

# Overview of Large Scale Scrum (LeSS)

@ Johnson & Johnson Agile CoP



# About Gene

Gene Gendel is an organizational design specialist, agile/lean coach and trainer, consultant and adviser to senior leadership. Almost 15 of 20+ years of his professional experience – Gene dedicated to working with companies of various sizes and lines of business, trying to help them improve internal dynamics, organizational structure and becoming a better place to work. Gene engages at all organizational levels: senior- and mid-level management, teams and individuals. In his work, Gene uses various methods, tools and techniques to amplify learning by others and to ensure that people gain autonomy after Gene “coaches himself out of the job”.

Gene is Certified Agile Coach (CAC), by Scrum Alliance – small elite community of Scrum Alliance Certified Enterprise Coaches (CEC). Today, he is the only CEC who resides in NY State. Gene is also one of the co-creators of Team Level Coaching Certifications (CTC) program for Scrum Alliance.

Gene is also one of very few Certified LeSS Trainers (CLT) that are based in the United States.



He is also well-recognized blogger and publisher. He is the author of Adaptive Ecosystems: Collection of Independent Essays About Agility and co-author of:

- Agile Coaching: Wisdom from Practitioners (free pdf).
- Best Agile Articles of 2017 and 2018



Gene significantly contributes to global and local agile communities, where he influences people by running workshops, webinars, seminars, panel discussions, lunch & learn, coaching retreats, professional gatherings and other events. While teaching and coaching Large Scale Scrum (LeSS) to organizations, Gene has built the biggest in world (today) LeSS meetup community that is used as a free educational hub about LeSS available to many people.

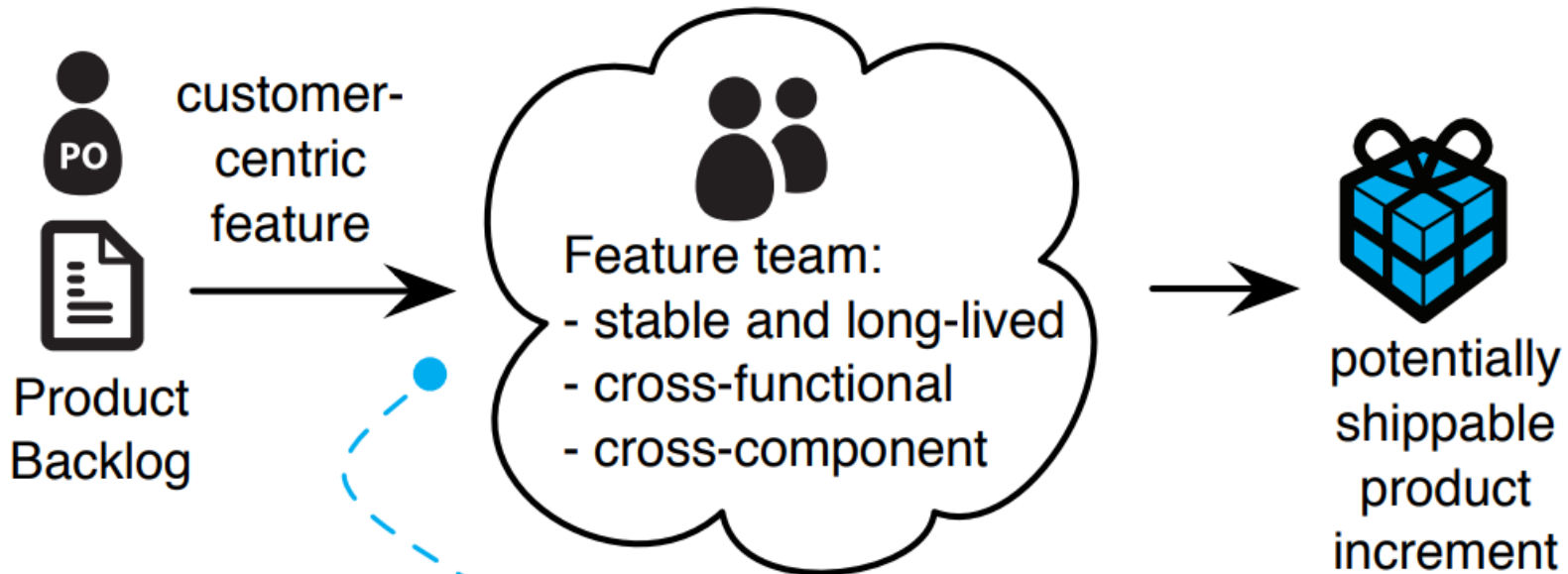
# Assets To Consider

- [gg\\_nyc@yahoo.com](mailto:gg_nyc@yahoo.com) (underscore)
- <https://less.works/>
- <https://www.meetup.com/Large-Scale-Scrum-LeSS-in-NYC/> 
- [https://join.slack.com/t/kstsconsulting/shared\\_invite/zt-ege93ww2-h\\_IAPArcR8CyujNi2umlKw](https://join.slack.com/t/kstsconsulting/shared_invite/zt-ege93ww2-h_IAPArcR8CyujNi2umlKw) (Gene's free Slack channel – ask questions about LeSS)
- [\*\*6/10-12 \(12:00 EST\) Certified LeSS Basics\*\*](#) 

# Soft Agenda

- High –level overview of Large Scale Scrum (LeSS) as an organizational design framework:
- Big picture: Principles → Rules → Guides → Experiments
  - Three Adoption Principles:
    - Deep and narrow over broad and shallow
    - Top-down and bottom-up
    - Use volunteering
- Organizational design implications (HR, Budgeting/Finance, vendor management and site strategies)
- Key differences between LeSS and other commercially known scaling approaches (SAFe, Spotify, S@S, Nexus)
- Most classically known pitfalls with scaling

# Good-Old Scrum

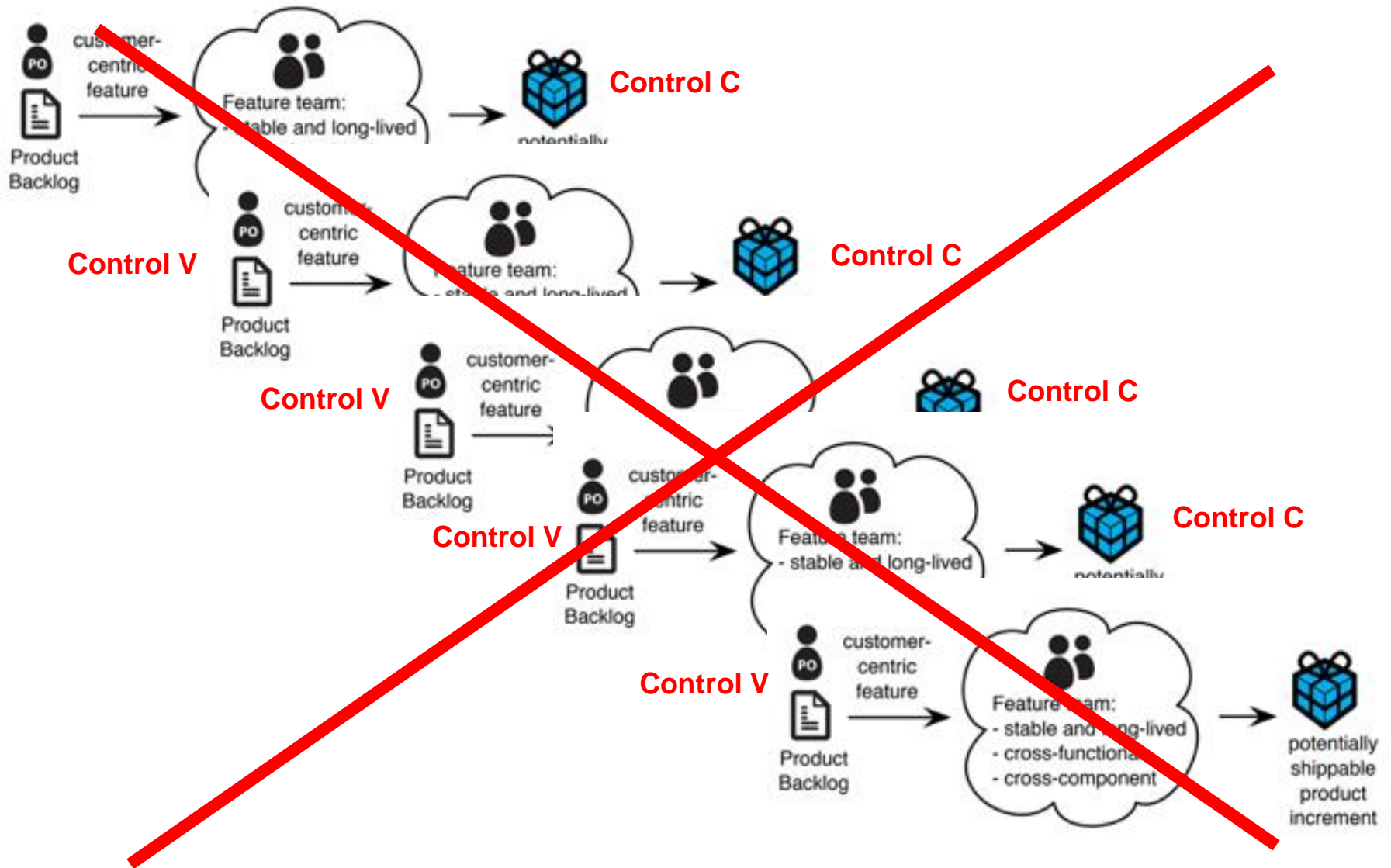


Team has the necessary knowledge and skills to complete an end-to-end customer-centric feature. If not, the team is expected to learn or acquire the needed knowledge and skill.

[www.craiglarman.com](http://www.craiglarman.com)  
[www.odd-e.com](http://www.odd-e.com)

Copyright © 2010  
C.Larman & B. Vodde  
All rights reserved.

# Copy & Paste Is NOT Scaling





# Burying Agility Under Layers of Waste

## Scrum Anti-Patterns

In Scrum, in every Sprint, a team delivers Potentially Shippable Product Increment (PSPI). **This is fundamental for Scrum.** In order for this to happen, each team must possess all necessary attributes (skills, knowledge, domain expertise) required to get work fully DONE (potentially shippable). This is what makes Scrum - real Scrum. Many teams that lack the required Scrum attributes still attempt to sprint, however, effectiveness of such "sprint-like activities" is significantly reduced. **Not all anti-patterns of Scrum are equally obvious.**

- separate, phase-specific backlogs or single backlog with phase-specific items
- local optimization by single-skill specialists (e.g., PM, BA, QA, Architect, Developer)
- hand-overs, toll-gates, "internal contracts"
- long periods of down-time by specialists, when it is not "their phase" to work
- "water-scrum" / "scrum-fall"
- Very weak Definition of Ready & Done
- PSPI - takes many sprints to produce



More obvious anti-pattern  
Less obvious anti-pattern

- separate, component-specific backlogs or single backlog with component-specific items
- local optimization by component specialists (e.g. UI/UX, middle-tier, back-end, web service, architecture)
- hand-overs, toll-gates, "internal contracts"
- multiple non-development sprints needed to integrate all components and fix bugs
- Weak Definition of Ready & Done
- PSPI - takes many sprints to produce



- single, shared, customer-centric backlog
- single, empowered Product Owner
- shared ownership of work, no siloes
- swarming by T-shaped people
- Strong Definition of Ready & Done
- PSPI - every sprint

