

# Large Scale Scrum(LeSS): More with LeSS

By Ram Srinivasan, CST

<https://linkedin.com/in/ramvasan>

<http://twitter.com/ramvasan>

<http://InnovAgility.com> (blog)



## The (Shameless) About Me Slide...

- Certified Scrum Trainer and practitioner
- Contributor to the upcoming 1-day LeSS introductory course (to be published by Scrum Alliance and The LeSS Company)
- Expert reviewer –PMI Agile practice Guide
- Passionate about complexity, neuroscience of coaching, systems coaching
- Speaker, reviewer, and co-organizer of many Agile conferences
  - Slides at <http://slideshare.net/ramvasan>

# Special Discount for ANE

**Agile New England Special:** Register for the ***Weekend Certified Scrum Master (CSM)*** workshop in Boston on August 5<sup>th</sup> and 6<sup>th</sup>, 2017 and get the early bird special for \$795 (first 5 tickets only) –

Register here - <http://bit.ly/2017-08-CSM>

Take whatever I say with a pinch of salt....



Craig Larman

Alistair Cockburn

# The Story we Believe...

We are a BIG organization and/or our product is BIG, so we need to use a scaling framework (preferably, the bigger the better)



# Scaling Agile



# Scaling Agile - YAGNI

Large Scale... Don't  
Multi Site... Don't  
Offshore... Don't



Bas Vodde and Craig Larman

“Scaling Agile  
methods is the last  
thing you should  
do”



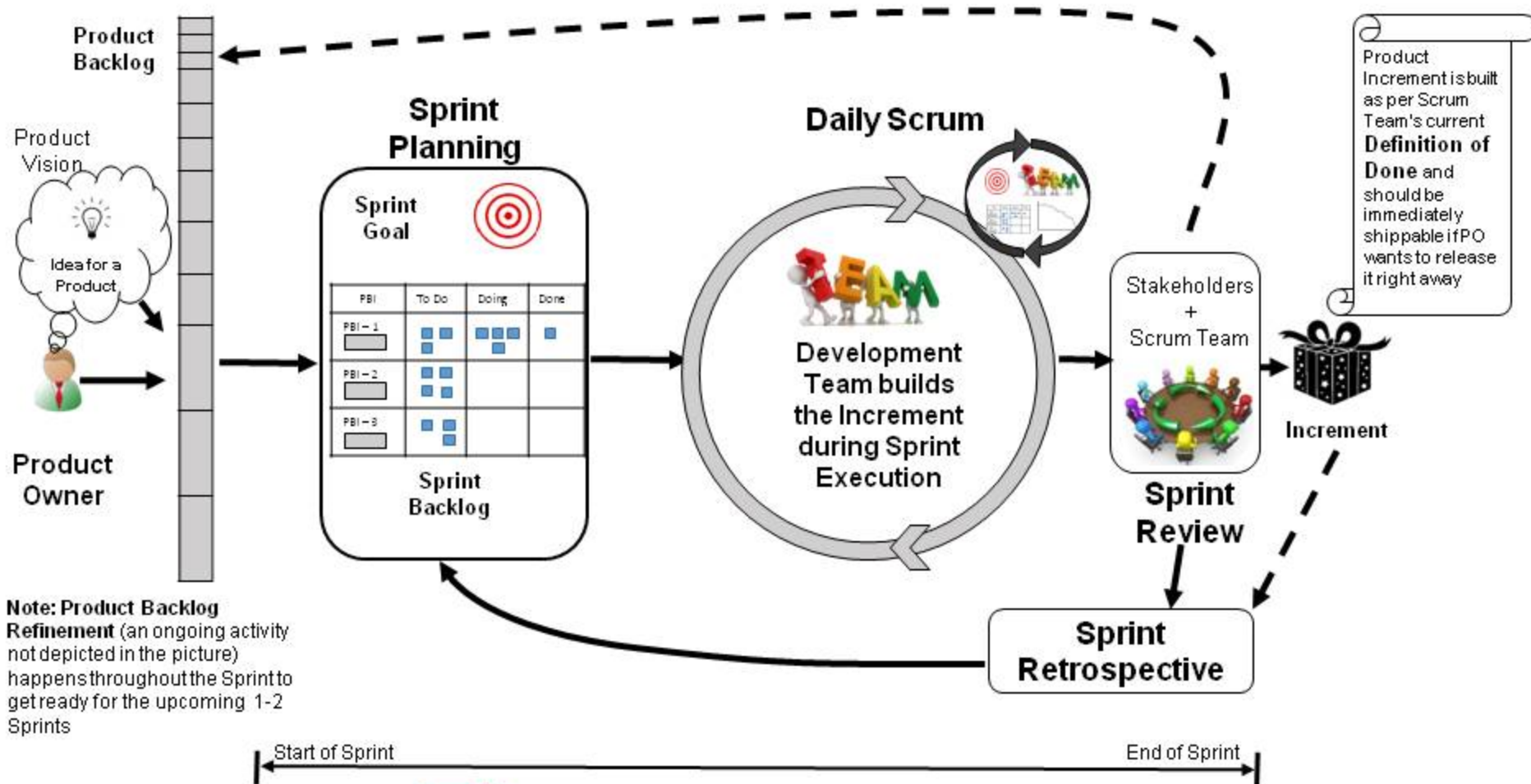
Martin Fowler

# First... The Basics: Check your Scrum-fu





# The Scrum Framework



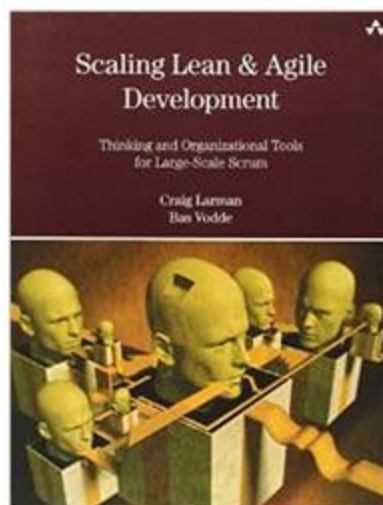
Scrum Master is responsible for “installing” Scrum and removing impediments



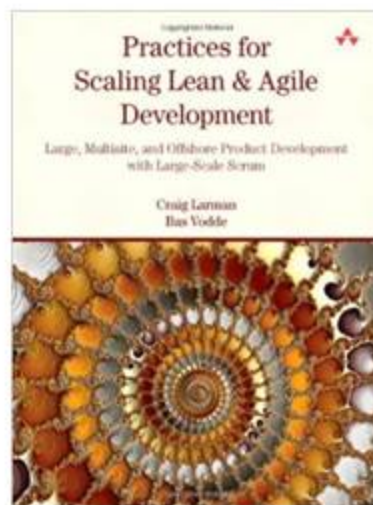
# The History of Large Scale Scrum

“Since 2005, we've worked with clients to apply the LeSS (Large-Scale Scrum) framework for scaling Scrum, lean and agile development to big product groups. We share that experience and knowledge through LeSS so that you too can succeed when scaling.”

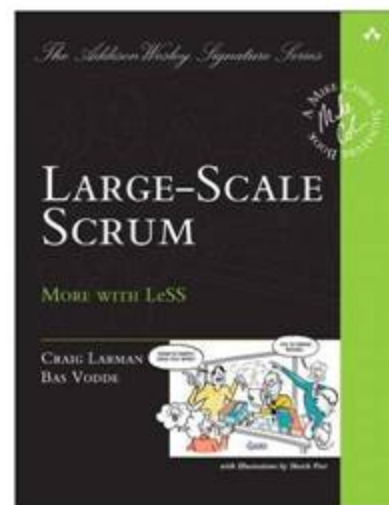
- Craig Larman and Bas Vodde



Book 1  
2008



Book 2  
2010



Book 3  
2016

# Prescriptiveness of Frameworks

LeSS: some rules guided by principles, many guides and experiments, no false dichotomies



**None – just soft principles**

Hard for novice organizations to start, easy to “fake change”

**Many roles, rules, processes, techniques, ...**

Not contextual enough, inhibits maximizing learning through empiricism & experiments

# LeSS is About Refactoring the Organization

LeSS is a (meta-process) framework for organizational design.

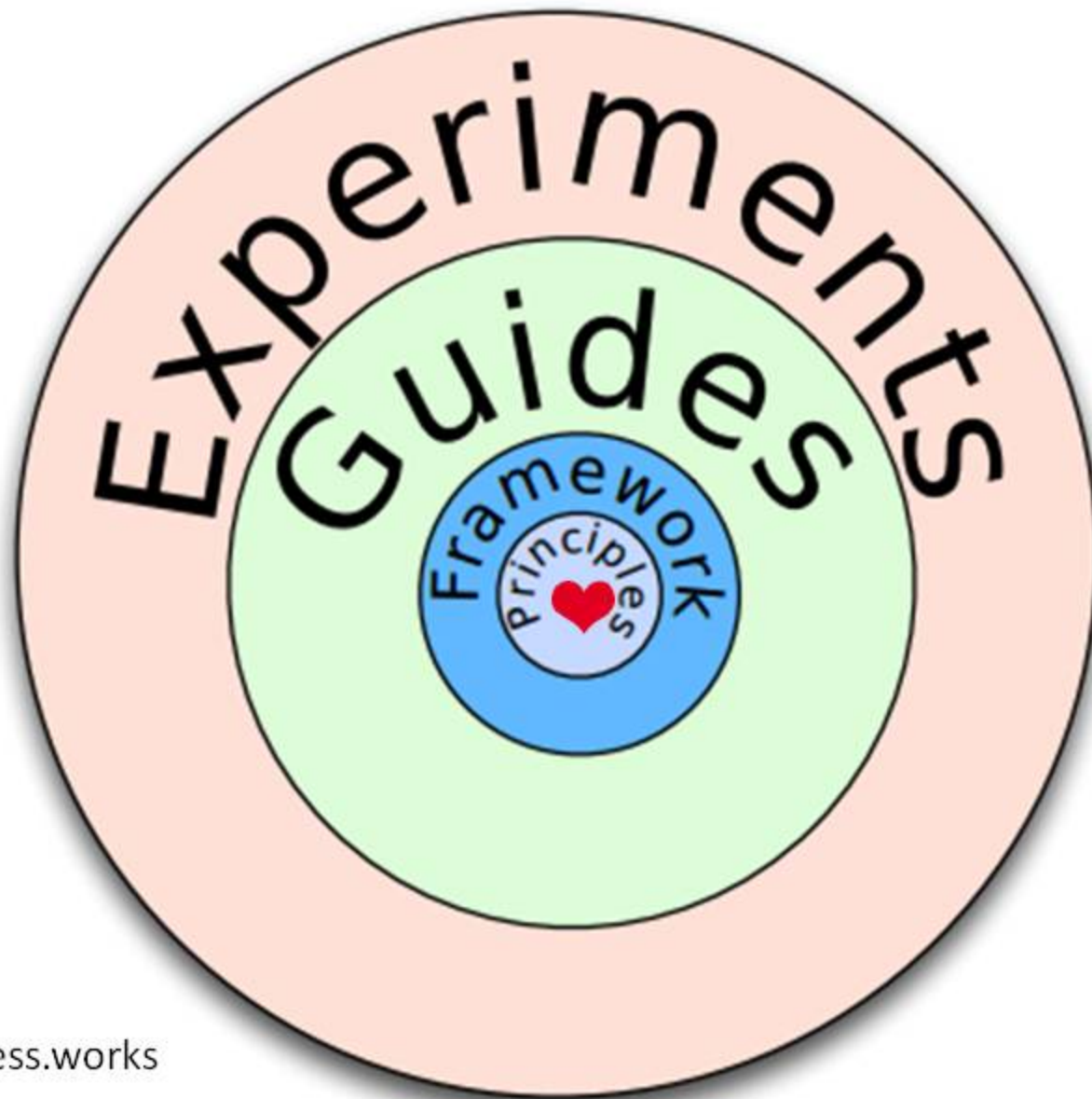
LeSS, like any other debugging tool, will **expose the defects and wastes** existing in the organization (but in a **painful way**)



LeSS simplifies scaling up Scrum by descending (unwanted) complexity

- Descending roles and organizational hierarchy
- Descending organizational structures, policies, etc.
- Descending architectural complexity

