

The background of the slide is a grayscale aerial photograph of the University of Porto's campus, showing a dense cluster of modern and traditional university buildings.

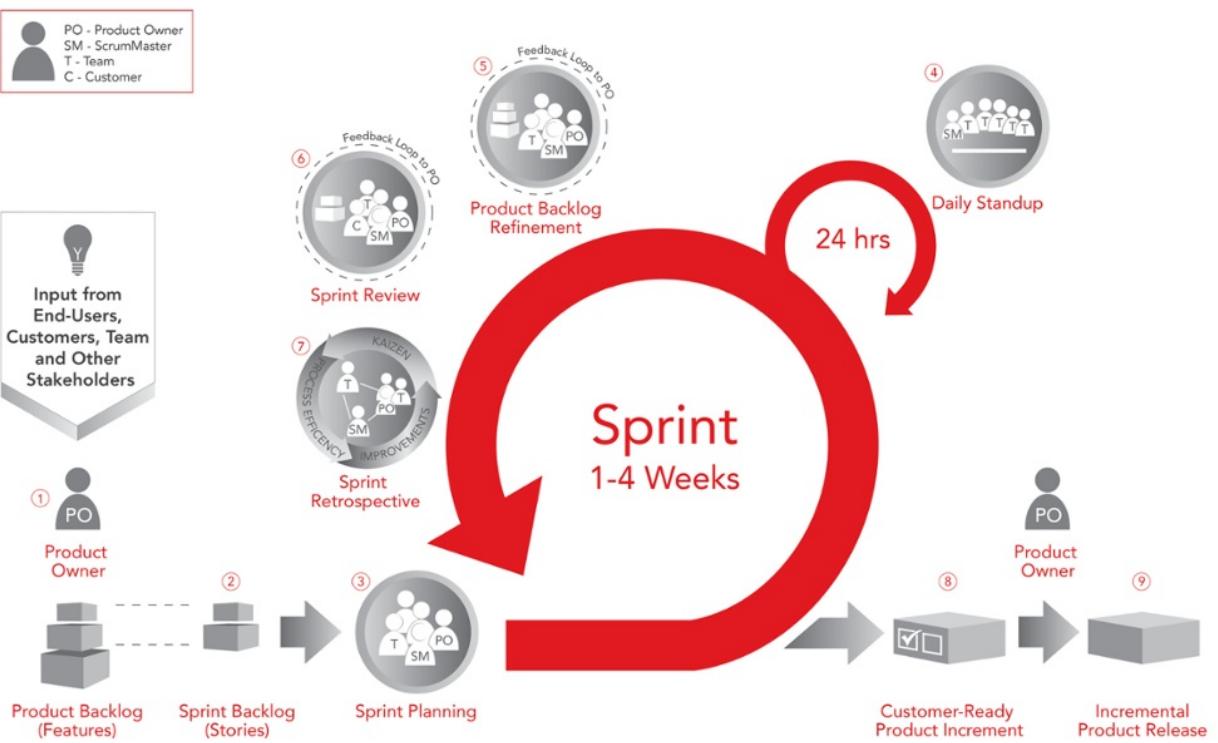
M.EIC, 2022-23

# Large Scale Software Development

Filipe Correia, Daniel Pinho, João Pedro Dias



# Scrum



# Scrum Patterns



# ¶18 Mitosis



# ¶18 Mitosis

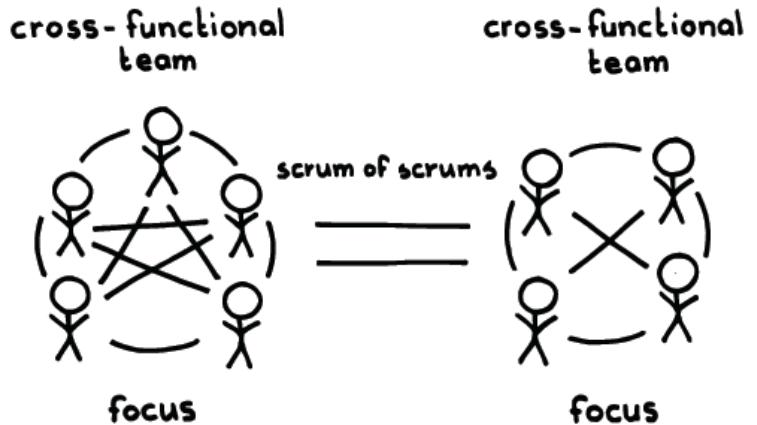
One should grow a *Scrum Team* in an incremental, piecemeal fashion, but eventually the team just becomes too large to remain efficient. *Small Teams* are the most efficient, but sometimes you need to grow.

*Therefore:*

Differentiate a single large *Development Team* into two small teams after it gradually grows to the point of inefficiency — about seven people in the old team.

# ¶18 Mitosis

Carry over the **experience, domain and product knowledge, and culture** of the original team into each new team.



# ¶34 Scrum of Scrums



## ¶34 Scrum of Scrums

When multiple teams work independently of each other they tend to focus myopically on their own concerns and lose sight of any common goals.

*Therefore:*

Give the **right and the responsibility to collaborate on delivering common goals** identified by the Product Owner to the **Development Teams** themselves. Permit the teams to figure out the best way to coordinate their efforts.