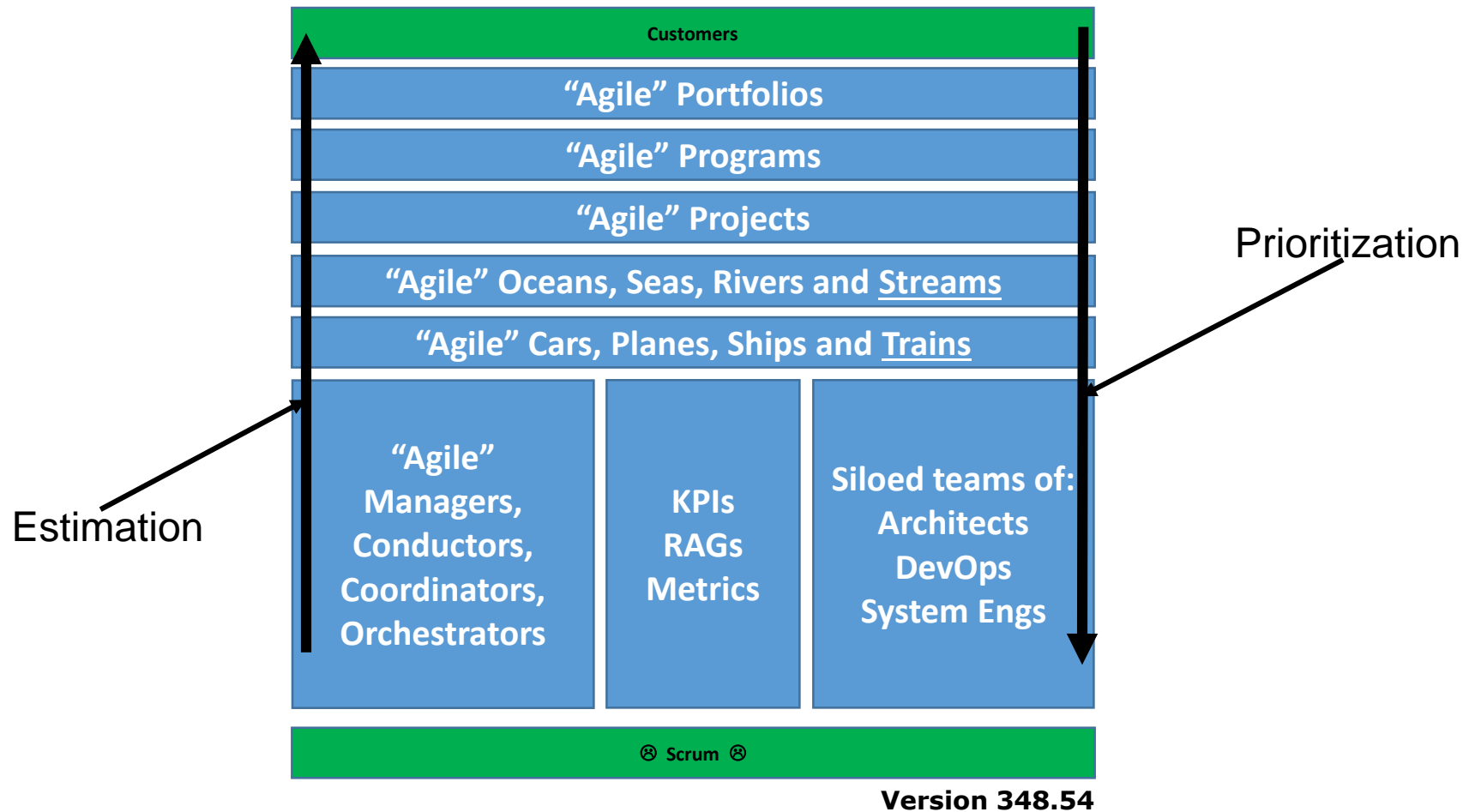


Try This:

Brought to you by [www.keystepstosuccess.com](http://www.keystepstosuccess.com)

KSTS  
Consulting

# Typical Scaling Mistakes



- Where is customer – centricity?
- Where is a feedback loop between customers and developers?
- Where is real organizational agility/adaptiveness?
- What has changed from what you saw prior to “installation”?



# Typical Scaling Mistakes

## *Birth Certificate*

This Certifies That

<hr/>	
(Name)	
<hr/>	<hr/>
(Mother)	(Father)
<hr/>	<hr/>
(Weight)	(Length)
<hr/>	
(Location)	
<hr/>	<hr/>
(City)	(State)
<hr/>	<hr/>
(Signed)	(Signed)

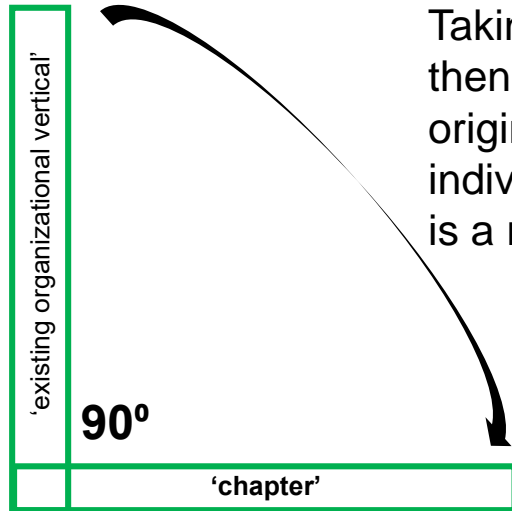
---

**(#@&e Certification Number)**

---

**(#@&e Version)**

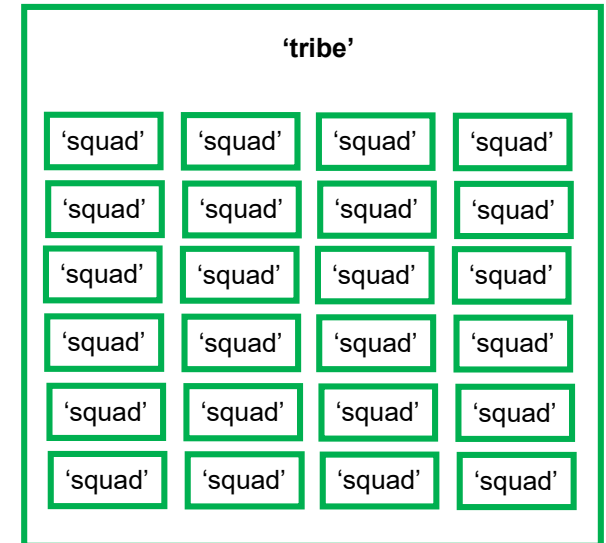
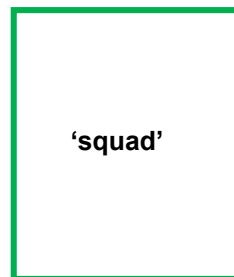
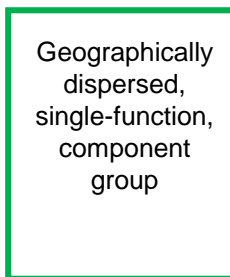
# Typical Scaling Mistakes



Taking an existing organizational vertical, flipping it on its side, and then calling it a 'chapter', while preserving all elements of the original structure (reporting lines, management relationships, individual performance by 'chapter leads', mandatory participation) - is a **masquerade**

## Re-labeling

Relabeling large, geographically dispersed, single-function specialist/component group, managed by engagement managers and technical leads (with groups, frequently coming from external vendors, bound by fixed-everything SOWs) into a 'squad' - is a **masquerade**



Taking an existing *portfolio* of *programs* and *projects* (usually, they weakly support customer-centric products) that are worked upon by used-to-be component groups, and now, so called 'squads', while relabeling a portfolio into a 'tribe' - is a **masquerade**

# Typical Scaling Mistakes

## Client Company

Engagement Lead

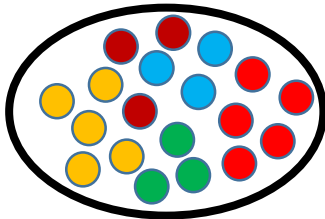
Engagement Lead

Engagement Lead

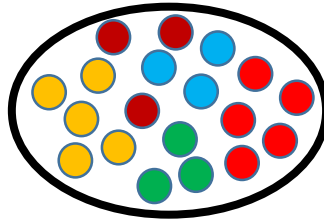
Engagement Lead

Engagement Lead

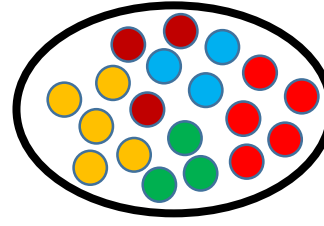
...managing and providing estimates 'on behalf of' their people - to a client company...



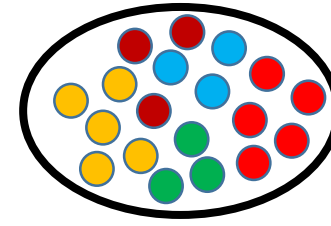
"team"



"team"



"team"



"team"

...somewhere in a cloud, doing mandate laborious work...

Vendor A

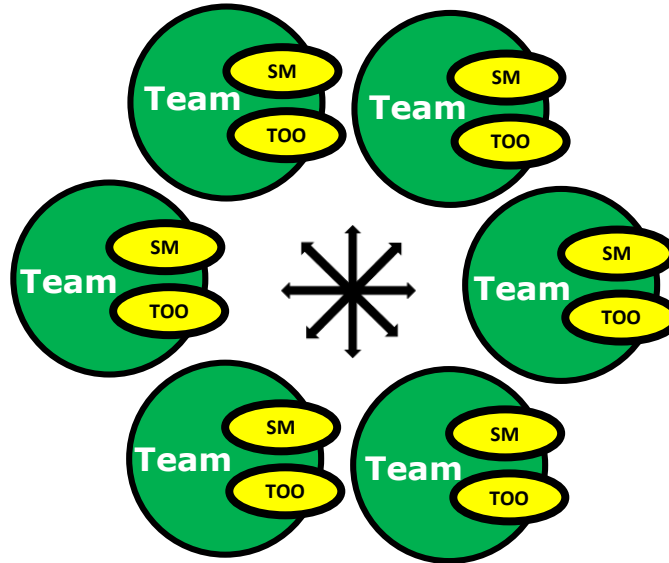
Vendor A

Vendor A

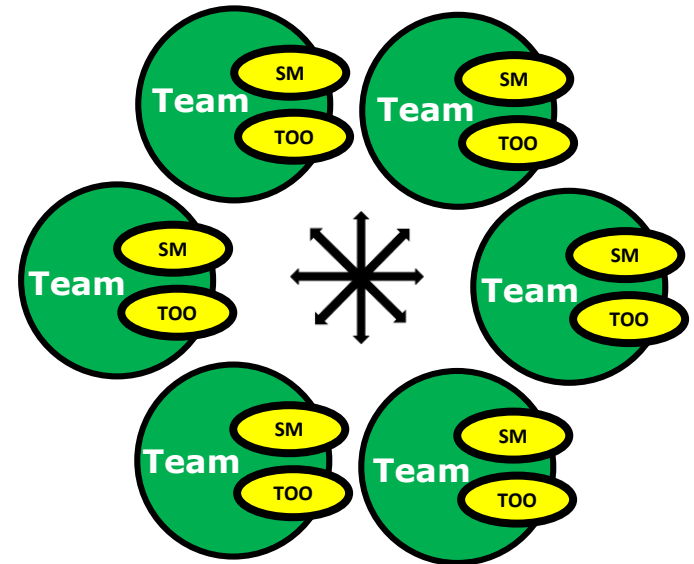
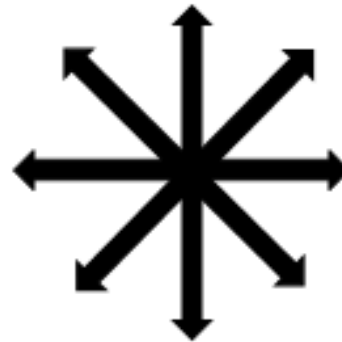
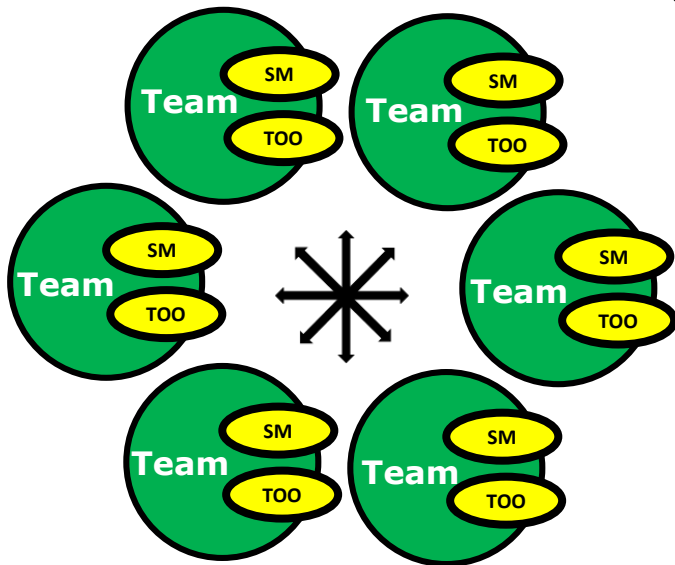
Vendor A