# The LeSS Complete Picture

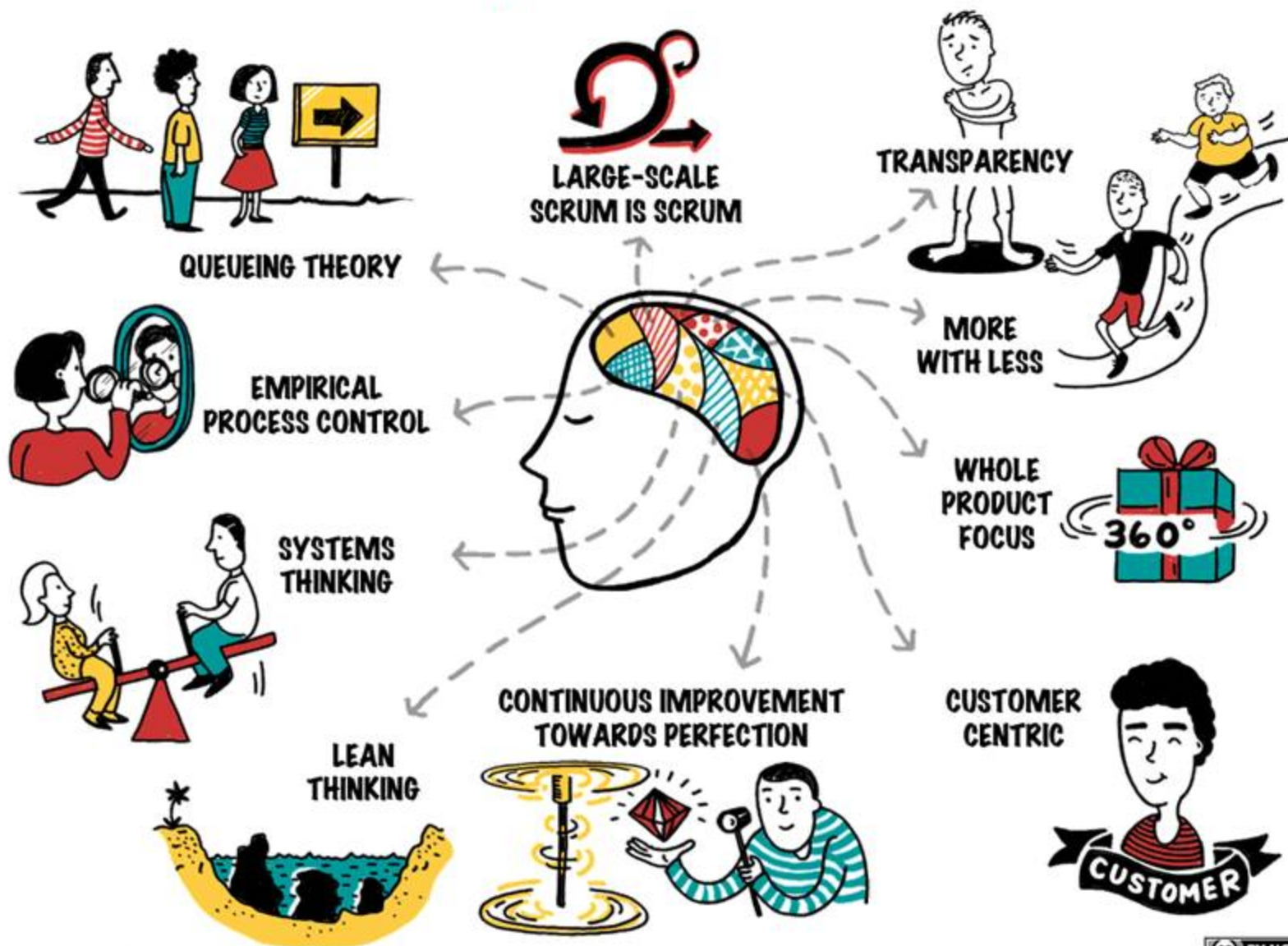


Source: <http://less.works>

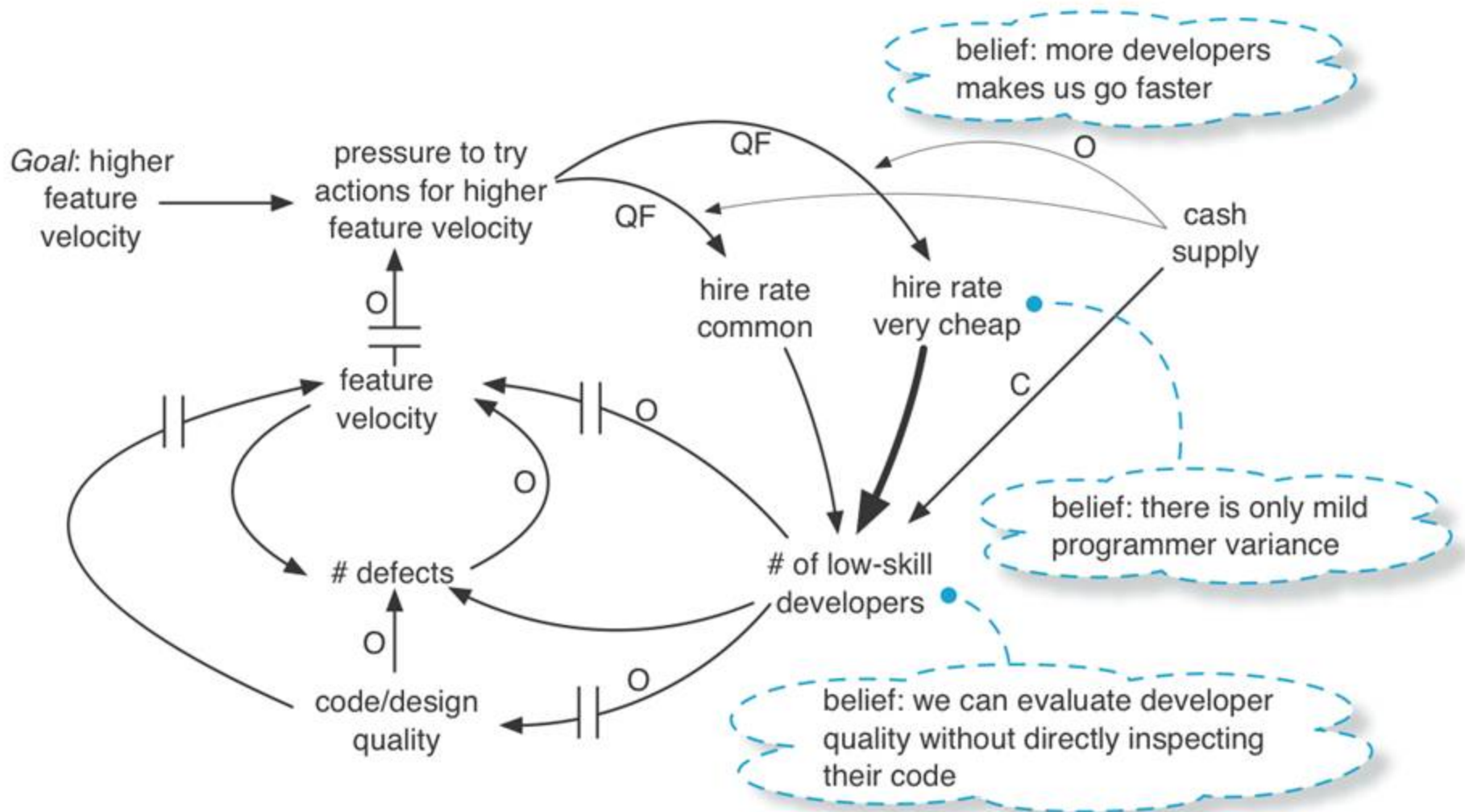
## **LeSS Complete Picture.**

**LeSS has been used with a dozen people, hundreds of people, and thousands of people.**

# LeSS Principles – Heart of LeSS



# Systems Thinking – An example with Causal Loop Diagrams





# LeSS Principles: Queueing Theory

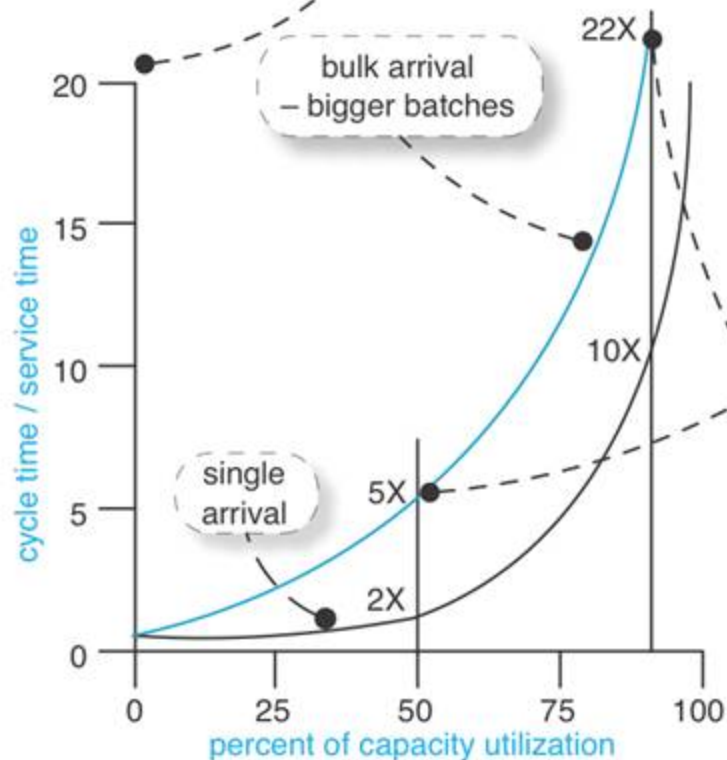


# LeSS Principles: Queueing Theory

ratio of cycle time (queue time + service time) divided by service time. for example:

- service time: 2 weeks      - queue time: 2 weeks  
- cycle time = 4              - **ratio = 2**

- service time: 2 weeks      - queue time: 8 weeks  
- cycle time = 10             - **ratio = 5**



**Analogy to Traditional Product Development with Big Batches**

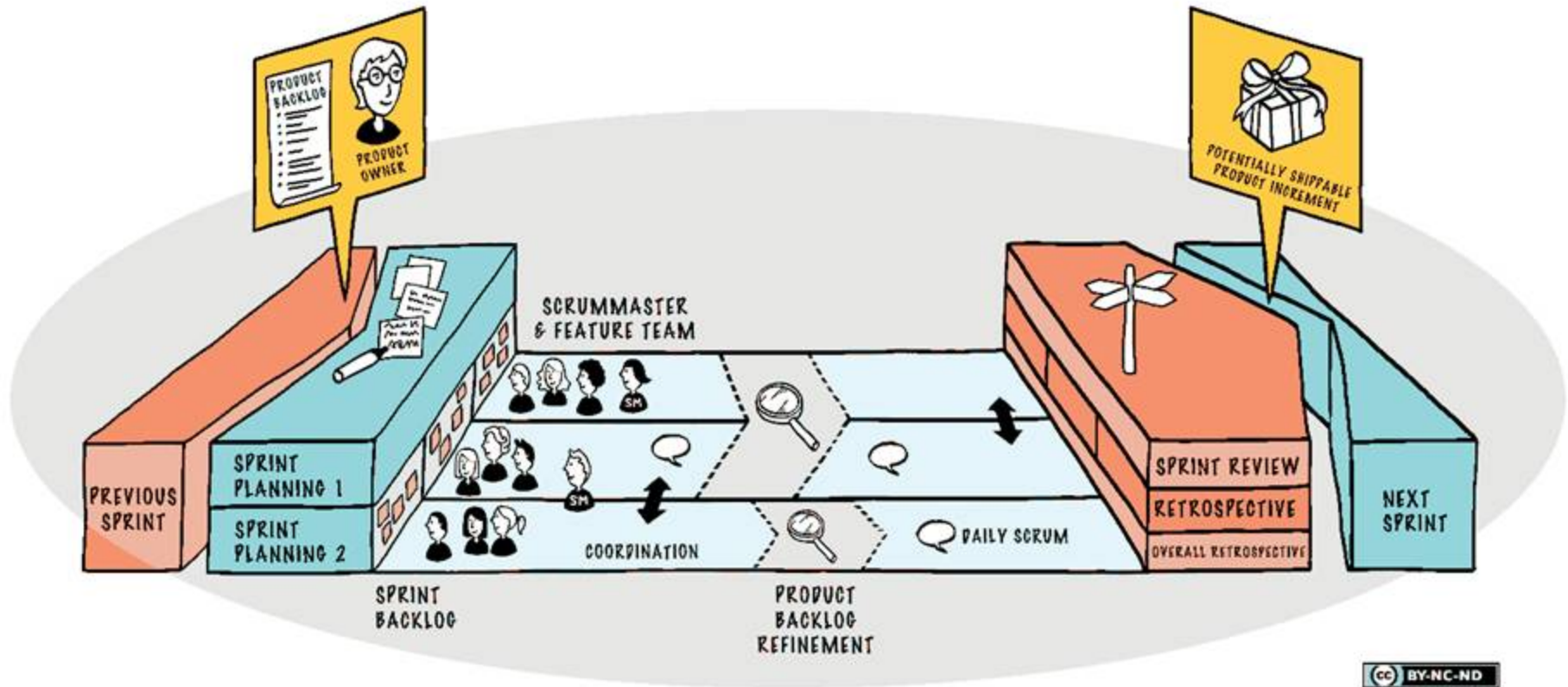
50% utilization: cycle time averages 5 times the time in service