## ¶34 Scrum of Scrums

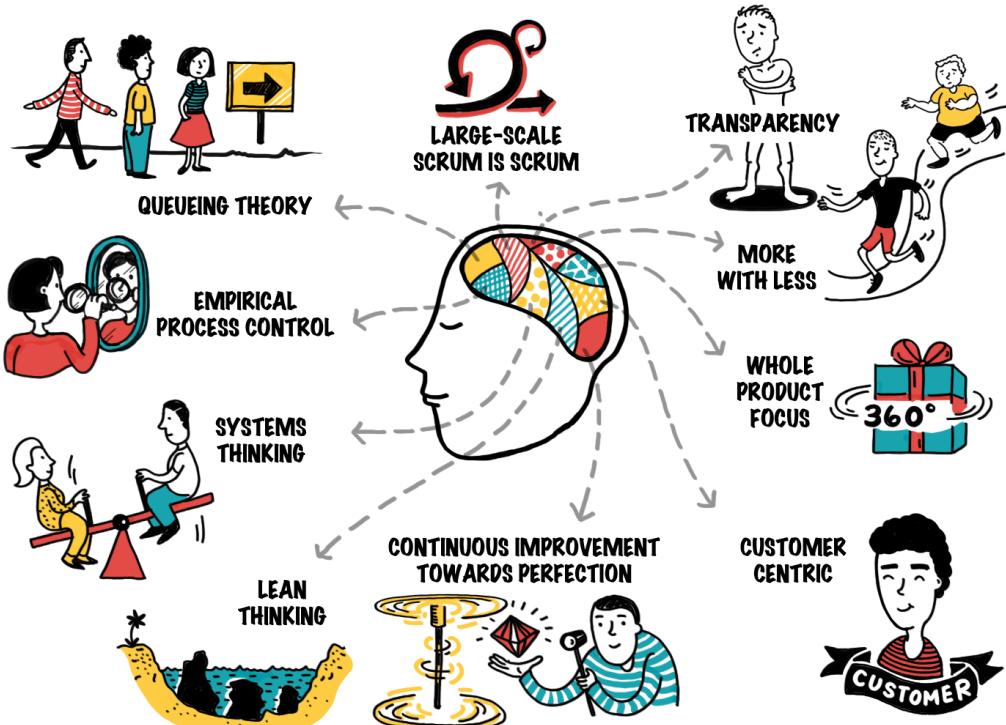
Scaling, when it does occur, is always situational, so the exact forms of collaboration will be determined by the Development Teams, but typical tactics include:

- Sprinting together—at the same cadence, at the same time [...]
- Maintaining a common ***Definition of Done***
- Common ***Sprint Planning, Sprint Reviews*** and other mandatory Scrum events
- Holding ***Backlog Refinement*** events in common
- Creating semi-formal optimizing networks [...] utilizing common competencies such as architecture across the teams to proactively handle issues that are known in advance
- Establish a regular ***Scrum of Scrums*** event, perhaps daily, after the teams' *Daily Scrum* events, to resolve emergent dependencies and issues, and to get things to Done (see *Definition of Done*).

