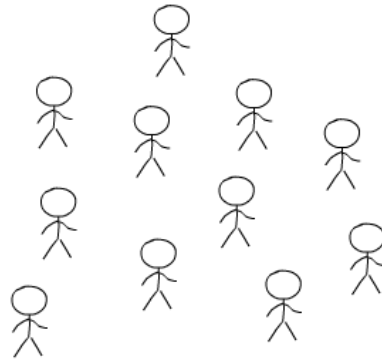
onboarding

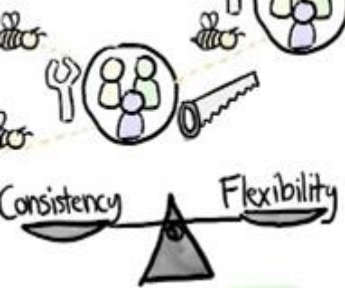


Embedded support

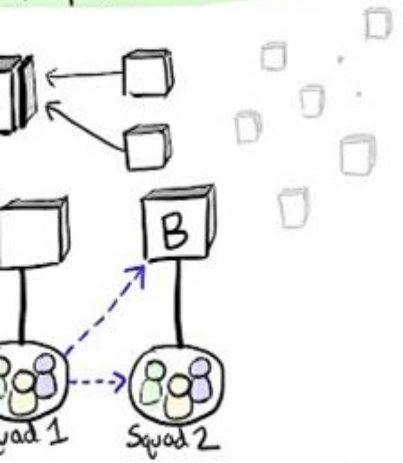
Fundamentals training

Selection criteria

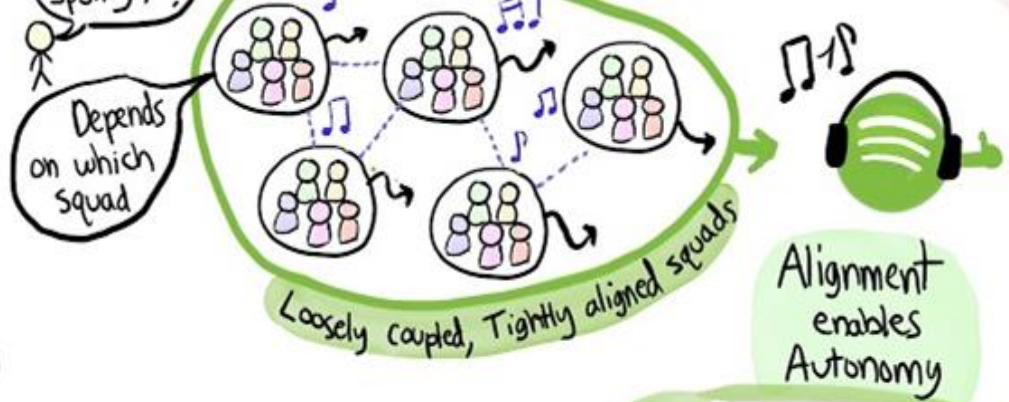




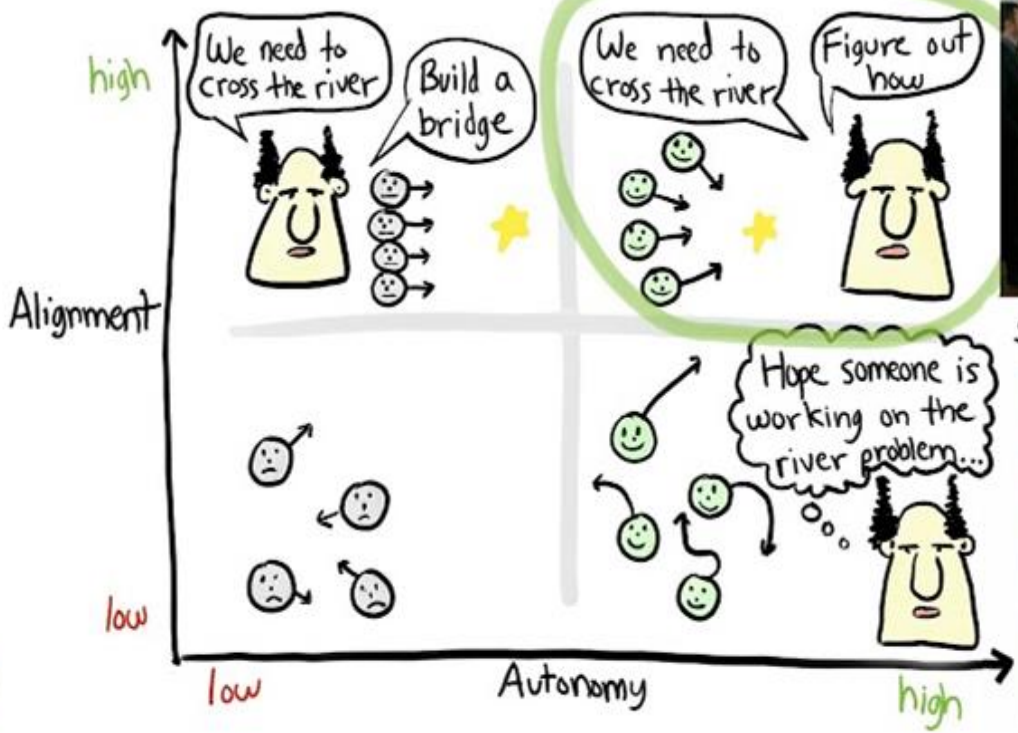
Open-source model



frequent releases



Leader's job: Communicate what problem needs to be solved. And...



Squads' job: Collaborate with each other to find the best solution.



Alignment



Shared
intent

Shared
decision
principles

Spotify Engineering Core Beliefs

- ▶ **Speed of learning and execution lead us to create a best in class product**
- ▶ **The fewer technologies we are world class on, the faster we get**
- ▶ **Shorter iterations yield faster learning, quicker value and higher quality**
- ▶ **Small empowered cross-functional teams (squads) is the foundation of fast paced product development and innovation**
- ▶ **Strong teams always beat rock stars**
- ▶ **Your ability to learn and adapt is more important than your knowledge, merit and experience**