90% utilization: cycle time averages 22 times the time in service

# Charades

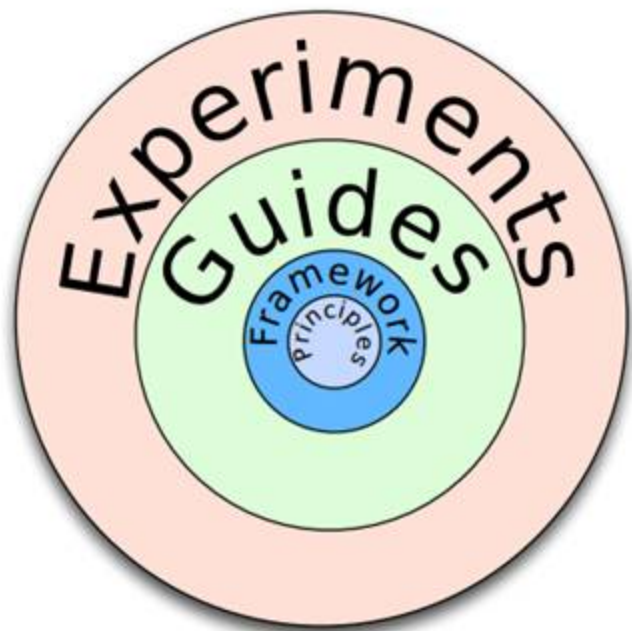


# The Frameworks – LeSS (“Basic” LeSS)



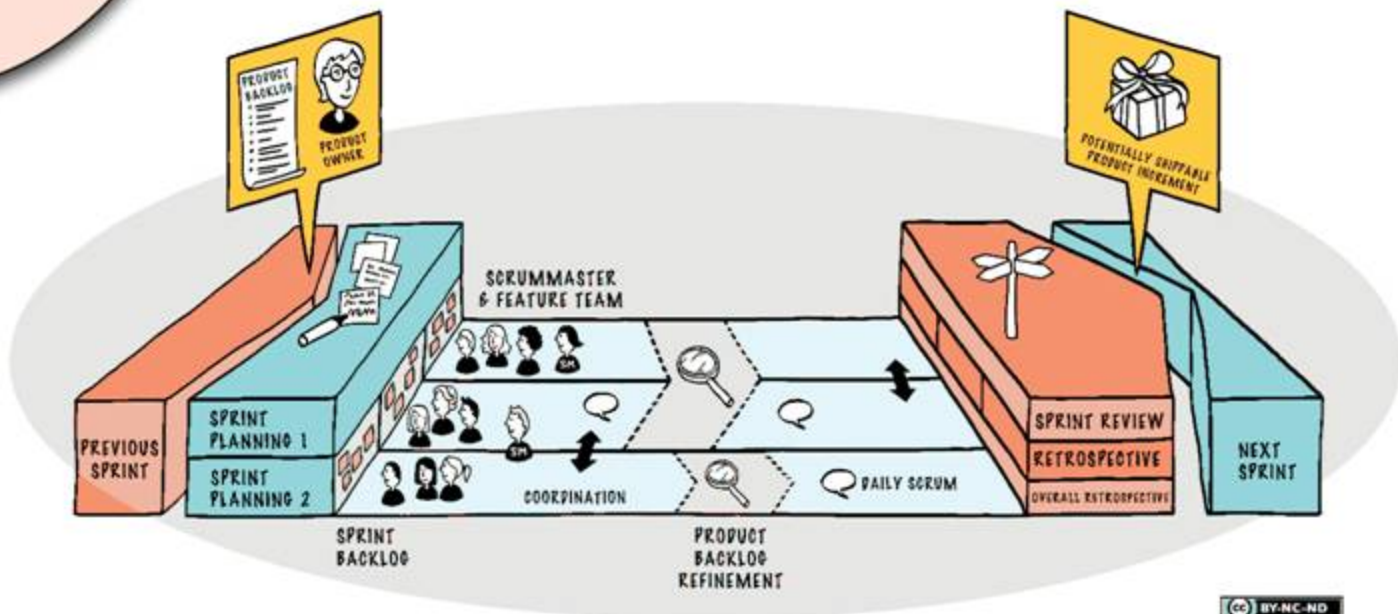
Applicable from 2 to “8” teams

# The term “LeSS” is over-loaded

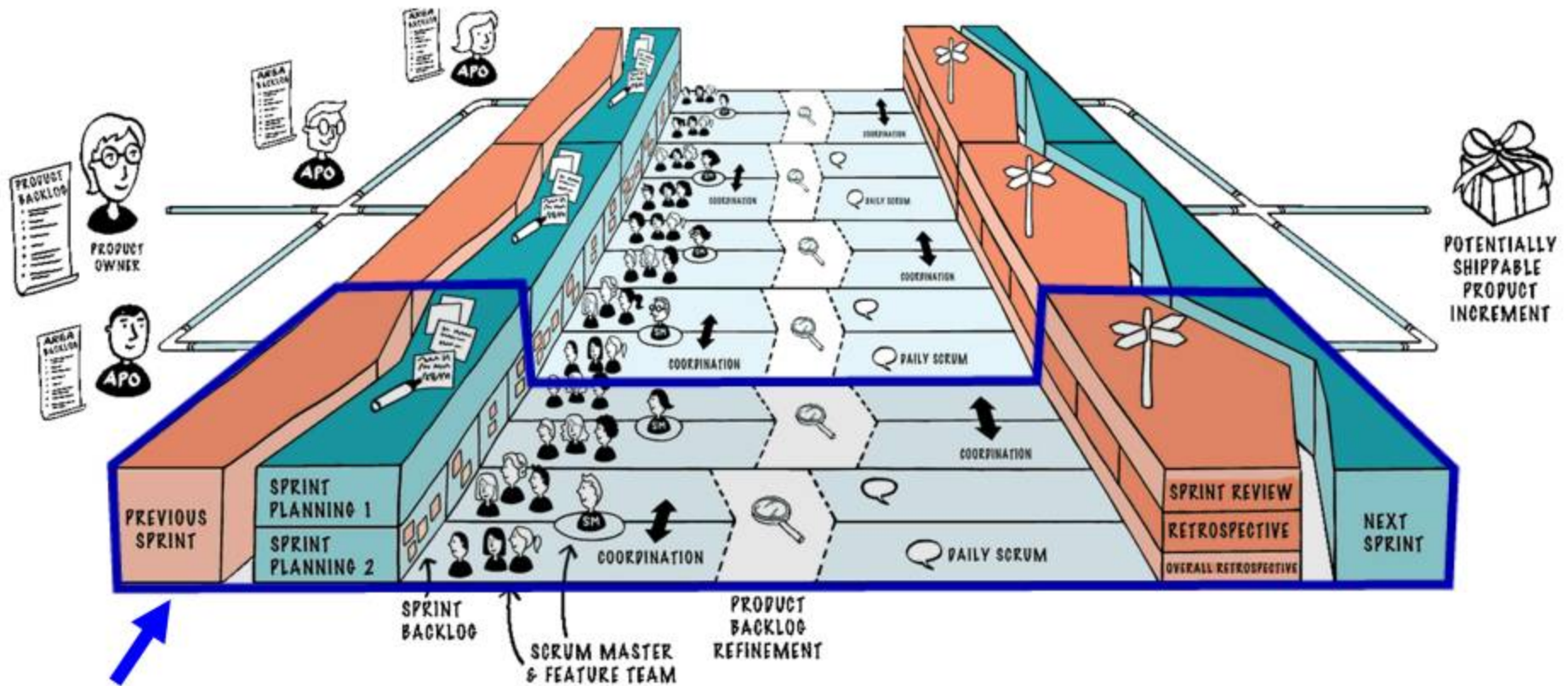


LeSS : Principles + Framework + Guides + Experiments

“Basic” LeSS  
framework or LeSS  
framework  
(formerly  
Framework 1 in  
book 2)



# The Frameworks – LeSS Huge



“Basic” LeSS Framework

Applicable to “8+” teams

# LeSS Rules define the LeSS Frameworks



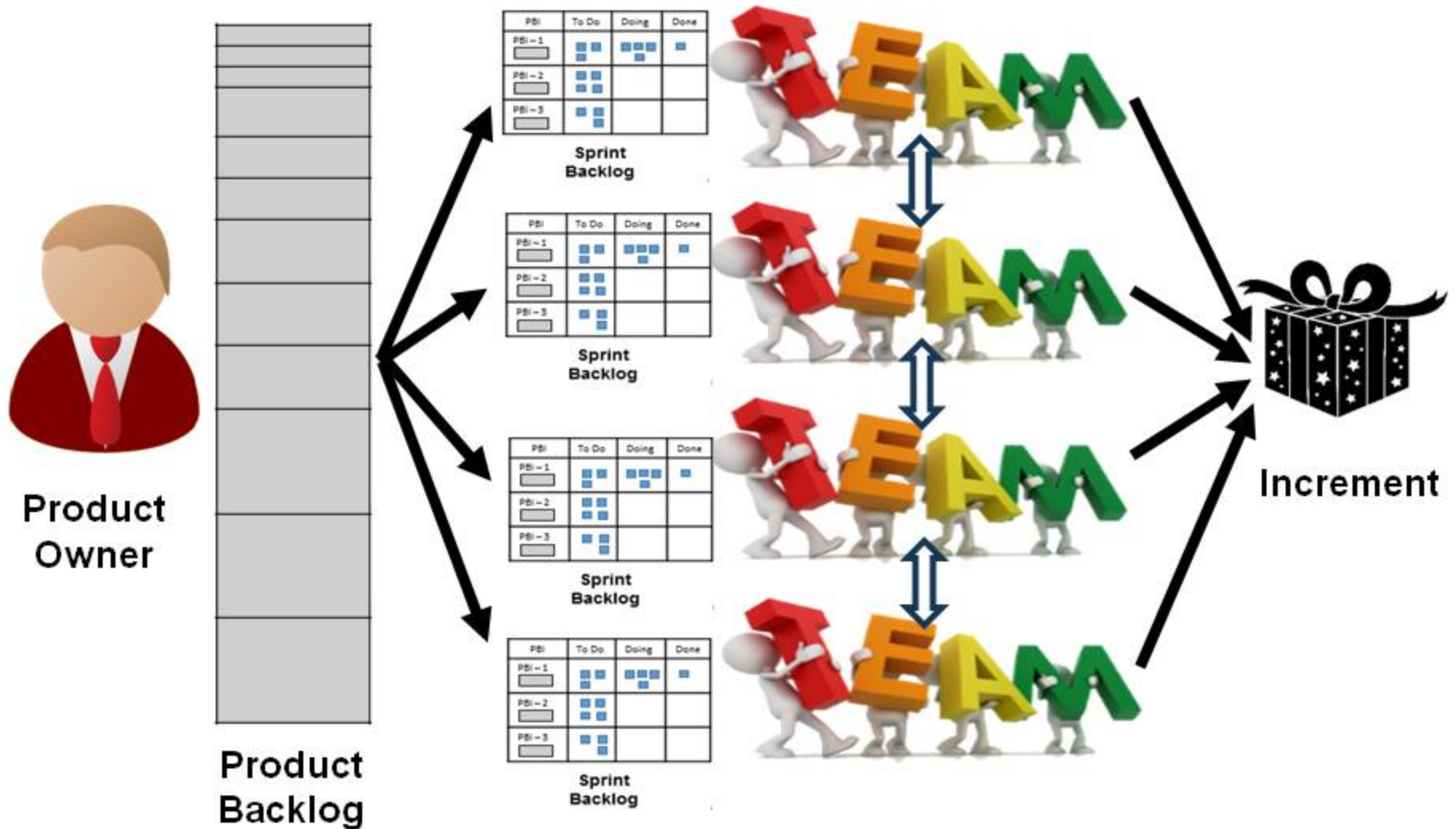
*LeSS is about many teams  
scrumming together on  
the same Product*



LeSS is NOT about many teams doing  
their own Scrum, or Scrum at the bottom  
for each team and something different  
layered on top

*LeSS Rules (structure, product, and Sprint) serve as  
an “enabling constraint” and provides coherence  
when scaling Scrum*

# Product Backlog, Sprint Backlog, and Increment





# The Scrum Guide

Multiple Scrum Teams often work together on the same product. One Product Backlog is used to describe the upcoming work on the product

The Product Owner is the sole person responsible for managing the Product Backlog.... the Product Owner remains accountable (for the product backlog)

The Product Owner is one person, not a committee

# Tragedy of the Commons - One Product Owner per Team

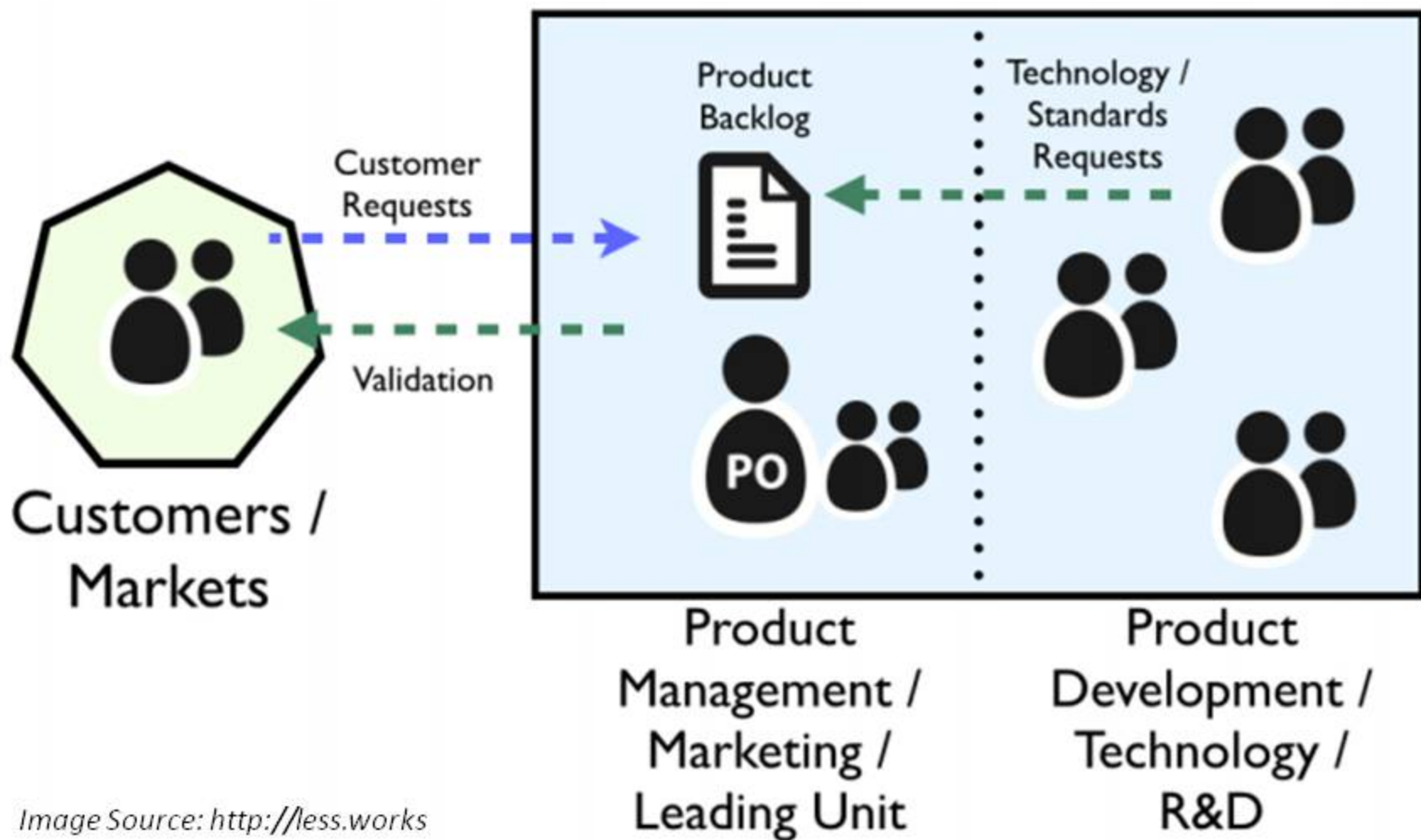


# Just One Product Owner? Yes !!!

- **LeSS Principle:** Customer centric, “Whole Product” focus
- Prioritization (Ordering) over clarification
  - Product Owner may do (product backlog management) work , or **have the Development Team do it** – The Scrum Guide
- Avoid burnouts
  - Product Owner as a connector, not a bottleneck in communication
  - Teams can and should directly work with stakeholders and other teams
- Focus on
  - Direction, prioritization, politics, market dynamics
- Watch out for
  - “more details”, administrativia, co-ordination
- Delegate (to teams)
  - Managing dependencies and inter-team coordination, challenging estimates, relaying information