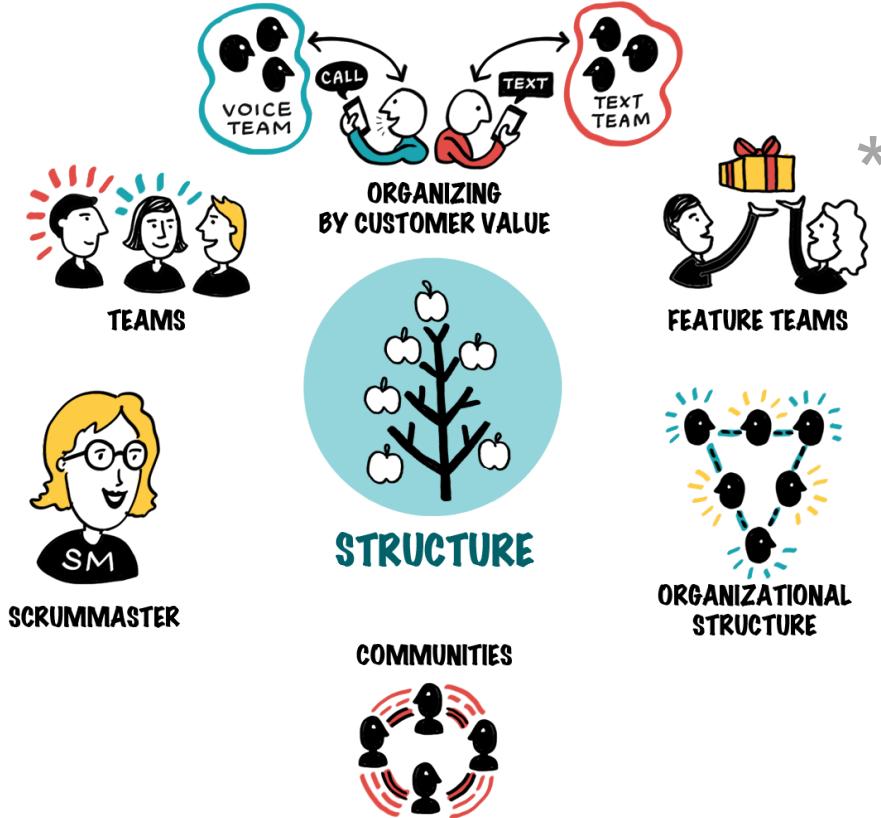
# LeSS – Large Scale Scrum

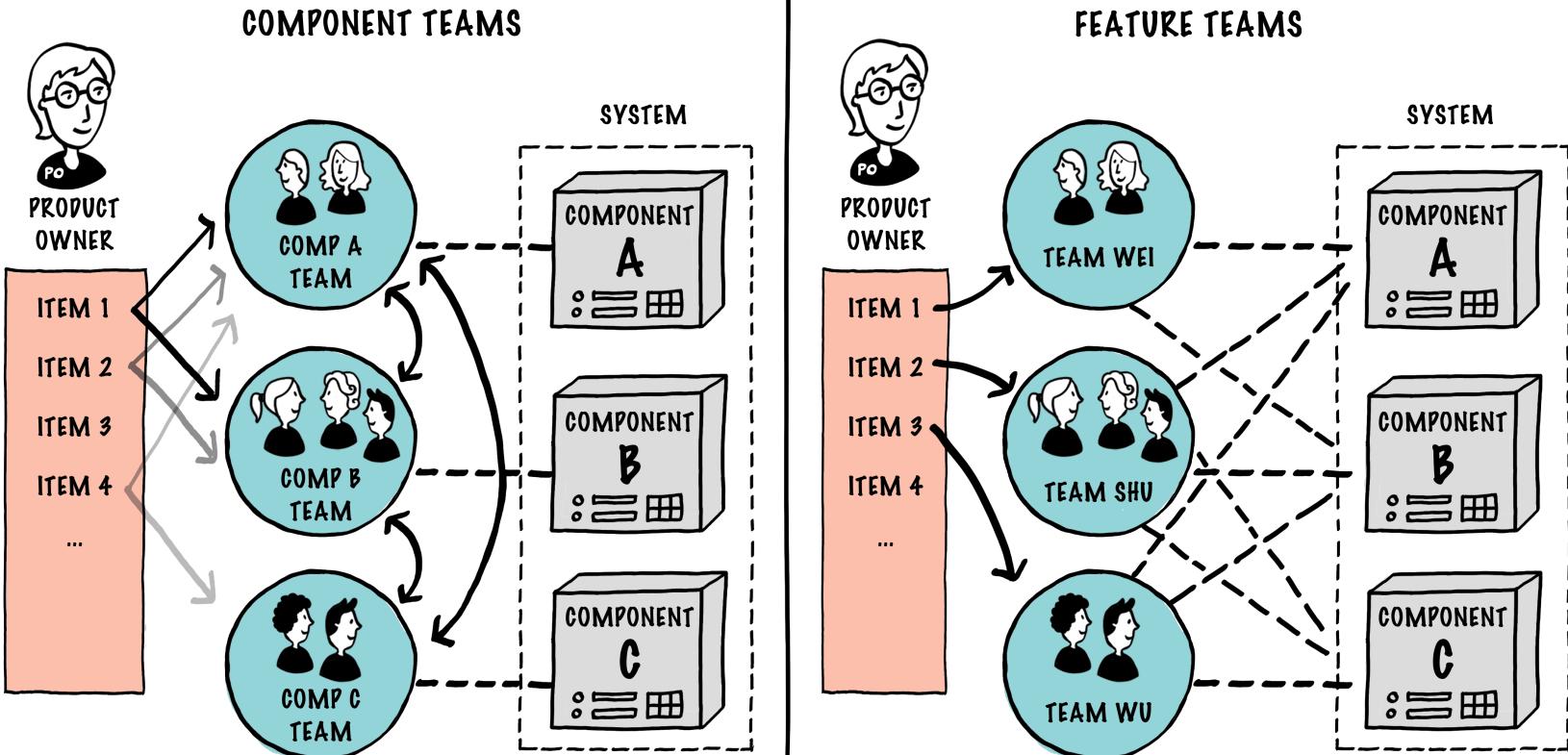
<https://less.works/less/framework>

# Principles

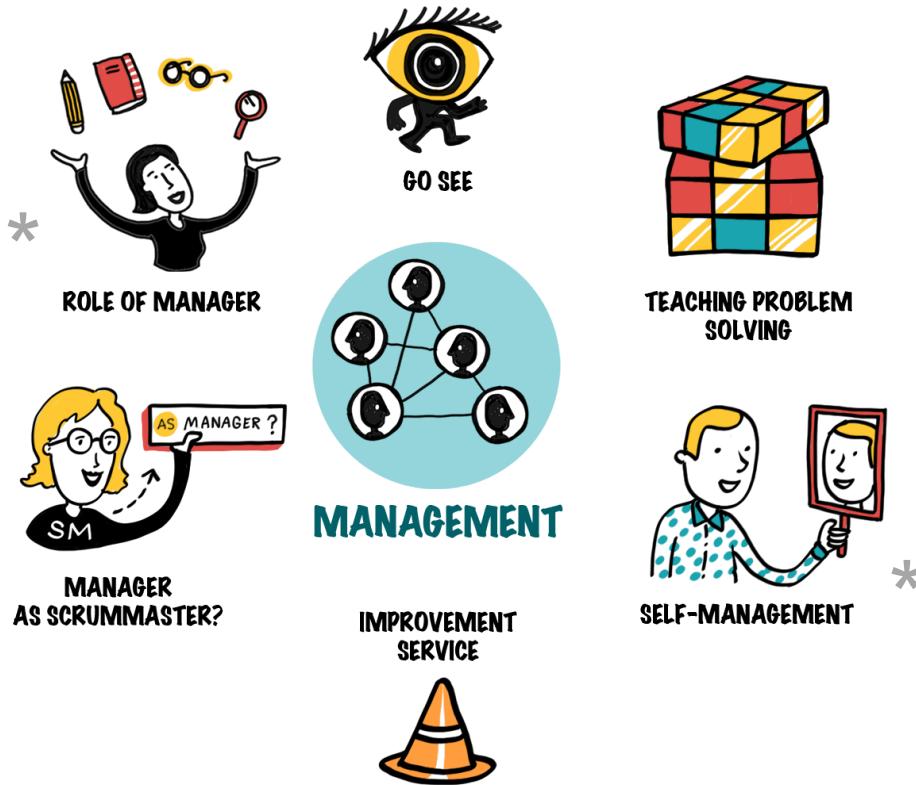


