SSW-555: Agile Methods for Software Development

Agile Practices at

Spotify®

Dr. Richard Ens
Software Engineering
School of Systems and Enterprises

Today's topics

Background on Spotify

Agile practices used at Spotify

Videos by Spotify

Acknowledgements

"Scaling Agile @ Spotify with Tribes, Squads, Chapters, and Guilds", Kniberg and Ivarsson, 2012.

"Spotify Engineering Culture Part 1-2", https://sit.instructure.com/courses/15619/modules/items/311229

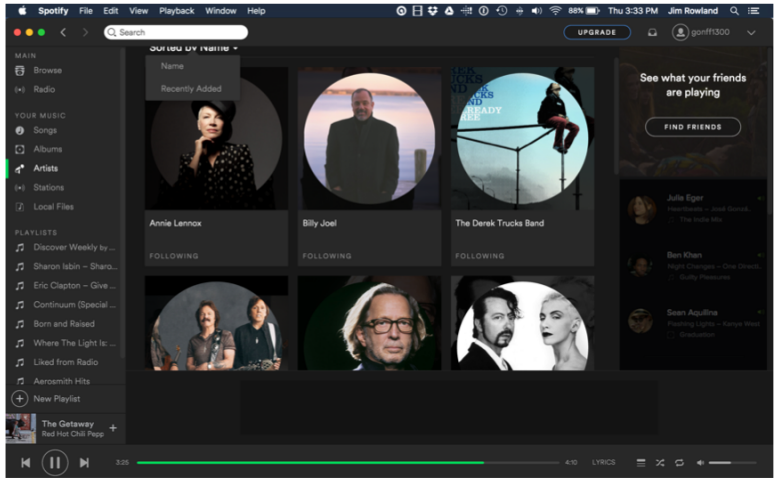
Spotify: Music Streaming Service

Freemium model

- Basic account is free with ads
- Premium accounts cost money

History

- 2006: company founded in Sweden
- 2008: service launched in Europe
- 2011: launched in US



Platforms

- Desktop (Windows, Mac, Linux)
- Mobile (IOS, Android)

Agile practices at Spotify

2000+ developers working on the Spotify product (2017)

- 70+ squads across 5 cities (2015)

Spotify developers ***do not*** follow a single Agile method

Began with Scrum but found that it "got in the way"