Vendor A

# Typical Scaling Mistakes

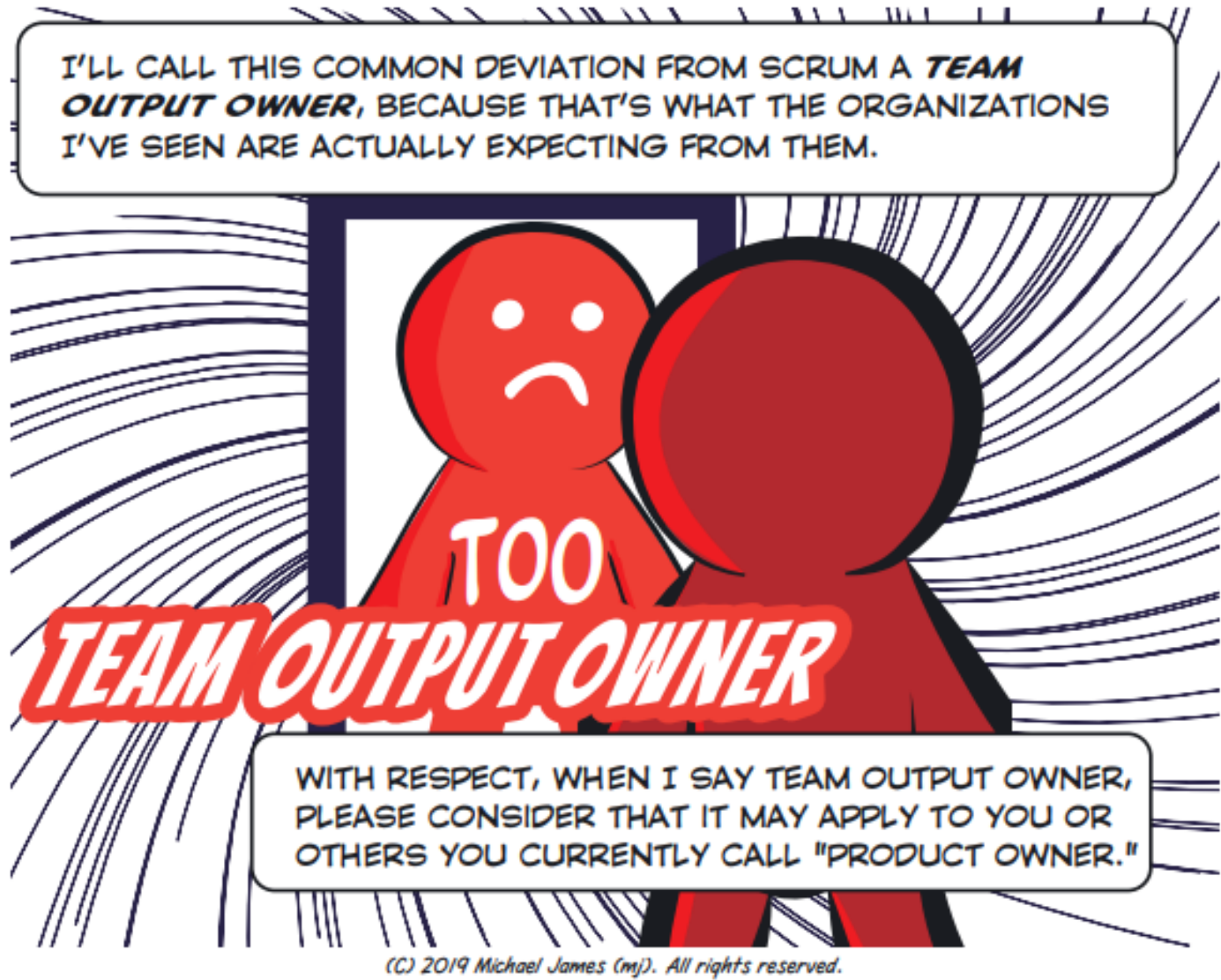


**Btw,  
who is  
TOO?**



**Each team: private backlog, dedicated TOO, SM – estimating in a silo**

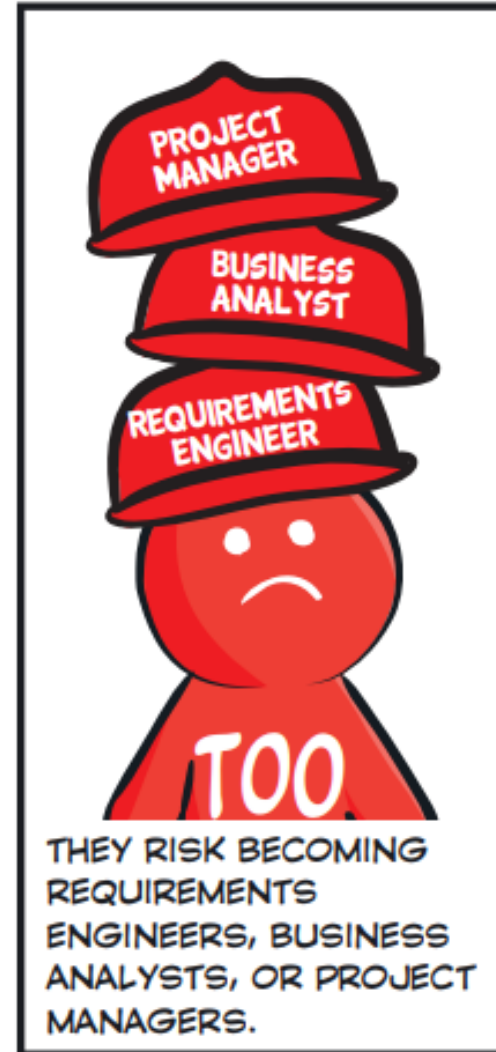
# Who is Team Output Owner (TOO)?



4

<https://seattlescrum.com/downloads/Why-Scrum-Isnt-Making-Your-Company-Very-Agile-v2.pdf>

# Who is Team Output Owner (TOO)?



(C) 2019 Michael James (mj). All rights reserved.

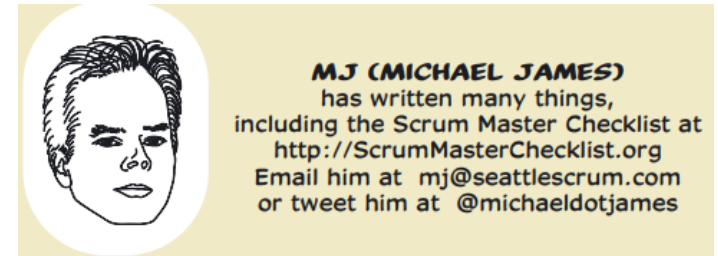
18



# Who is Team Output Owner (TOO)?

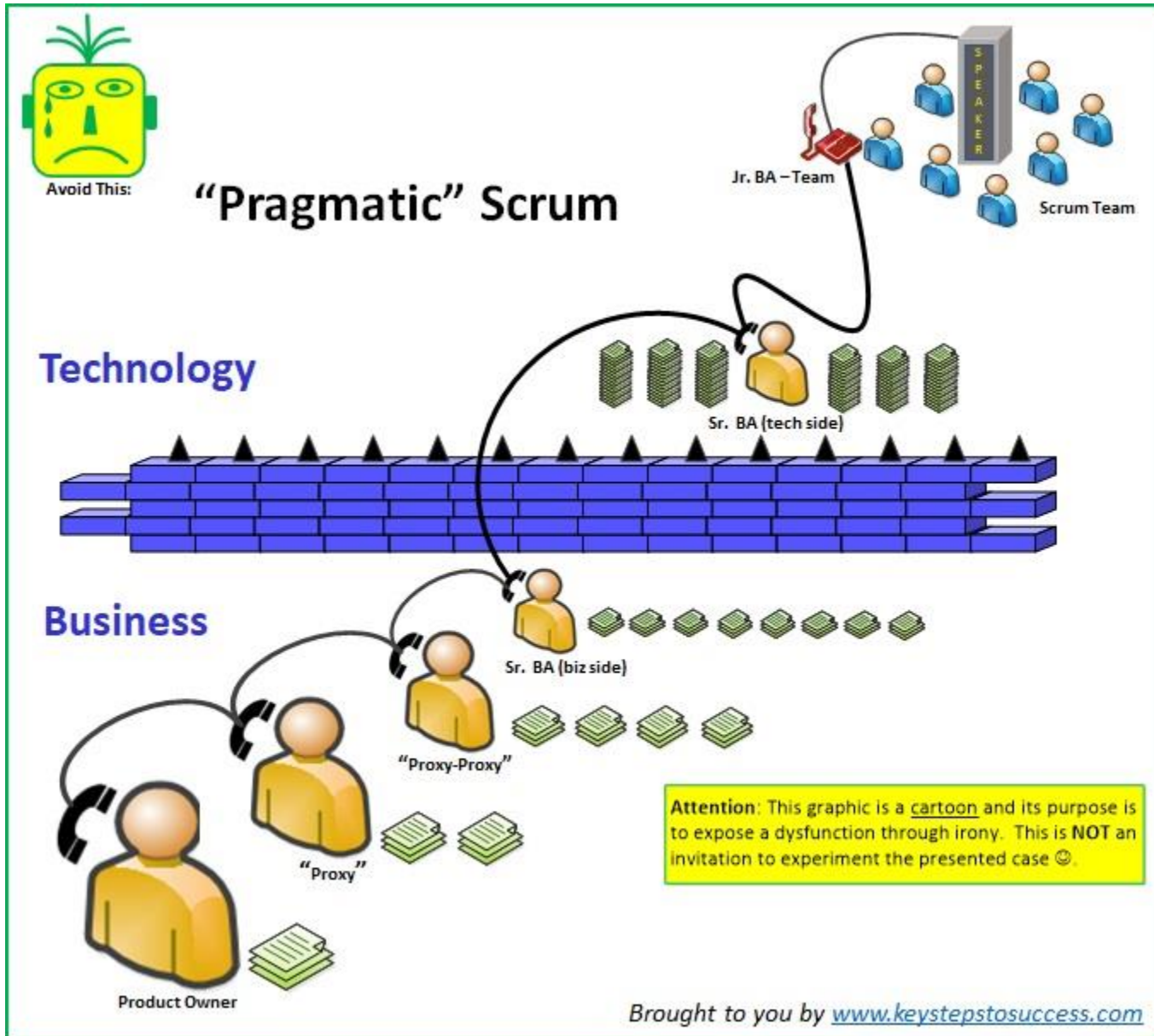


Why "Scrum" Isn't Making Your Organization Agile: Harmful Misconceptions About Product Owner Role



<https://www.youtube.com/watch?v=cr2rjaGmUzo>

# DO NOT



# DO NOT

## Scrum Blind Date [with Mask Show]

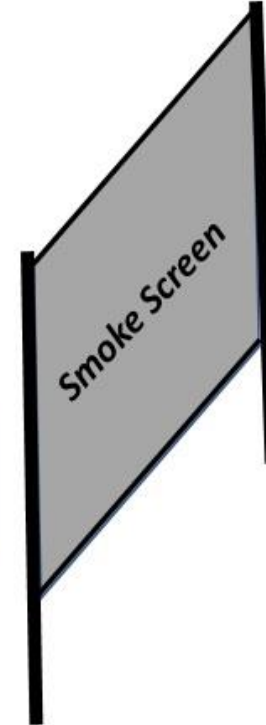


Scrum Team

...while **Team** wishes to have a real **Scrum Master** (servant-leader, promoter and supporter of Scrum process), AND a real **Product Owner** (empowered person, responsible for management and prioritization of Product/Backlog)....



***This is  
What  
Team  
Gets***

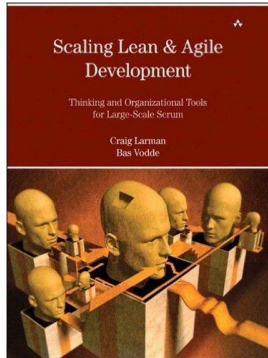


...organization assigns any historically abundant, but frequently unsuitable candidates to fulfill these two important Scrum roles, and by doing so, adds challenges, dysfunctions and inefficiencies to the Scrum process... Such classic organizational mistakes, as internal contractual relationships, bureaucracy, overhead and local optimization, find their way into Scrum and adversely impact its dynamics...