# Structure

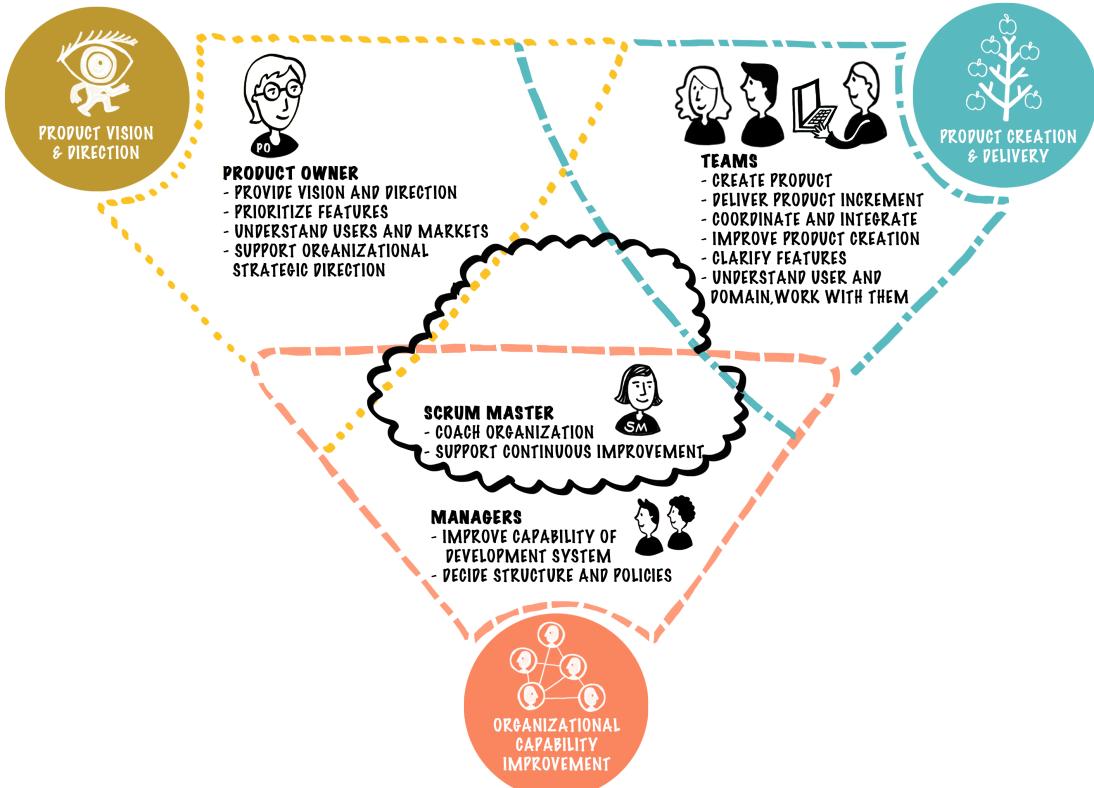




# Management

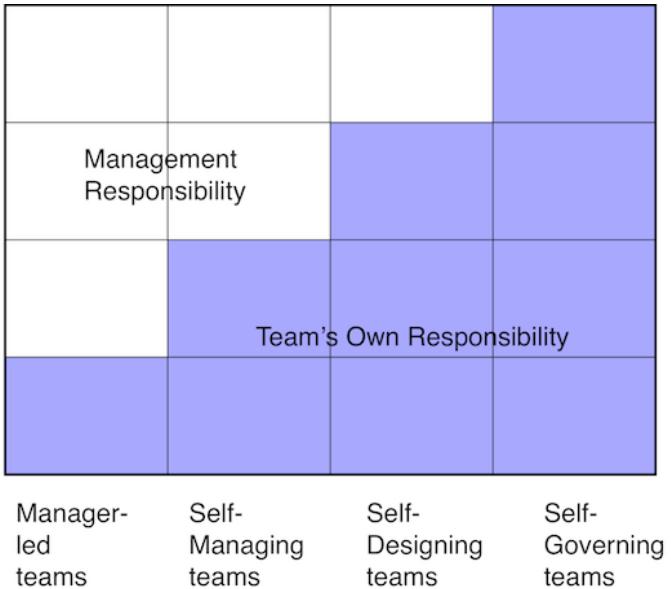


# Role of Manager

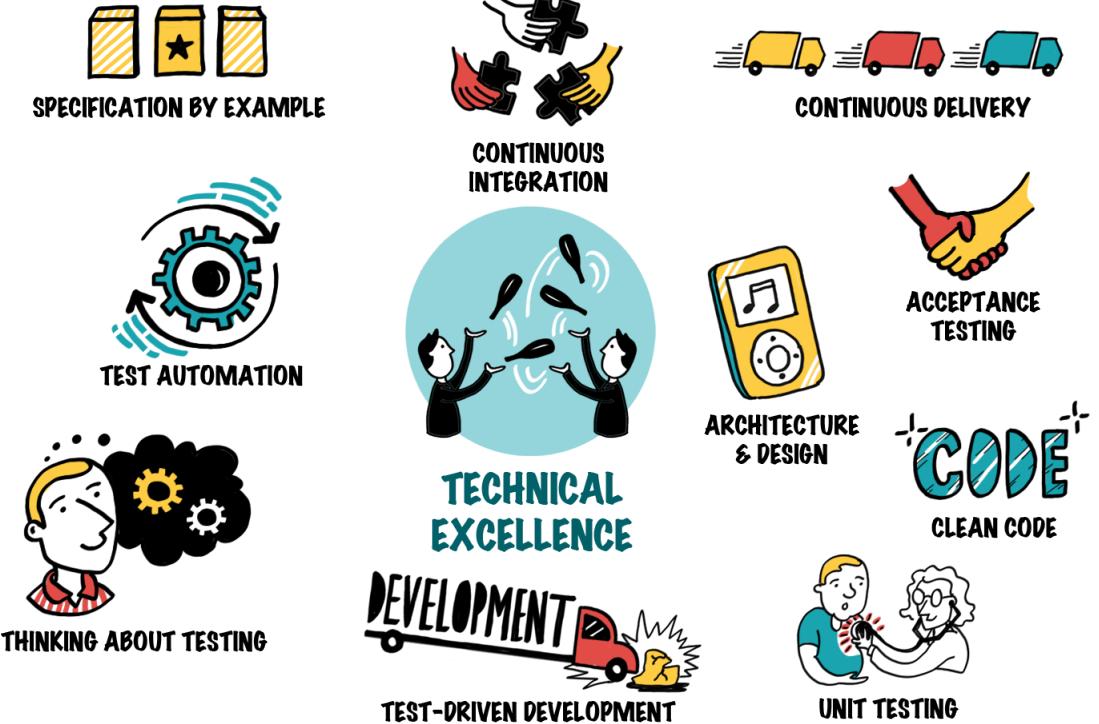