Spotify Engineering Core Beliefs

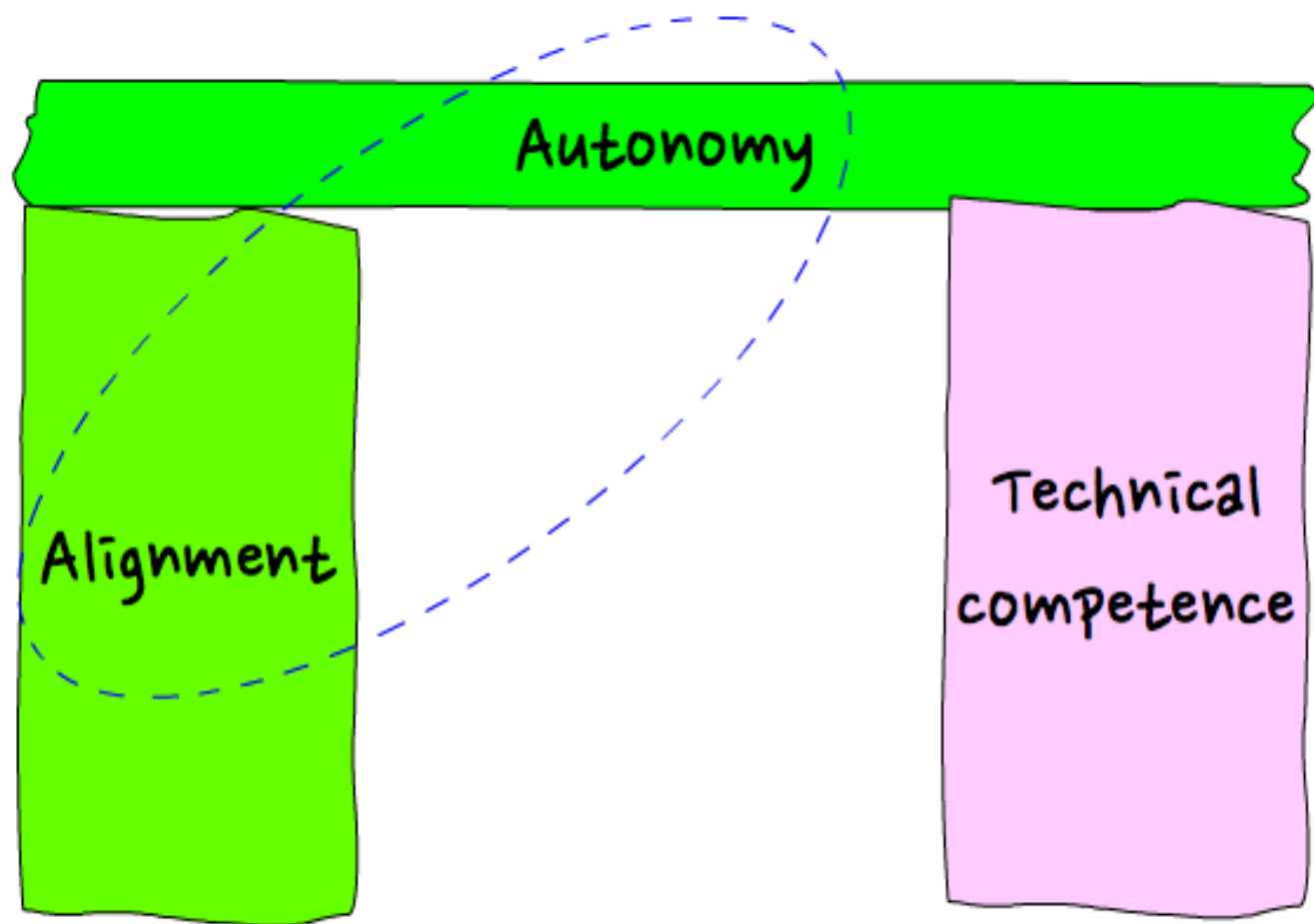
- ▶ Speed of learning and execution lead us to create a best in class product
- ▶ The fewer technologies we are world class on, the faster we get
- ▶ Shorter iterations yield faster learning, quicker value and higher quality
- ▶ Small empowered cross-functional teams (squads) is the foundation of fast paced product development and innovation
- ▶ Strong teams always beat rock stars
- ▶ Your ability to learn and adapt is more important than your knowledge, merit and experience

How might this reliably affect decisions?

Good shared decision principles are:

- ▶ **Role modeled by leaders and influencers**
- ▶ **Supported by systems and policies**
- ▶ **Communicated explicitly**
- ▶ **Mentioned regularly**

