



Certified LeSS Basics (CLB)

1- Day Review



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 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.

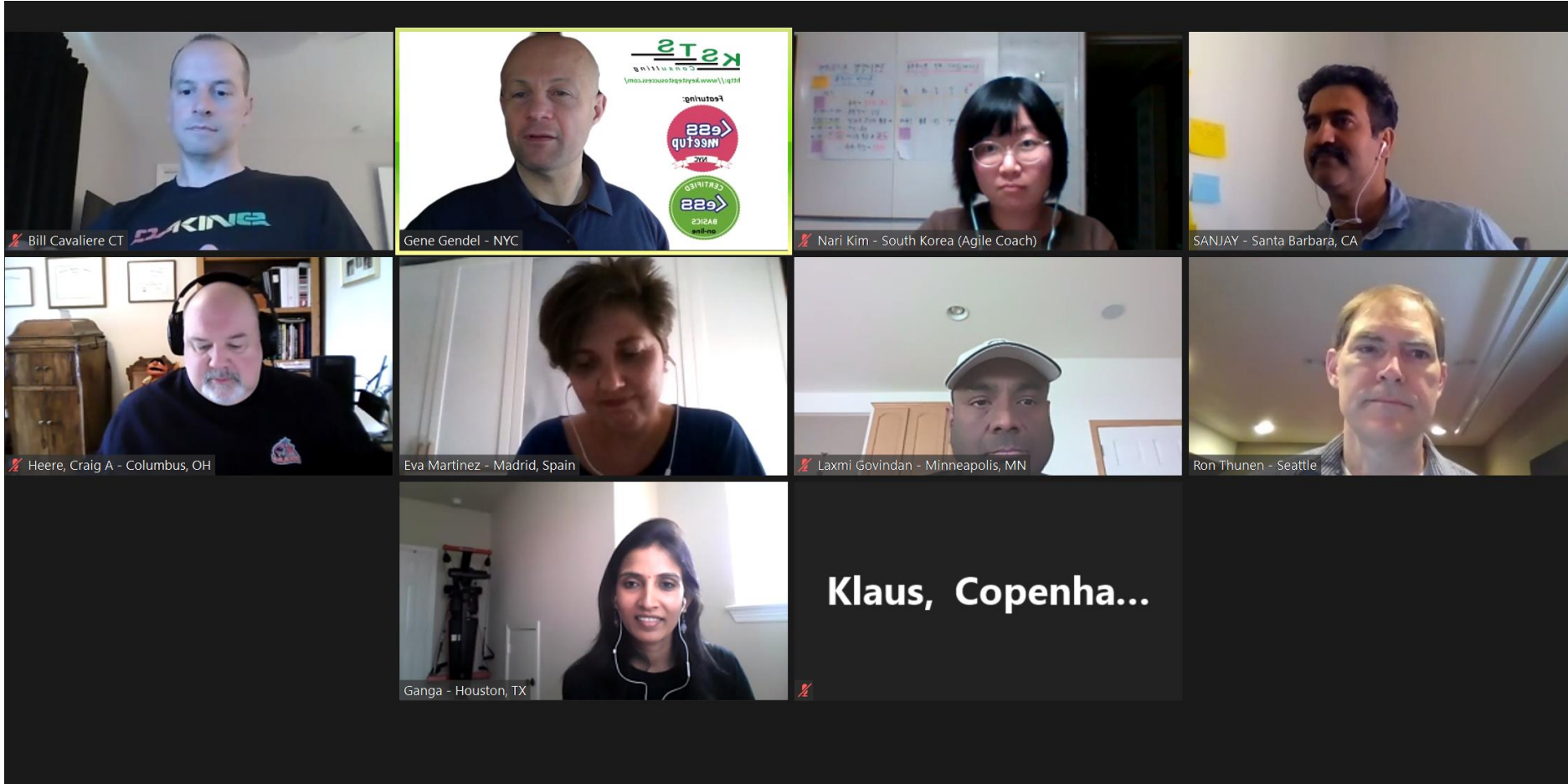
He is also well-recognized blogger and publisher. He is the author of Adaptive Ecosystems: Collection of Independent Essays About Agility and co-author of:

- 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.

Our Class



Taking Care of Key Logistics



Meeting ID: 990 8460 8282
Meeting Password: 948046
<https://zoom.us/j/99084608282?pwd=SjlJa2ZYbU9sZEHUcU5QMk9mQmdWQT09>



<https://miro.com/welcomeonboard/ghck4LmOT7YxexbXZHWPsvmxC67japxAdyuHp8bHpeJHehnrXR8AHOfIwnLXiUMR>



https://docs.google.com/spreadsheets/d/1N5fMp11qKijDzP04pKvg4yY_Q0lkUvpqbl554-6wKRY/edit?usp=sharing

Introduction

Topics Covered and Functional Learning

- Introduction:
 - What is LeSS Complete diagram: Principles, Rules, Guides, and Experiments?
 - What are some expected challenges with LeSS adoption?
 - Why is System Thinking and System Modelling so important in LeSS?
- What makes LeSS is a Scrum-based approach on scaling?
- What is Local Optimization & Contract Game?
- What impact does LeSS have on the organizational design (structures, policies)?
- How can LeSS scale over to ~ 8 teams? Team Size, Sprint Cadence?
- What differences exist in dynamics of component vs. feature teams?
- What are LeSS roles and their purposes?
 - Why is there only real PO and not so-called team POs ("TOO")
 - Why SM is a full-time role in LeSS?
- What is the different between Product vs Project/Program/Portfolio?
- What are LeSS Events & Coordination?
- What about management in LeSS?
- What is LeSS Huge?



How to Maximize Benefit from CLB?

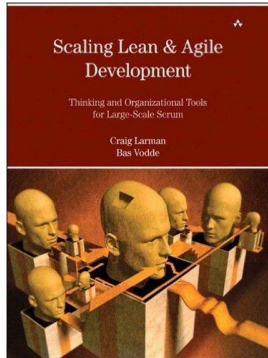
- Must Have: Foundational understanding of Scrum
- Preferred: basic Scrum training/certification from a credible source (ScrumAlliance.org or Scrum.org)
- Preferred: hands-on experience with Scrum

Warm welcome to:

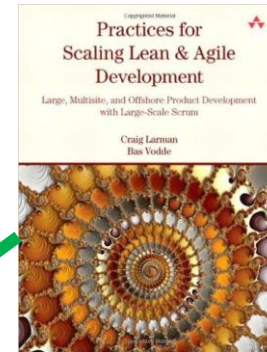
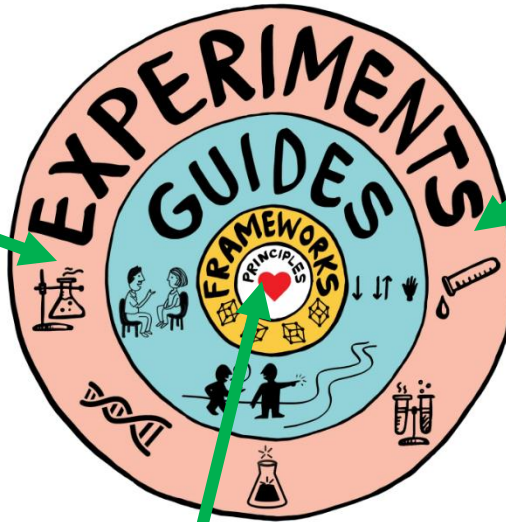
- Senior Management
- HR, Finance/Budgeting, Vendor Management
- Coaches/Scrum Masters, Product Owners, Developers
- Business: Customers and/or Users

LeSS: The Big Picture & History

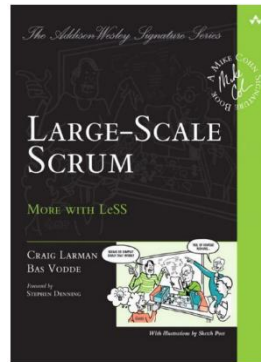
LeSS – The Big & Picture History



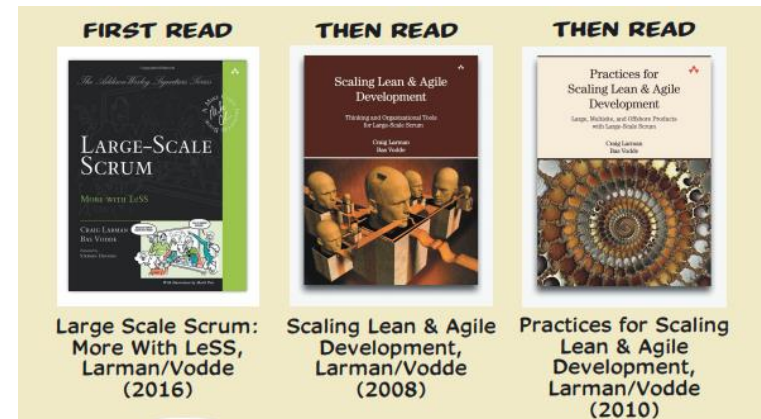
2008



2010



2016



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

Anticipated Obstacles

Challenges That LeSS Is Likely To Face

- “Doing” agile
- Terminology over-loading
- “Best Practices”
- Agile CoE (refurbished from other CoEs)
- Re-writing Playbooks, Manuals, Internal Guides, etc
- ...other FADs....

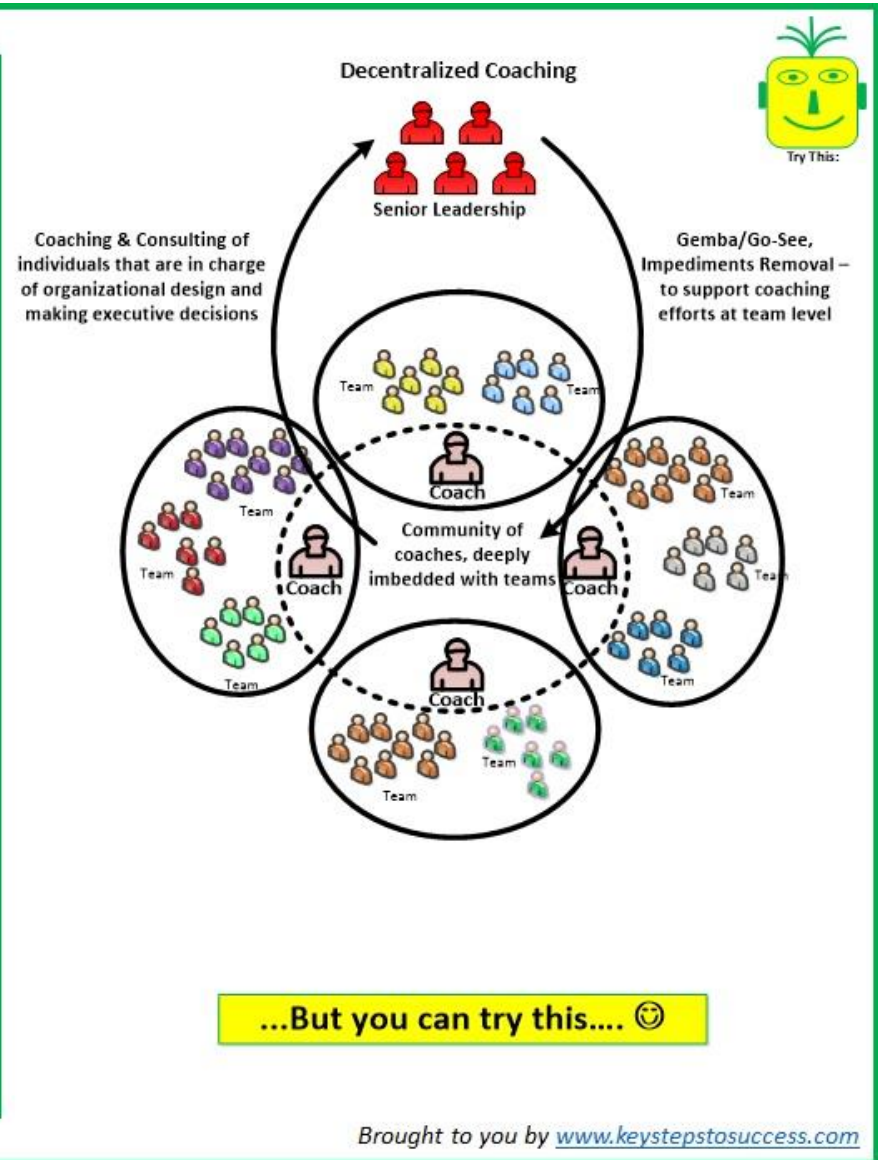
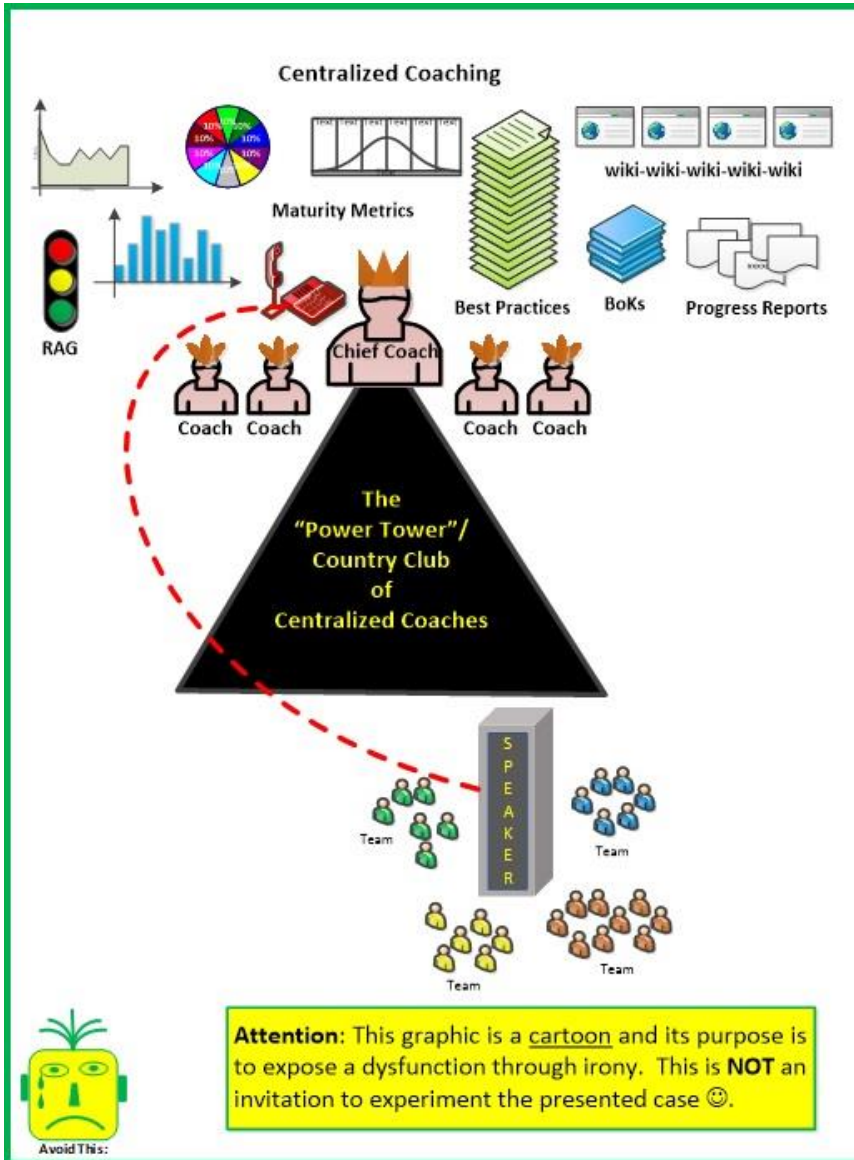
- Embracing organizational complexity
- Protecting existing power structures
- Promoting Local Optimization
- Protecting Role Security (not to be confused with Job Security)
- Creating dependencies on big commercial scaling solutions that are “in trend and style”

