Scaling Agile

Thomas Luvö
@tomluvoe
tom@samoht.se

Today's lecture

- Recap Scrum and Project Management
- Agile
- Scaling Agile
- Case study: Spotify

Scrum

- Software project framework for self-organizing teams, to run iterative and incremental development

Project Management

- Plan, organize, motivate and control a project
- To reach the project's objectives within the project's constraints

- Remember, that when Agile is described, it is almost always done in the light of that we are most familiar with Waterfall software development

Agile

"'Agile' is not a practice. It's a quality of the organization and its people to be adaptive, responsive, continually learning and evolving - to be agile, with the goal of competitive business and rapid delivery of economically valuable products and knowledge"

Larman & Vodde, Scaling Lean & Agile Development

Agile manifesto

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

http://agilemanifesto.org

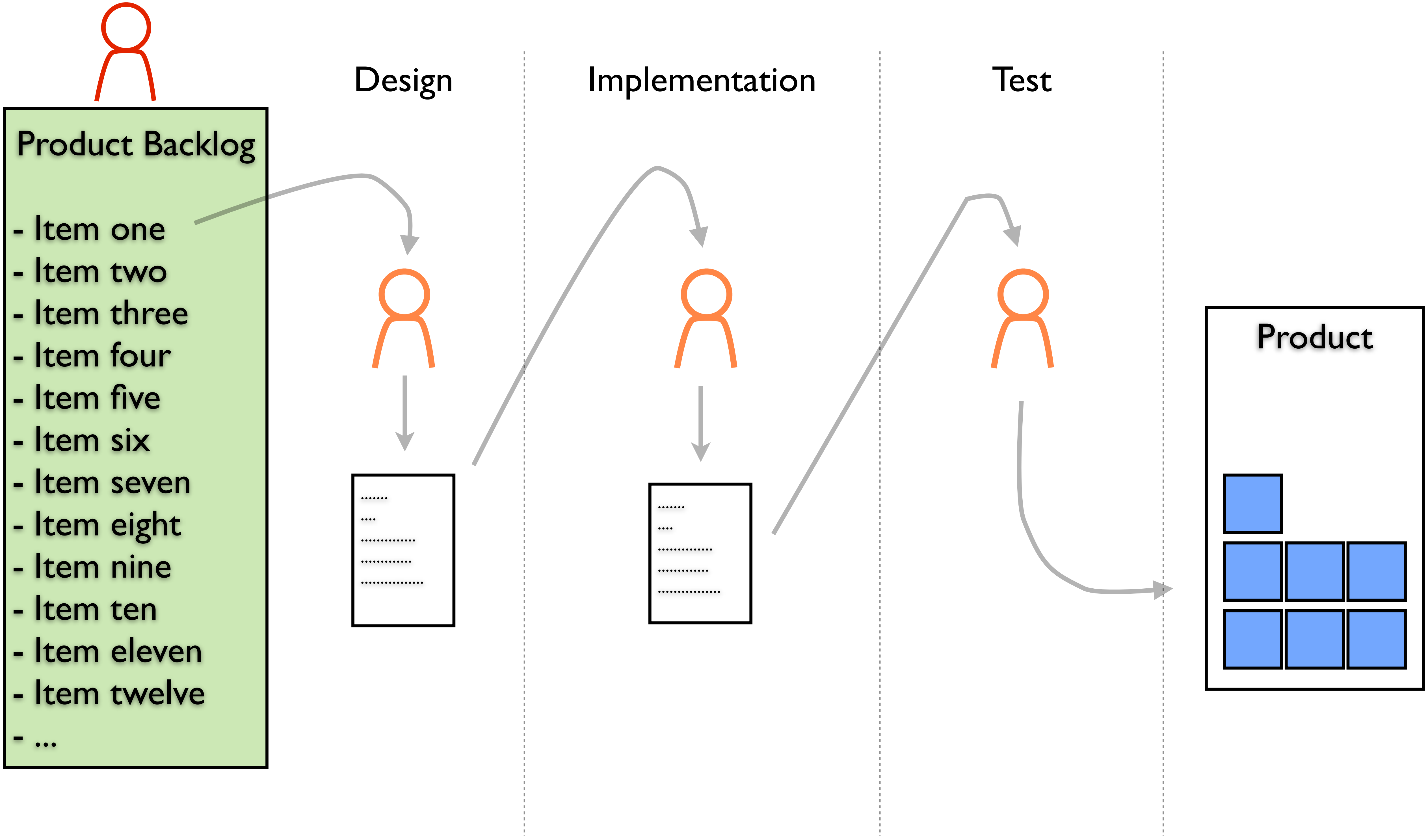
Agile principles

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development.Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development.The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity--the art of maximizing the amount of work not done--is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Scaling Agile

- Complex products
- Large organizations

“Waterfall”



Product Backlog

- Item one
- Item two
- Item three
- Item four
- Item five
- Item six
- Item seven
- Item eight
- Item nine
- Item ten
- Item eleven
- Item twelve
- ...