# Self-management

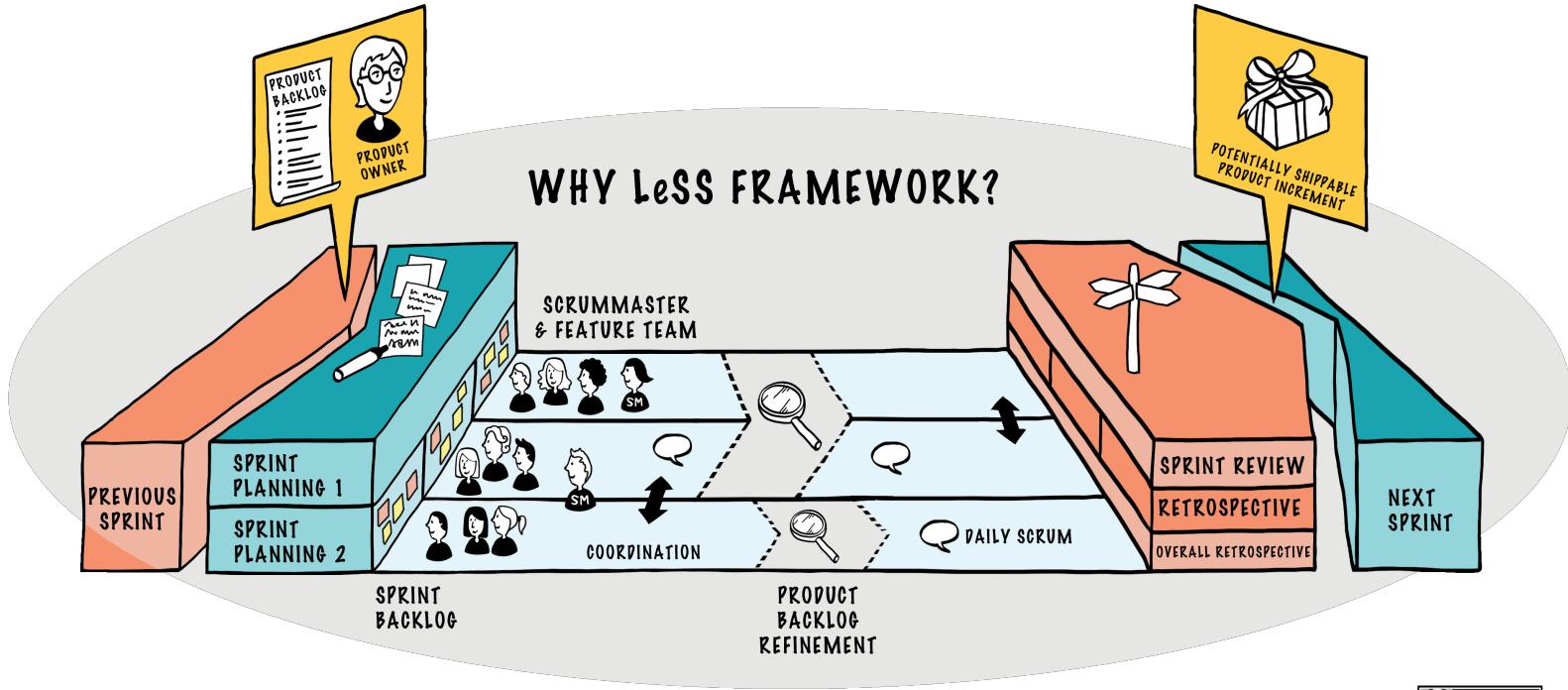
- Setting overall direction
- Designing the team and its organizational context
- Monitoring and managing work process and progress
- Executing the team task