Beware of Agile “Theater”



• [John Kotter – Resistance to Change](#)

Awareness of Agile “Theater”




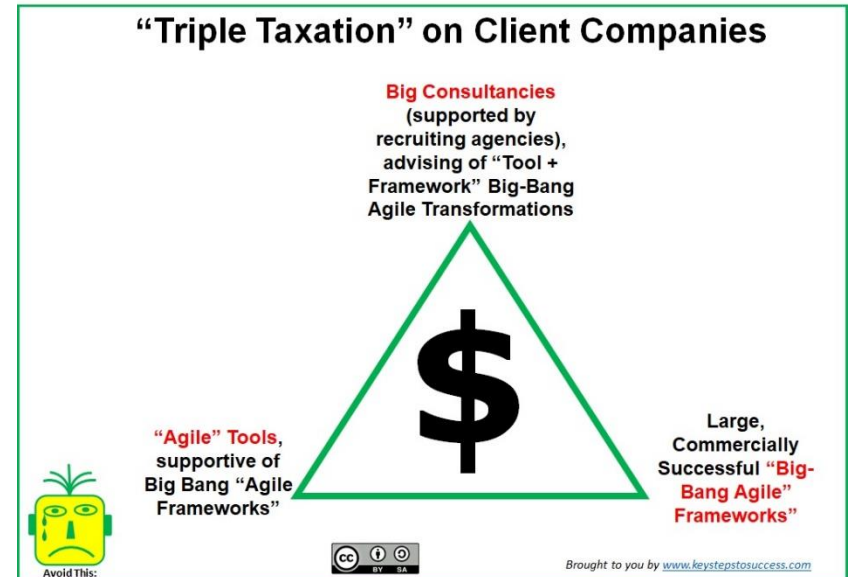
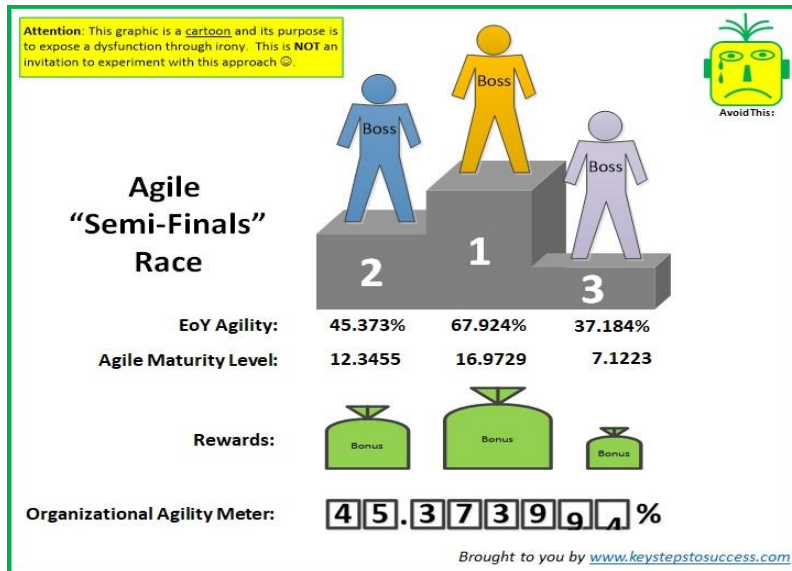
Brought to you by www.keystepstosuccess.com

Awareness of Agile “Theater”

Brought to you by www.keystepstosuccess.com

Avoid
(Deck – Driven Transformations)
DDT

Knowledge retention and quality of learning are inversely proportional to thickness of a power point 'agile transformation' deck and directly proportional to font size used in a deck ;)

Management In LeSS

Informed Consent



Management NOT Supportive of LeSS

Frederick W. Taylor



Frederick Taylor is known today as the father of scientific management.

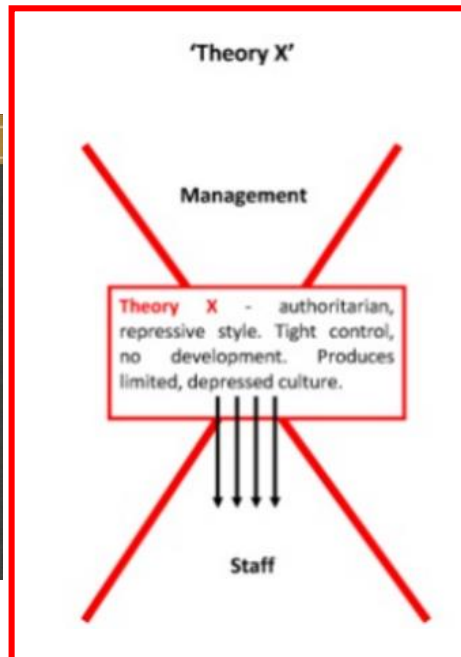
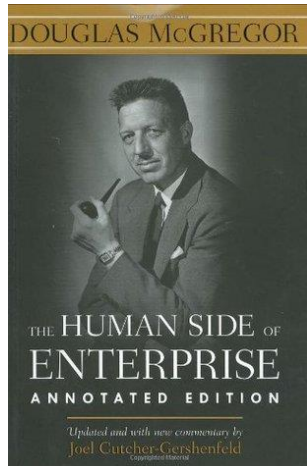
One of his many contributions to modern management is the common practice of giving employees rest breaks throughout the day.

Henry Fayol



Pioneered the study of principles and functions of management. Developed management functions:

- Planning
- Coordinating
- Organizing -Controlling
- Commanding



Managers Teachers

manager-teacher



“my manager can do my
job better than me”
Toyota (Lean) saying

“One of the real problems
with American Business is
this notion that you can be
trained in management, in
some kind of generic form
of management, and that
then you can manage any
operation.

But that absolutely
doesn't work in a
technical situation.”

Robert Noyce -
Founder of Intel

“...if you are telling me *what* to
do, show me *how* to do it...”

599

- Managers - to do 'Go See' (Gemba)
- Managers, as capability builders and enablers
- Less command & controlling; more empowerment
- Teachers and Mentors

Management in LeSS

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).**

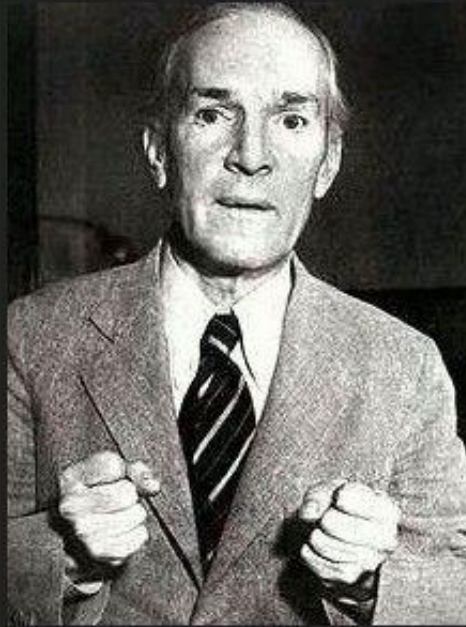
5. Culture follows structure.

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.*"

Management in LeSS



It is difficult to get a man to understand something when his salary depends upon his not understanding it.

(Upton Sinclair)

izquotes.com

Management in LeSS

A common problem with Scrum adoption occurs when managers do not give up these responsibilities, thereby causing an organizational conflict between the Team, Product Owner, Scrum Master, and managers.

Job & Salary Safety
but not
Role Safety

Why LeSS?