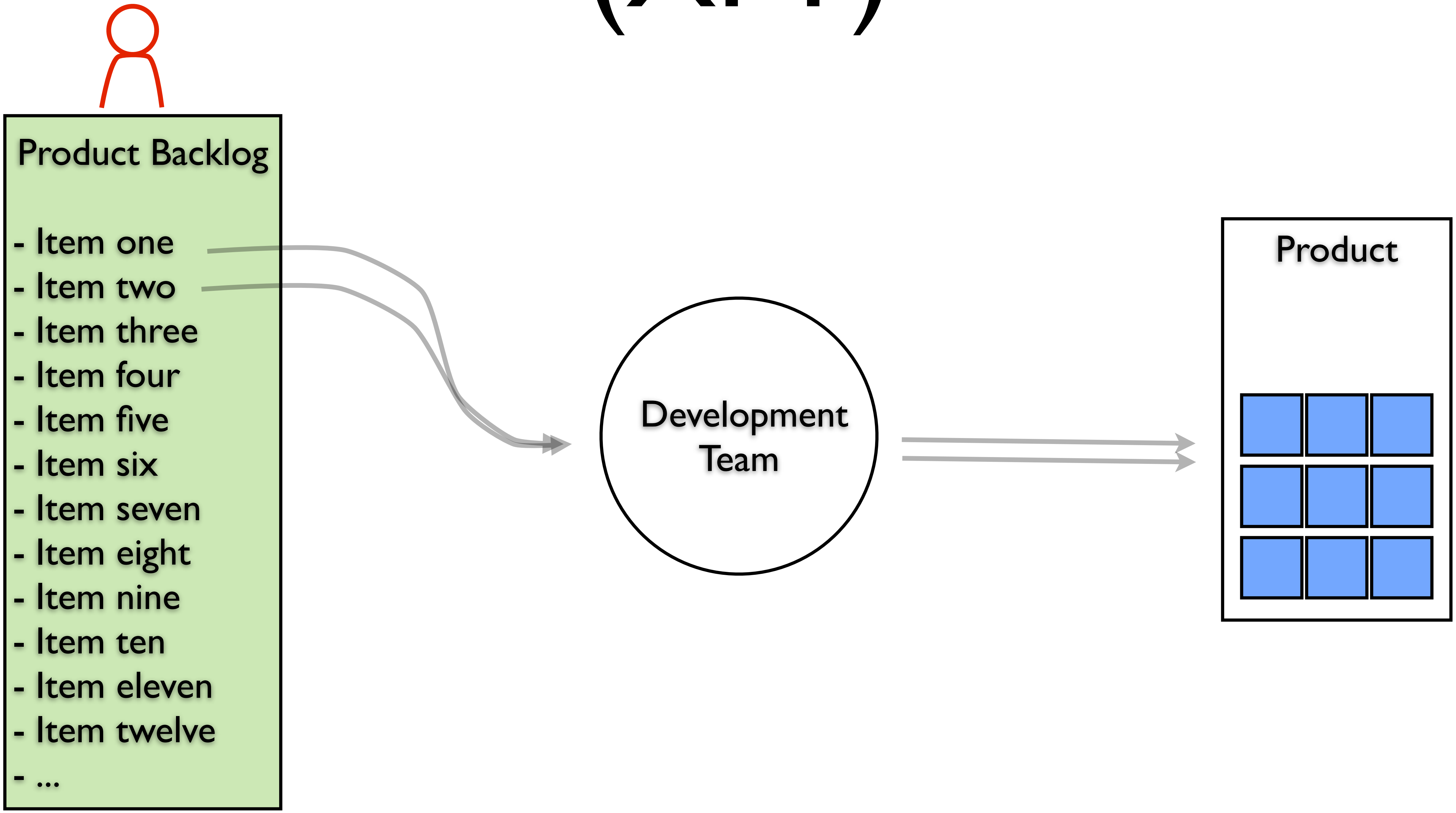
Design

Implementation

Test

Product

Example 1 (XFT)