# Technical Excellence

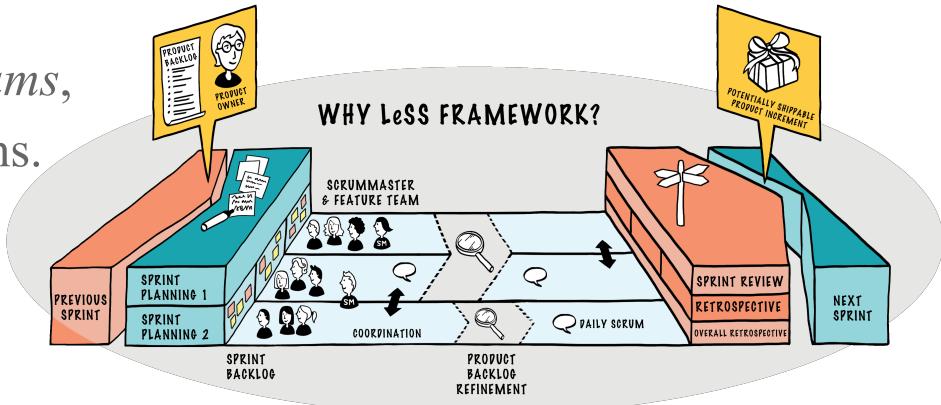


# LeSS Framework



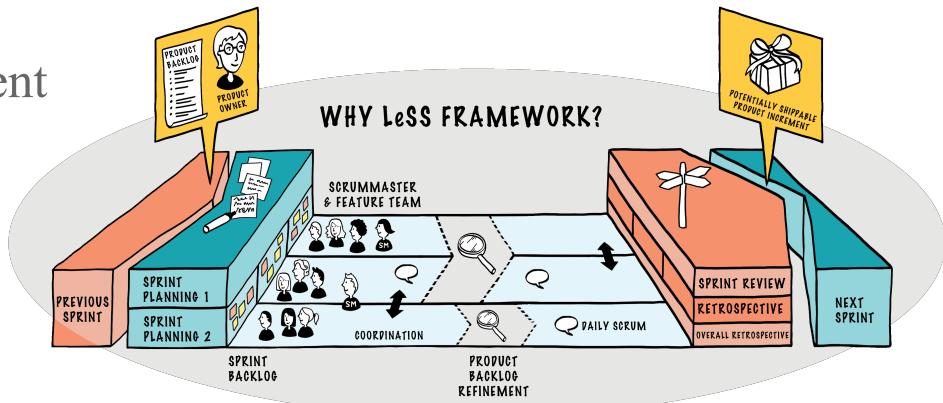
# LeSS Framework – what is the same as One-Team Scrum?

- A single *Product Backlog* (it's for a product, not a team),
- A single *Definition of Done* for all teams,
- A single *Product Increment* at the end of the Sprint,
- A single *Product Owner*,
- Many *Cross-functional teams*,
- A single *Sprint* for all teams.