Three Adoption Principles

LeSS Adoption, as a Sushi Roll

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>

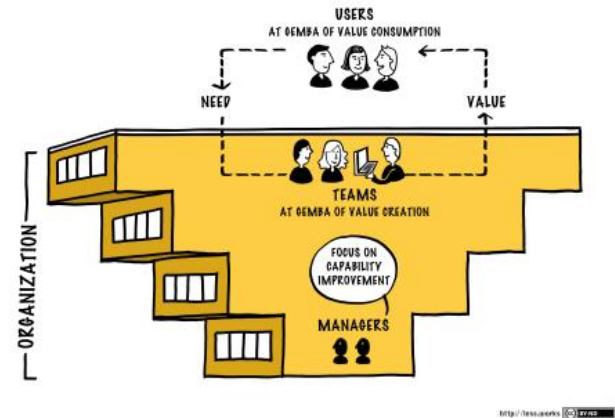


<http://www.keystepstosuccess.com/2017/08/agile-organization-as-a-sushi-roll/>

Three (3) Adoption Principles of LeSS

➤ Deep and narrow over broad and shallow

➤ Top-down and bottom-up



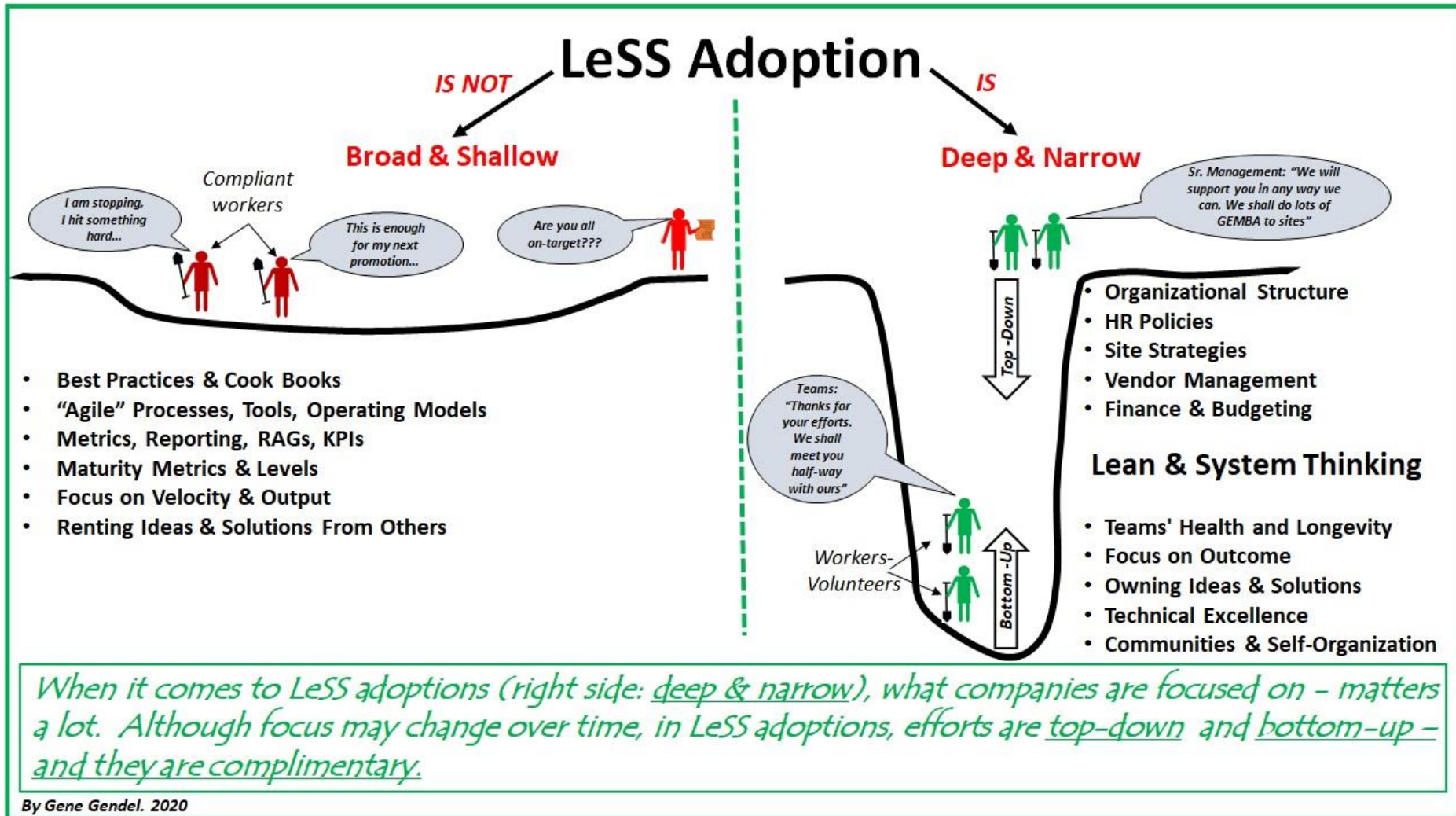
➤ Use volunteering



Broad & Shallow VS. Deep & Narrow

Superficial

Top → Bottom & Bottom → Up



Use Volunteering

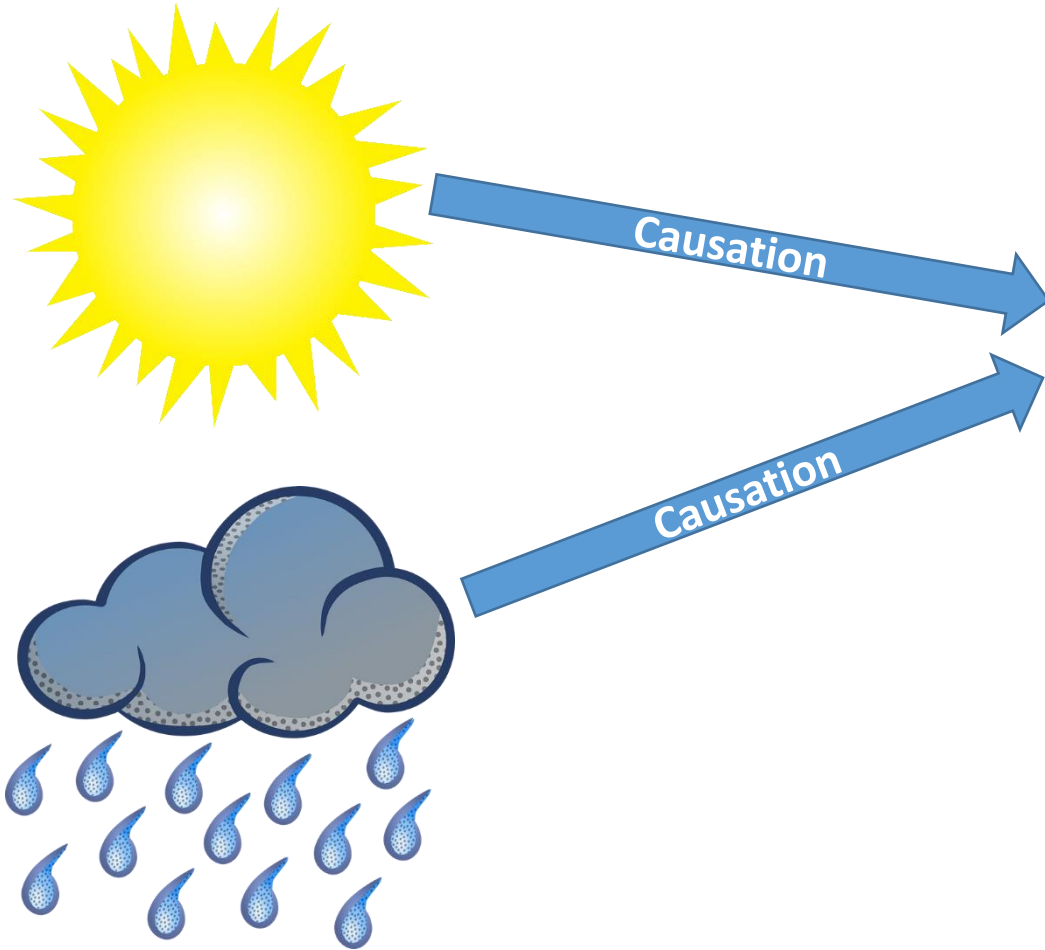
- **When selecting a product group for LeSS adoption**
- **When letting people in (product group)**
- **When letting people out of (product group)**
- **When forming teams**
- **When creating Communities**
- **When teams pick up backlog items during PBR**
- **When teams pick up backlog items during Sprint Planning**
- **When team members pick up backlog items during Sprint**



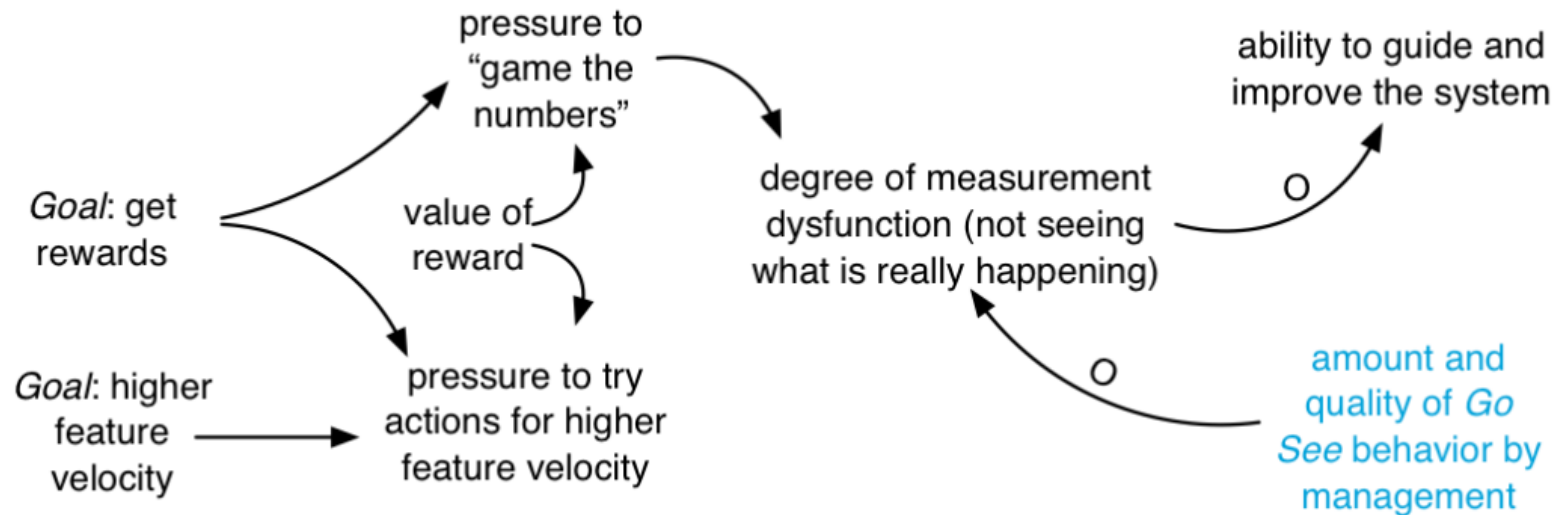
Understanding System Thinking *and* System Modelling

Causation vs. Correlation

What is the difference?

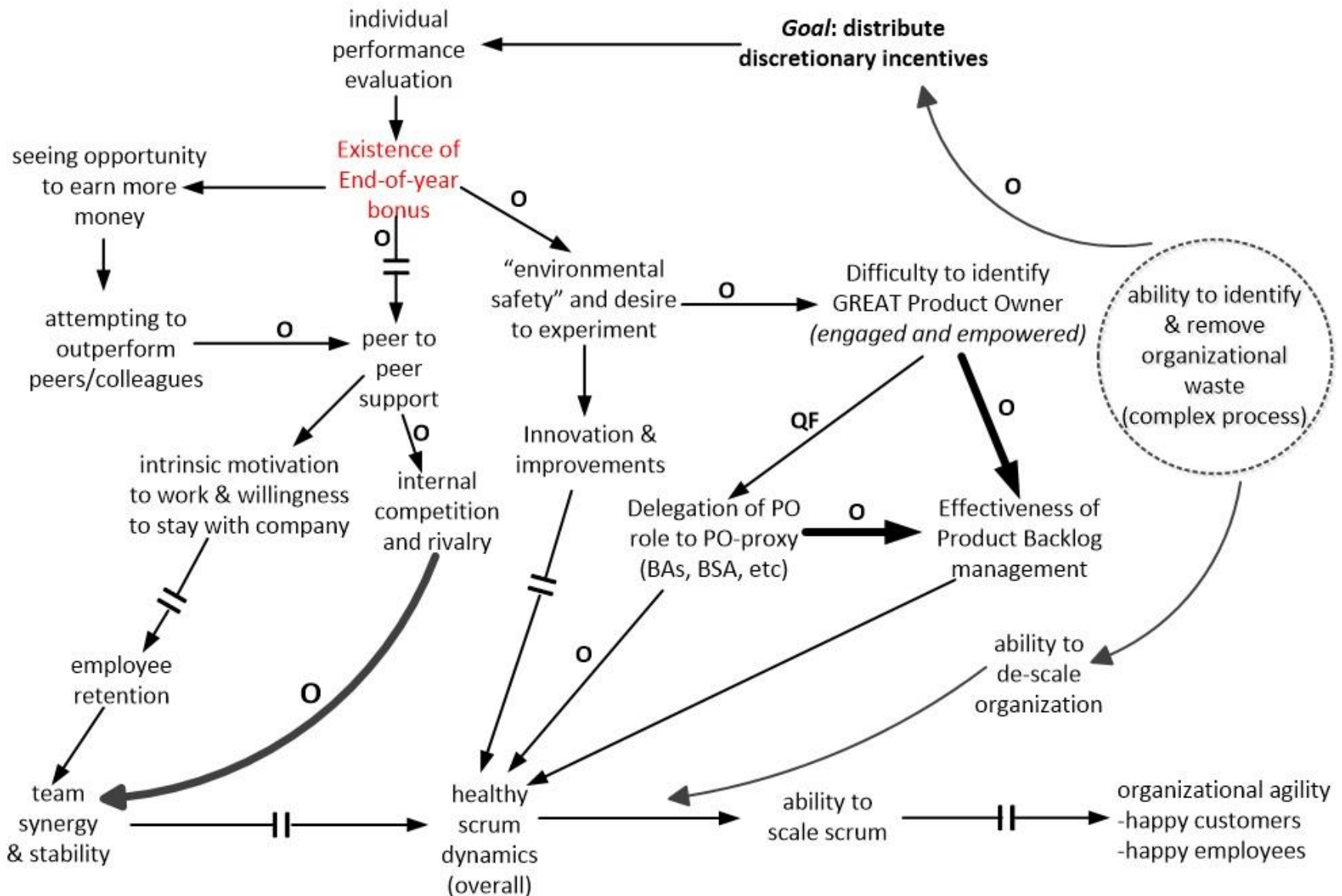


System Modelling with CLD (example)



Source : http://less.works/less/principles/systems_thinking.html

System Modelling with CLD (example)



Source: <https://www.scrumalliance.org/community/articles/2016/july/from-the-less-toolbox-causal-loop-diagrams-to-visu>

Causal Loop Diagrams (CLDs): Annotations

Here are some elements of CLDs that I use in my graphics:

- Goals — A high, overarching/strategic goal that needs to be achieved
- Variables — System elements that have an effect or influence on other system elements (other variables)
- Causal links — Arrows that connect two related variables
- Opposite effects — “O” annotation near an arrow; suggests that the effect of one variable on another is the *opposite* of what could be expected
- Delayed effect — “||” annotation that disrupts a causal link (arrow); it implies that there is a delayed effect of one variable by another variable
- Extreme effects — One variable has an extreme (beyond normal) effect on another variable; it is represented by a thick arrow
- Constraints — “C” annotation near arrow; implies that there is a constraint on a variable
- Quick-fix reactions — “QF” annotation near an arrow; action that brings about short-term, lower-cost effect

Group Activity (Agile “Big Bangs”)

Exercise

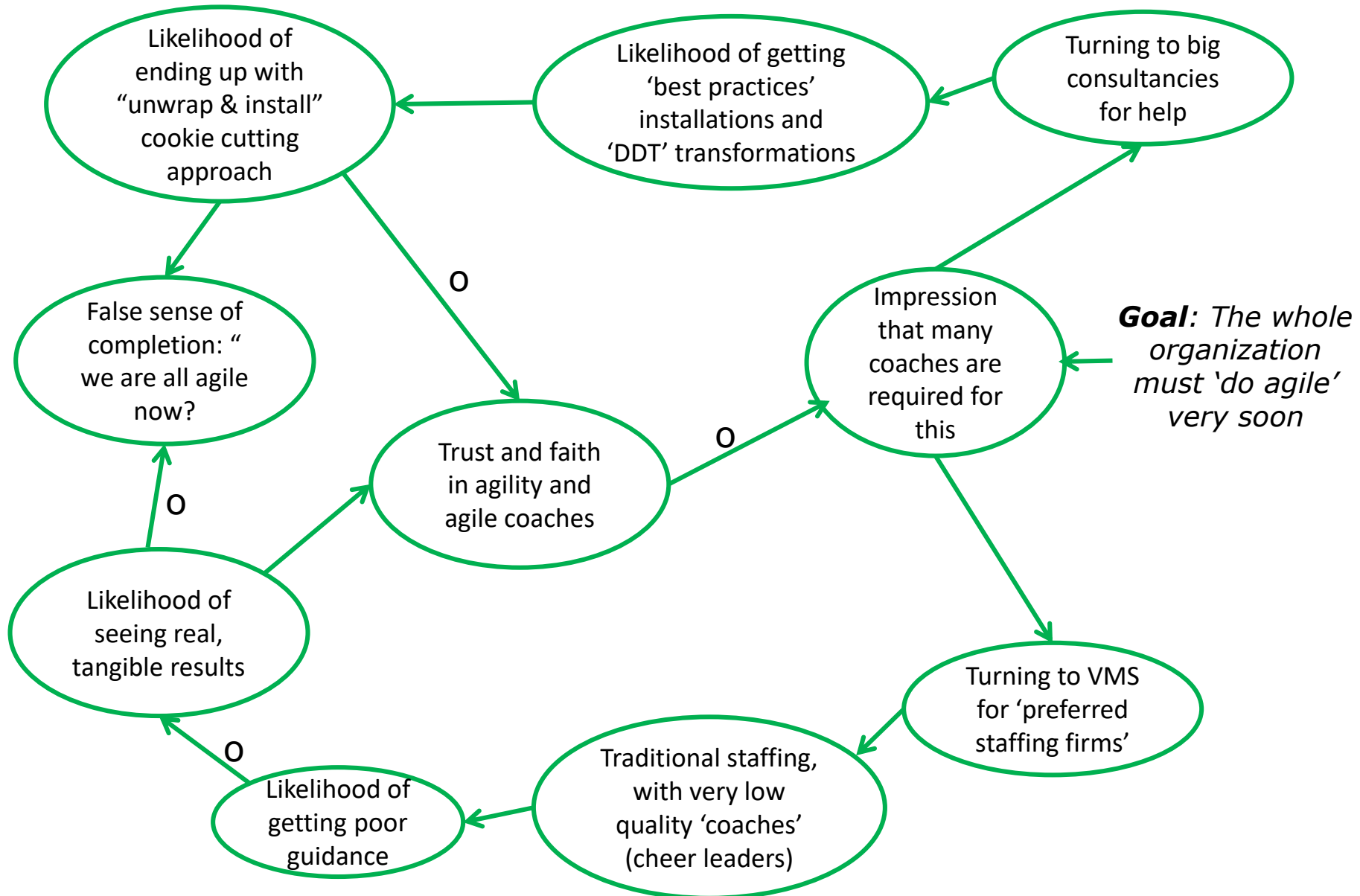
Duration: 10 min

Class: in-groups, brainstorm what could some reasons for **Big-Bang Superficial Agile Transformations** and what they may lead to.