- Agile > Scrum
- Principles > Practices

Each development group chooses methods that work for them

- e.g. Scrum, Kanban, Lean, …

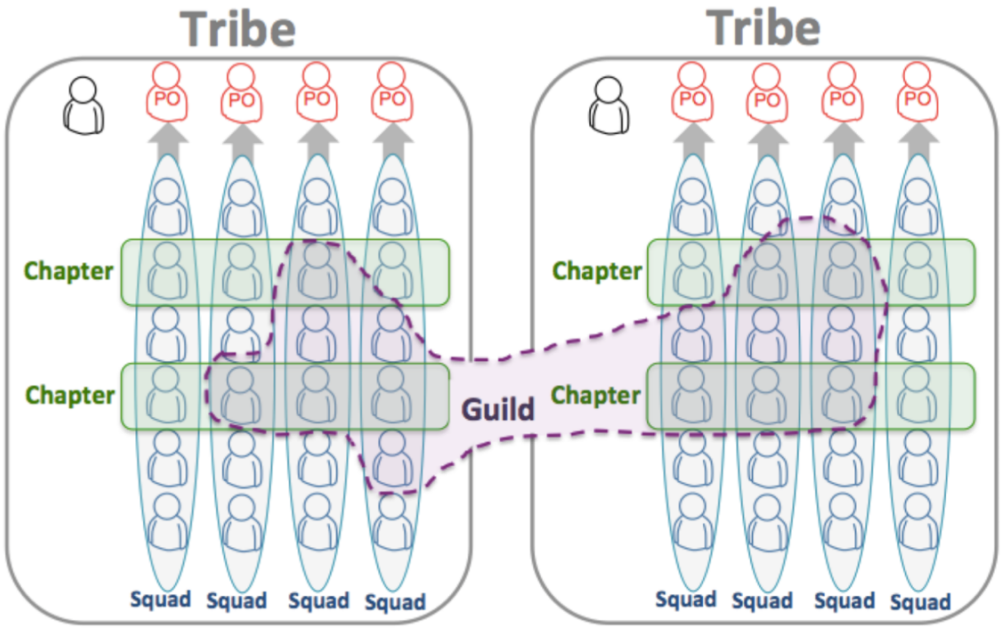
Spotify organizational structure

Squads
- Small teams: <= 8 people

Tribes
- Collections of squads

Chapters
- Groups within a tribe that share a specialty or responsibility

Guilds
- Cross-tribe communities of interest



Squad

Small, ***autonomous***, agile teams

- Loosely coupled, but tightly aligned

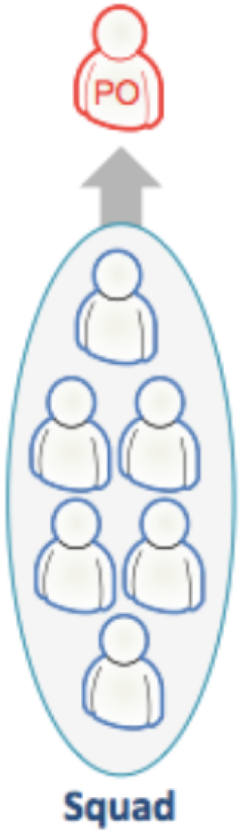
One long-term mission

- End to end responsibility: develop, test, deploy, maintain…
- E.g. IOS client, Android Client, DB backend, …

Self organizing

Choose their own methods and procedures

- Each squad may use a different approach



Squad Roles

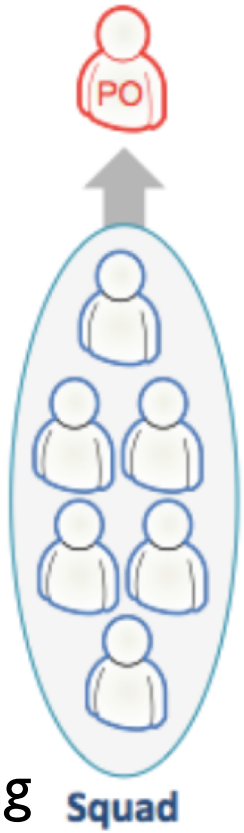
No formal leader

Product owner

- Prioritizes work from backlog
- Collaborates with other product owners to coordinate between squads
  - Squads interact through Product Owners

Agile coach

- Agile coach may work with multiple squads
- Help to run meetings, retrospectives, sprint planning
- Help squads with Agile processes



Quarterly Reflection Survey

Each squad conducts a quarterly survey to reflect on what's working and not working well and the trends

Area	Squad 1	Squad 2	Squad 3	Squad 4	Squad 5
Product owner					
Agile coach					
Influencing work					
Easy to release					
Process that fits team					
A mission					
Org. support					