# Pick the Right Product Owner

(Product Development)



# Pick the Right Product Owner

(Internal (Product) Development)

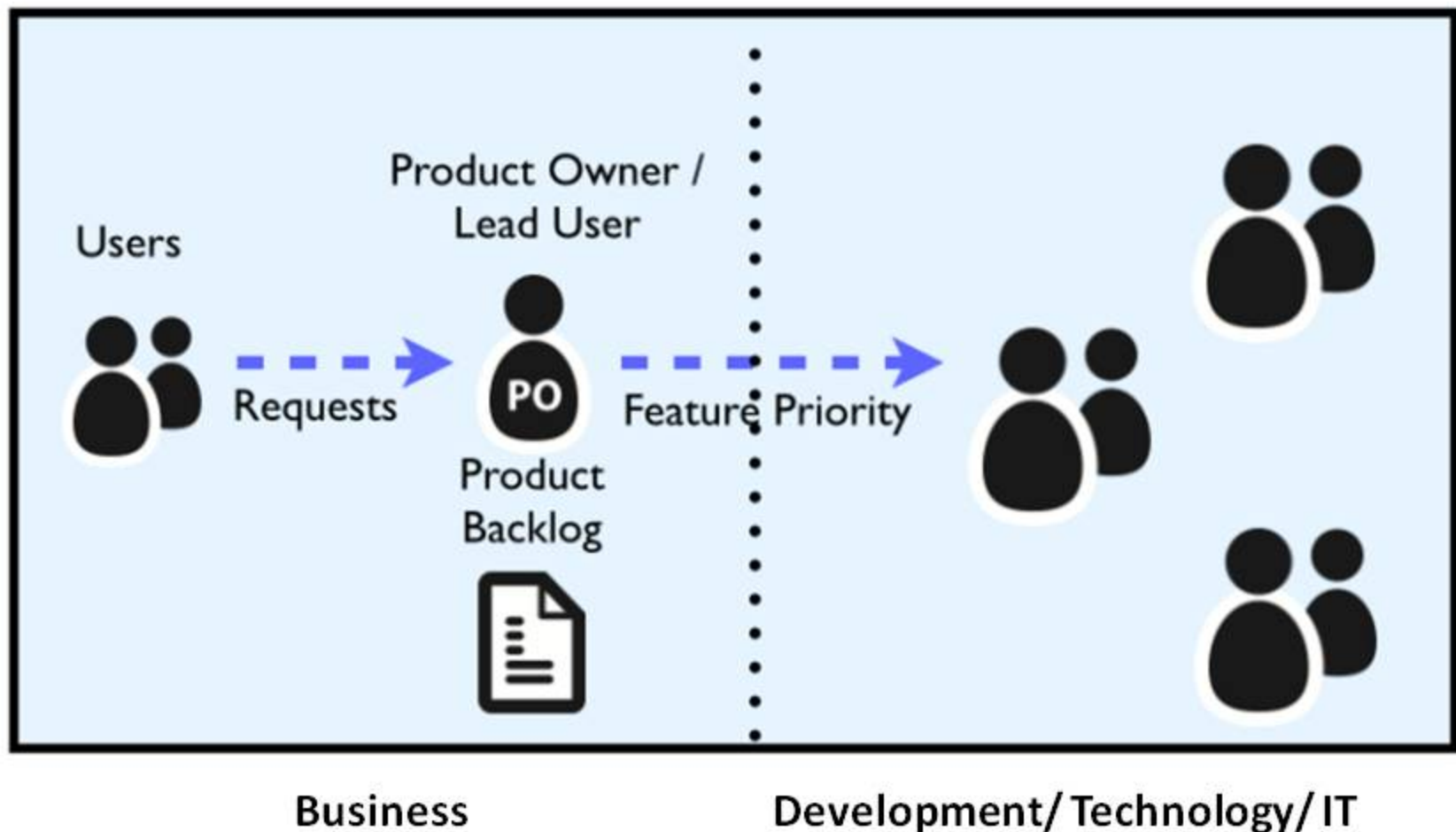
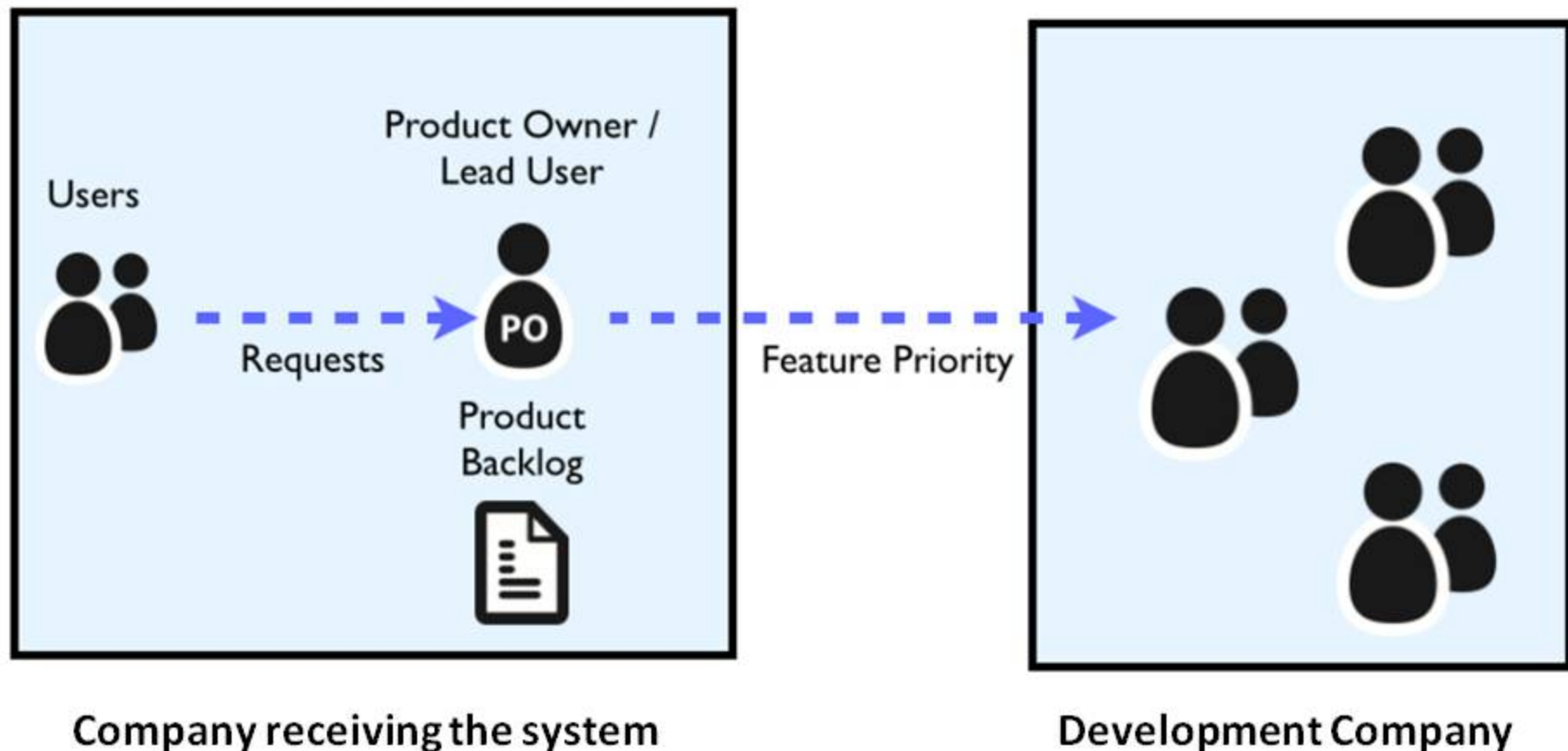


Image Source: <http://less.works>

# Pick the Right Product Owner

(Project Development / Client- Vendor Scenario)



Company receiving the system

Development Company

# LeSS Structure: Feature Team

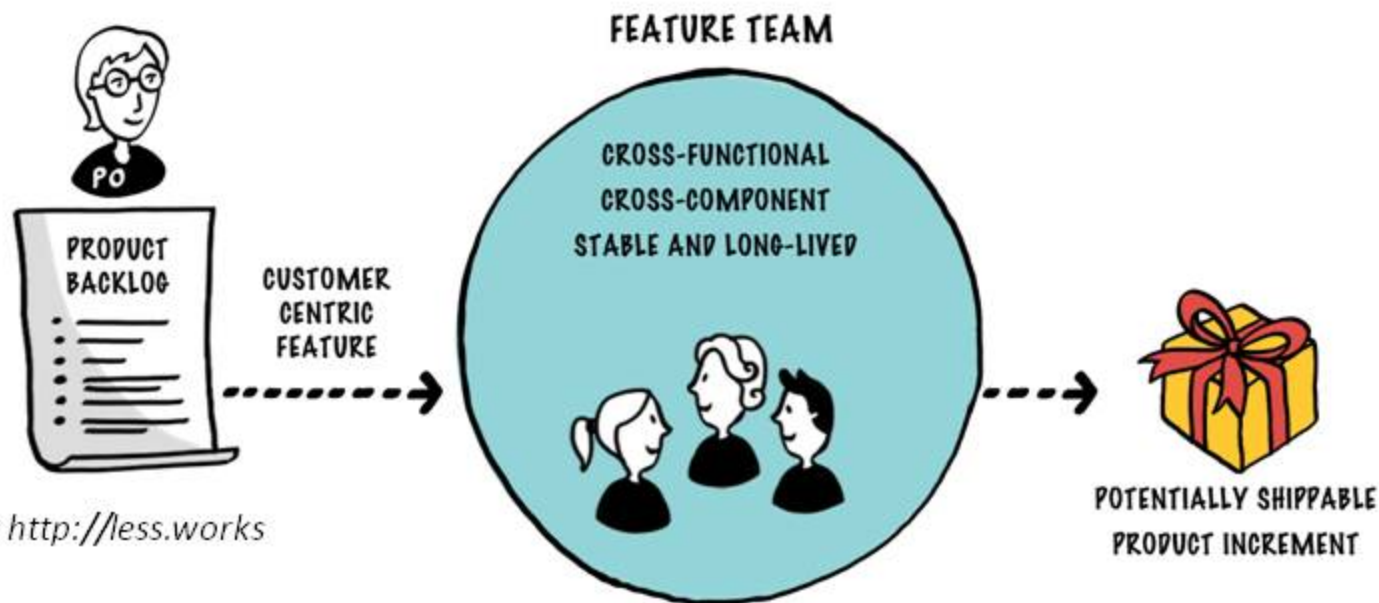


Image Source: <http://less.works>

- Has the necessary knowledge and skill to complete end-to-end customer centric features.
  - If not, expected to learn or acquire the needed knowledge and skills
  - Team members are “multi-learners” (technical, domain, etc.)
- Each team is self-managed, cross-functional, **co-located**, (multiple teams may be spread across geography) and long-lived .