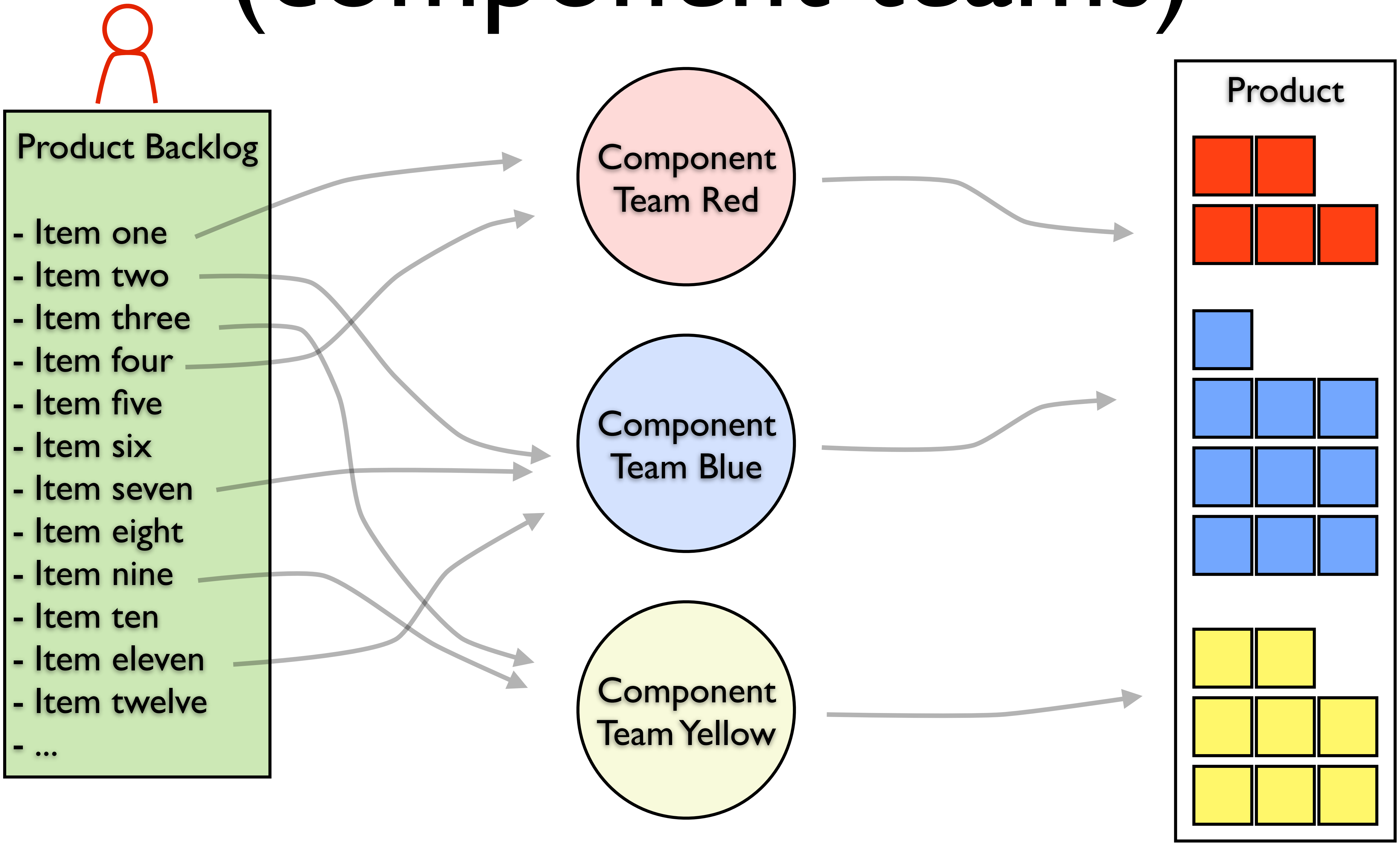
Product Backlog

- Item one
- Item two
- Item three
- Item four
- Item five
- Item six
- Item seven
- Item eight
- Item nine
- Item ten
- Item eleven
- Item twelve
- ...