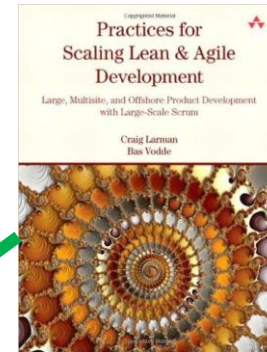
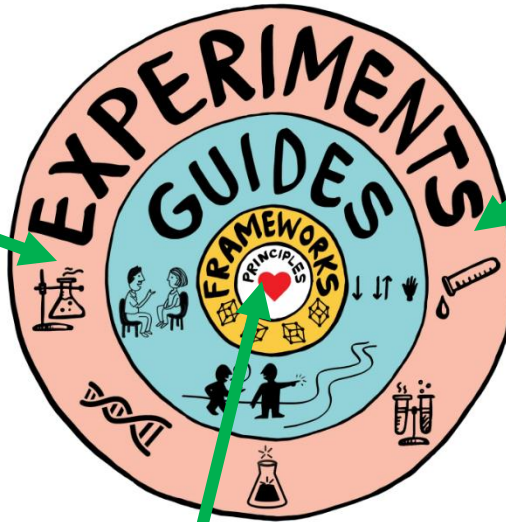
Brought to you by [www.keystepstosuccess.com](http://www.keystepstosuccess.com)



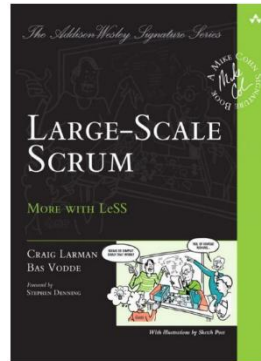
# LeSS – The Big Picture & History



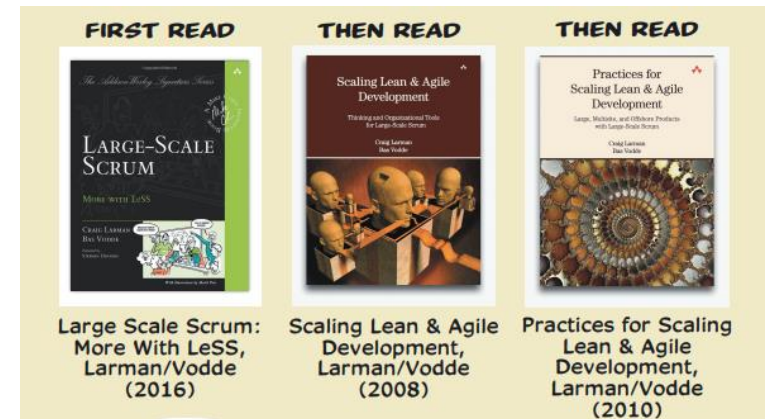
2008



2010



2016



<https://seattlecscrum.com/downloads/Why-Scrum-Isnt-Making-Your-Company-Very-Agile-v2.pdf>



# Important

## Organizational STRUCTURE –

is the 1<sup>st</sup> Order Factor (Variable) that has impact on everything else inside ECOSYSTEM.

*Everything else just follows: culture, mindset, behaviors, norms, values, processes, policies*

*e.g. metrics, velocities, OKRs, KPIs, RAGs, maturity levels – much less relevant*

This is what some less experienced agile coaches do not stress, when they coach senior management 😞



# DO NOT (1-4)

## Larman's Laws of Organizational Behavior

([Russian translation here](#))

After decades of observation and organizational consulting, here are *Larman's Laws of Organizational Behavior*. These are observations rather than laws to follow ;)

- 1. Organizations are implicitly optimized to avoid changing the status quo middle- and first-level manager and “specialist” positions & power structures.**
- 2. As a corollary to (1), any change initiative will be reduced to redefining or overloading the new terminology to mean basically the same as status quo.**
- 3. As a corollary to (1), any change initiative will be derided as “purist”, “theoretical”, “revolutionary”, “religion”, and “needing pragmatic customization for local concerns” — which deflects from addressing weaknesses and manager/specialist status quo.**
- 4. As a corollary to (1), if after *changing the change* some managers and single-specialists are still displaced, they become “coaches/trainers” for the change, frequently reinforcing (2) and (3).**

### **5. Culture follows structure.**

Or, *Culture/behavior/mindset follows system & organizational design*. i.e., If you want to really change culture, you have to start with changing structure, because culture does not really change otherwise. By the way, this is an observation in large-scale; in small start ups, it's the reverse: structure follows culture (org design follows mindset).

And “culture follows structure” (in large scale) is why deep systems of thought such as *organizational learning* are not very sticky or impactful by themselves in large scale, and why systems such as Scrum (that have a strong focus on structural change at the start) tend to more quickly impact culture — if the structural change implications of Scrum are actually realized.

I discovered that the well-known systems-thinker/advocate John Seddon also observed this: “*Attempting to change an organization's culture is a folly, it always fails. Peoples' behavior (the culture) is a product of the system; when you change the system peoples' behavior changes.*”

# UNDERSTAND

*It is vital to appreciate that organizational agility cannot be achieved by a development team in isolation -- it is a system challenge for organizational redesign. Especially when you are interested in LeSS within an R&D department of thousands, where each product group may have 200 or 700 people distributed in two or five sites around the world. If an engineering team has the technical capacity to adapt or change quickly, but requirements management, legal practices, product management, HR policies, site strategies, and deployment processes all emphasize rigidity, conformance to original plans, conformance to the status quo, and slow practices, then how can there be real agility?*

Source: <https://www.scrumalliance.org/community/spotlight/craig-larman/june-2015/less-agile-or-less-agile>