# LeSS Structure: Feature Team

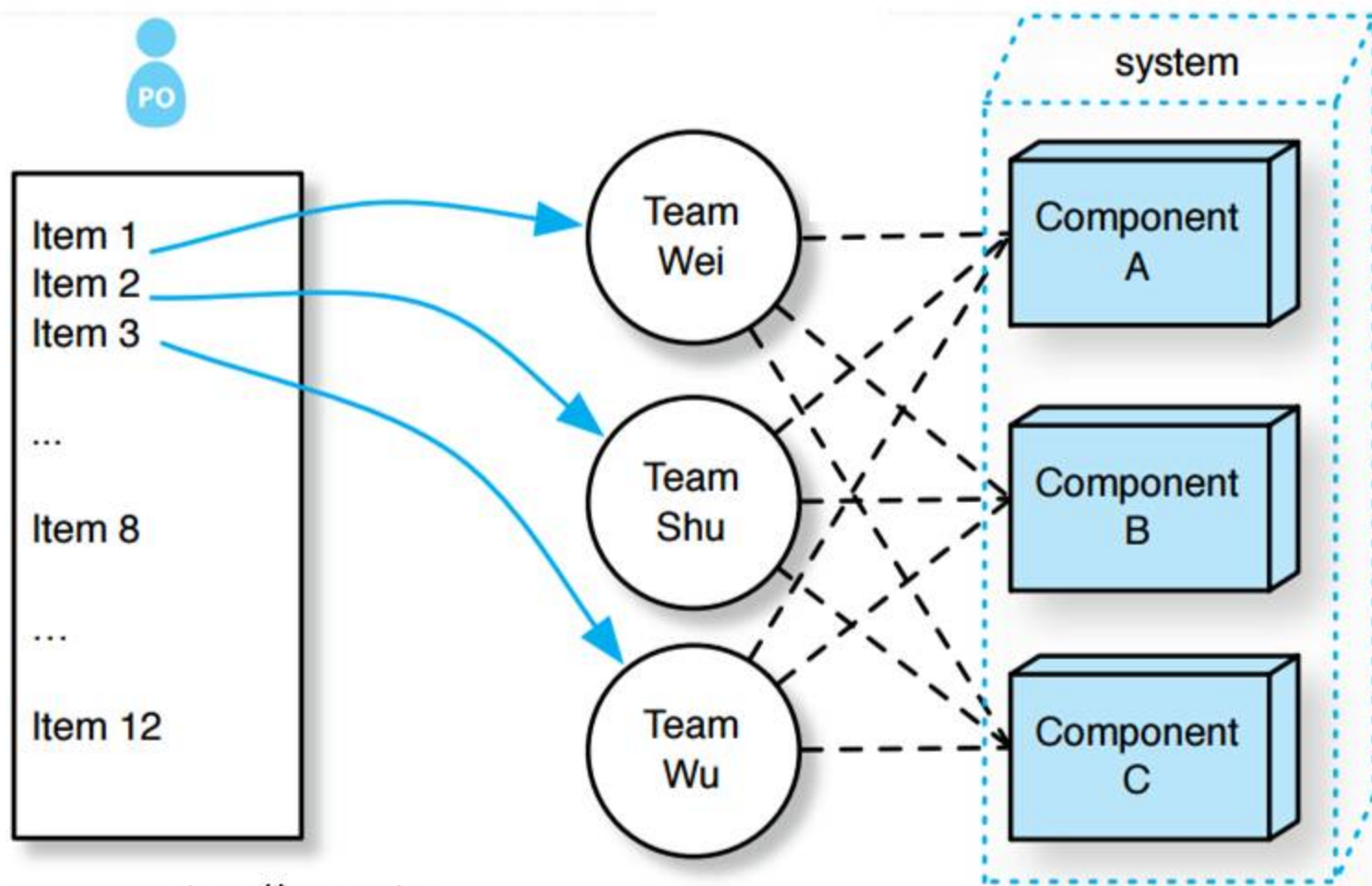
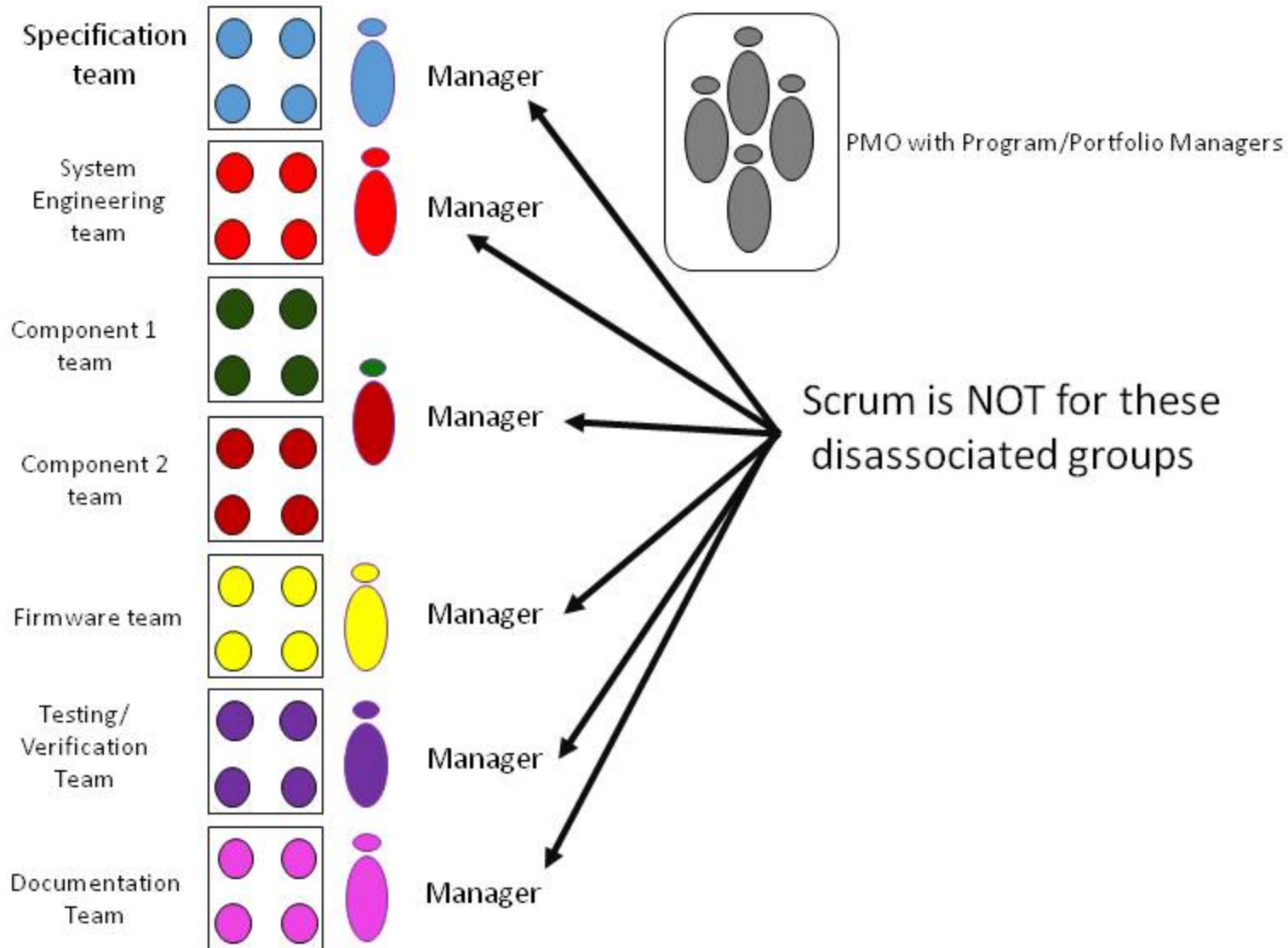


Image Source: <http://less.works>

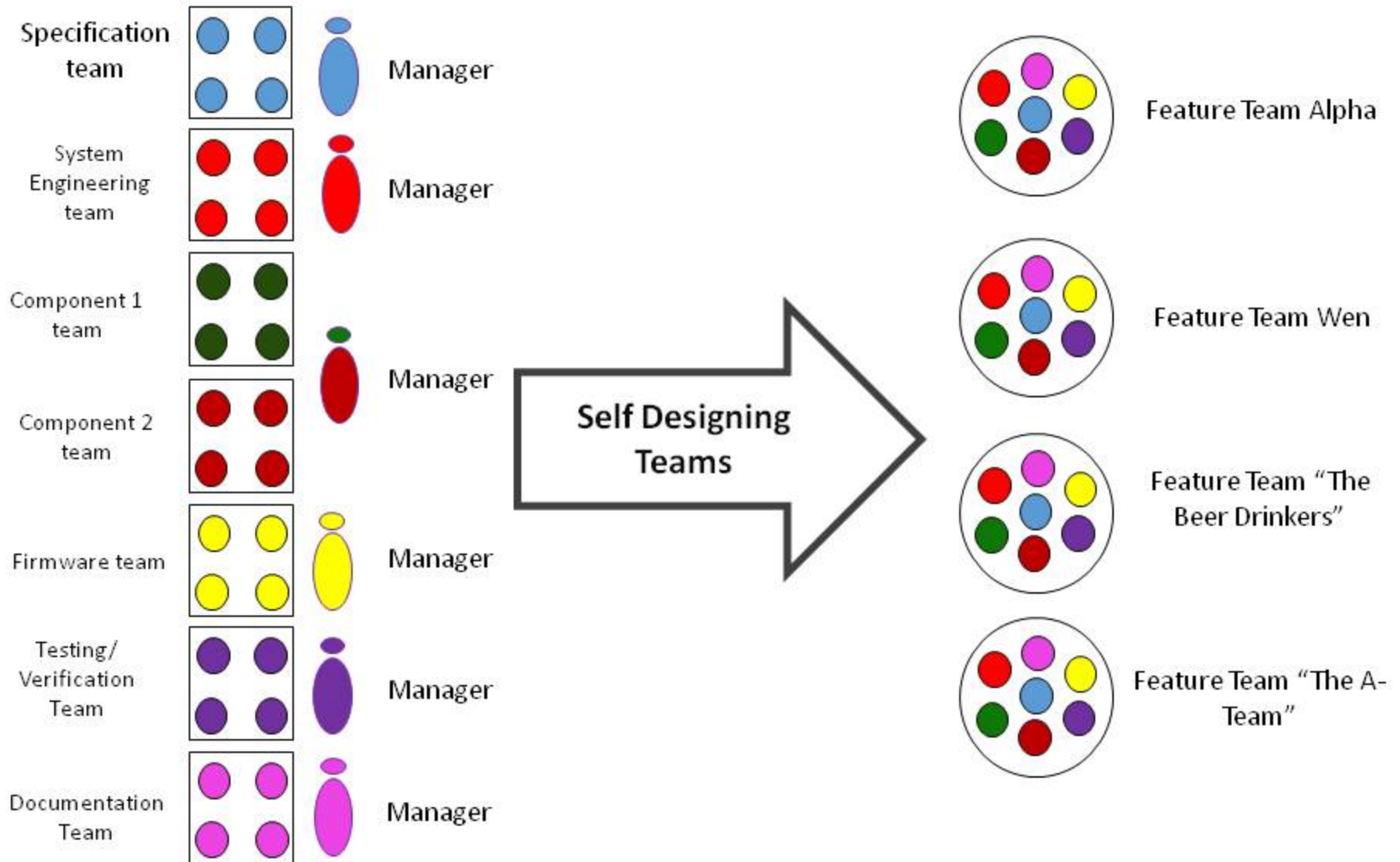
Hint: Continuous attention to technical excellence and good design enhances agility



# Component Teams to Feature Teams

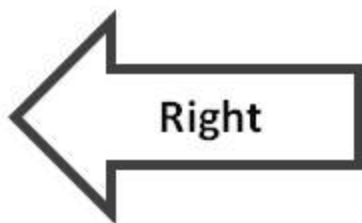


# Component Teams to Feature Teams



# Discussion: Change Implication of Component Teams to Feature Teams

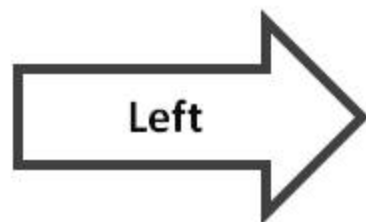
How will ex-functional managers, ex-component managers, and specialists react to this change?  
What will their new role be?



# Discussion: Change Implication of Component Teams to Feature Teams

Scrum Guide: ...no title other than (Product) Development Team Member regardless of the work being performed....  
No exception to this rule

What are the implication of this on performance reviews, HR policies and processes?

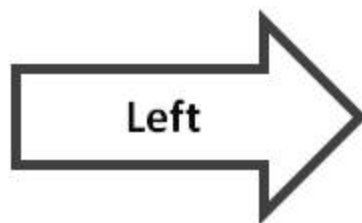
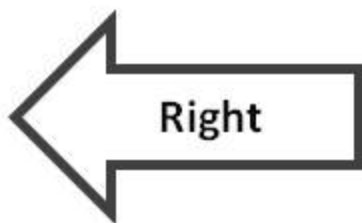


# Discussion: Change Implication of Component Teams to Feature Teams

How will ex-functional managers, ex-component managers, and specialists react to this change?  
What will their new role be?

Scrum Guide: ...no title other than (Product) Development Team Member regardless of the work being performed....  
No exception to this rule

What are the implication of this on performance reviews, HR policies and processes?





**“It is difficult to get a man to understand something when his job depends on not understanding it.”**

*—Upton Sinclair*

# Lean Principle – Respect for People

Job safety and salary  
safety but no role safety

# Scrum Master is a Full-Time role

(One Scrum Master can serve 1-3 teams)

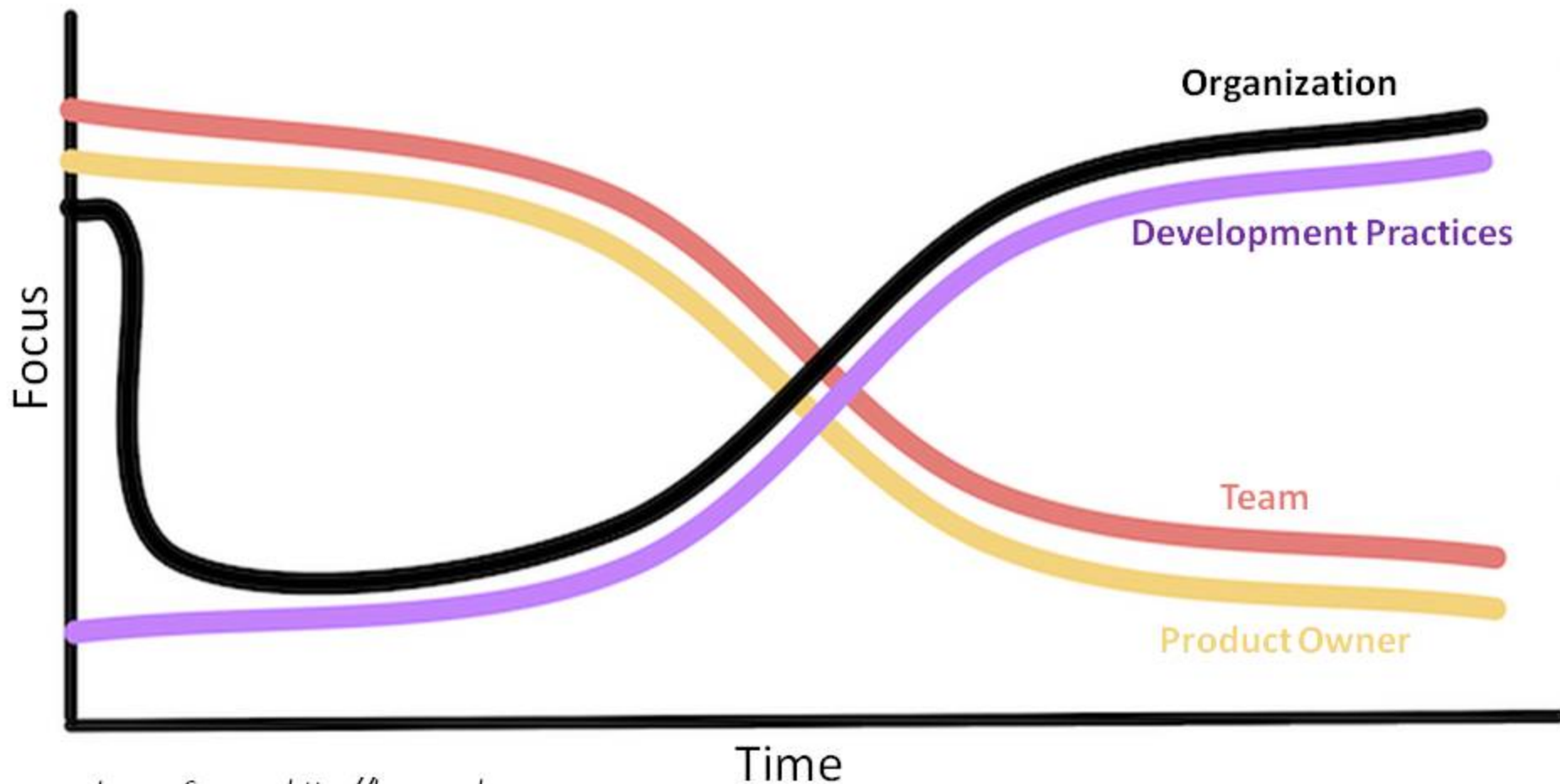
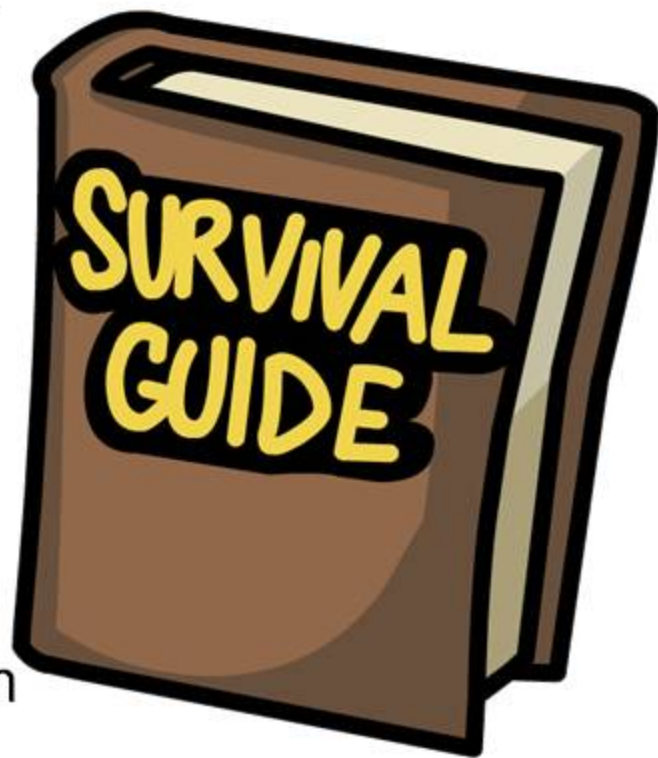