Method/Tool: System Modeling with CLD

Instructor: review with Class

Suggested Solution (Agile “Big Bangs”)



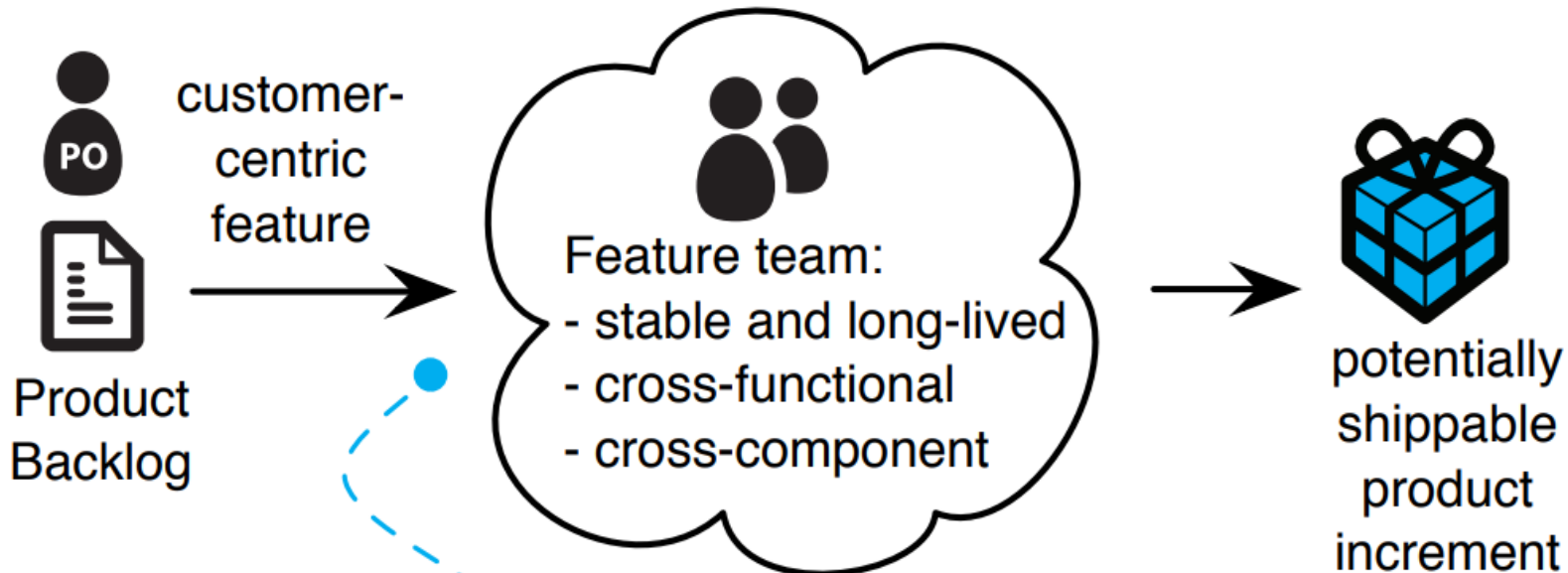


5 min Break

LeSS IS Scrum

(Size, Sprint Cadence, Dynamics)

LeSS is Scrum-based Approach

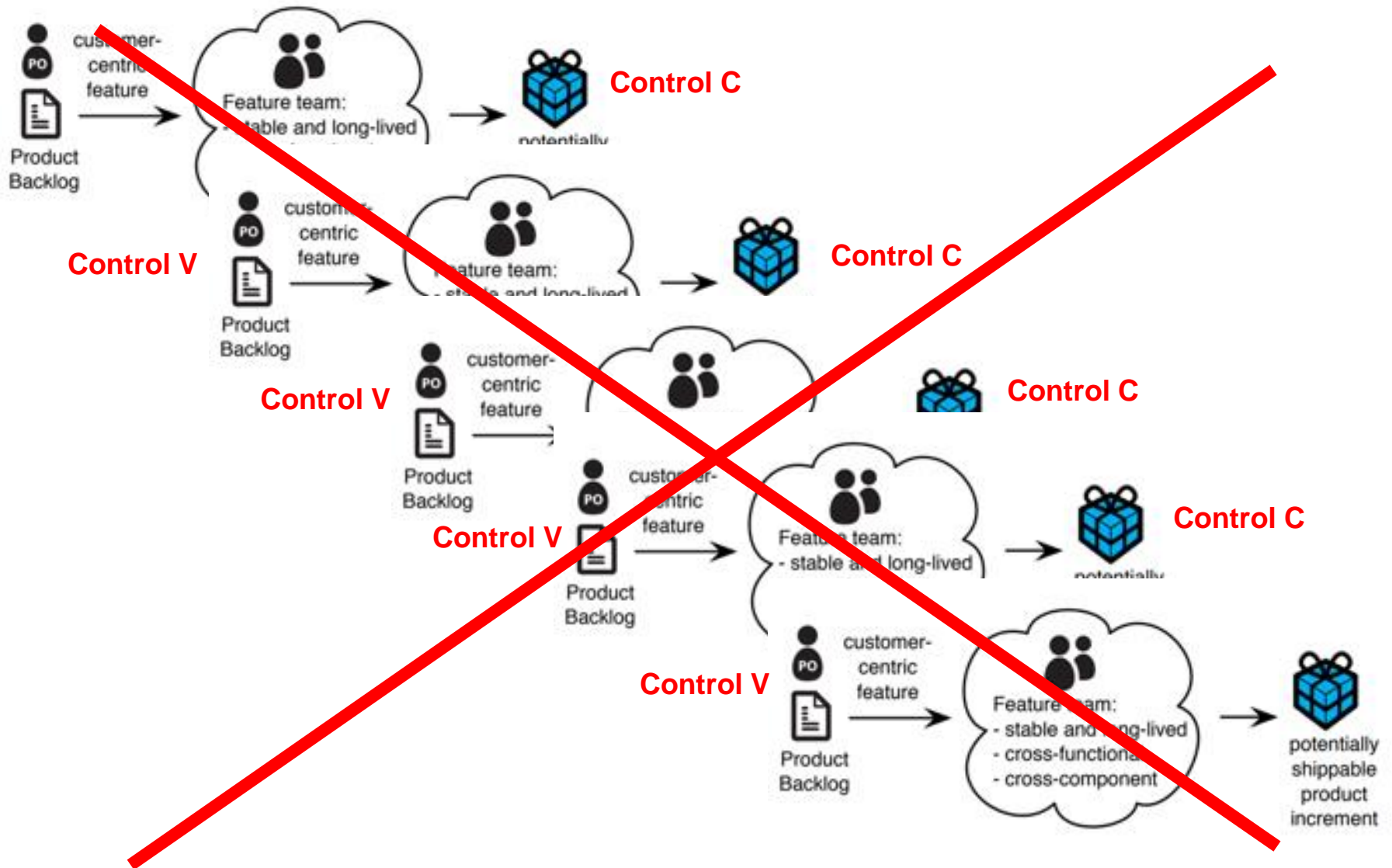


Team has the necessary knowledge and skills to complete an end-to-end customer-centric feature. If not, the team is expected to learn or acquire the needed knowledge and skill.

www.craiglarman.com
www.odd-e.com

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

LeSS is NOT



Beware of Scrum Anti-Patterns

Scrum Anti-Patterns

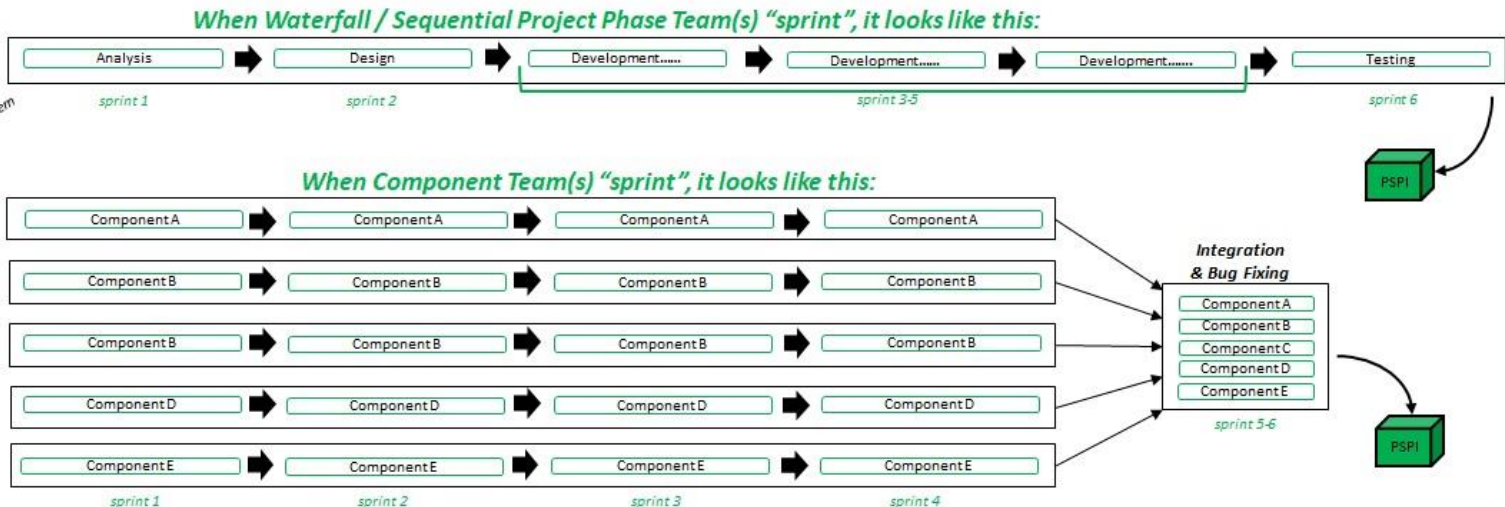
In Scrum, in every Sprint, a team delivers Potentially Shippable Product Increment (PSPI). **This is fundamental for Scrum.** In order for this to happen, each team must possess all necessary attributes (skills, knowledge, domain expertise) required to get work fully DONE (potentially shippable). This 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



Try This:

- 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

Brought to you by www.keystepstosuccess.com

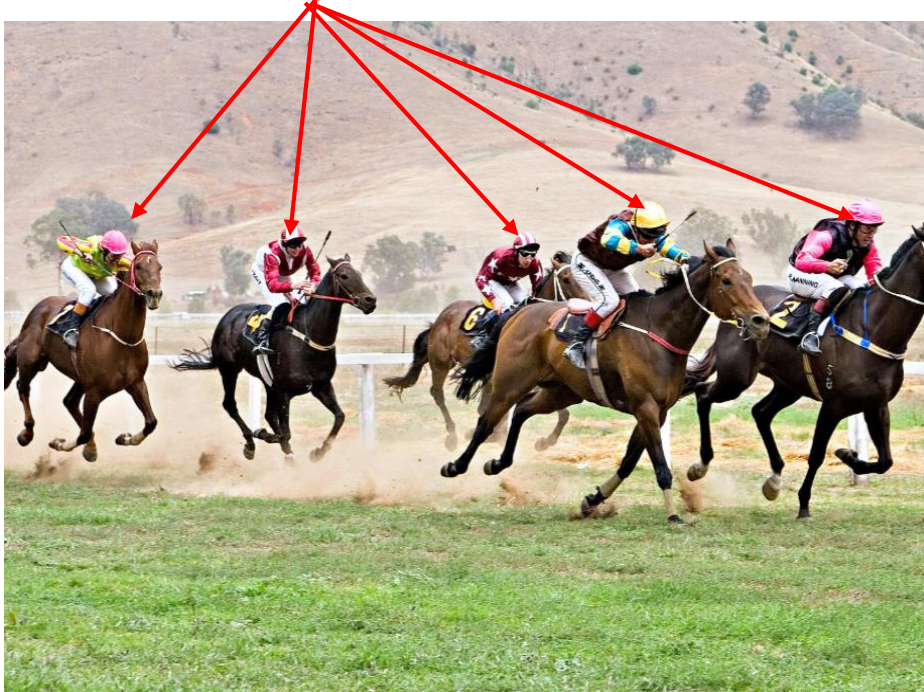
KSTS
Consulting

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

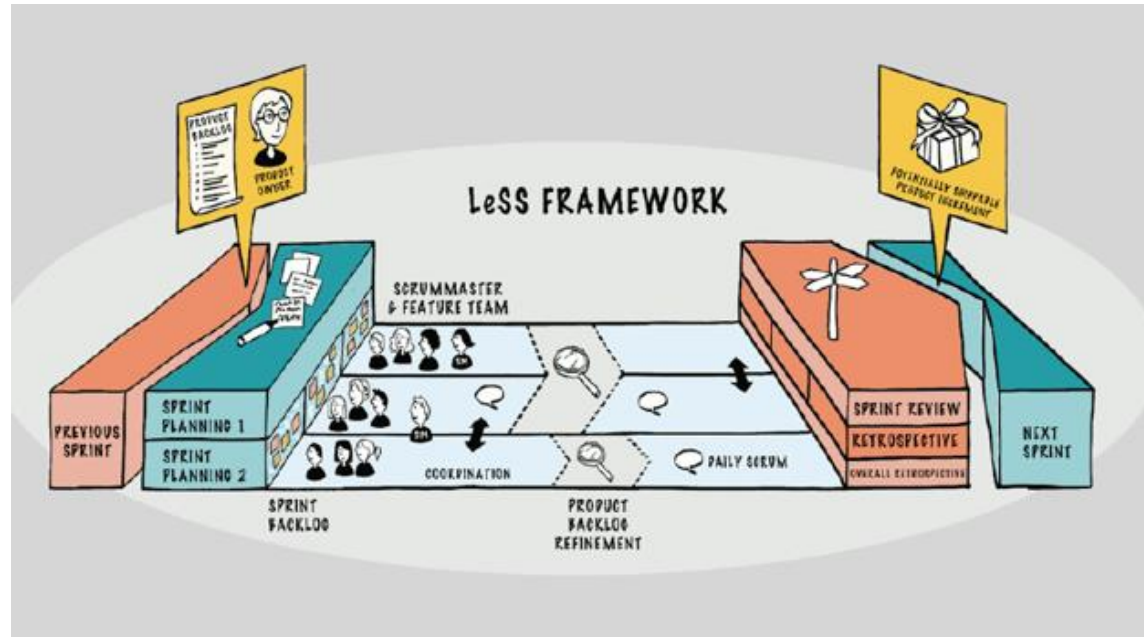
Same person with strategy, and vision, setting a unified course