<http://www.keystepstosuccess.com/2017/08/agile-organization-as-a-sushi-roll/>

# DO

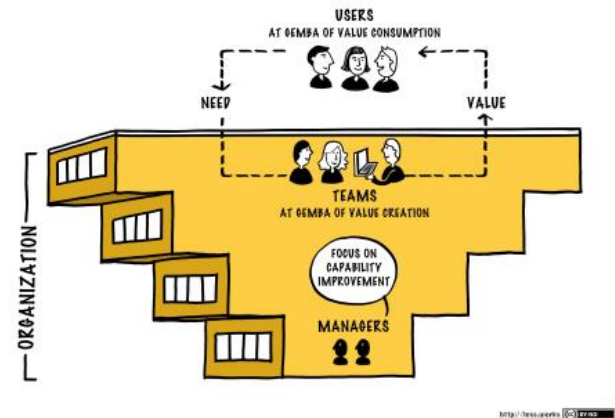


# UNDERSTAND

## Three (3) Adoption Principles of LeSS

➤ Deep and narrow over broad and shallow

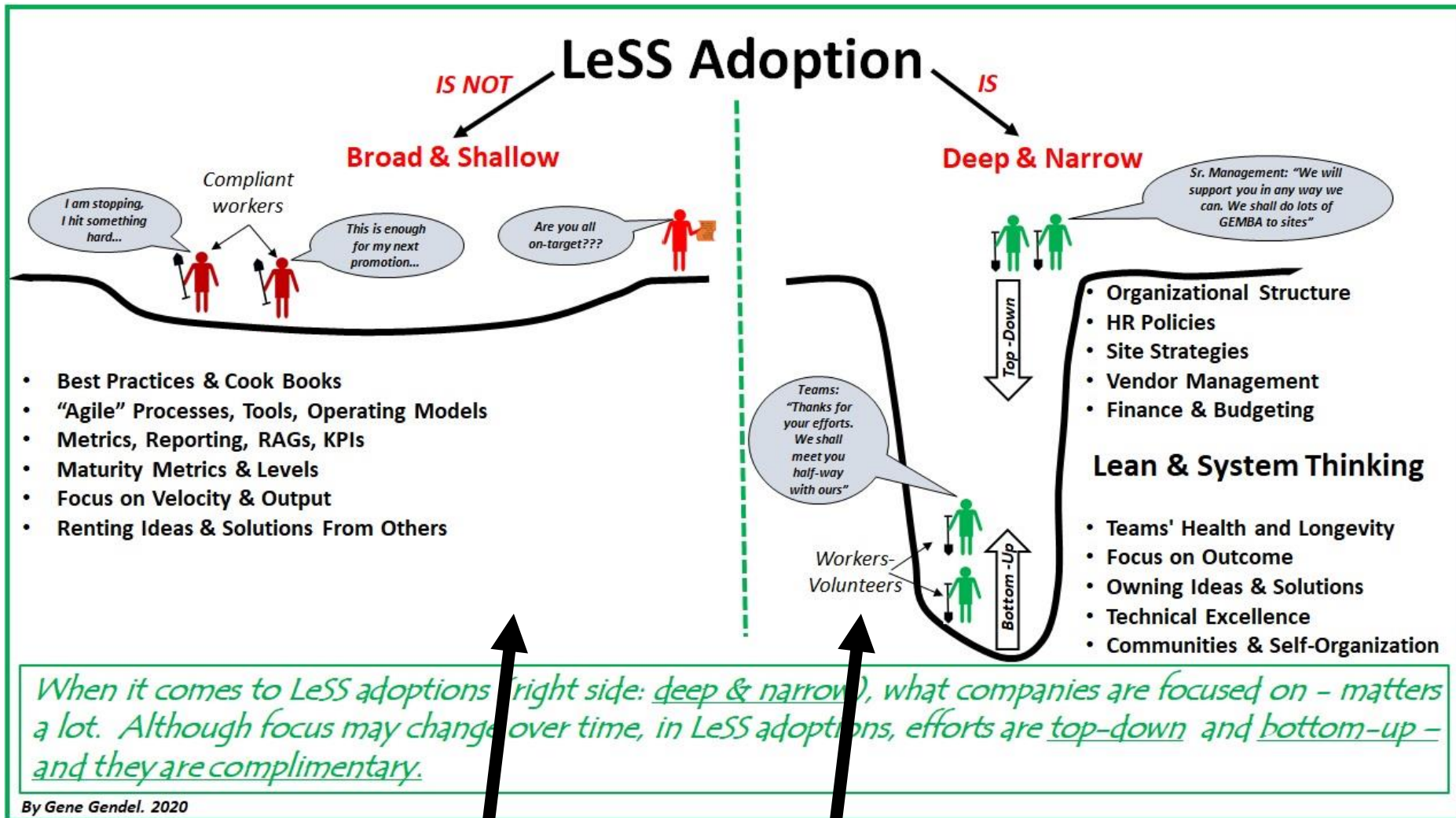
➤ Top-down and bottom-up



➤ Use volunteering



# Three (3) Adoption Principles of LeSS



**NO**

**YES**

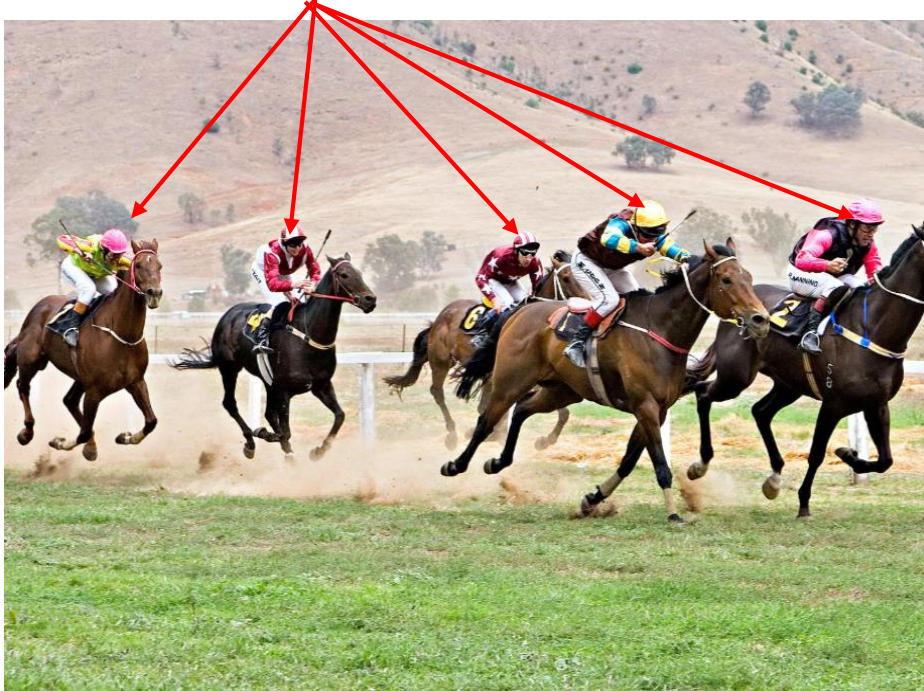


# LeSS Is Scrum with 2-8 Teams

**LeSS is not:** Many teams doing their own Scrum

**LeSS is:** 2-8 teams scrumping together (on same Product)

Different people with strategy, and vision, setting a different course



**NO**

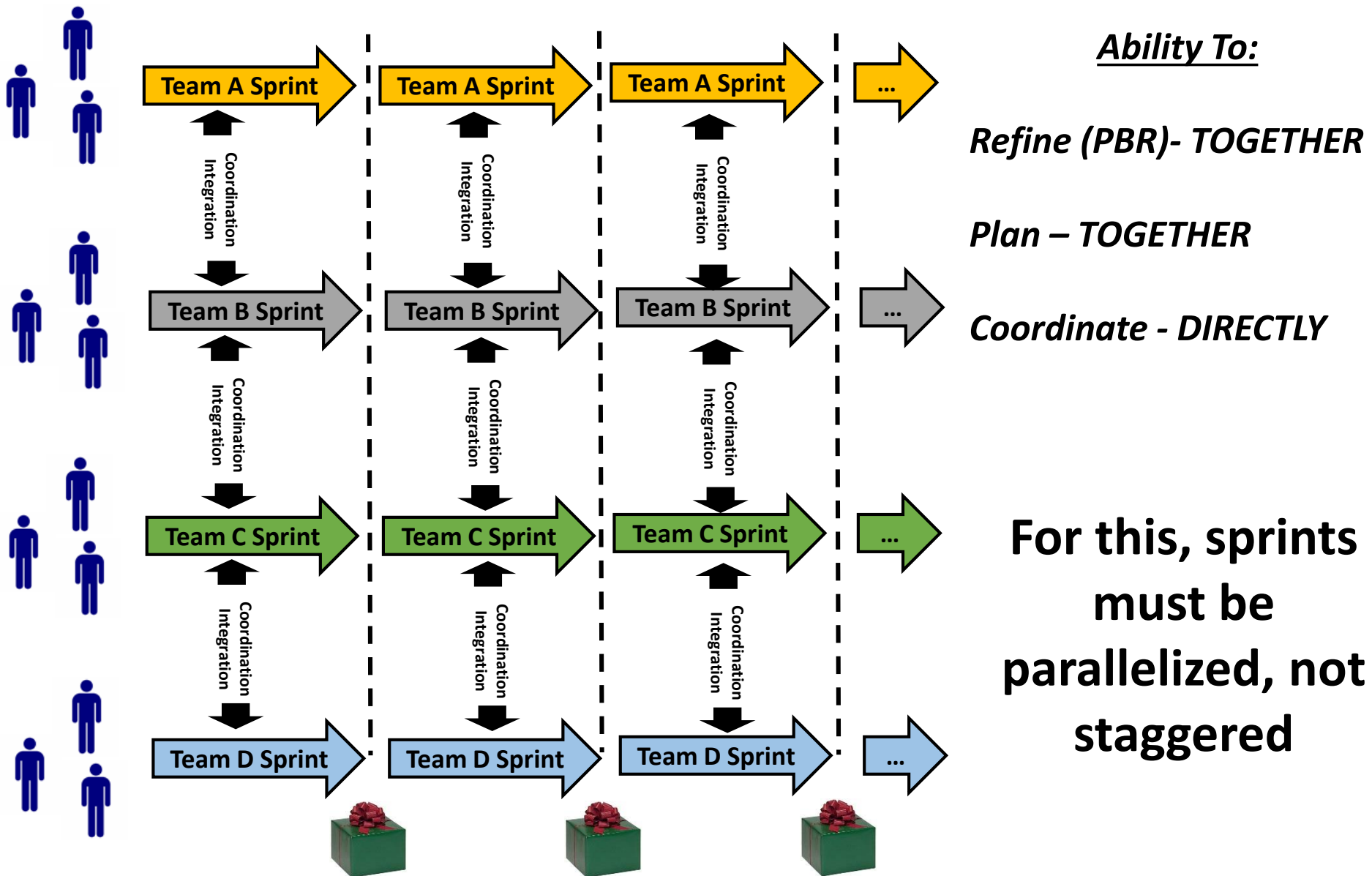
Same person with strategy, and vision, setting a unified course



**YES**



# One LeSS Sprint: 2 – 8 Teams

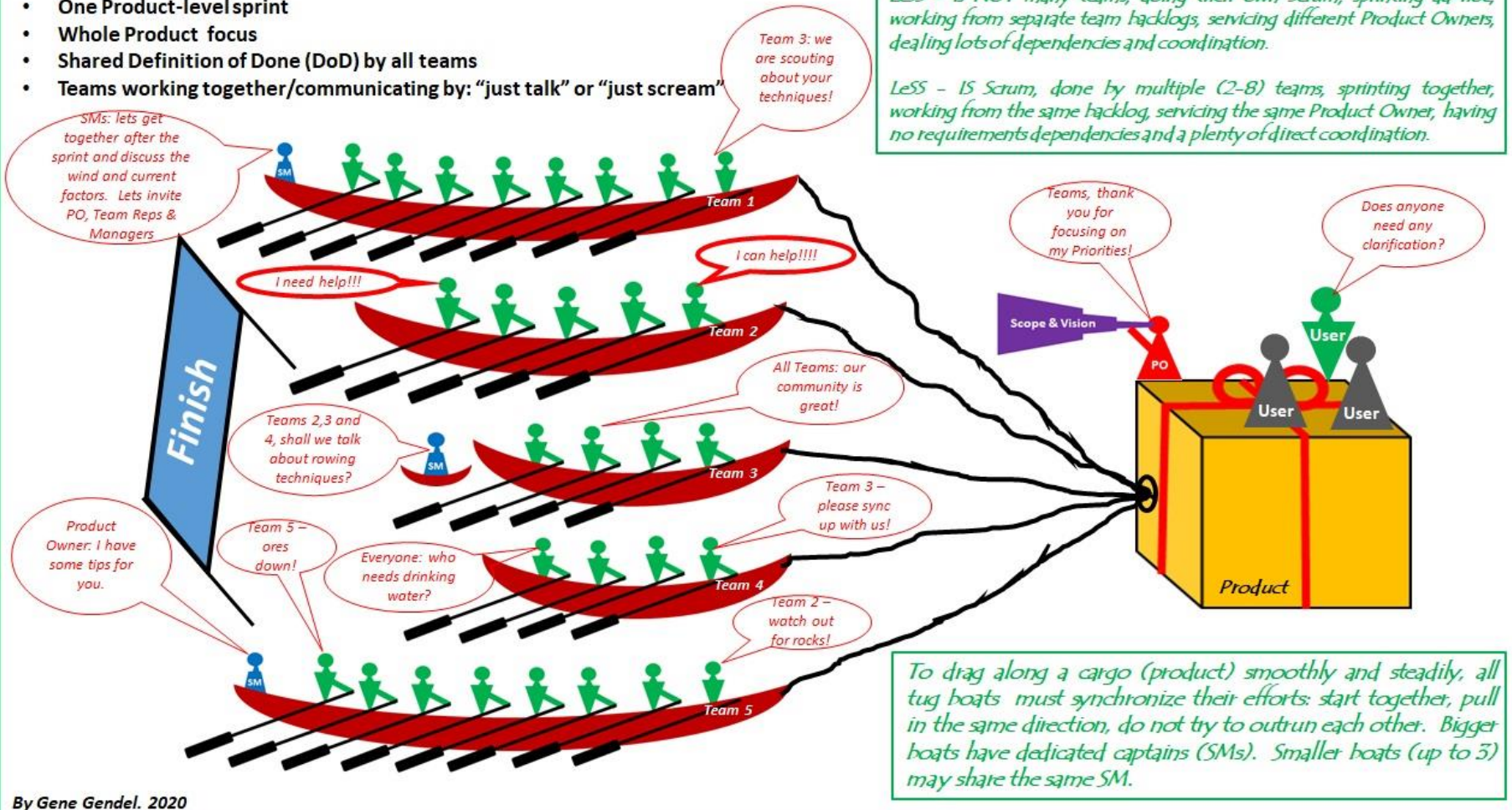


# One LeSS Sprint: 2 – 8 Teams

- One Product-level sprint
- Whole Product focus
- Shared Definition of Done (DoD) by all teams
- Teams working together/communicating by: “just talk” or “just scream”

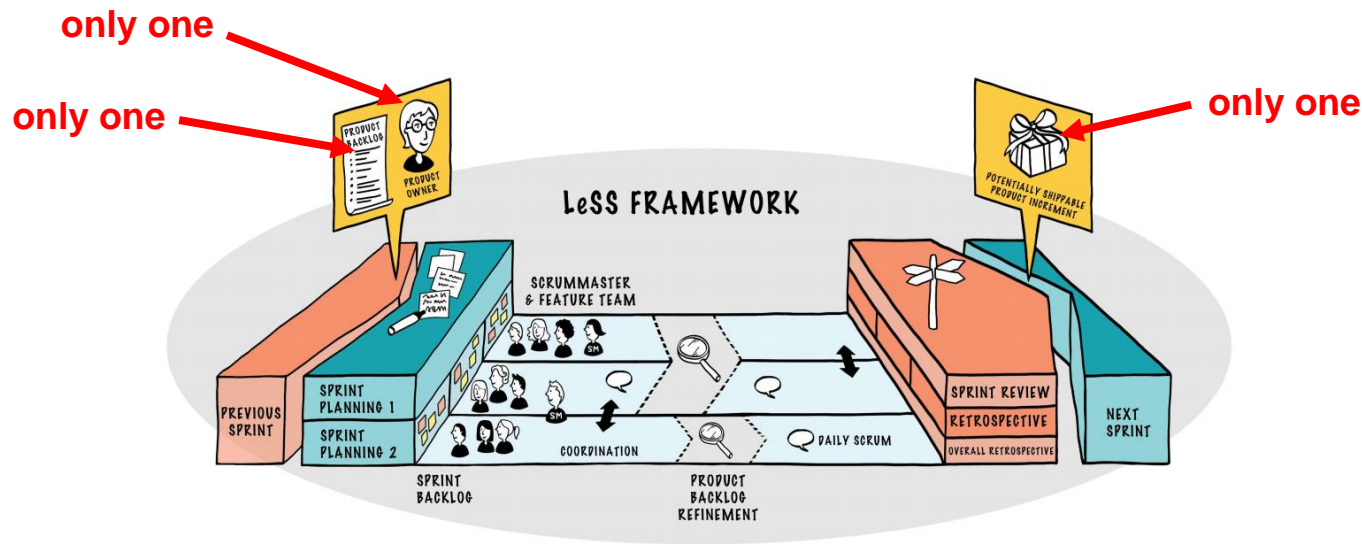
*LESS - is NOT many teams, doing their own Scrum, sprinting ad-hoc, working from separate team backlogs, servicing different Product Owners, dealing lots of dependencies and coordination.*

*LESS - IS Scrum, done by multiple (2-8) teams, sprinting together, working from the same backlog, servicing the same Product Owner, having no requirements dependencies and a plenty of direct coordination.*