Image Source: <http://less.works>

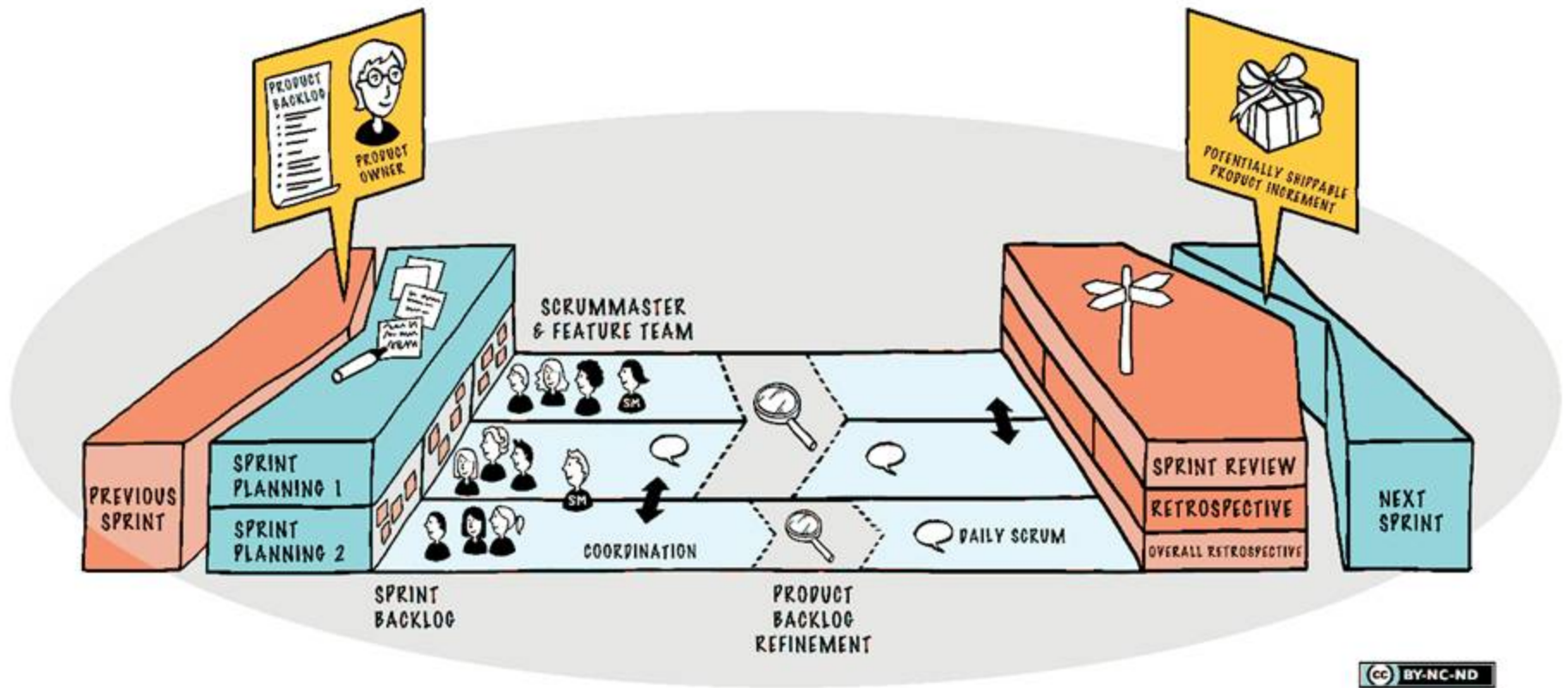


# LeSS Guide for Scrum Master

- Don't be the teams' personal assistant
  - Enable co-ordination between teams
  - Have teams' take more ownership
- Avoid being an "anti-Scrum Master"
- Promote multi-learning
- Create LeSS Community
- Create Scrum Master Community
- Learn large-group facilitation techniques
- Survive (and thrive) in the organization
  - Patience, low expectation, courage, persistence, open and humble, sense of humor
- Partner with Managers and executives



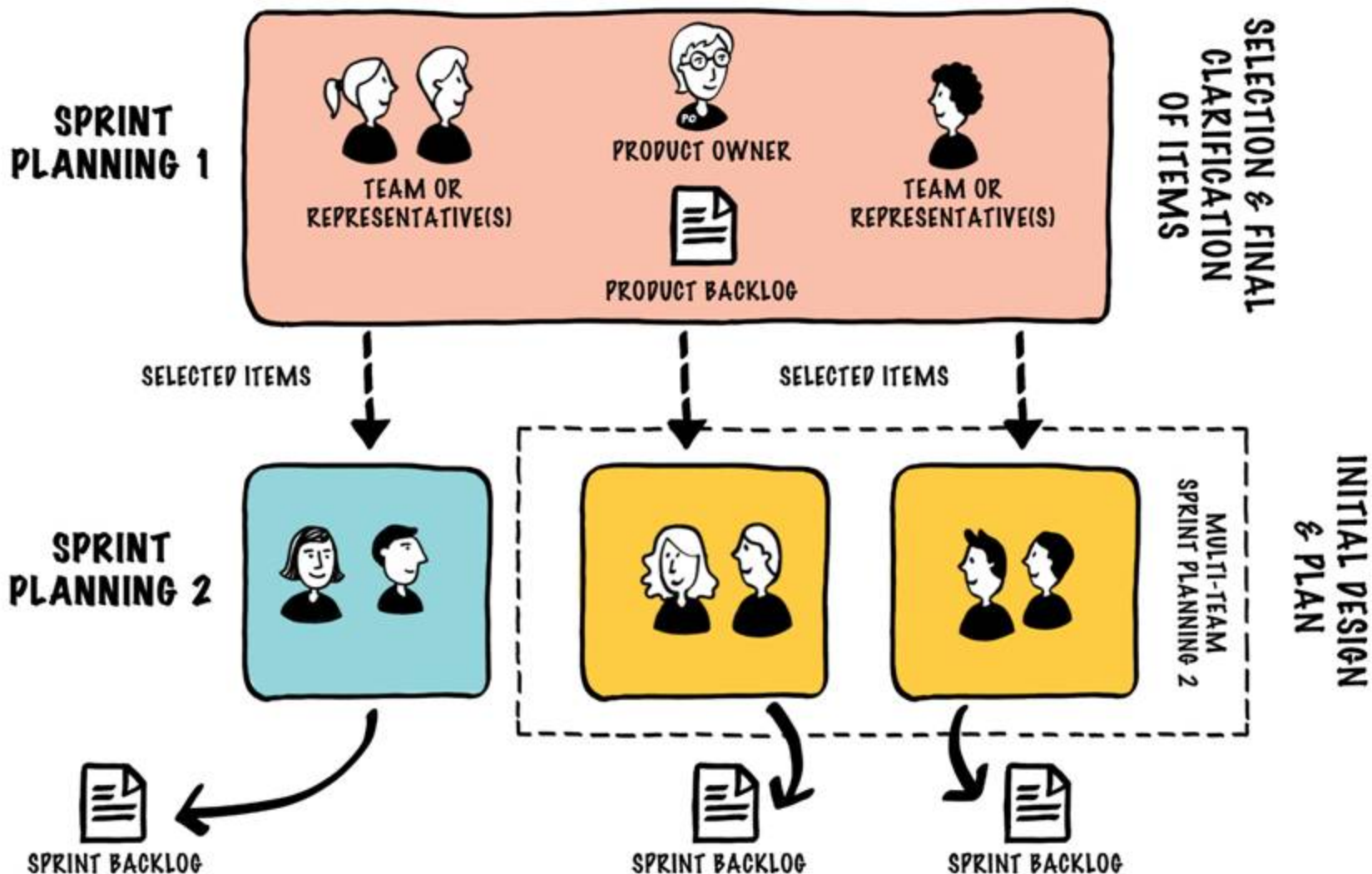
# There is only ONE sprint



Source: <http://less.works>

Sprint Planning, Review and Retrospective synchronized  
Product Backlog Refinement is coordinated between the teams

# LeSS Meetings: Sprint Planning



# LeSS Meetings: Sprint Planning 1

- Product Owner brings the ordered Product Backlog
- PO / SMEs clarify Product Backlog Items, if needed
- Teams (if only 2) / team representatives (more than 2 teams) decide which items to pick
- Spread risk among teams. If items higher up in the backlog are not equally spread out, PO can change the distribution
- The teams should be able to complete PBIs end-to-end with minimal dependencies

## Tips

If teams have many questions during SP1, PBR is not rigorous

If there are too many cross-team coordination issues, the teams are not true feature teams

# LeSS Meetings: Sprint Planning 2

- Teams decide “How”
- One room, multiple teams
- Shared design
- Each team creates their own Sprint Backlog
- Team members interact and co-ordinate throughout the Sprint



# LeSS Meetings: Daily Scrum

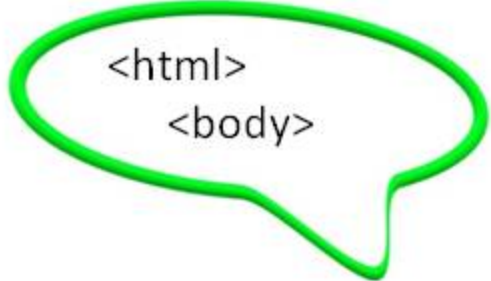
- ... is no different from a single-team Scrum
- ... can be used for coordination between teams by having people from other teams join in to observe.
- ... is a meeting for the Team, not for the Scrum Master or managers.

# Coordination during the Sprint




**Just Talk !!!**

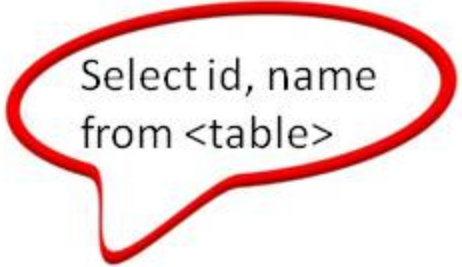
# Coordination during the Sprint



```
<html>  
<body>
```



```
#include <stdio.h>
```



```
Select id, name  
from <table>
```



Barriers to  
integration are  
barriers to  
coordination

**Communicate in Code**



# Coordination during the Sprint



Send Observers to the Daily Scrum

# Coordination during the Sprint



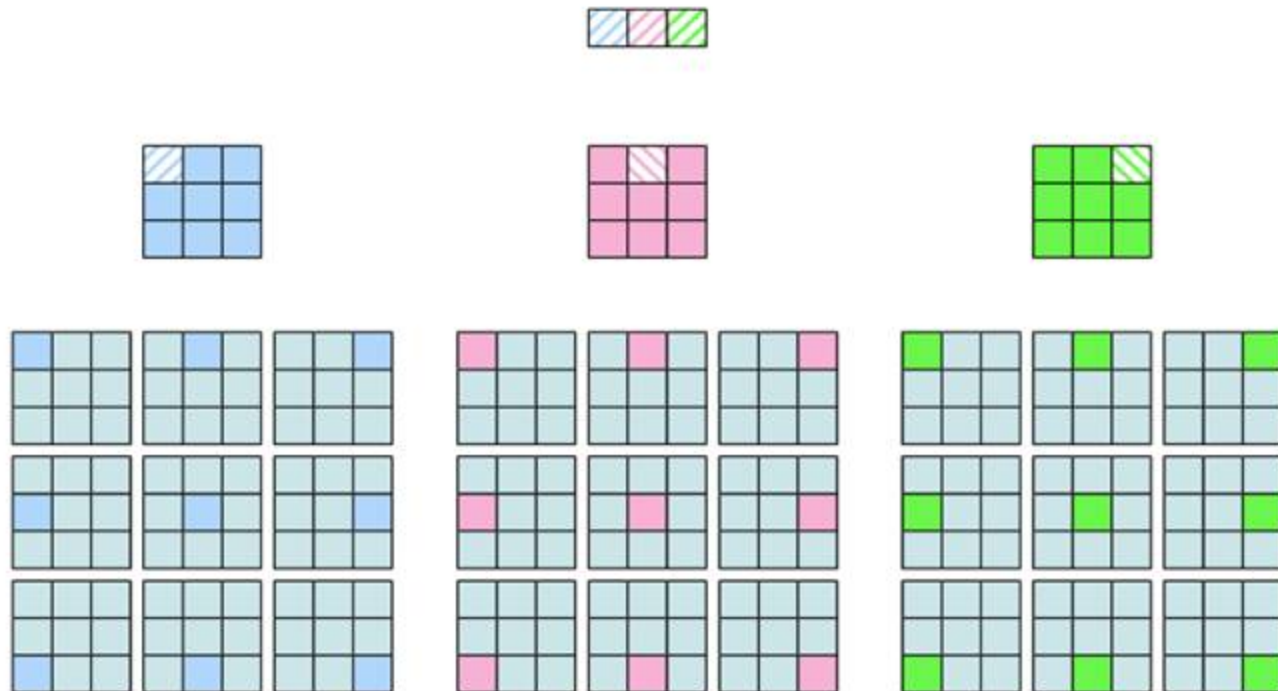
**Avoid Branching**



**Use Communities**

# Coordination during the Sprint

## Scrum of Scrums



**A caution:** The desire to hold a Scrum of Scrums can be a sign of unnecessary dependency or coordination problems caused by single-function groups and component teams, or by teams not able or willing to identify and do shared work.