Alignment isn't enough to enable autonomy



Existing initiatives to level up technical competence

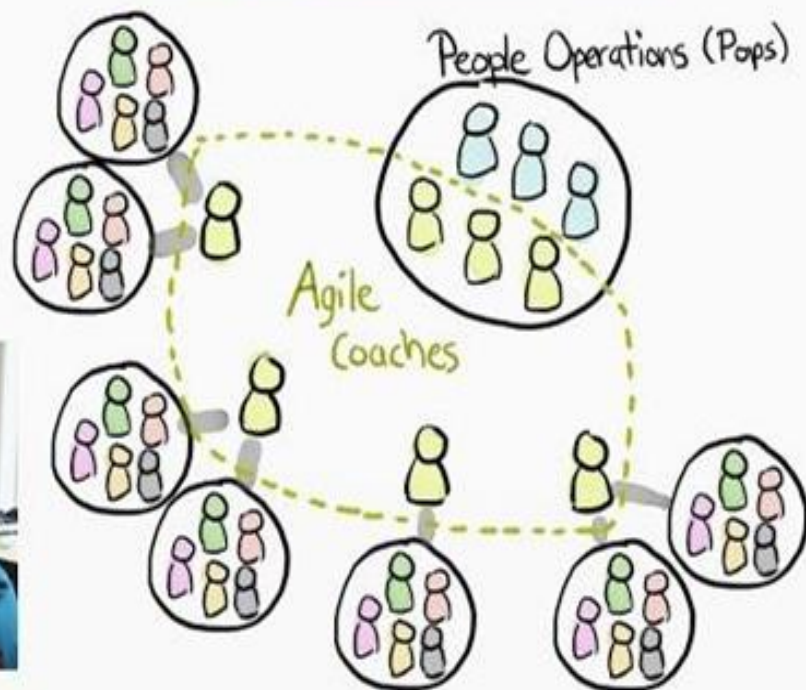
- ▶ **#technical-excellence**
 - Get more people talking about it
- ▶ **“Fearless Client Development”**
 - Turn this into a prioritised TPD Bet

are super-passionate about.



Value delivery / team performance

Culture-focused Roles



Healthy Culture
heals Broken Process



Pains + Ch

FIXED

FI

- w w w
- w w w
- w w w
- w w w

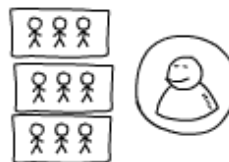
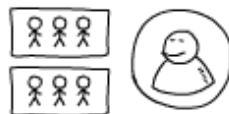


Boot Camp

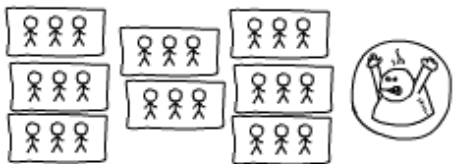
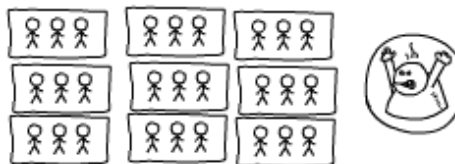
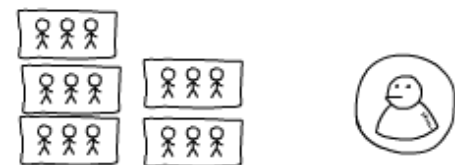




ScrumMasters



Agile coaches

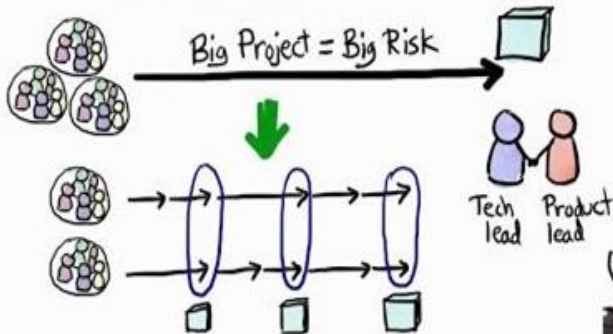


Agile coaches with
rapid growth

can't just rely on Agile coaches to cover
"Agile culture" -> selection, onboarding, etc.

Minimize the need for Big Projects

- Task estimates
- Useless meetings
- Corporate BS



Weekly Demo

cool things real!

Party on Friday!

Visual Progress



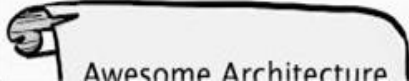
Daily Sync



Definition of Awesome

Awesome Architecture

DEFINITION OF AWESOME - NILS



**When you can't avoid the
big project...**

**Spotify is not designed
for this**

How to reliably suffer with big projects

"We're autonomous!"



"We're autonomous!"



"We're autonomous!"



*Not actually
decoupled efforts



Final integration!