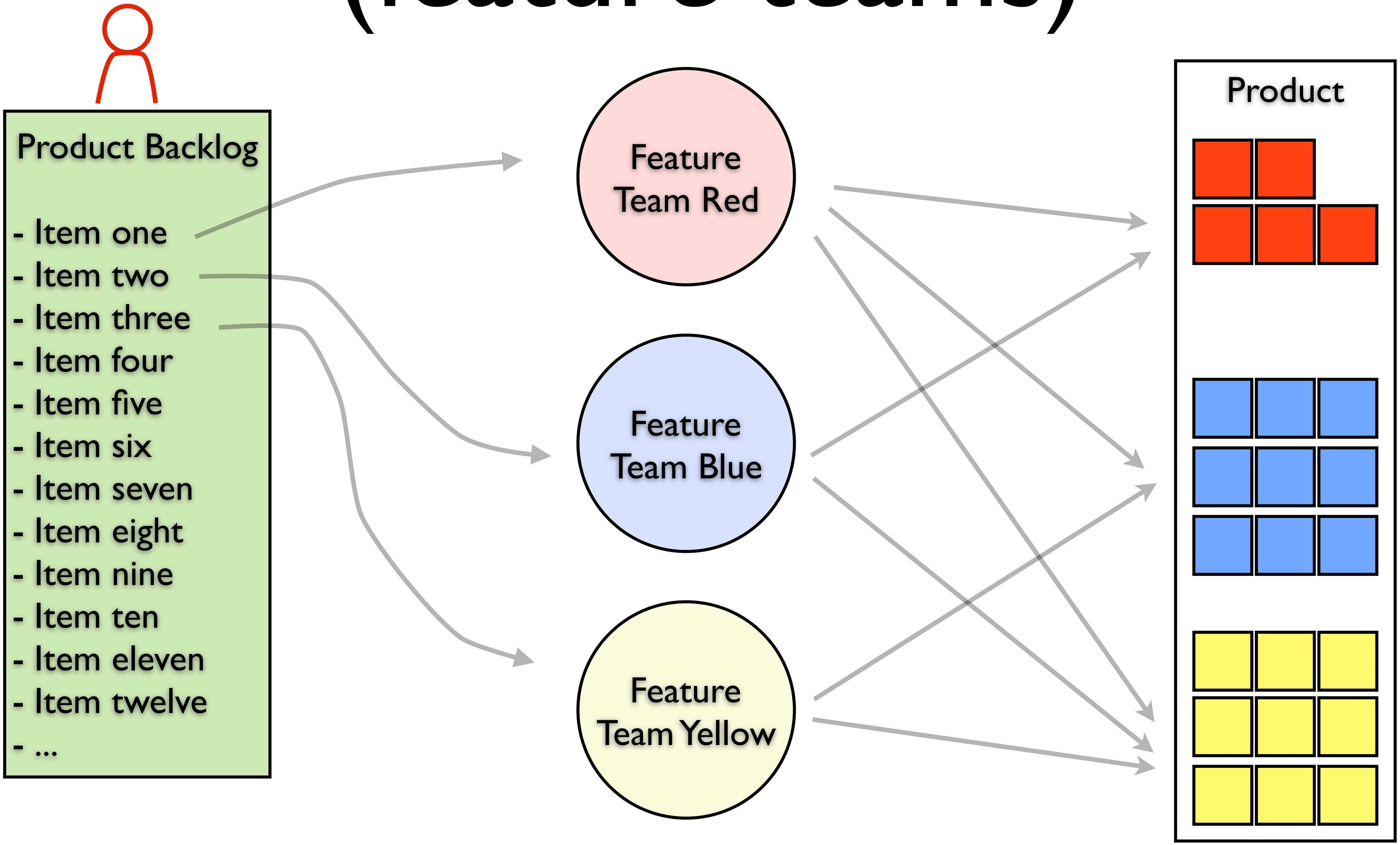
Development Team

Product

Example 2 (component teams)



Example 3 (feature teams)



Product Backlog

- Item one
- Item two
- Item three
- Item four
- Item five
- Item six
- Item seven
- Item eight
- Item nine
- Item ten
- Item eleven
- Item twelve
- ...