Tribe

Collection of squads in related areas

- ≦~ 100 people

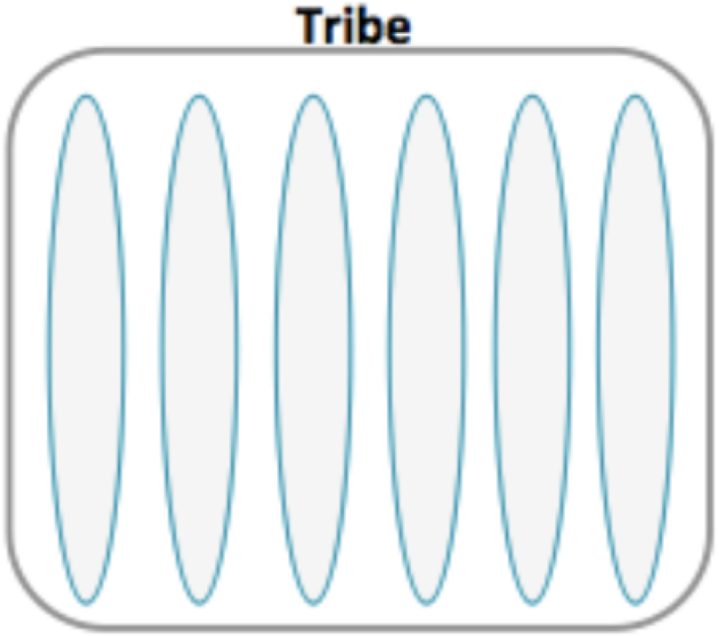
Physically co-located

May have a System Owner

- Represents Product Owners across Tribe

May have a Chief Architect

- Responsible for overall architecture and design across Tribe



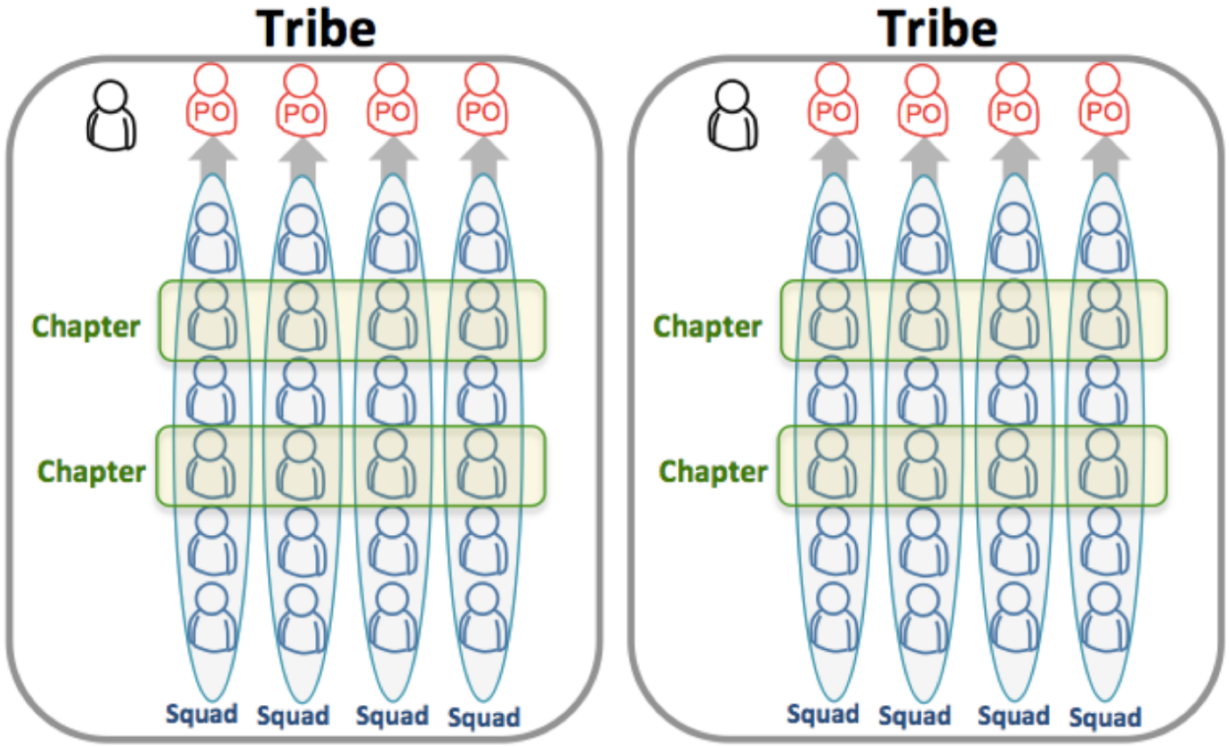
Chapter

Take advantage of economies of scale without loss of autonomy

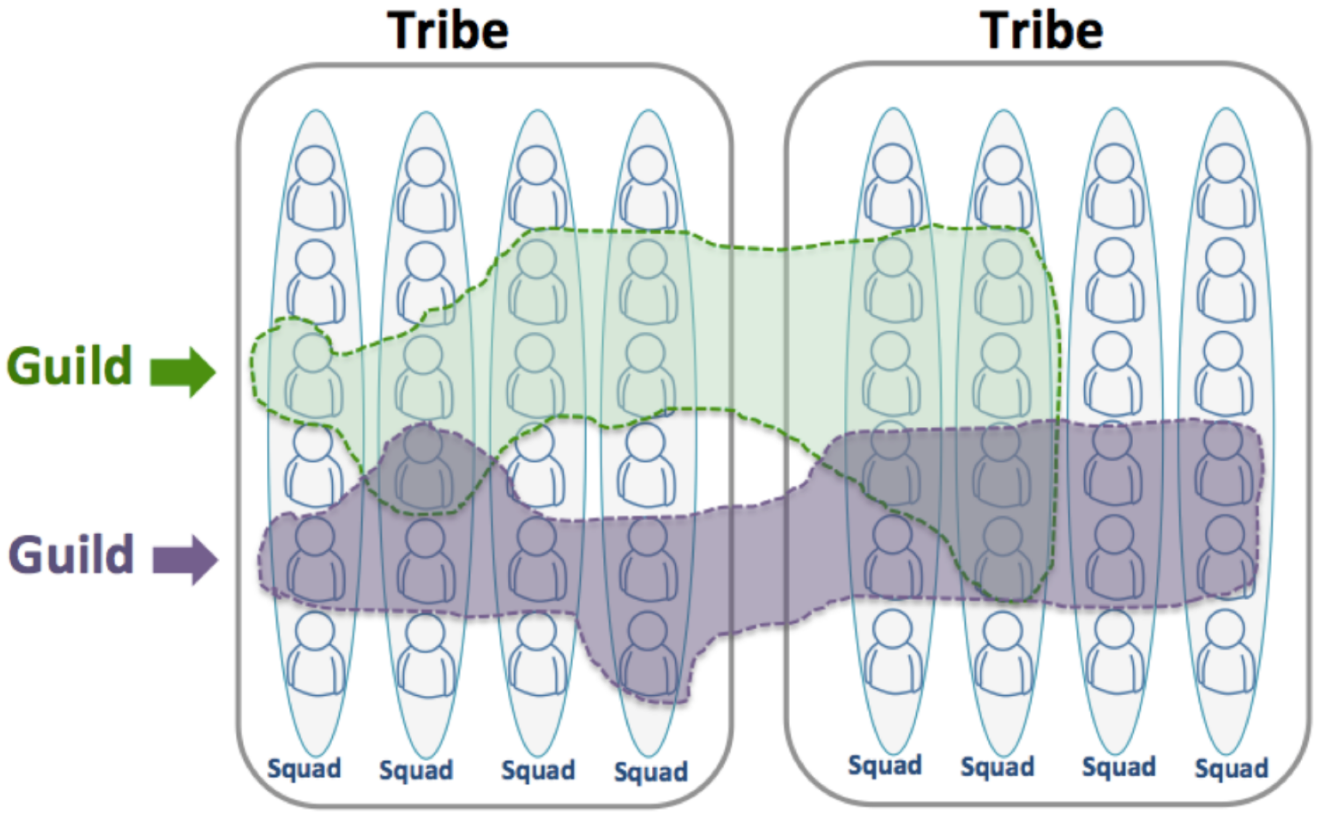
People with similar interests, e.g. testing, databases, etc…

Focus on competencies

Chapter lead is line manager



Guild – Community of Interest



Guild members share a common interest and share information