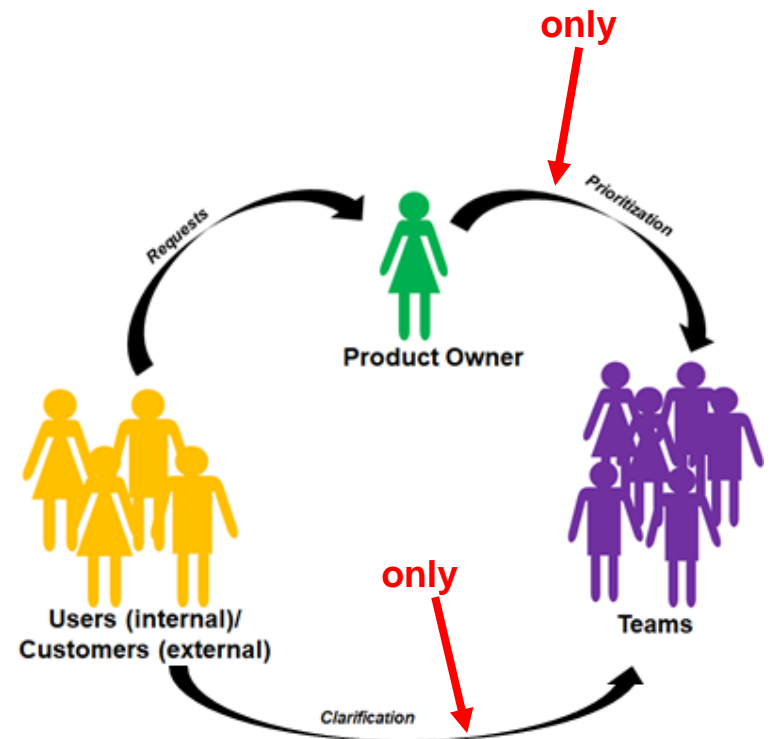
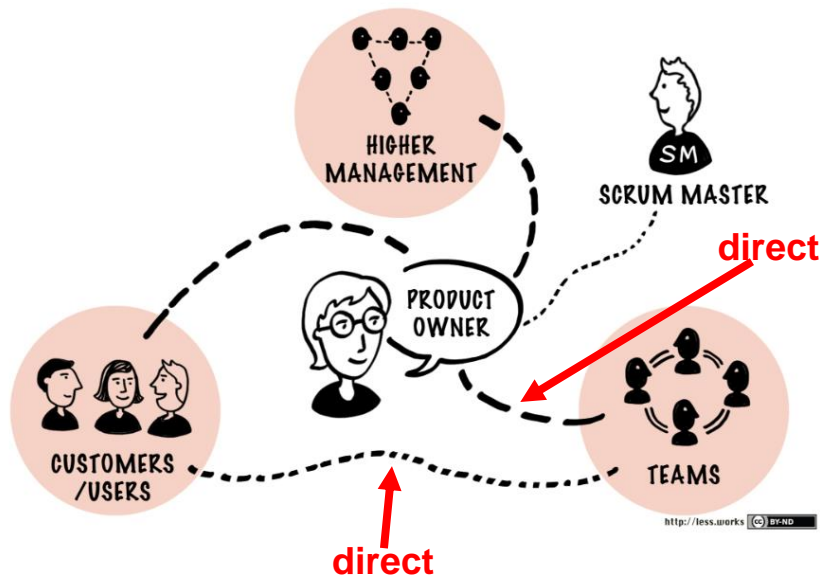
# Size of LeSS Adoption

- LeSS adoptions require months of preparation and then a 'FLIP'.
- LeSS is no more than 70+/- a few, people: don't try to bite more than you can swallow
- 2-8 Teams
- **One Product. One Product Owner. One Backlog**
- **NB:** (LeSS HUGE adoptions are incremental and take years)



<http://less.works> (CC) BY-ND

# Product Owner in LeSS



# Scrum Master in LeSS



## Bad SQL

(in HR Database)

It is amazing if your company, creates the following important titles (marked in red) in its HR database, to support agile roles and careers. However, it is a sign of a deep systemic dysfunction and 'agile masquerade', if the following SQL queries are ran, as a batch job, overnight...🤖

```
UPDATE table_employee  
SET table_employee.title = 'Senior Agile Coach'  
WHERE table_employee.title = 'Senior Project Manager'
```

```
UPDATE table_employee  
SET table_employee.title = 'Scrum Master'  
WHERE table_employee.title = 'Junior Project Manager'
```

```
UPDATE table_employee  
SET table_employee.title = 'Product Owner'  
WHERE table_employee.title = 'Business Analyst'
```

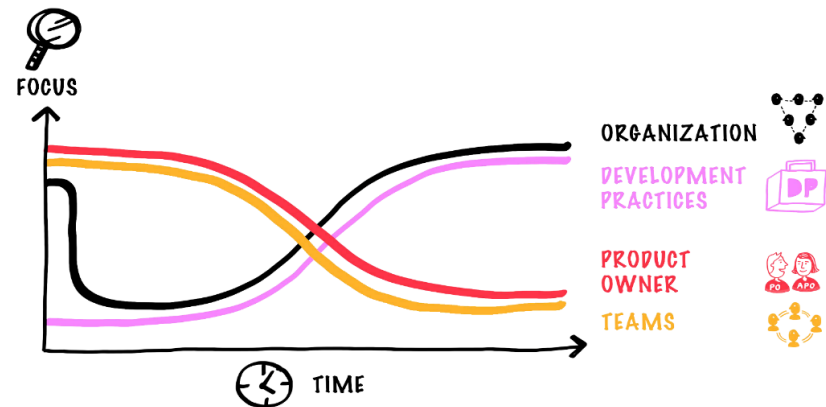
Brought to you by [www.keystepstosuccess.com](http://www.keystepstosuccess.com)

NO

YES

## Dedicated full-time role

In LeSS, the ScrumMaster role is vital. We've seen many organizations try part-time ScrumMasters, which usually leads to no ScrumMasters at all. This then affects the LeSS adoption enormously. In LeSS the ScrumMaster is a dedicated, full-time role in the same way that being a Scrum Team member is a dedicated, full-time role. Having said that, it is possible for one full-time ScrumMaster fill the role for up to three teams, depending on any number of factors.



<http://less.works> CC BY-ND



# Scrum Master in LeSS

## Misunderstood

The ScrumMaster role is a new one and often not understood by teams and organizations who are adopting Scrum. A frequent response is to make the “leftover people” the ScrumMasters. They might be nice people but often lack the right skills, motivation, and Scrum knowledge to be effective ScrumMasters. They morph the role into something else which then becomes accepted within the organization as the way a ScrumMaster should be. After all, the ScrumMaster should know if they’re doing things correctly, right? Sometimes well-meaning ScrumMasters who are new to Scrum or not a good fit for the ScrumMaster role cause things to happen that are actually counter to Scrum and detrimental to Scrum adoption, thereby transforming them into anti-ScrumMasters.

**Is there Career Path for F/T Scrum Master?**

## Dedicated full-time role

In LeSS, the ScrumMaster role is vital. We’ve seen many organizations try part-time ScrumMasters, which usually leads to no ScrumMasters at all. This then affects the LeSS adoption enormously. In LeSS the ScrumMaster is a dedicated, full-time role in the same way that being a Scrum Team member is a dedicated, full-time role. Having said that, it is possible for one full-time ScrumMaster fill the role for up to three teams, depending on any number of factors.

*Sourced from less.works*



# Team In LeSS

## Types of teams


Setting overall direction				
Designing the team and its organizational context	Management Responsibility			
Monitoring and managing work process and progress		Team's Own Responsibility		
Executing the team task				
	Manager-led teams	Self-Managing teams	Self-Designing teams	Self-Governing teams

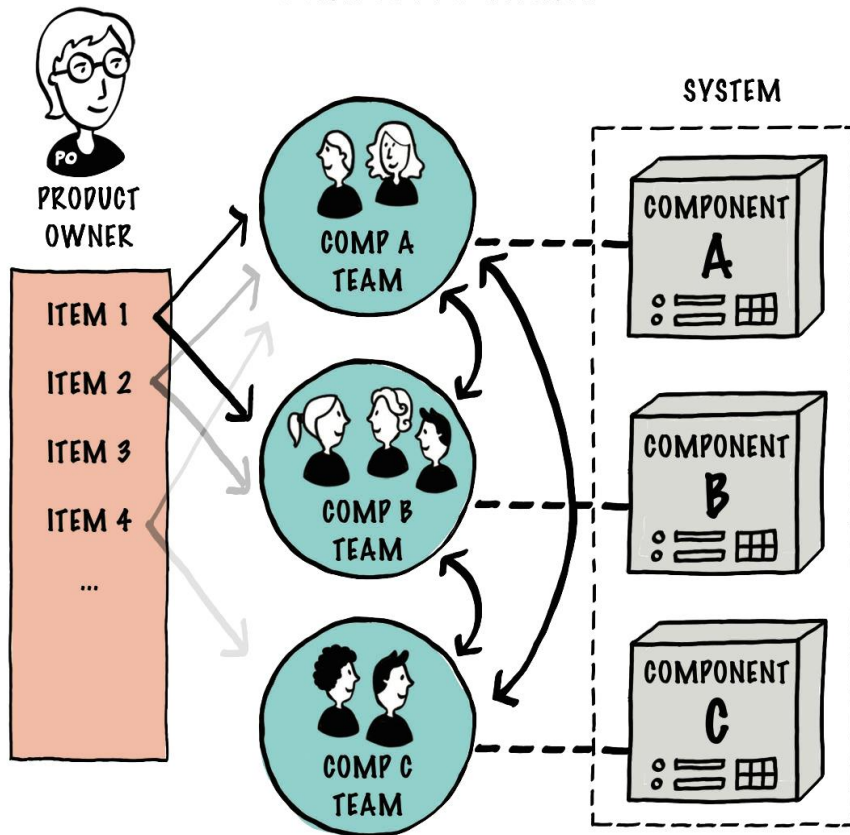
Figure 1. Types of Teams.