Feature Team Red

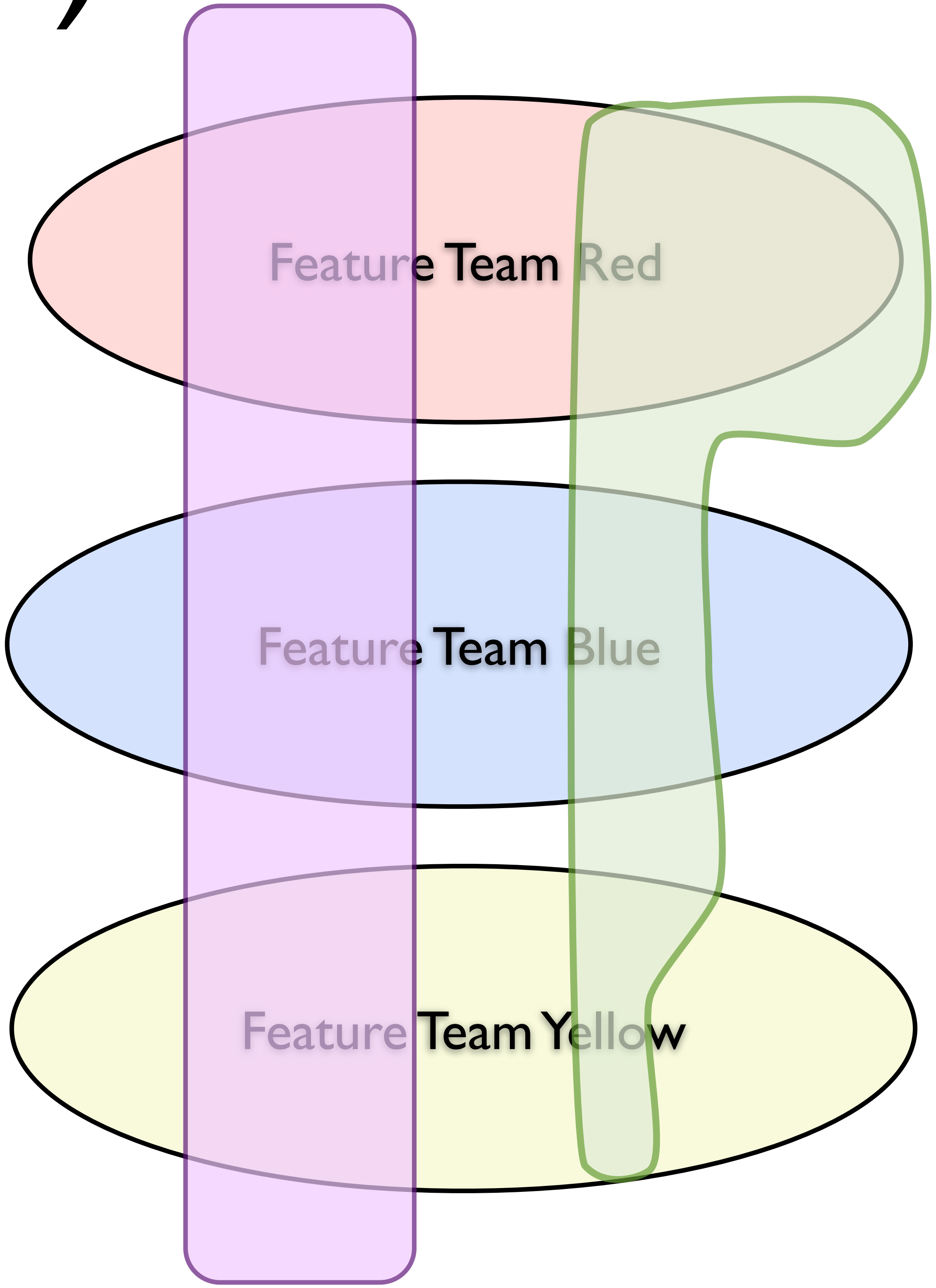
Feature Team Blue

Feature Team Yellow

Product

Community of practice (CoP)

- How do you share knowledge, experiences, solutions over each function (e.g. testing, software development)?
- How do you share information and knowledge for each product component?



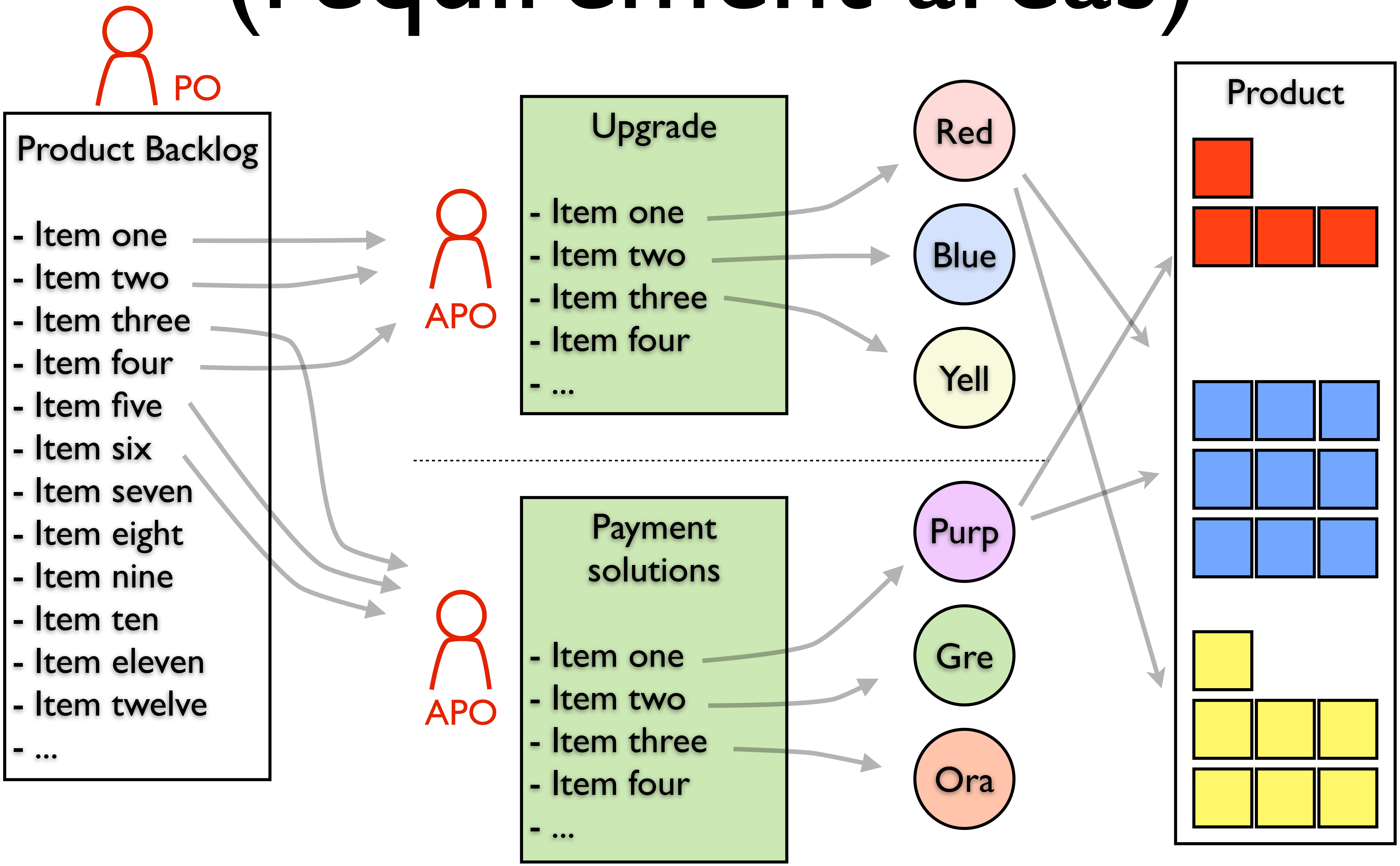
For even larger products

- When we have a large product and many feature teams, we will reach a point where
  - It will be difficult for a single feature team to be able to work in the whole product
  - It will be difficult for a single product owner to work with so many teams in parallel
- Necessary to somehow divide the product backlog

Requirement areas

- Customer centric areas of the product backlog
- Example
  - Upgrade
  - Performance
  - Reliability
  - Ads
  - Payment solution
  - User support system
  - ...