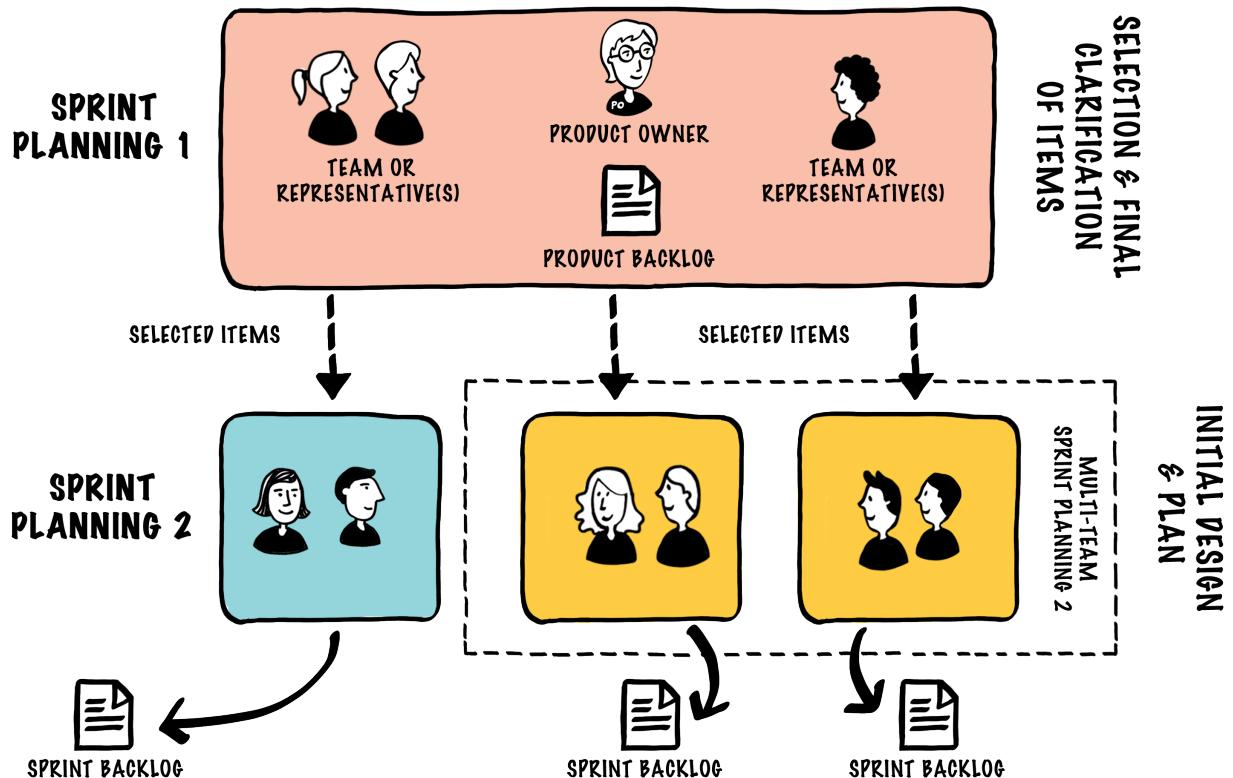


# LeSS Framework – what differs from One-Team Scrum?

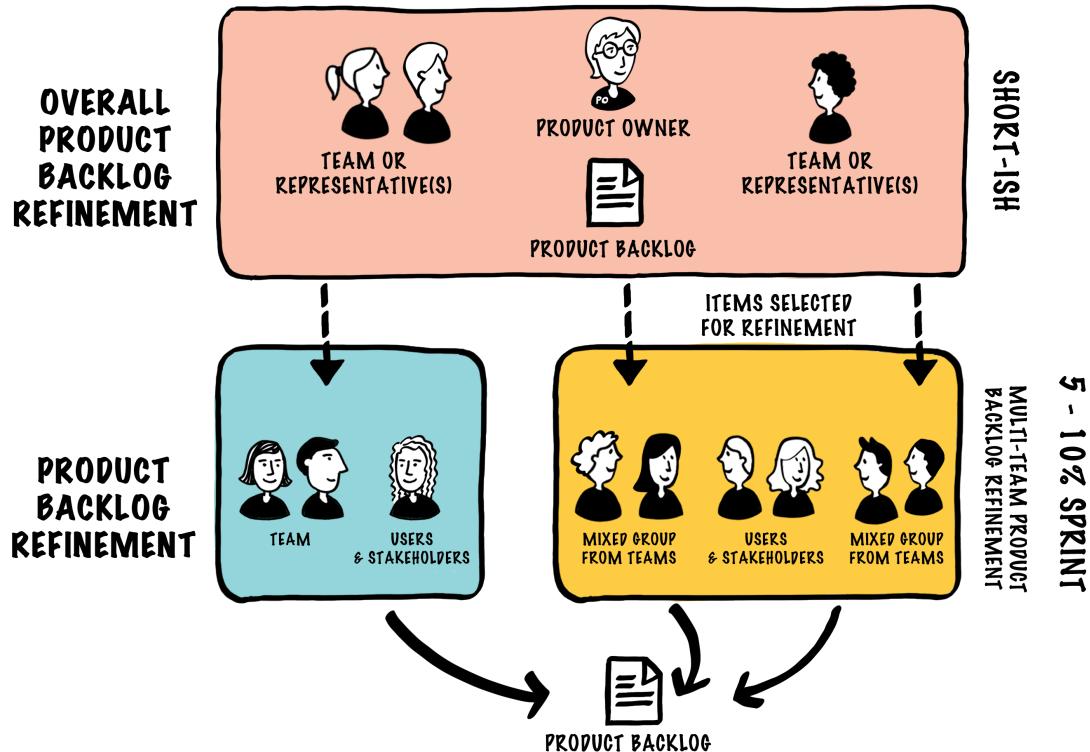
- Sprint Planning 1
- Sprint Planning 2
- Coordination
- Daily Scrum
- Product Backlog Refinement
- Sprint Review
- Overall Retrospective