e.g. database technology, processes, etc.

Birds of a Feather (BOF) groups

System Owner

A large, complex system requires technical oversight to insure technical integrity

- Need an overall architecture

Each major system has a System Owner

System Owner coordinates decisions about the system

Owns quality, documentation, technical debt, reliability, scalability, etc.

Part time responsibility along with development

Chief Architect

Responsible for high-level architecture across multiple systems

Review new development for integrity and alignment

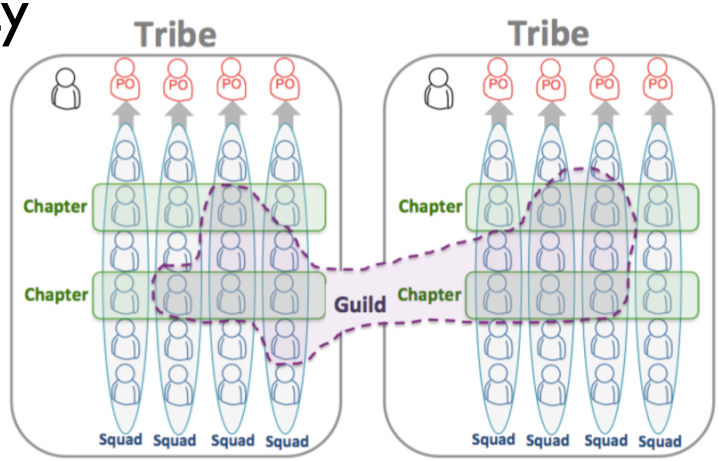
Offers advice – squad is ultimately responsible