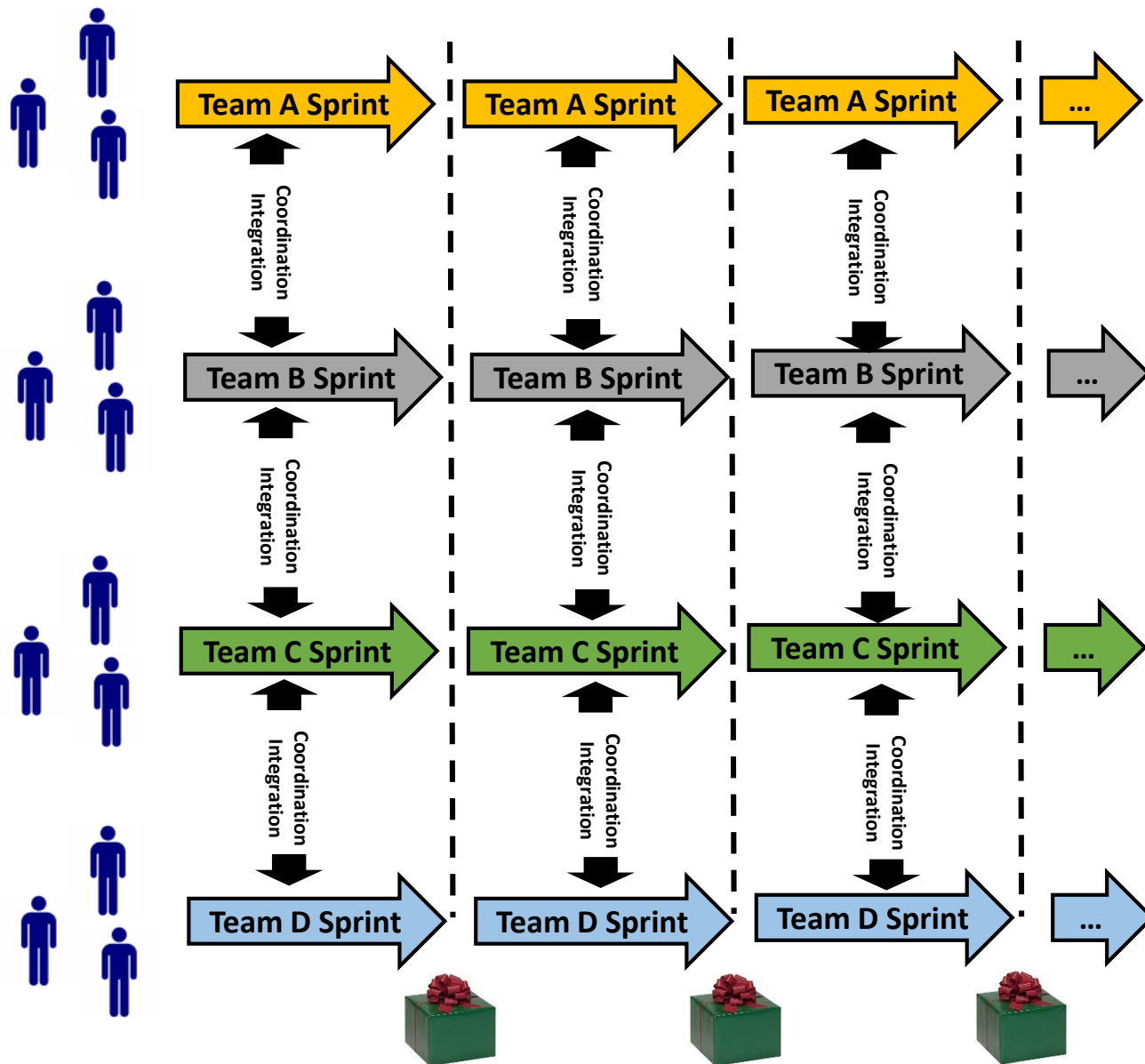
YES

Size of LeSS Adoption

- LeSS adoptions require months of preparation. (LeSS HUGE adoptions are incremental, not “all-at-once”)
- 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.**



Sprint Cadence in LeSS: One LeSS Sprint



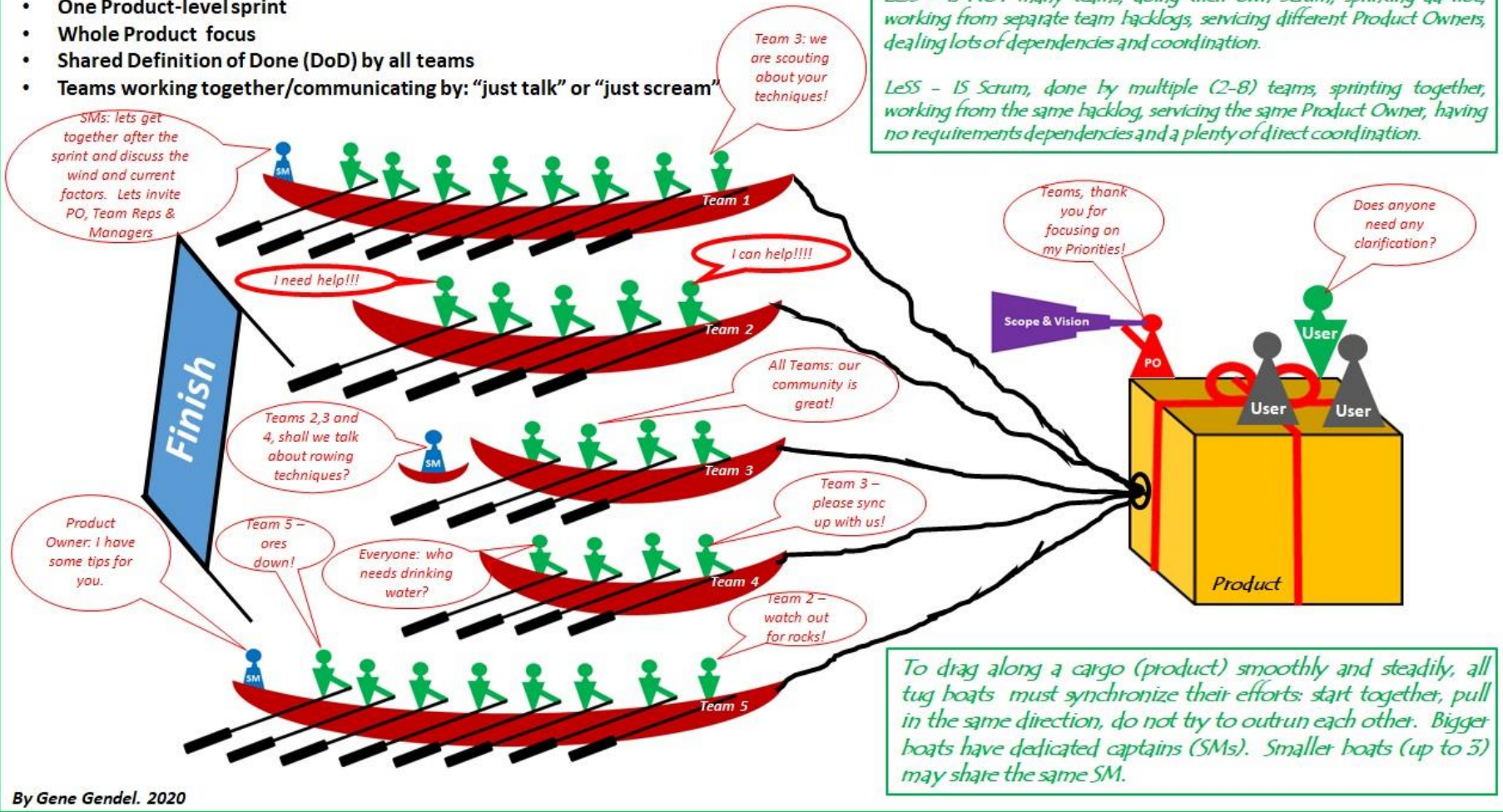
**Sprints are
parallelized,
not staggered**

Sprint Cadence in LeSS

- 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”

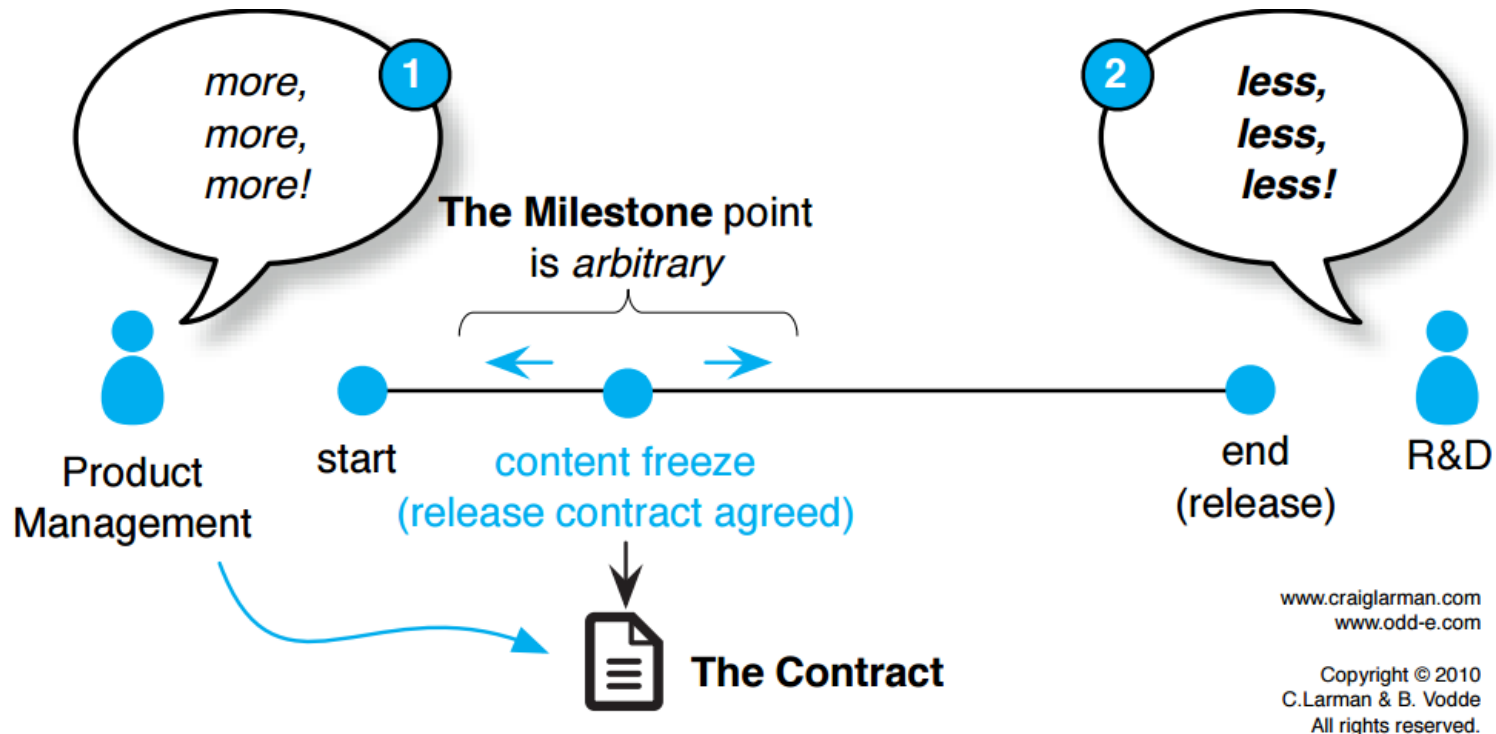
LeSS - is NOT many teams, doing their own Scrum, sprinting ad-hoc, working from separate team backlogs, servicing different Product Owners, dealing lots of dependencies and coordination.

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.



By Gene Gendel. 2020

The Contract Game



[The “Contract Game” \(22-44min\)](#), by Craig Larman

**What is INTERNAL
? CONTRACT?**

Group Activity (Internal Contracts)

Exercise

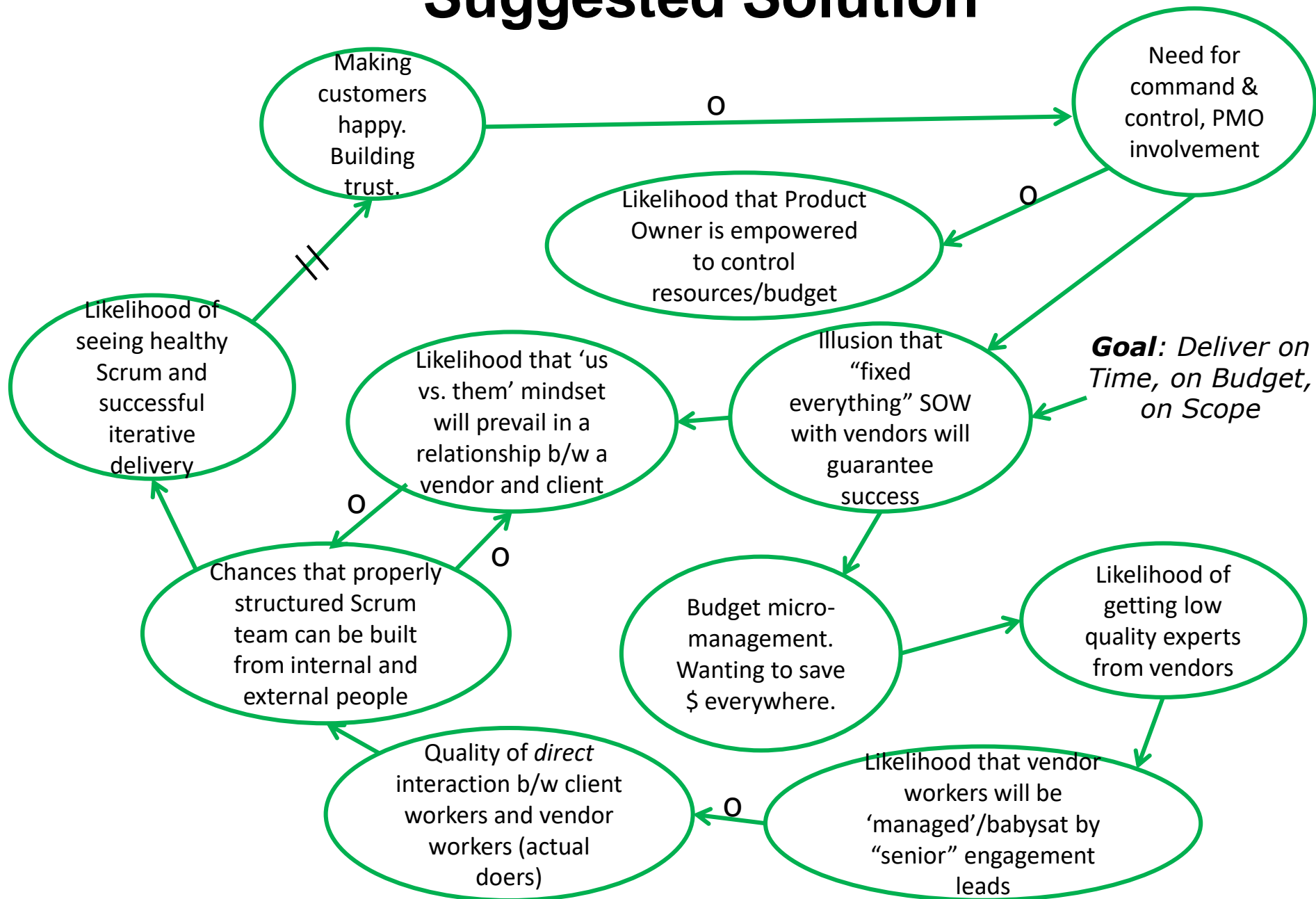
Duration: 10 min

Class: in-groups, think of how (legal) contracts, if '*translated*' into an **internal contracts**, could lead to organizational dysfunctions. Work with provided system variables to create a model.