Example 4 (requirement areas)



Case study: Spotify

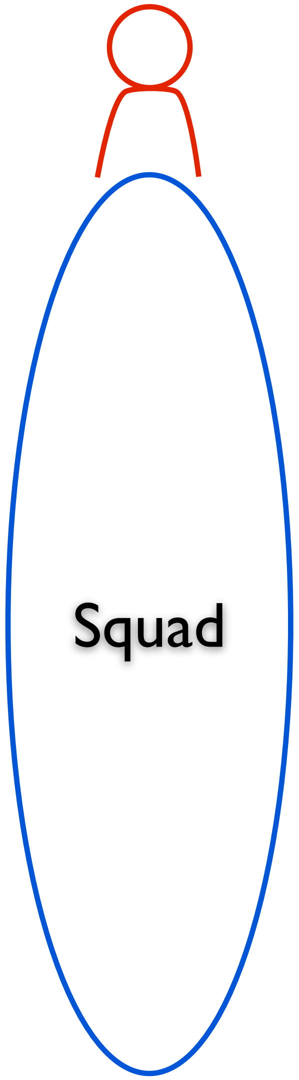
- Kniberg & Ivarsson, 2012, “Scaling Agile @ Spotify with Tribes, Squads, Chapters & Guilds”

Spotify

- Available since 2008
- 6 million paying subscribers
- 24 million active users
- 20 million songs
- Available in 28 countries

- 30 agile teams across 3 cities (2012)

Development



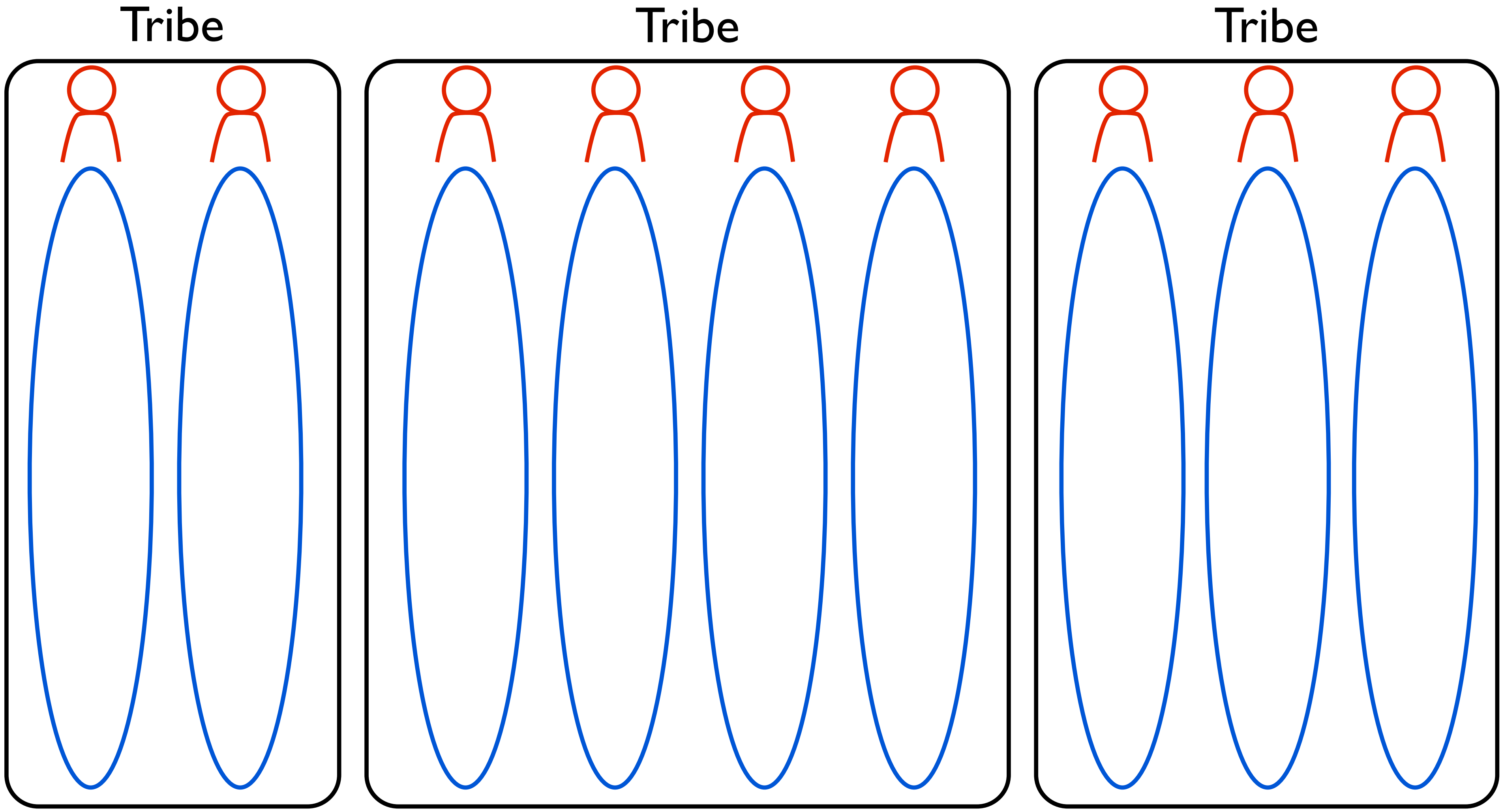
- Development is done in Squads
- A Squad is a self-organizing, cross-functional, team, that use Scrum, Kanban or other Agile methods
- A Squad works with one Product Owner (PO)
- Every Squad has a long-term mission and responsibility (e.g. build and improve mobile app experience, provide payment solutions)
- Encouraged to use MVP (minimum viable product)
- Squads become experts in their area!