# LESS SPRINT PLANNING

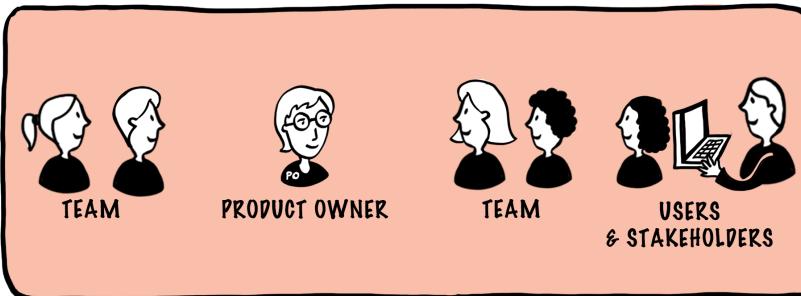


# LeSS PRODUCT BACKLOG REFINEMENT

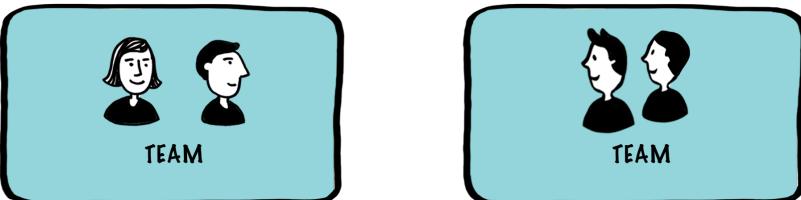


# LESS SPRINT REVIEW & RETROSPECTIVE

## SPRINT REVIEW



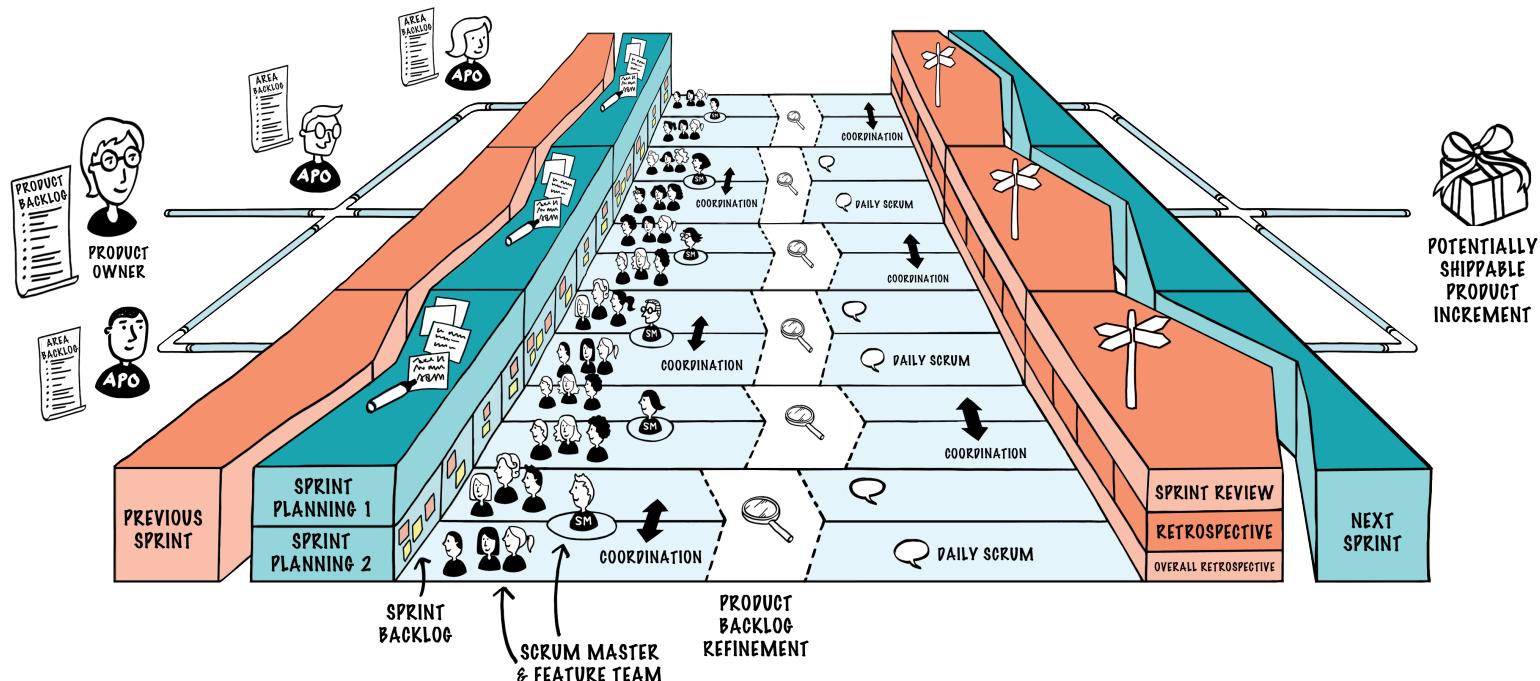
## TEAM RETROSPECTIVE



## OVERALL RETROSPECTIVE



# LeSS Huge



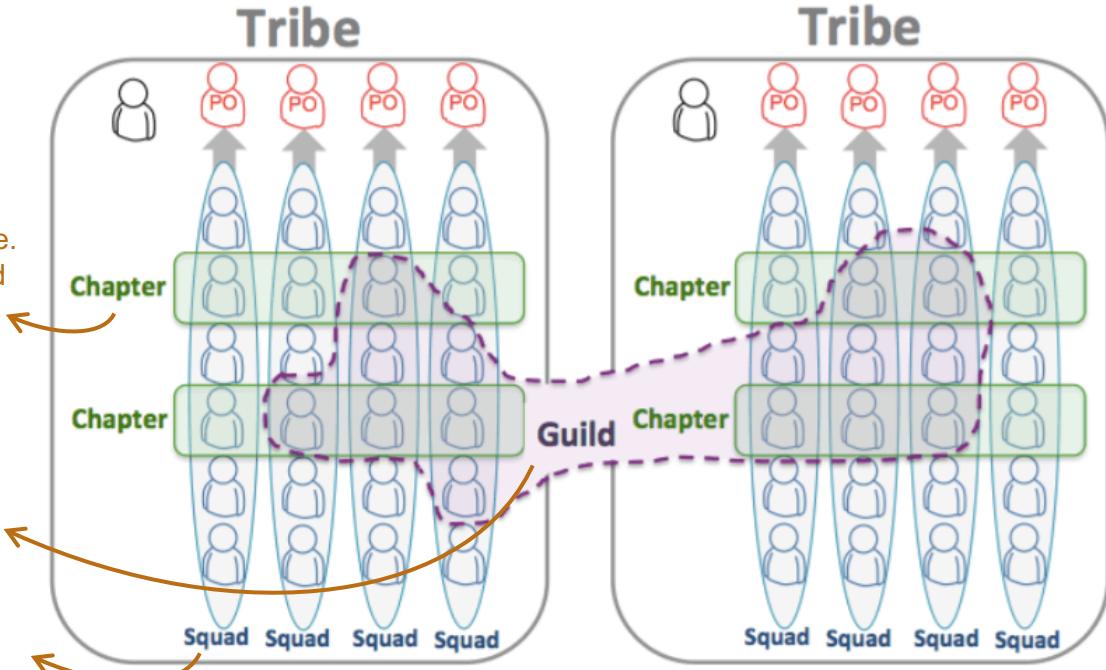
# Scaling Agile @ Spotify

# Scaling Agile @ Spotify

Skills community. Local to a tribe.  
Leader focuses on coaching and  
mentoring. E.g., testing, agile  
coaching, web development

Community of interests. Can  
be around any topic. Has the  
goal to share knowledge,  
tools, or practices.

Equivalent to a Scrum Team.  
Cross-functional, self-organizing  
and autonomous



Recommended reading: [Scaling Agile @ Spotify](#)

Recommended watching: [Spotify Engineering Culture \(13min\)](#)

Henrik Kniberg & Anders Ivarsson – Oct 2012

# FEUP.DEI



DEI - DEPARTAMENTO DE ENGENHARIA INFORMÁTICA  
[dei.fe.up.pt](http://dei.fe.up.pt) | [secdei@fe.up.pt](mailto:secdei@fe.up.pt) | +351 225 082 134  
Director **João MP Cardoso** | Full Professor  
[jmpc@fe.up.pt](mailto:jmpc@fe.up.pt) | +351 220 413 284

FEUP - FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO  
Rua Dr. Roberto Frias | 4200-465 Porto | PORTUGAL  
[www.fe.up.pt](http://www.fe.up.pt) | [feup@fe.up.pt](mailto:feup@fe.up.pt) | +351 225 081 400  
 @DEI\_FEUP