Method/Tool: System Modeling with CLD

Instructor: review with Class

Suggested Solution



Day 2

Organizational Design & Local Optimization

This May Not Be Obvious to Everyone

Organizational STRUCTURE –

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

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

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

This May Not Be Obvious to Everyone

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.*”

Local Optimization in Roles ("This Is Not My Job")



What is Local Optimization?

"Everyone is busy and working so hard. Yet, the system is delivering slow and Users are not happy"

How could that be?



Zero Sum Effort

Local Optimization In Roles

Tonight's Play in Agile Theater

Title: *"Rearranging Deck-Seats on Titanic"*

Summary: *Intermediate ~~Senior !!!! Business Analysts~~ Product Owners (proxies) talk to Users and write ~~BRDs~~ User Stories for developers, and answer developers' questions* 🤔

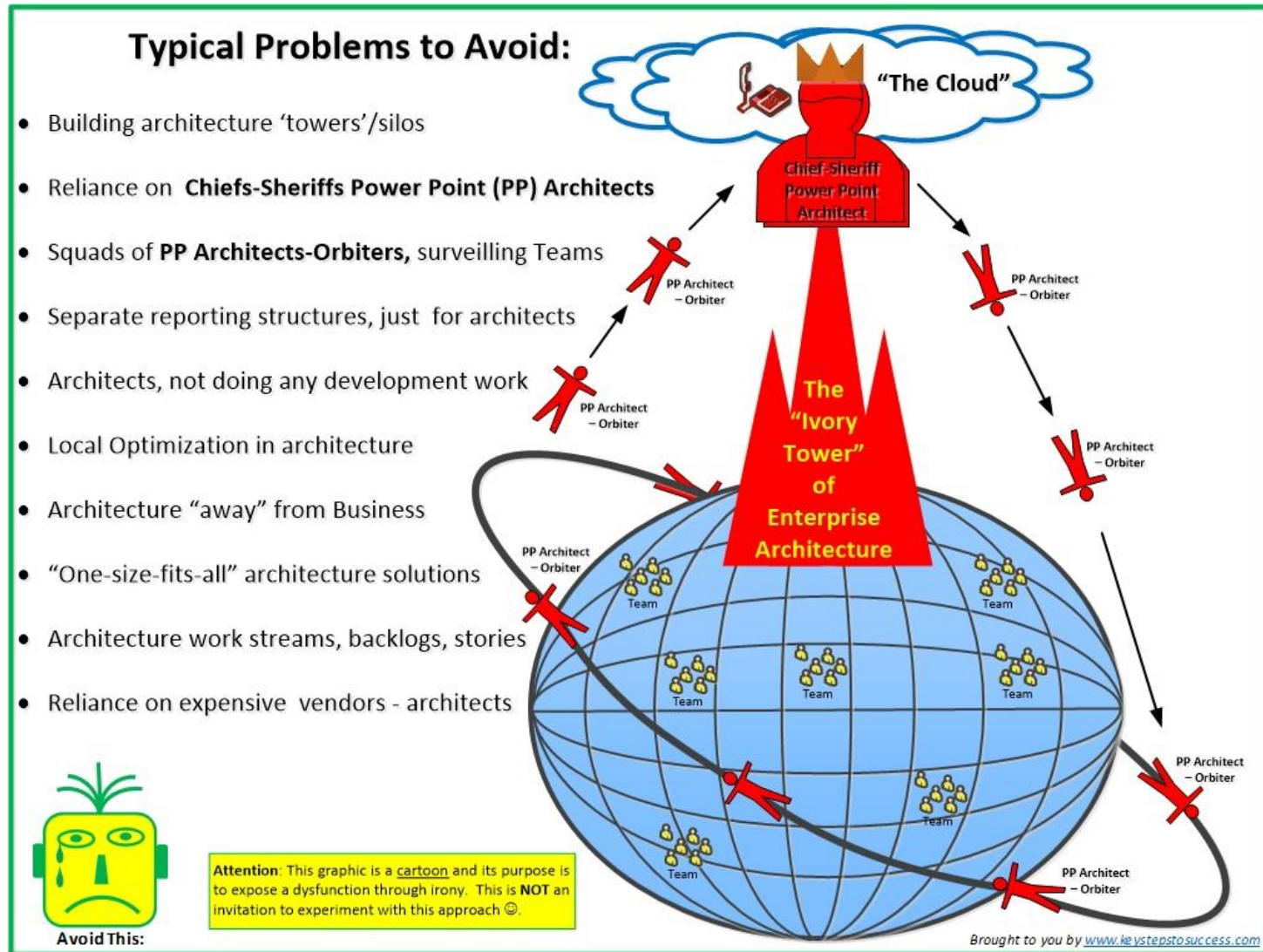
Now we are Agile!!!

And arguments we hear:

- Efficient
- Productive
- Best
- Good
- Optimized
- Ideal
- Cost-Effective

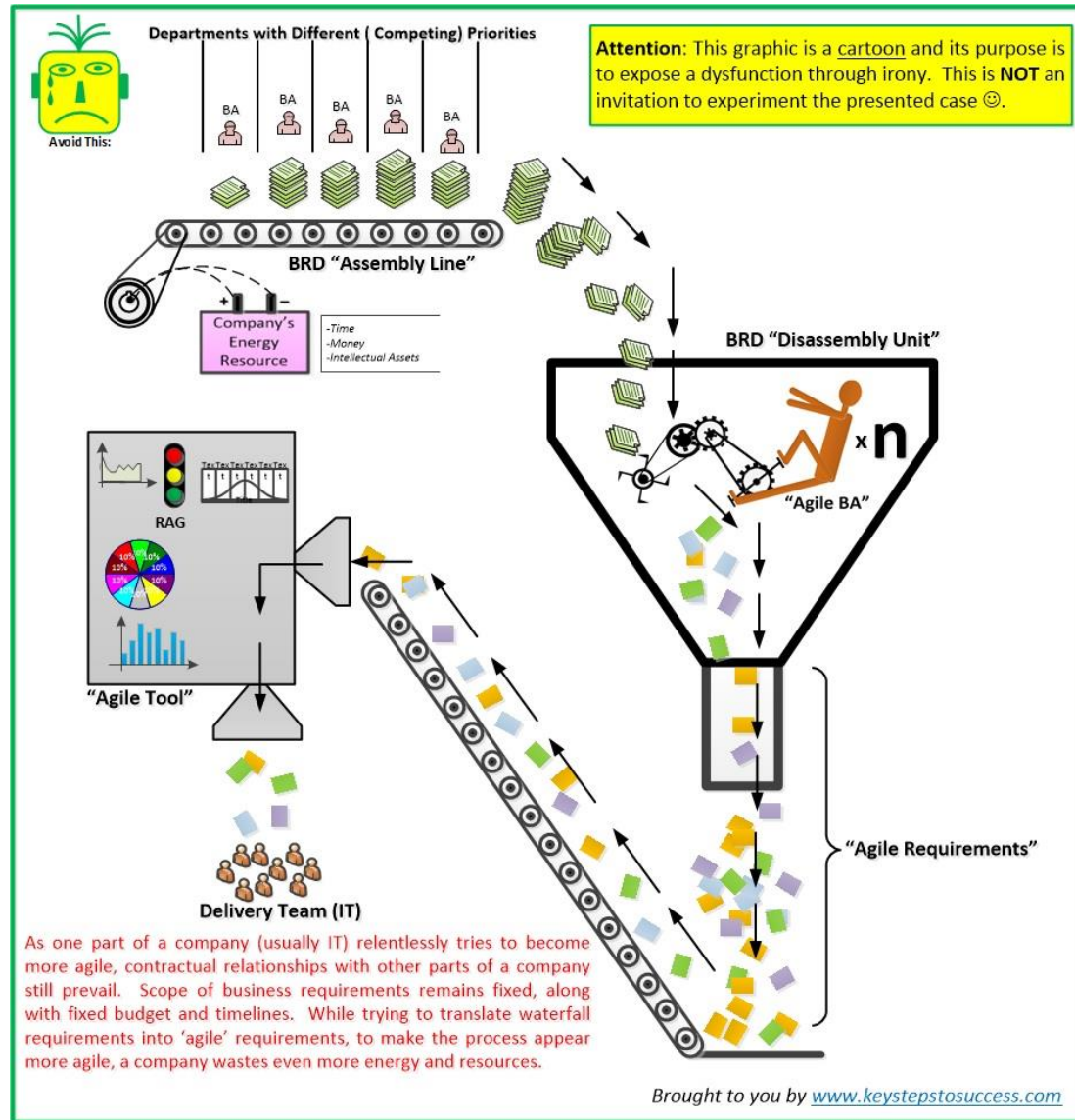


Local Optimization In Roles



Source: http://www.keystepstosuccess.com/wp-content/uploads/2019/01/pp_architects.jpg

Local Optimization In Roles



Source: http://www.keystepstosuccess.com/wp-content/uploads/2018/07/cartoon_brd_assembly_line.jpg

Seeing (Hearing) Local Optimization in...

Team Structures

Org. Structures

Documentation

Definition of Done

Backlogs

Role Definitions

Product Design

Goals & Metrics

Also at: [https://less.works/less/principles/systems-thinking.html#Seeing\(andHearing\)LocalOptimization](https://less.works/less/principles/systems-thinking.html#Seeing(andHearing)LocalOptimization)

Group Activity (Local Optimization)

Exercise

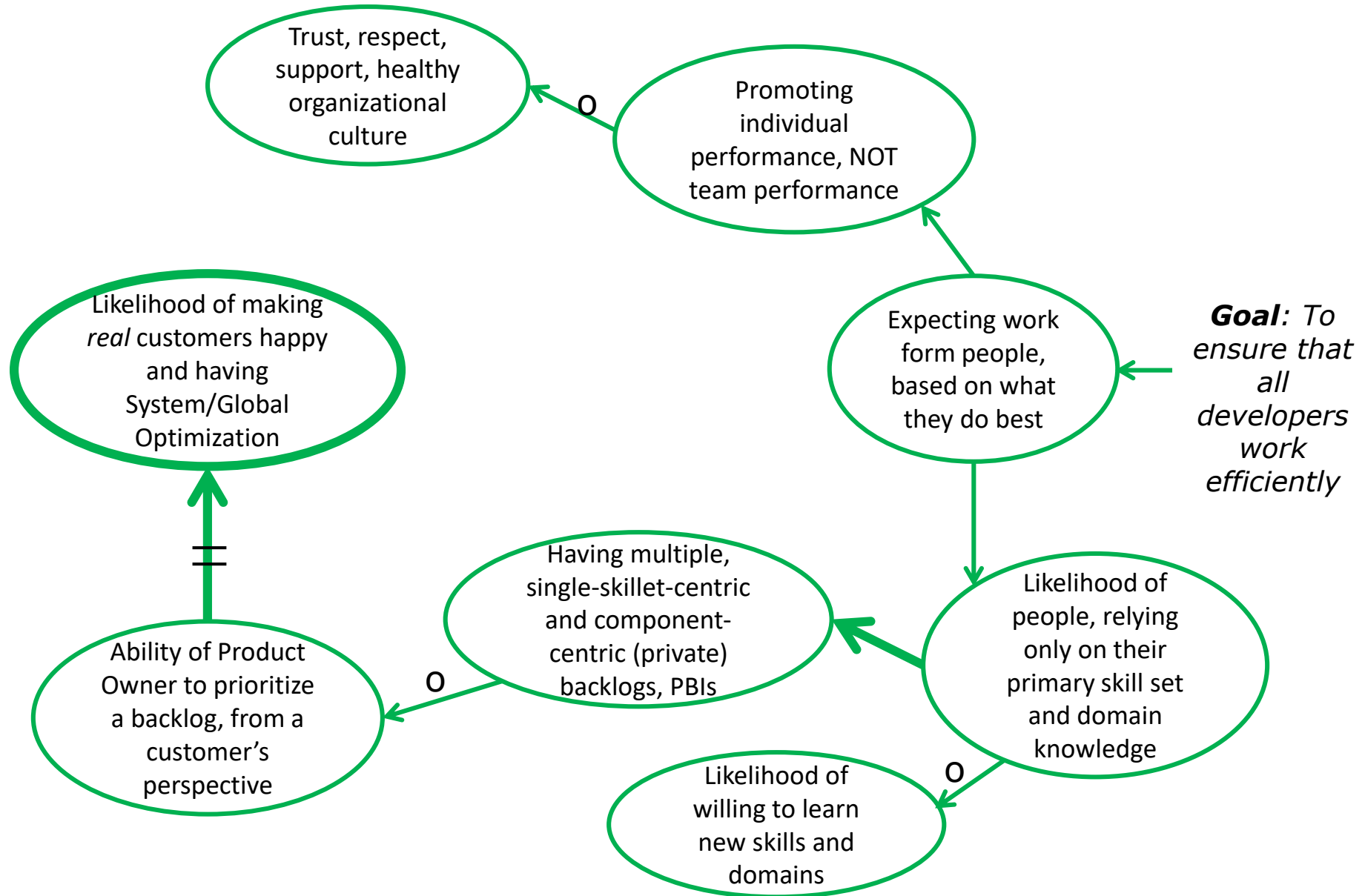
Duration: 10 min

Class: in-groups, brainstorm some of the most common examples of ***Local Optimization*** in your respective organizations. Work with provided system variables to create a model.