Squads, Roadmap and Coaching

- Each Squad is encouraged to spend 10% of development time on "hack days"
- There is no Squad leader, true self-organization
- Product owner only prioritizes the backlog, does not tell Squads what to do
- Products owners align their backlogs and present Spotify roadmap
- Agile coaches available to support Squads with Sprint Planning, Retrospectives
- Quarterly surveys with each Squad

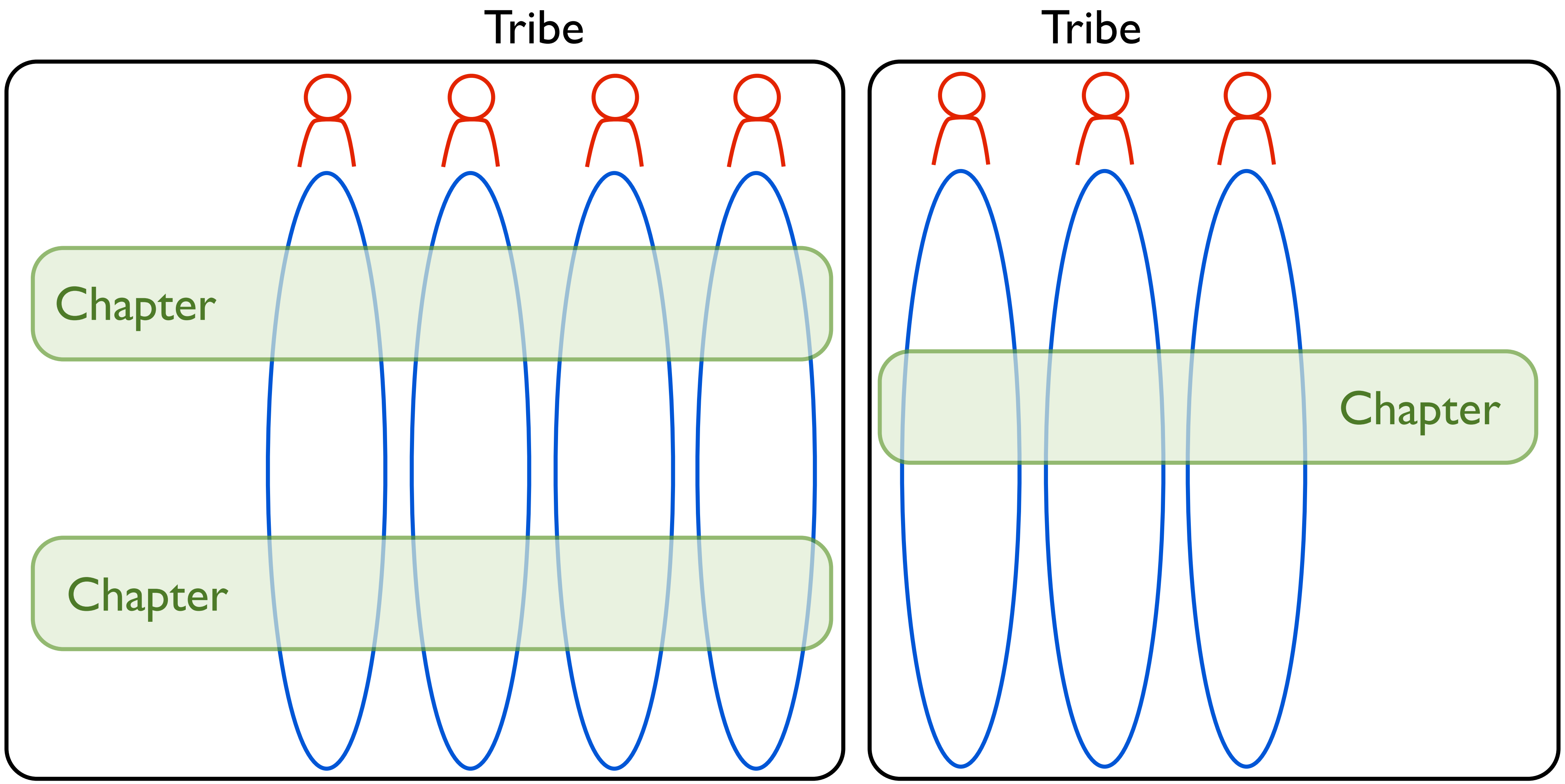
Tribes

- Collection of Squads that work in similar areas
- Sit together, has a Tribe leader, has Gatherings