# Coordination during the Sprint

## Decentralized (preferred)

- Send the observer (to other teams' Daily Scrum)
- Travellers
- Multi-site Shared Space (e.g. "always on" webcams)
- Environment Mapping (explore teams' dependencies)
- Coordination Working Agreements

## Centralized

- Open Spaces
- Scrum - of - Scrums
- Town Hall

# Product Backlog Refinement in LeSS

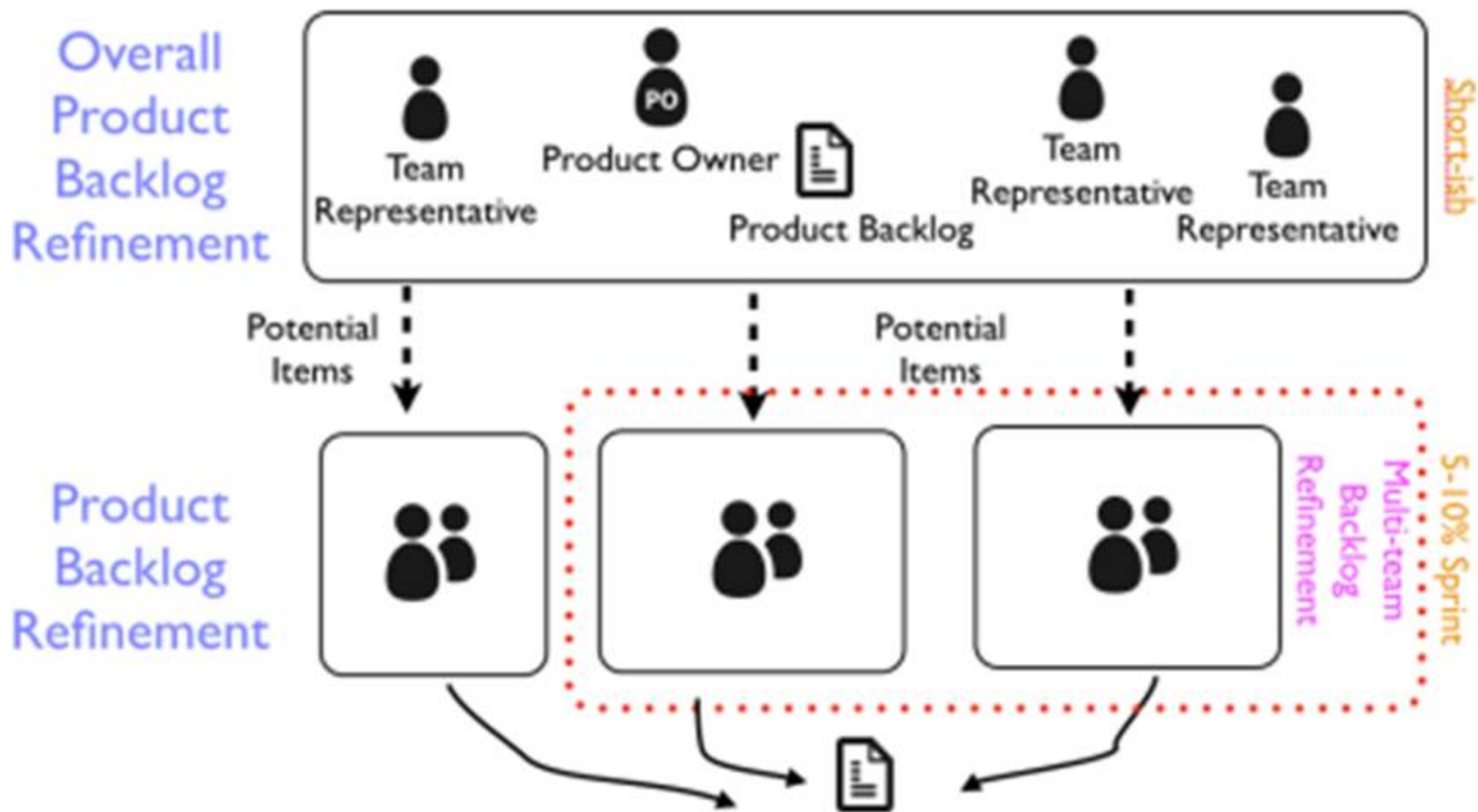


Image Source: <http://less.works>

# LeSS Meetings: Sprint Review



For larger teams, use  
Diverge-Merge techniques

# LeSS Meetings: Sprint Retrospective

**TEAM  
RETROSPECTIVE**



**OVERALL  
RETROSPECTIVE**



*Image Source: <http://less.works>*

In an Overall Retrospective, the systemic and organizational issues explored are above the level of a single team.

# ONE Definition of Done for the Product

- Not one DoD per team, but Teams can extend the “base” version of DoD
- Must be created before the first Sprint during Initial Product Backlog Refinement by Teams and Product Owner
- Steps
  - Identify all activities (technical, non-technical) that should be performed to ship the product to end customer (Perfect DoD for PSPI)
  - Identify activities that can be done within each Sprint considering current capabilities, context, skills, etc. (Current DoD)

Work “done” ( in iteration) = current DoD x Product Backlog Item

**“Undone” work = work for Perfect DoD – work “done” as per current DoD**

## LeSS Principle - Continuous Improvement Towards Perfection

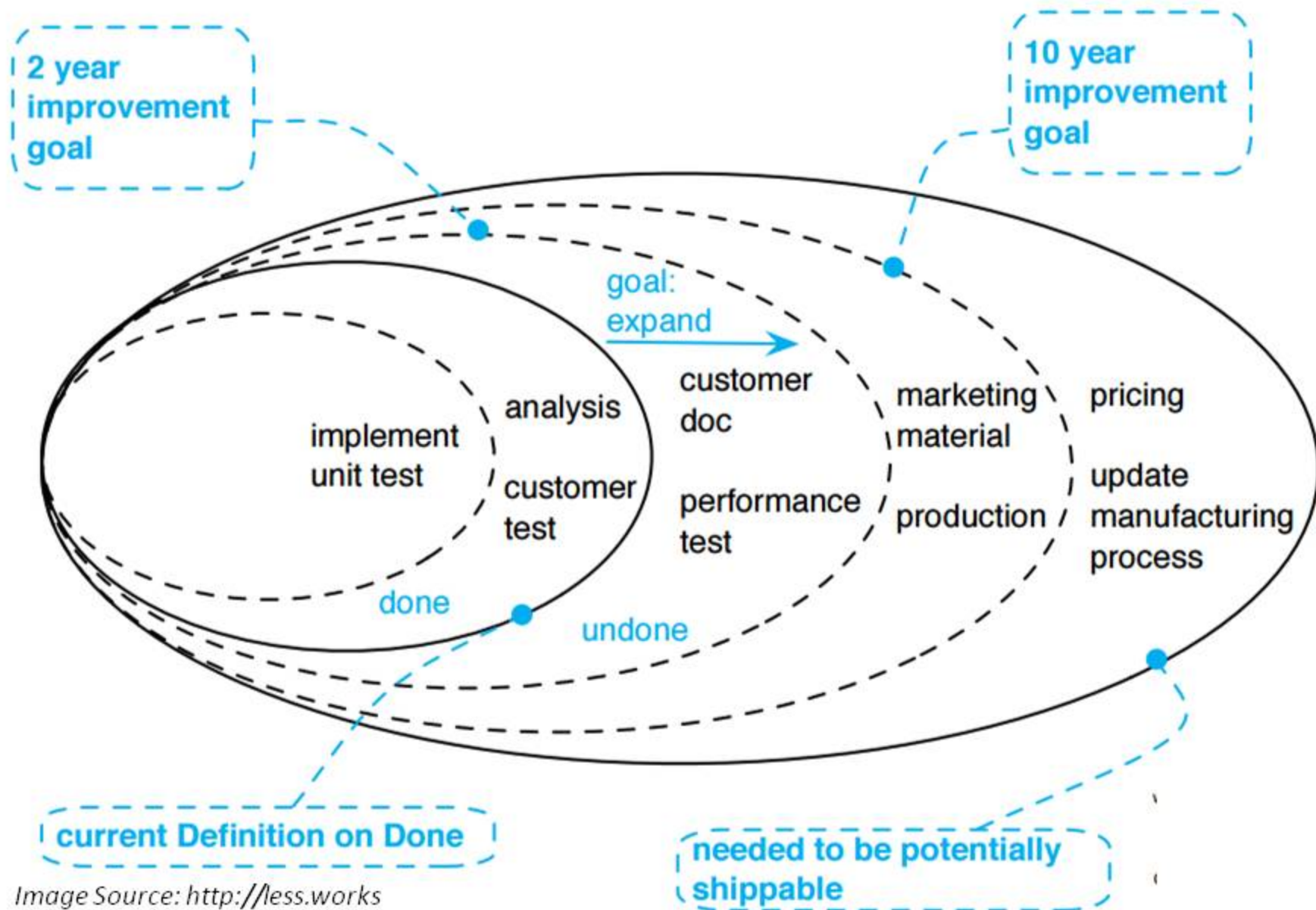
“Undone” Increment reduces transparency. So as time progresses, current DoD becomes more rigorous and there is no “Undone” work



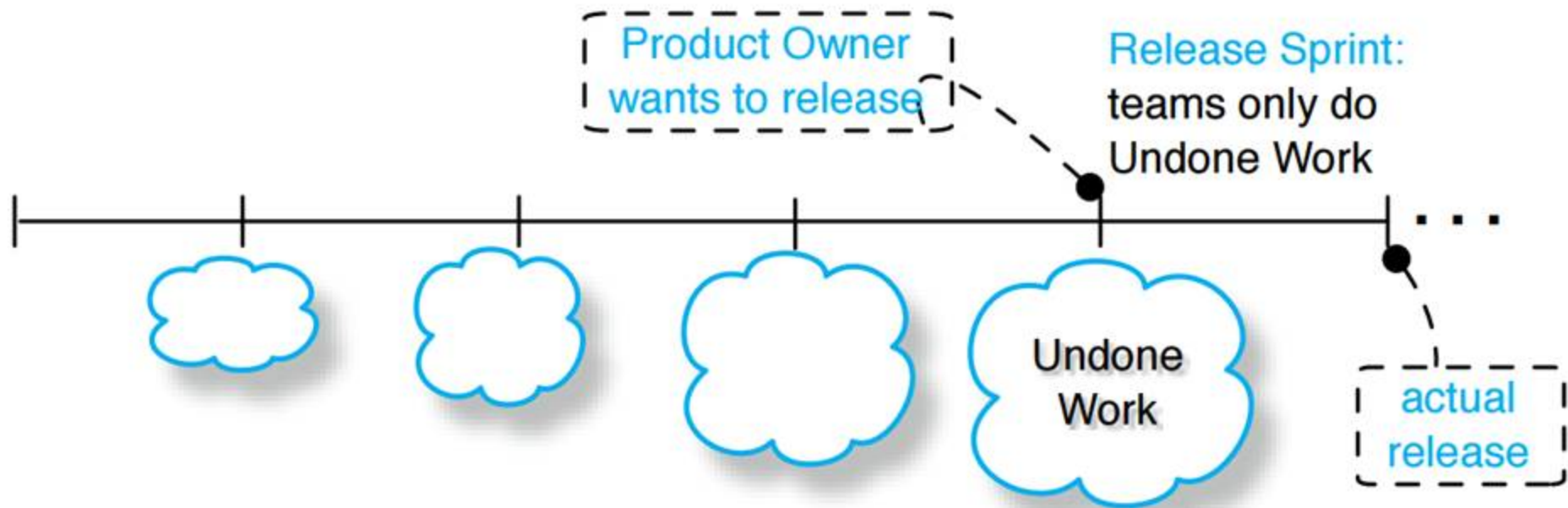
If you can, get a perfect DoD before Sprint 1