Source: <https://less.works/resources/graphics/index.html>

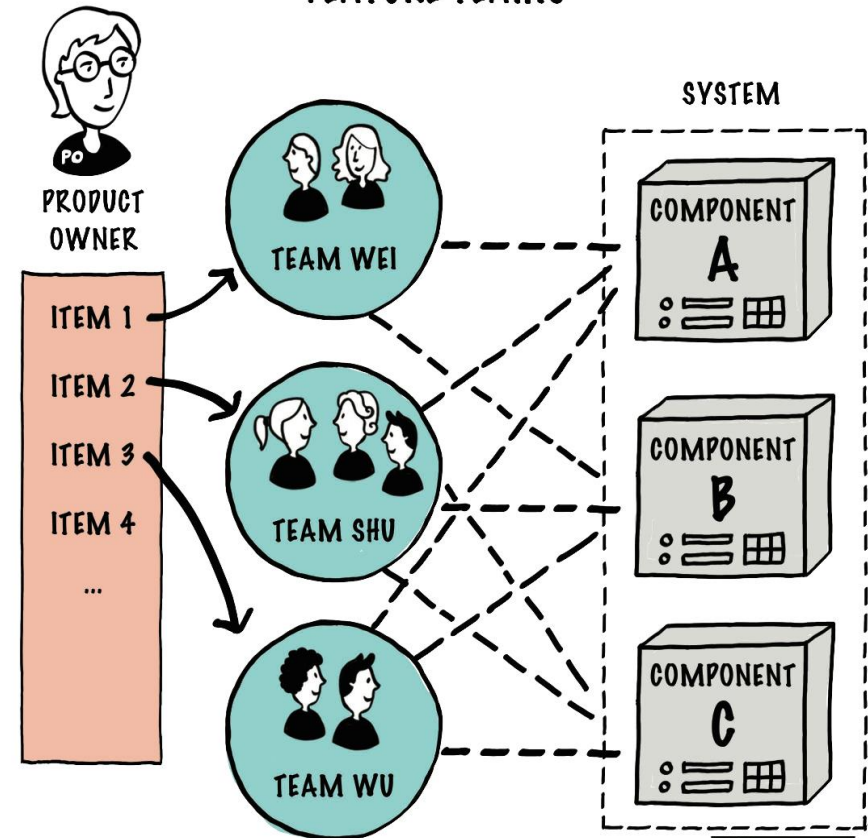
# Team In LeSS



## COMPONENT TEAMS

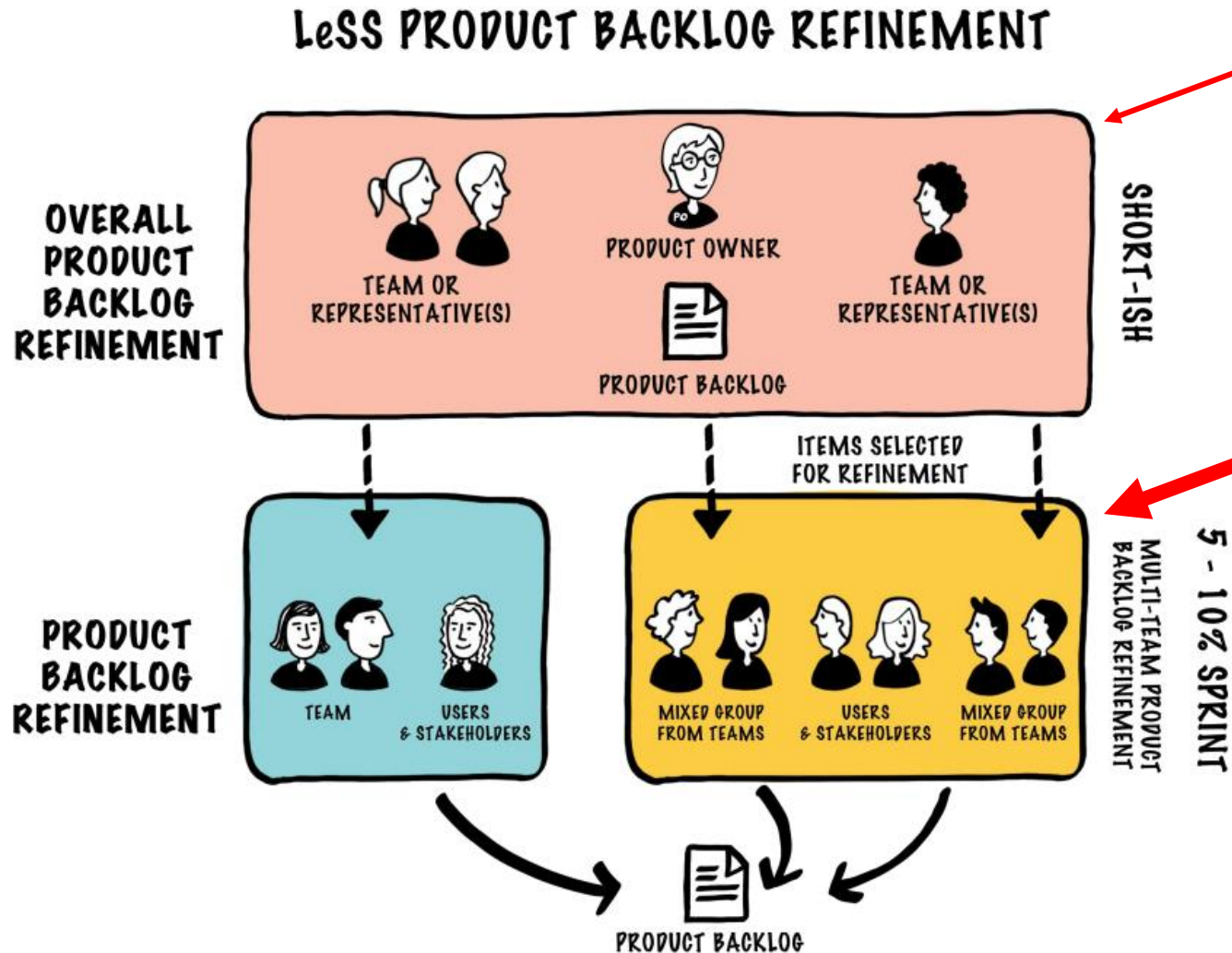


## FEATURE TEAMS



<http://less.works> BY-ND

# Opportunities For Coordination In LeSS



# Opportunities For Coordination In LeSS

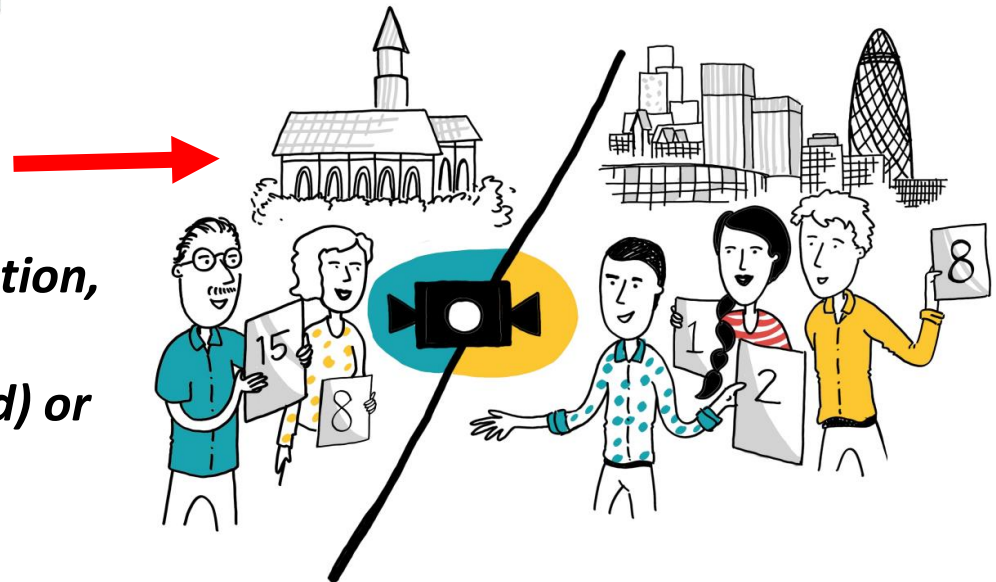
## Multi-team Product Backlog Refinement



- *Open space (physical or virtual)*
- *“Promiscuous” collaboration*
- *Directly, with users and customers*

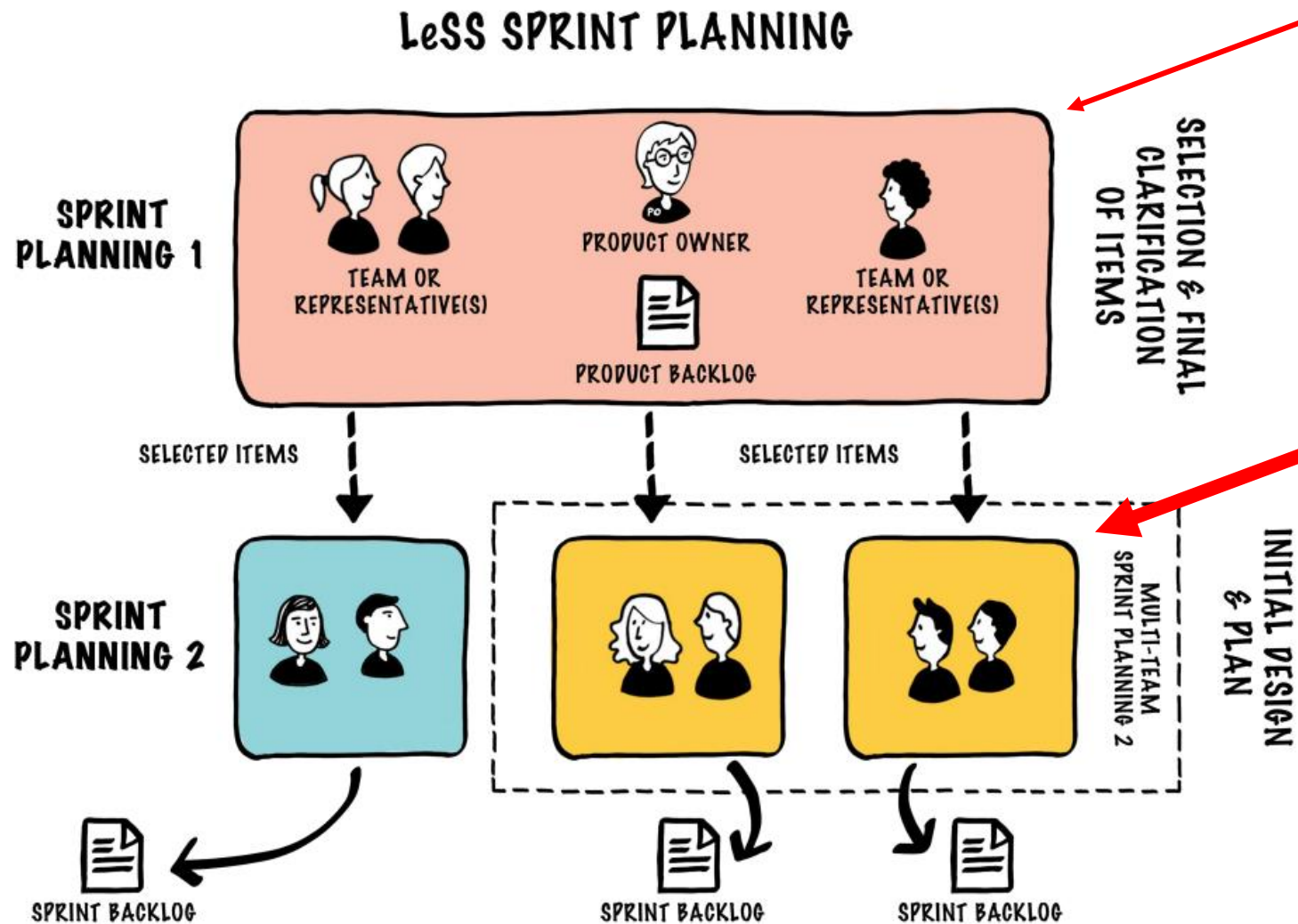
## Multi-site estimation with Planning Poker

- *By “doers” (team members)*
- *Everyone is involved*
- *The main goal: CCC (Card, Conversation, Confirmation)*
- *No need to be too conservative (pad) or aggressive, while estimating*