Method/Tool: System Modeling with CLD

Instructor: review with Class

Suggested Solution (Local Optimization)



Impact of LeSS On Org. Design/Structure

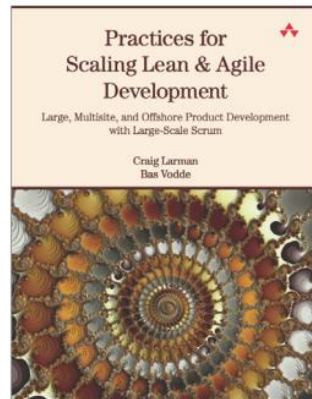
AGILE CONTRACTS PRIMER

Derived from the book...
*Practices for Scaling Lean & Agile Development:
Large, Multisite, & Offshore Product Development with Large-Scale Scrum*

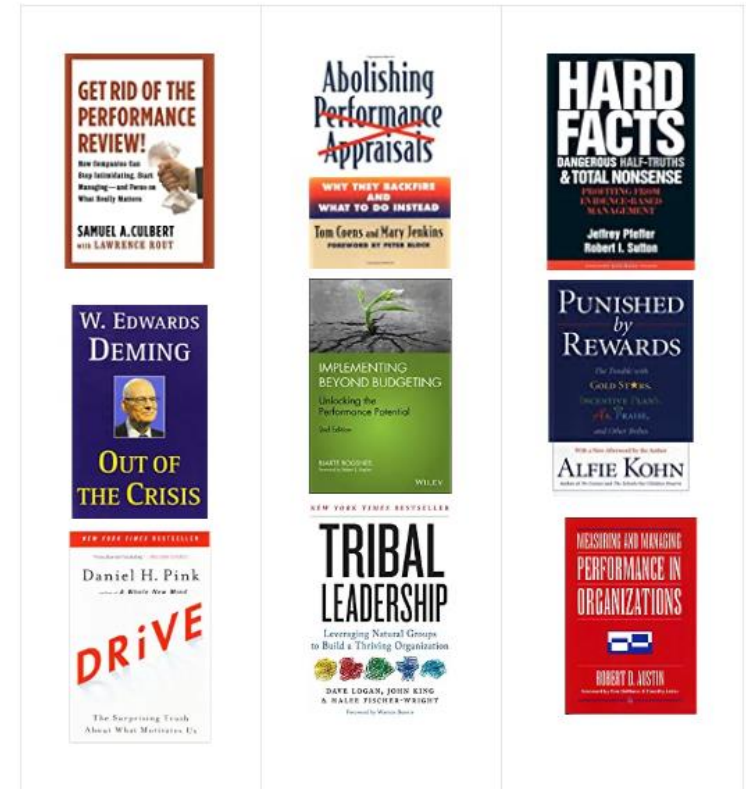
by Tom Arbogast, Craig Larman, and Bas Vodde

Version 5

Please send us comments for future versions, at www.agilecontracts.org.
Note: Check website for latest version; share the URL (rather than file) to keep up-to-date.



1	Introduction	1
2	Large-Scale Scrum	9
Action Tools		
3	Test	23
4	Product Management	99
5	Planning	155
6	Coordination	189
7	Requirements & PBIs	215
8	Design & Architecture	281
9	Legacy Code	333
10	Continuous Integration	351
11	Inspect & Adapt	373
12	Multisite	413
13	Offshore	445
14	Contracts	499



• Book Summaries

- <http://www.keystepstosuccess.com/2016/08/implementing-beyond-budgeting/>
- <http://www.keystepstosuccess.com/2016/02/quotes-from-get-rid-of-the-performance-review-how-companies-can-stop-intimidating-start-managing-and-focus-on-what-really-matters-by-culbert-samuel-a-laurence-rout/>
- <http://www.keystepstosuccess.com/2016/02/quotes-from-punished-by-rewards-the-trouble-with-gold-stars-incentive-plans-as-praise-and-other-bribes-by-alfie-cohn/>

• Proper Scaling of Scrum and Dynamic Financial Forecasting

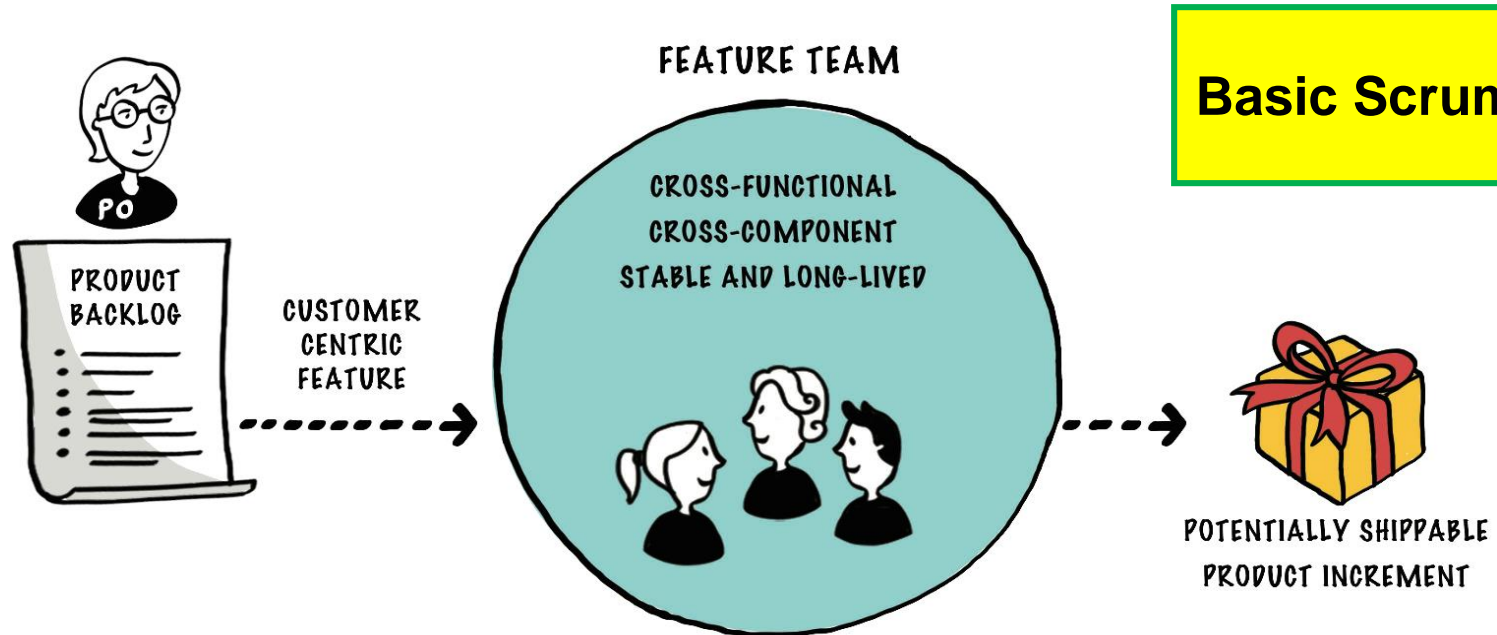
- <http://www.keystepstosuccess.com/2018/02/proper-scaling-of-scrum-and-dynamic-financial-forecasting/>

• Gap Between Science and Business

- <http://www.keystepstosuccess.com/gap-between-science-and-business/>

Feature Teams vs. Component Teams

Feature Team Key Characteristics

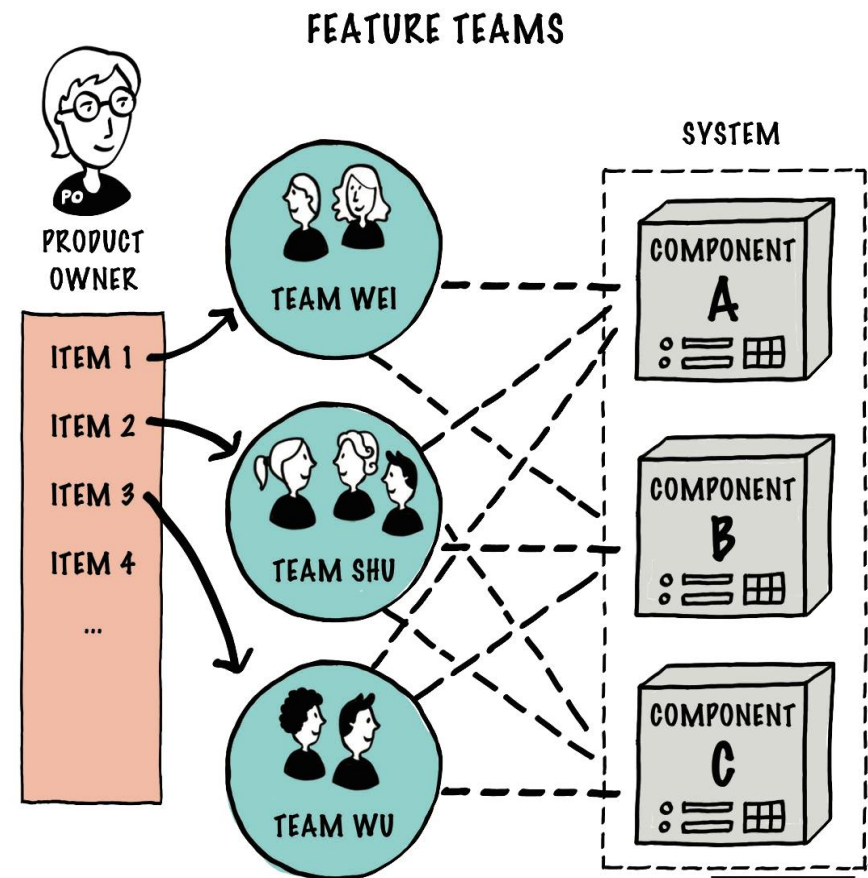
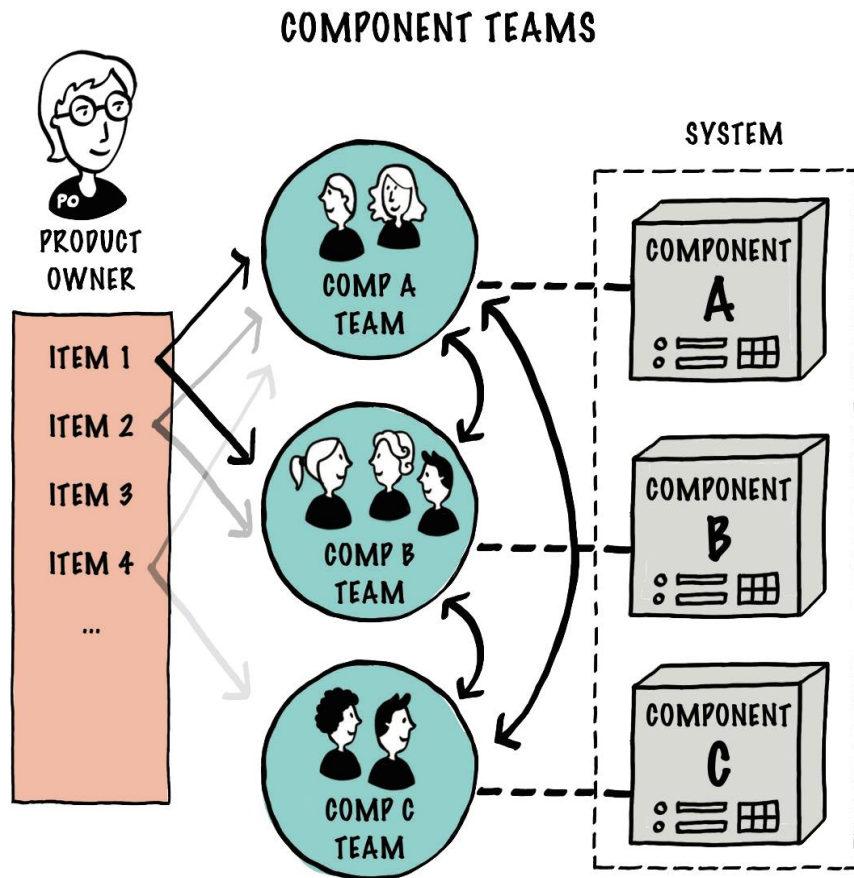


TEAM HAS THE NECESSARY KNOWLEDGE AND SKILLS TO COMPLETE AN END-TO-END CUSTOMER-CENTRIC FEATURE. IF NOT, THE TEAM IS EXPECTED TO LEARN OR ACQUIRE THE NEEDED KNOWLEDGE AND SKILL.