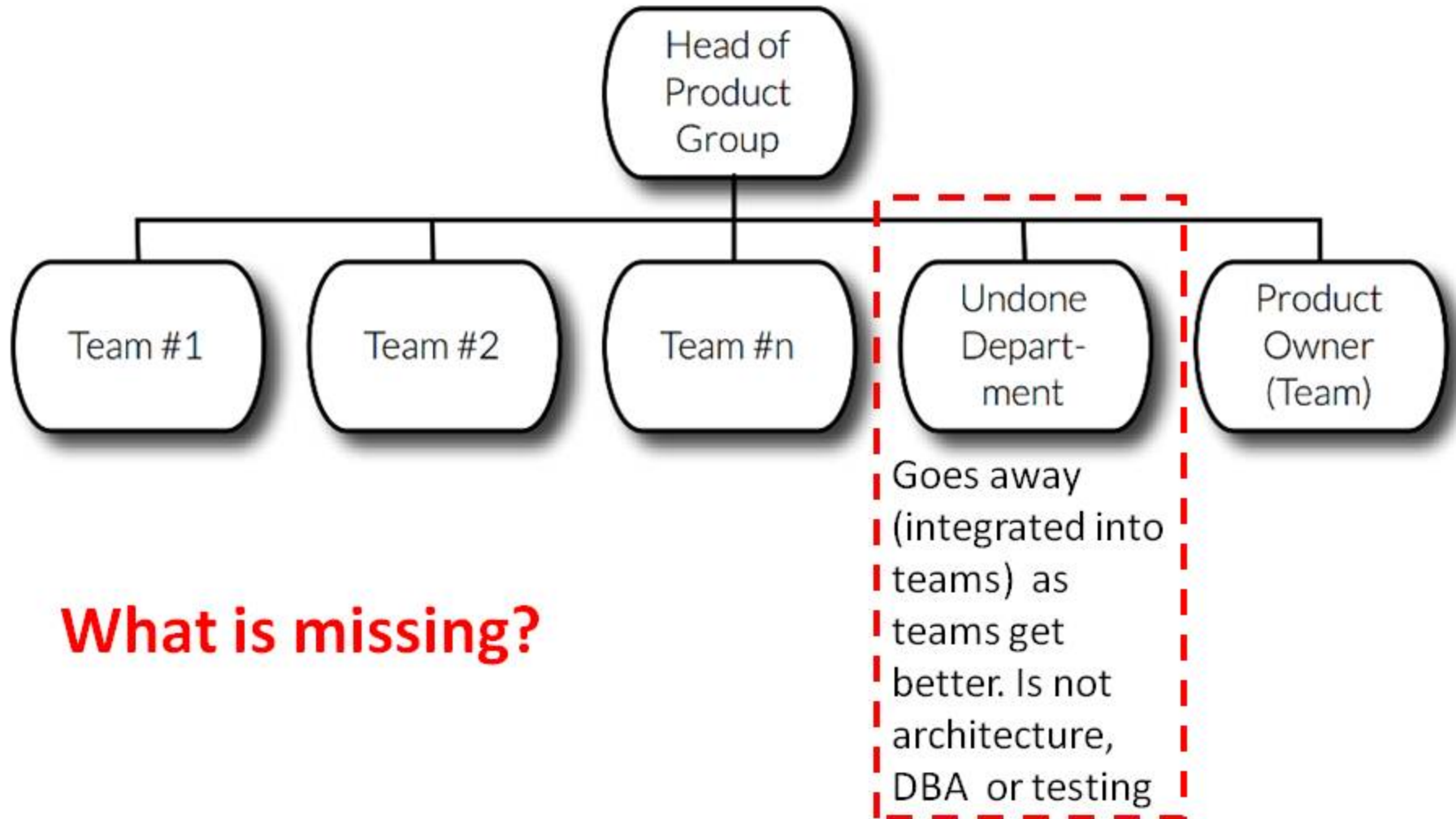
# If you end up having "Undone" work....



# If you end up having “Undone” work....



# LeSS Organizational Structure



# Management



**ROLE OF MANAGER**



**GO SEE**



**TEACHING PROBLEM SOLVING**



**MANAGER AS SCRUMMASTER?**



**MANAGEMENT**



**SELF-MANAGEMENT**

**IMPROVEMENT SERVICE**



# Organizational Agility is constrained by Technical Agility



**SPECIFICATION BY EXAMPLE**



**CONTINUOUS INTEGRATION**



**CONTINUOUS DELIVERY**



**TEST AUTOMATION**



**TECHNICAL EXCELLENCE**



**ARCHITECTURE & DESIGN**



**ACCEPTANCE TESTING**



**THINKING ABOUT TESTING**



**CLEAN CODE**



**TEST-DRIVEN DEVELOPMENT**



**UNIT TESTING**

Image Source: <http://less.works>

# Adopting LeSS



THREE PRINCIPLES



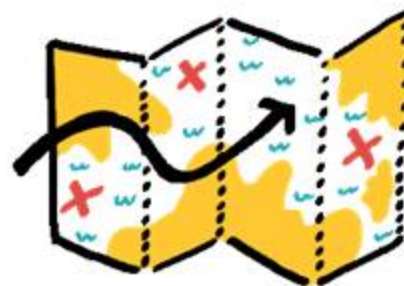
GETTING STARTED



COACHING



ADOPTION



FEATURE TEAM  
ADOPTION MAP



CONTINUOUS IMPROVEMENT

# Adoption: Three Principles



THREE PRINCIPLES



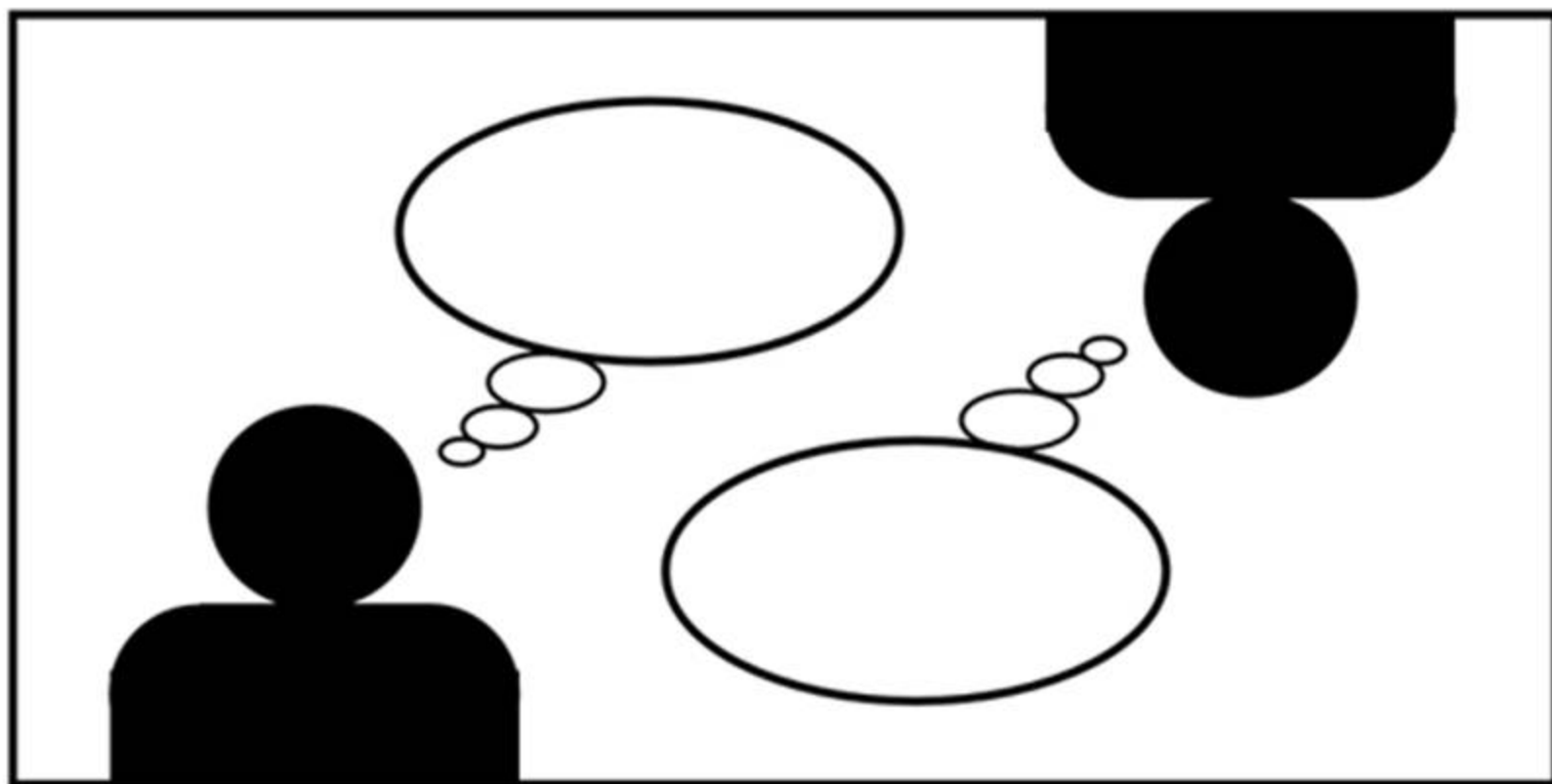
Deep and narrow over broad and shallow



# Adoption: Three Principles



THREE PRINCIPLES



Bottom-up and Top-Down

# Adoption: Three Principles



THREE PRINCIPLES



Use Volunteering

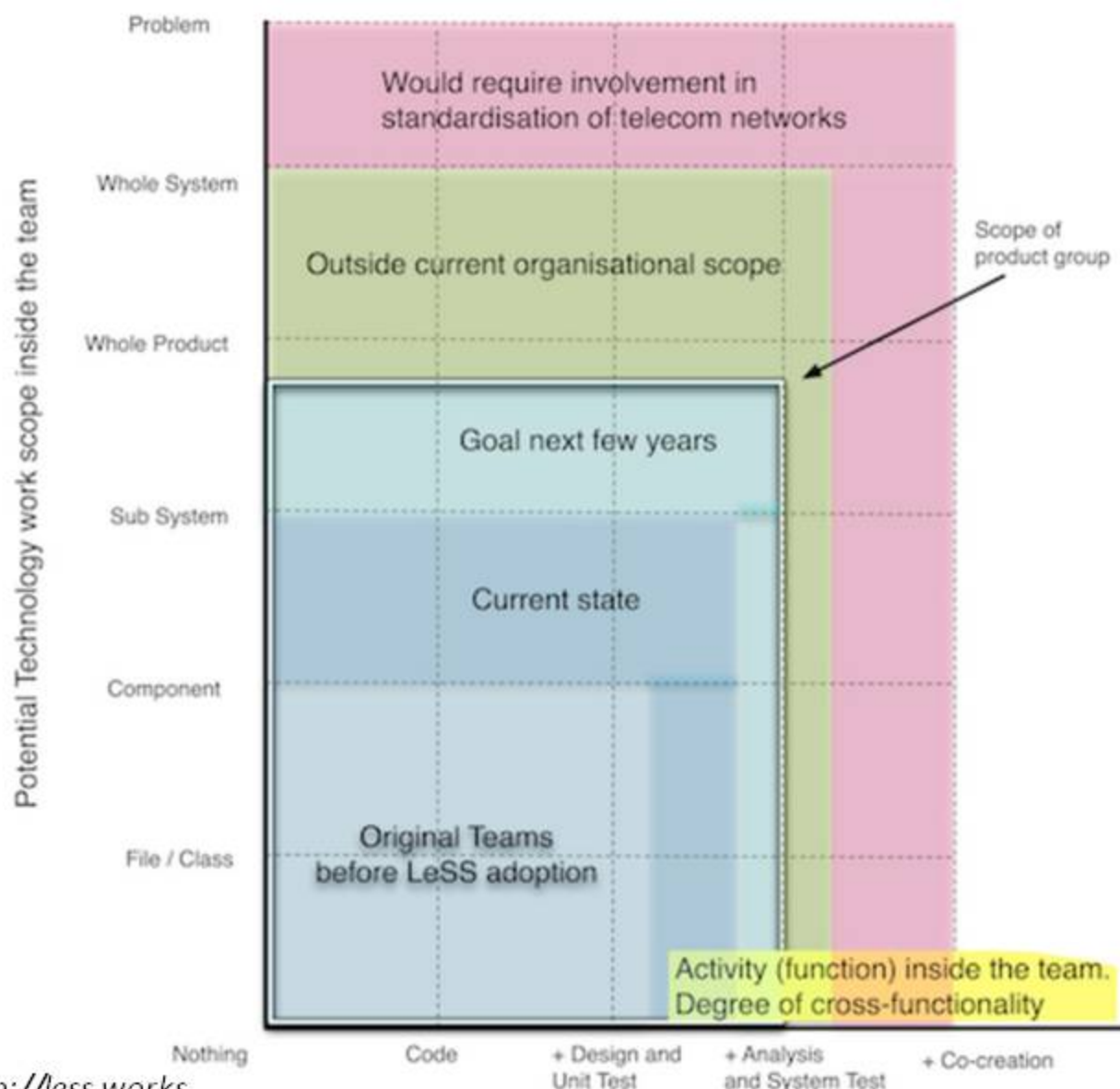
# Coaching

- Internal vs External coaching
- Three levels of Coaching
  - Organizational Coach
  - Team level (or LeSS coach)
  - Technical practices



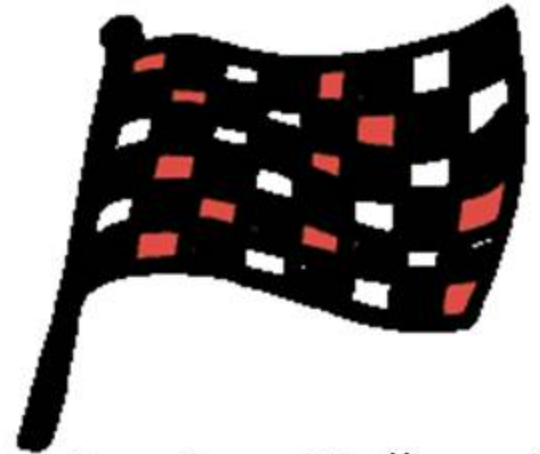
*Image Source: <http://less.works>*

# Feature Team Adoption Map



# Getting Started

1. Educate everyone
2. Define 'product'
3. Define 'done'
4. Have appropriately-structured teams
5. Only the Product Owner gives work to the teams
6. Keep project managers away from the teams



*Image Source: <http://less.works>*