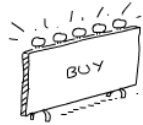


"I wonder what they're up to?"



Business teams

"Let's not bog down the
teams with the messy stuff"

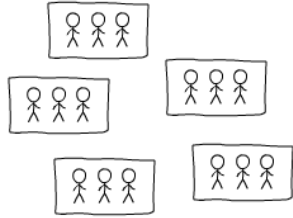


Launch!

*"messy stuff" =
relevant context



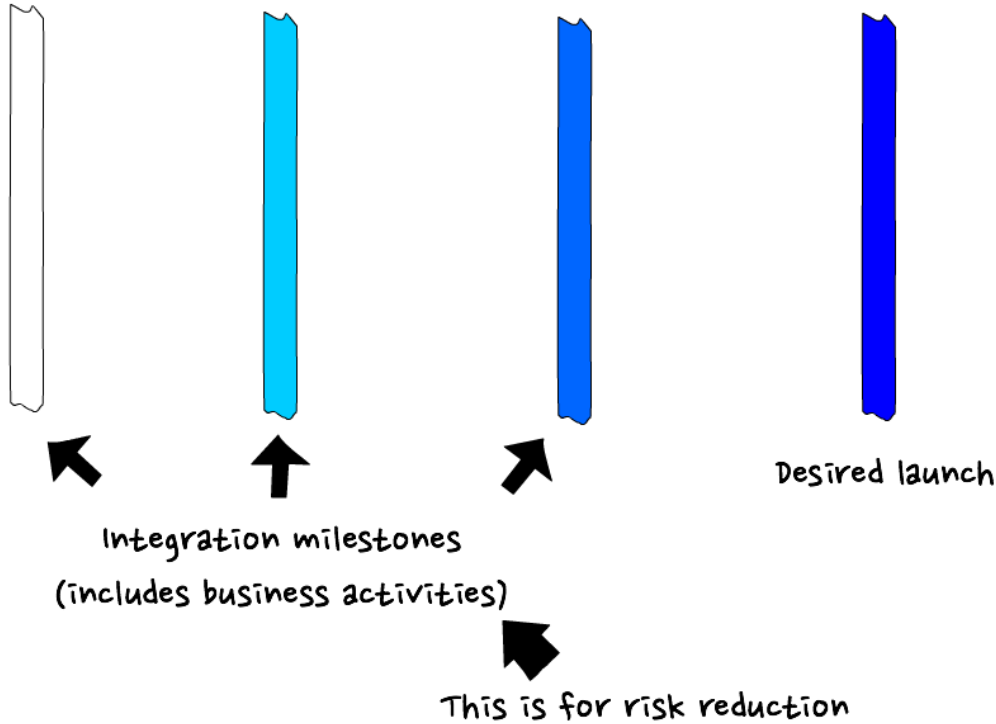
Kickoff



What are we trying to accomplish?

What do we need to coordinate?

How might we design this to make coordination easier?



Summary:

Recommended priorities

Summary

- ▶ **Selection, onboarding, embedded support to cover Agile fundamentals gap with rapid growth**
 - Agile coaches cannot compensate
- ▶ **Level up autonomy by more explicitly addressing shared decision principles and technical excellence**
- ▶ **Establish kick-offs and integration milestones for bigger projects**

Appendix



Who was that guy?

- ▶ **Jason Yip**
- ▶ **Agile Coach at Spotify since Feb 2015**
- ▶ **Previously Principal Consultant at ThoughtWorks (started in February 2001)**
- ▶ **First encountered Extreme Programming in 1998**

Spotify Engineering Culture videos

- ▶ **Part 1:**

<https://labs.spotify.com/2014/03/27/spotify-engineering-culture-part-1/>

- ▶ **Part 2:**

<https://labs.spotify.com/2014/09/20/spotify-engineering-culture-part-2/>

**Thank you
for your
attention**