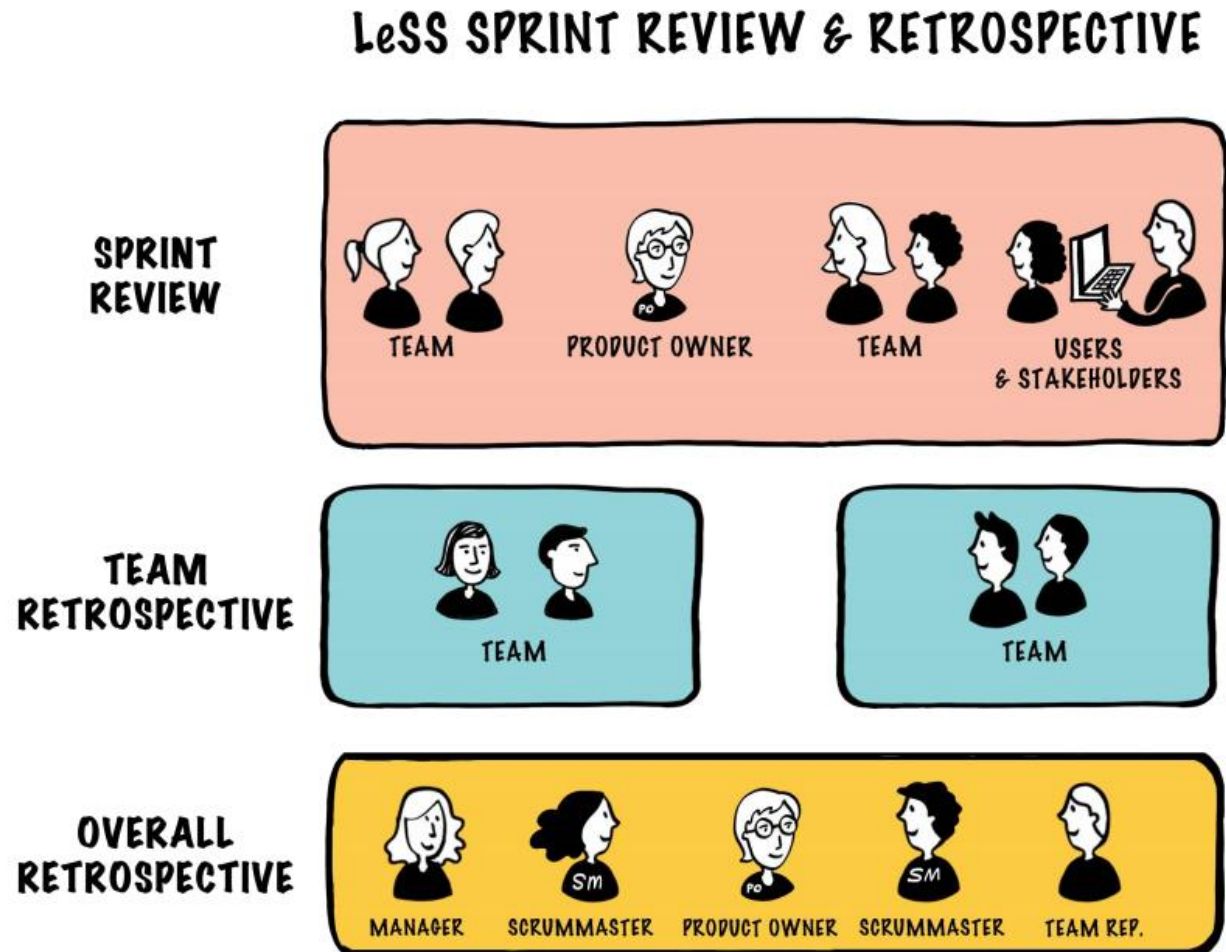


# Opportunities For Coordination In LeSS



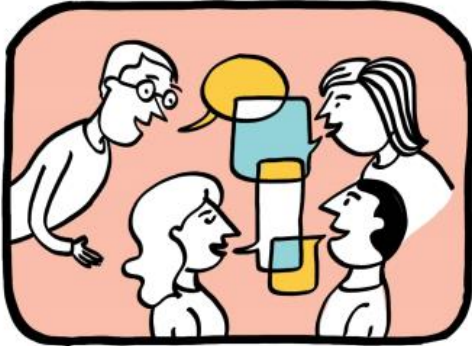


# Opportunities For Coordination In LeSS

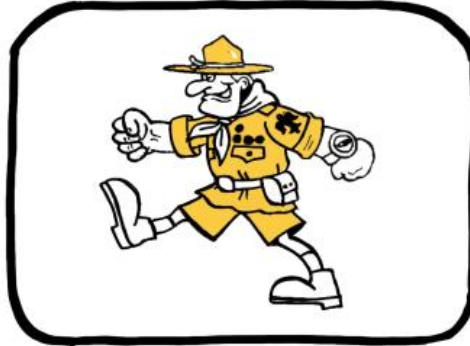


<http://less.works> CC BY-ND

# Opportunities For Coordination In LeSS



JUST TALK



SCOUT



OPEN SPACE



TRAVELER



COMMUNITIES



COMPONENT MENTOR

# APPENDIX

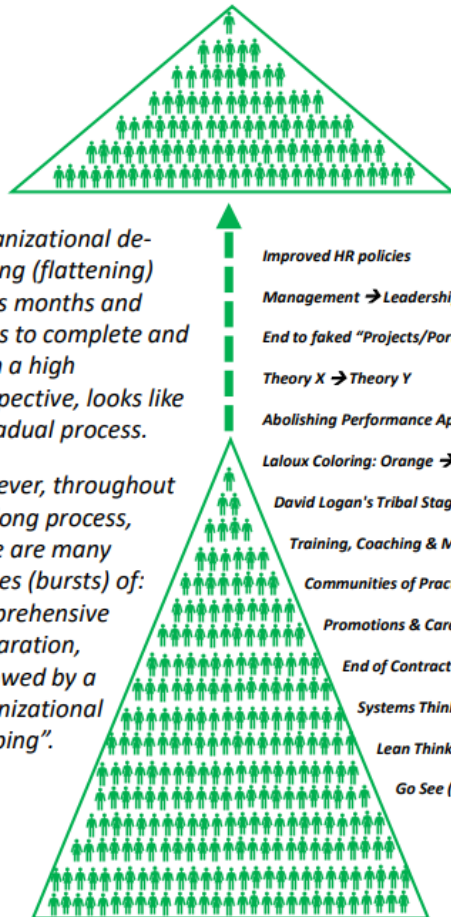
# Less is More

## Scaling Organizational Adaptiveness (a.k.a. “Agility”) with Large Scale Scrum (LeSS)

Organizational De-Scaling / Flattening

Organizational de-scaling (flattening) takes months and years to complete and from a high perspective, looks like a gradual process.

However, throughout this long process, there are many phases (bursts) of: comprehensive preparation, followed by a organizational “flipping”.



Improved HR policies  
Management → Leadership  
End to faked “Projects/Portfolios”  
Theory X → Theory Y  
Abolishing Performance Appraisals  
Laloux Coloring: Orange → Green  
David Logan’s Tribal Stage: 3 → 4  
Training, Coaching & Mentoring  
Communities of Practice  
Promotions & Career Paths  
End of Contract Game  
Systems Thinking  
Lean Thinking  
Go See (Gemba)

Diagram created by Gene Gendel

### LeSS

**Simplified** (flatter) and **inverted** (senior leadership is in supportive function) **organizational design**. **System Optimization**. Reduction of: silos, hand-overs. Scrum is implemented by coordinated, feature-centric teams (2-8), building the same, widely defined Product/serving the same Product Owner. Scrum is the main building block of IT org. structure. Teams are collocated. Teams strive to ‘own’ their decisions, not ‘rent’ from others. Adoption of LeSS is “**deep & narrow**”, not “**broad and shallow**”; no Big Bang transformation attempts.  
**Multi-site development** - used for multiple locations. Reliance of technical **Mentoring** and **Communities**. No subsystem code ownership. Reduction of ‘undone’ work. Focus on **Customer value**. Support of Senior Leadership. **Involvement of HR**, supporting the concept of **job security, not role security**.

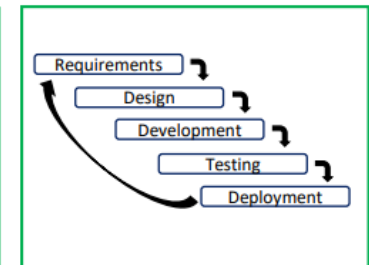
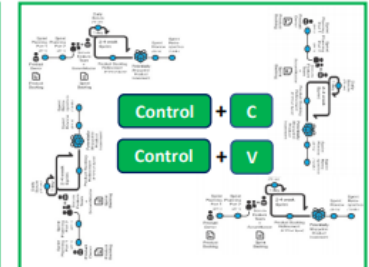
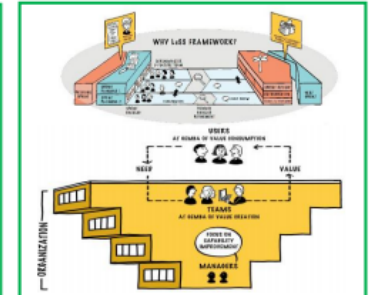
*Transition from independent basic Scrum, performed by multiple teams to LeSS, is a desirable approach, when a product is widely defined and real Product Owner is identified*

### Scrum

**Copy-paste scaling** (no conscious scaling strategy) of Scrum throughout an organization: many teams doing their ‘own’ Scrum. Frequently manifested, when there is an internal competition between departments or company areas, on e.g. “*who is going to have more scrum teams by e.g. end-of-year*”. True product definition is weak. Cases of using **Scrum for component-centric development** are frequent (often, the result of trying to meet goals of agile transformation (% annually), set at enterprise level. This results in massive integration problems that need to be resolved by adding “*integration/stabilization*” sprints. Importance of **Scrum dynamics and roles** is viewed as **secondary**, to existing organizational structures and blueprints. Too many single-specialty experts and very few T-shaped workers. **No meaningful HR changes**.

### Waterfall

**Complex organizational design**. Domains of single-function expertise, ownership and control. Numerous silos, hand-offs and translational layers between component teams, single-function specialty groups, and technology and business, overall. **Internal contracts and blame-shifting** strongly prevail. Long cycle “from concept to cash”. **Local optimization** by single-specialty workers/departments. **Theory X management** – is the primary way to manage/treat individuals. **HR** – putting a lot of emphasis on performance appraisals and individual ownership; manipulating workers with **subjective bonuses**; fostering an environment of **internal competition** and “*I am the best and you are not*” attitude. Weak definition of product value, from a stand-point of a paying customer. Manifestation of **Larman’s Laws of Organizational Behavior**.



Graphics are courtesy of <https://less.works>

Scaling Scrum

Waterfall

Source: [http://www.keystepstosuccess.com/wp-content/uploads/2018/02/scrum\\_scaling\\_org\\_descaling-1.pdf](http://www.keystepstosuccess.com/wp-content/uploads/2018/02/scrum_scaling_org_descaling-1.pdf)

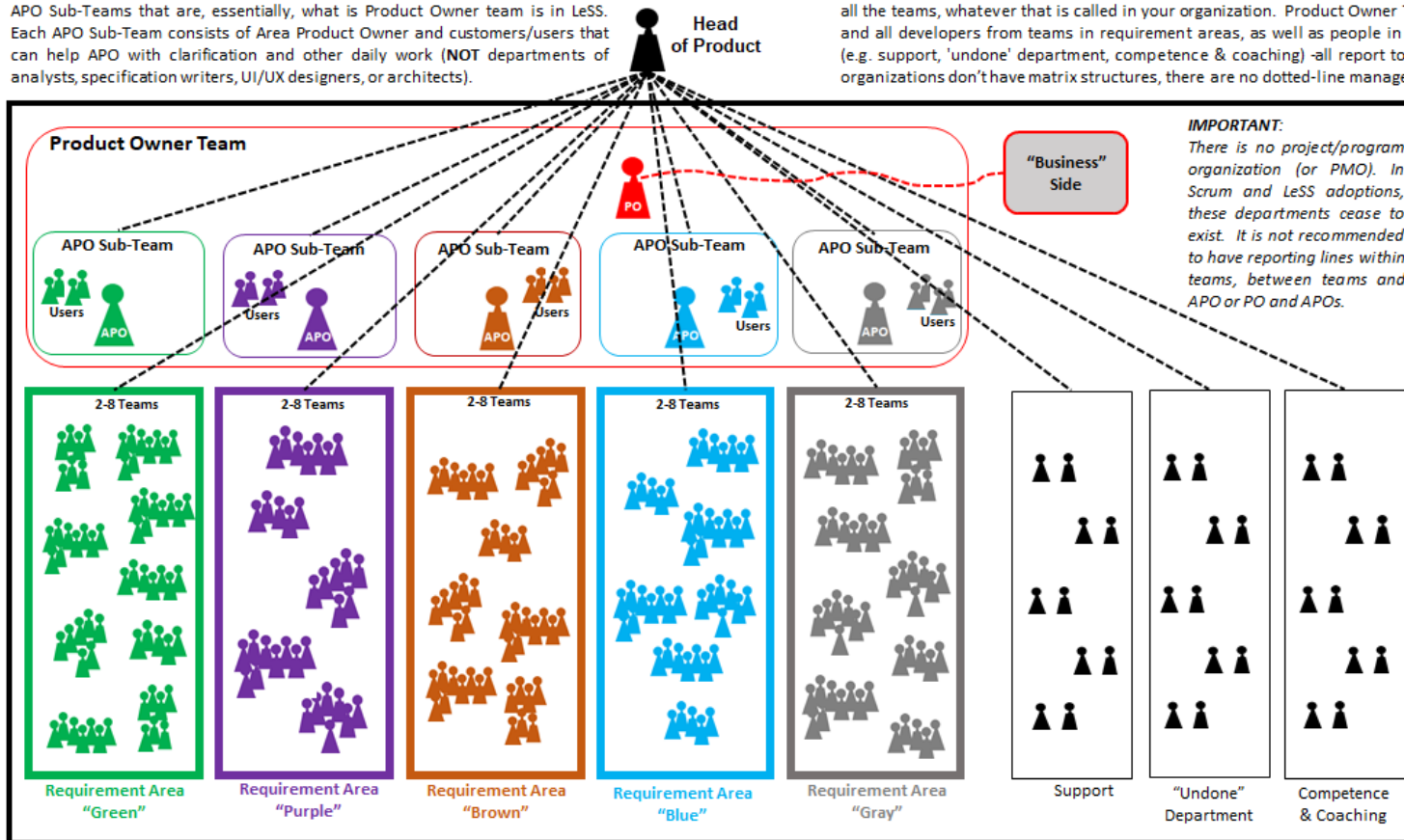


# Relationships in LeSS

**Guide: TRY.... Keeping LeSS Huge Organizational Reporting Structure As Flat and Simple As Possible**

**Product Owner Tam**— Product Owner plus all Area Product Owners. Conceptually, the same as in a LeSS structure. In huge LeSS Huges, there are also APO Sub-Teams that are, essentially, what is Product Owner team is in LeSS. Each APO Sub-Team consists of Area Product Owner and customers/users that can help APO with clarification and other daily work (**NOT** departments of analysts, specification writers, UI/UX designers, or architects).

**Head of Product Group** - most likely, a traditional "head of product group". Whereas the whole group supports teams by Go See and help them remove obstacles and improve, Head of Product Group plays the *role of the line manager* of all the teams, whatever that is called in your organization. Product Owner Team members (including APO Sub-Teams) and all developers from teams in requirement areas, as well as people in other supportive organizational structures (e.g. support, 'undone' department, competence & coaching) -all report to Head of Product Group. **Note:** Since LeSS organizations don't have matrix structures, there are no dotted-line managers.



**Undone department** - e.g. test, QA, architecture, or business analysts - they should not exist in the smaller LeSS framework groups; rather, they should be integrated into the teams. Undone department is still frequently seen, in some LeSS adoptions, as they often cross organizational boundaries. A goal in every LeSS adoption should be to remove an undone department.

**Support** In a LeSS Huge organization, it typically centralizes some support because of the massive volume of work. Still, this department should be as small as possible with the attitude *"How can we help?"* rather than *"Take it this way!"* This department is less likely to exist in smaller LeSS organizations, where teams support each other without the need for a separate, centralized group.

**Competence and Coaching (technical, organizational)** - group consists of skillful practitioner-experts who actively Go See and observe how people work. They pair up and work with people to discover their training and coaching needs.

*By Gene Gendel. 2020*