Summary of organizational structure

Squads
- Focus on product delivery and quality

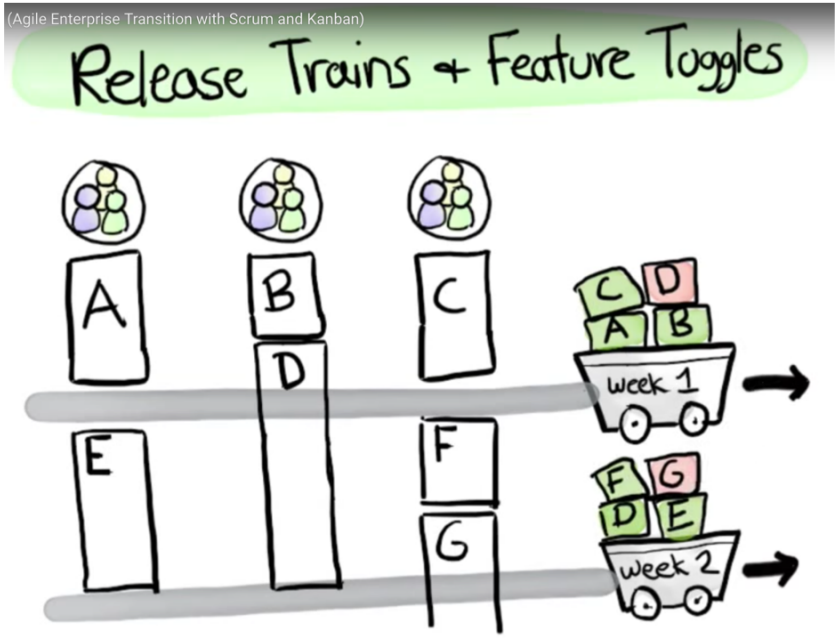
Tribes
- Lightweight matrix

Chapter
- Competency area
  - E.g. Agile Methods, web development, …
- Chapter Leader is the line manager



Guild
- Lightweight community of interest across the company
- Flexible membership

Release Management at Spotify



"Think it, build it, ship it, tweak it"

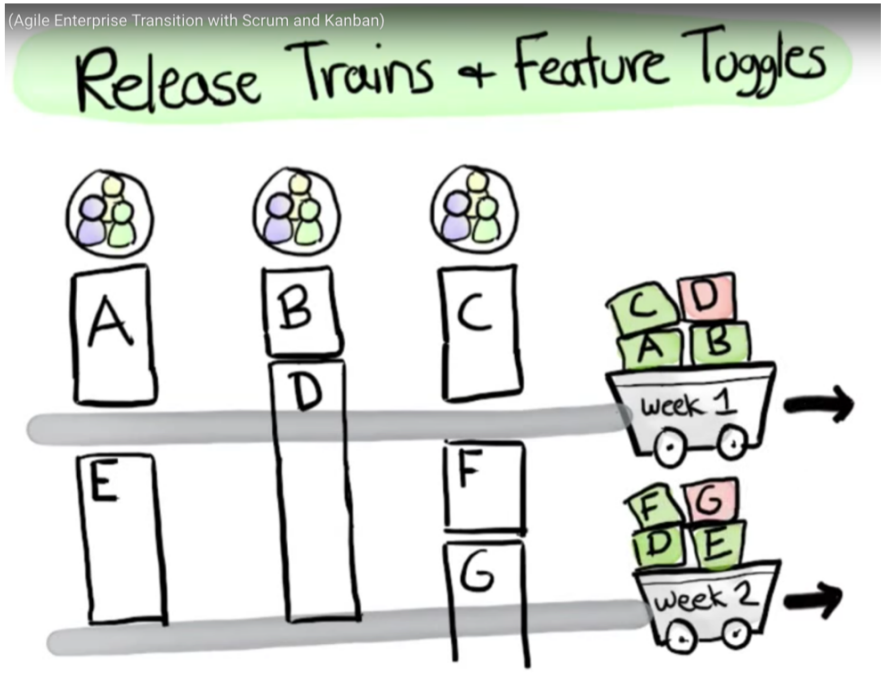
Focus on frequent releases with user feedback to choose best alternative