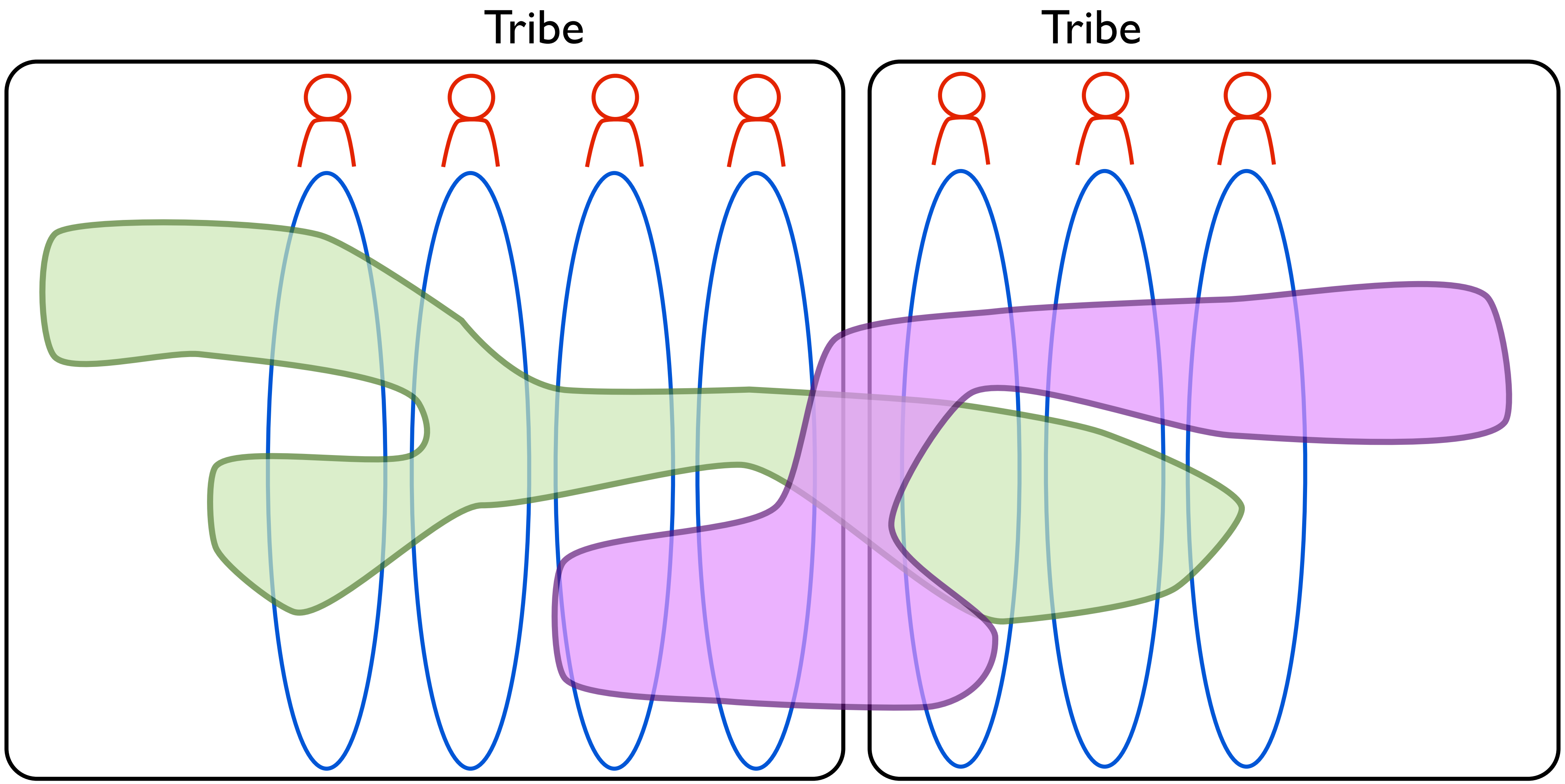
Chapters

- Chapter is your family within a Tribe of the people with the same competences as you
- Meets regularly to discuss the area and challenges
- Chapter leader is the line manager, responsible for people development, salary etc.



Guilds

- Similar to a community of practice, that spans the whole company
- E.g. web technology guild, tester guild, agile coaches guild



Architecture

- 100 distinct systems that can be deployed separately
- Squads are feature teams and can edit any part of the system
- System Owners
  - One person or one pair that keeps track of quality, performance, technical debt, release process, scalability for a system
  - A system maps approximately to the tribe areas
- Chief Architect
  - Handles high level architecture issues across multiple systems

Going forward

- Case study was written in 2012
- Spotify had grown from 30 to 250 people
- So far the model had scaled well!

Q&A