Weekly releases to production

Release early and often

Source: https://www.youtube.com/watch?v=Mpsn3Wal_4k

Release Management at Spotify



Source: https://www.youtube.com/watch?v=Mpsn3WaI_4k

Agile Release Trains

Each squad makes their own releases without central control

All complete features and some incomplete features

Incomplete features are not visible to users through **Feature Toggles**

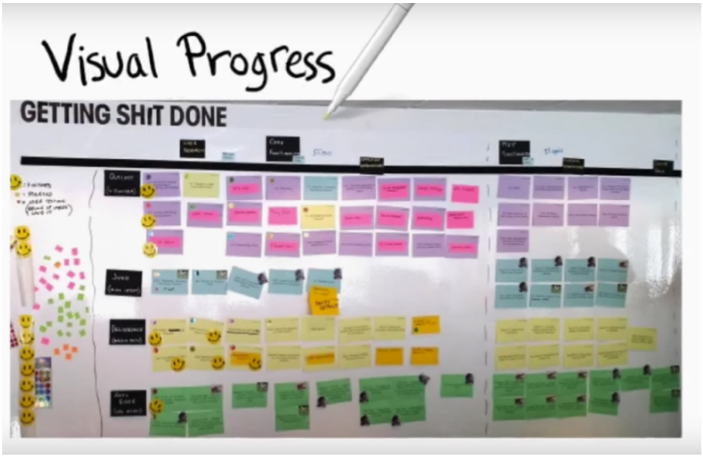
Identifies integration issues early

New features are rolled out to few users at a time until proven

Limited blast radius

Practices at Spotify

- Agile Release Trains
- Visualize progress, e.g. Kanban
- Weekly demos/Story Telling
- Daily sync (stand up meetings)
- Break big projects into small projects with frequent releases
- Boot camp
- Hack Time



Spotify Strategies

Emphasis on culture

"Healthy Culture heals Broken Process"

Eliminate waste

Balance between chaos and bureaucracy

Emphasis on innovation

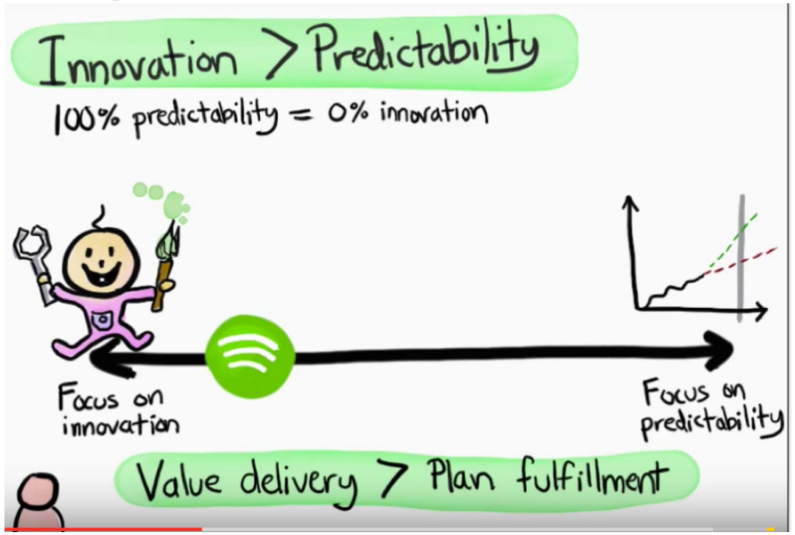
Trade innovation for predictability

Experiment-friendly culture

Limited Blast Radius of failure

Gradual rollout with feature toggles

Fail Fast, Learn Fast, Improve Fast



Spotify Videos

"Spotify Engineering Culture Part 1"

https://www.youtube.com/watch?v=4GK1NDTWbkY

"Spotify Engineering Culture Part 2"

https://www.youtube.com/watch?v=X3rGdmoTjDc

Questions?