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

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

Local Optimization In Backlogs and Teams Structure – Classic Example ☺



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

No Team 'Tsars'

Component Teams

System Component

Single-specialty
Workers

Component Manager

UI/UX

4 people



Business Layer

5 people



Component A

7 people



Component B

4 people



Web Service

3 people



DBA

6 people

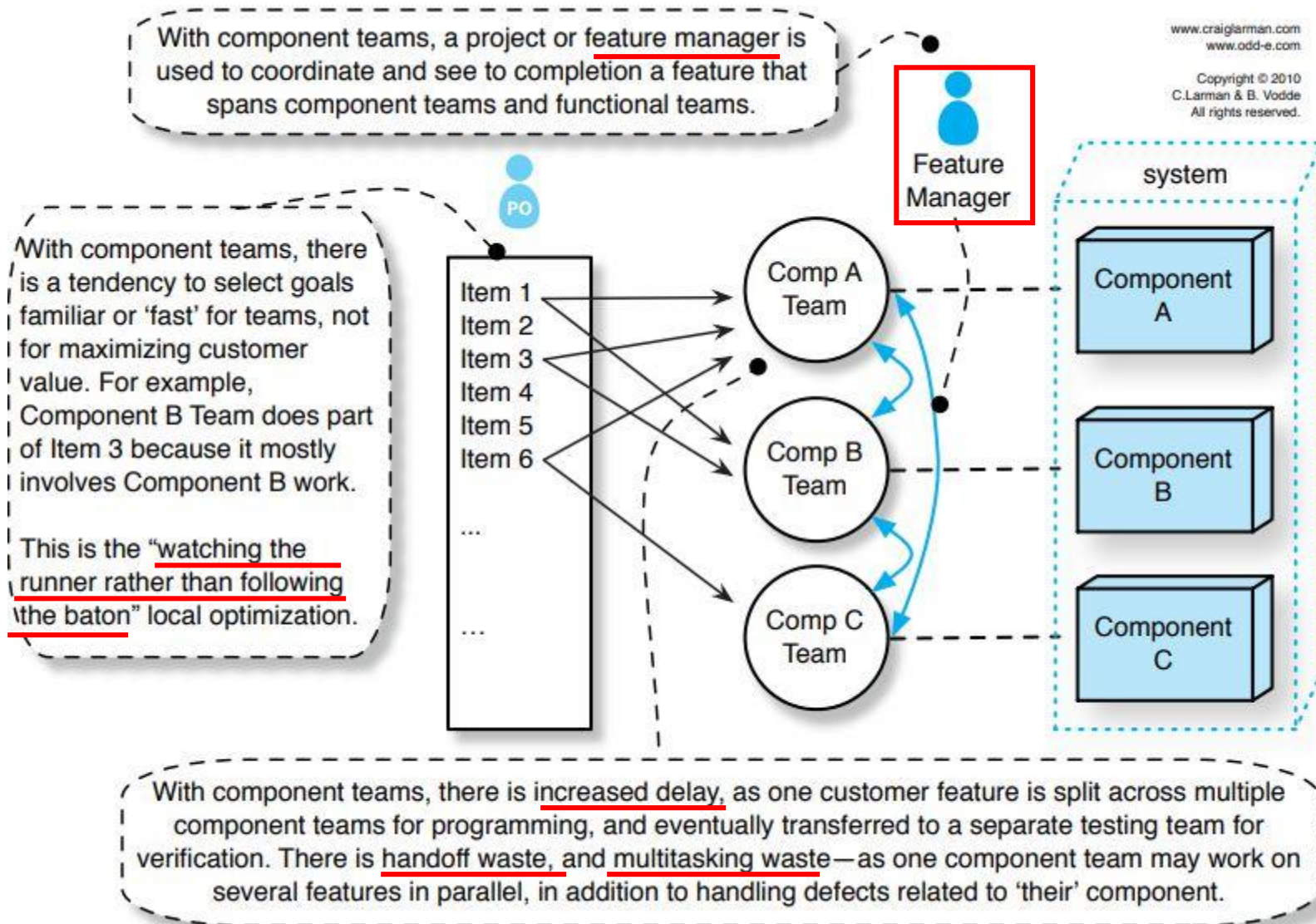


**Classic Example of
Local Optimization**

Component Team

www.craiglarman.com
www.odd-e.com

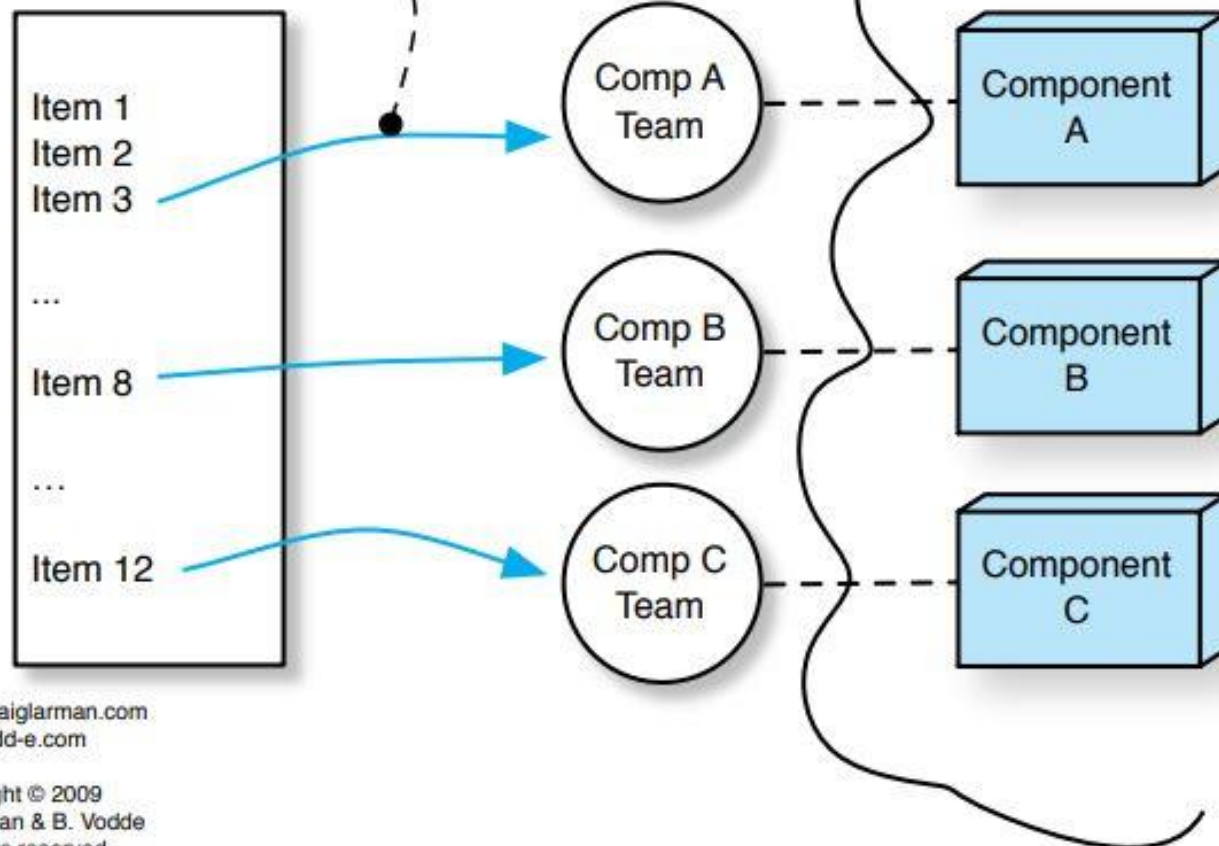
Copyright © 2010
C. Larman & B. Vodde
All rights reserved.



Component Team - Cont.

With component teams, there is a tendency to select goals familiar for people, not for maximizing customer value. For example, Component A Team does Backlog Item 3 because it mostly involves Component A work.

Local Optimization



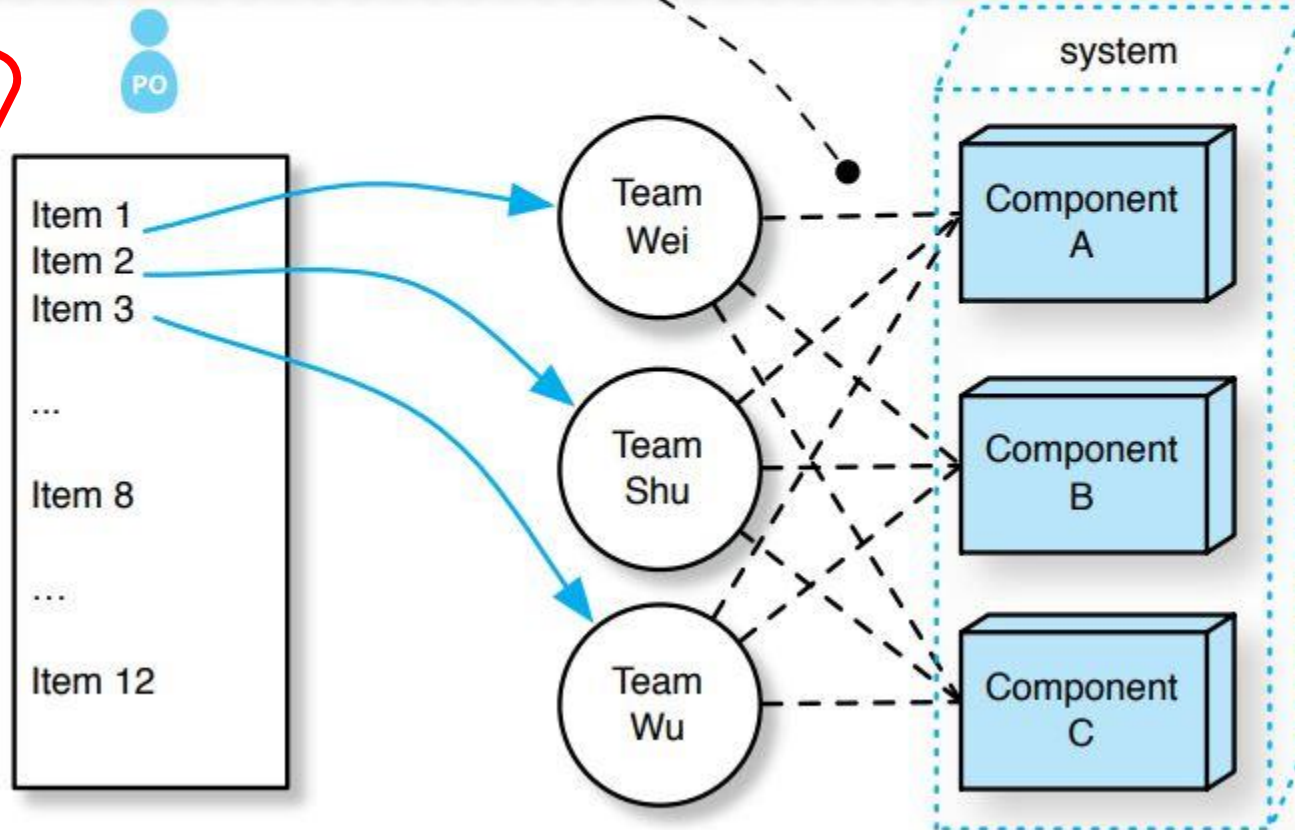
www.craiglarman.com
www.odd-e.com

Copyright © 2009
C. Larman & B. Vodde
All rights reserved.

Feature Team

With feature teams, teams can always work on the highest-value features, there is less delay for delivering value, and coordination issues shift toward the shared code rather than coordination through upfront planning, delayed work, and handoff. In the 1960s and 70s this code coordination was awkward due to weak tools and practices. Modern open-source tools and practices such as TDD and continuous integration make this coordination relatively simple.

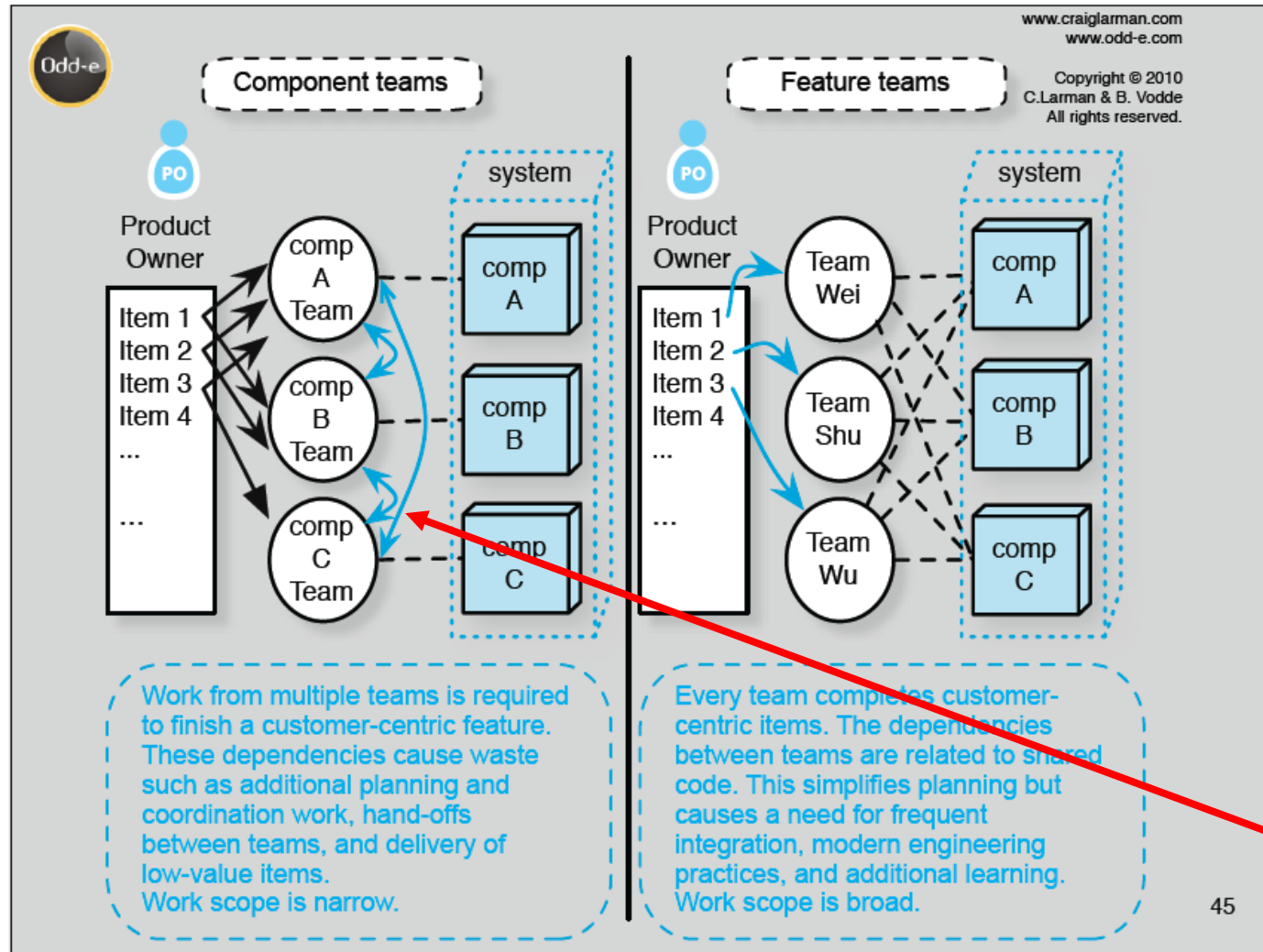
System Optimization



www.craiglarman.com
www.odd-e.com

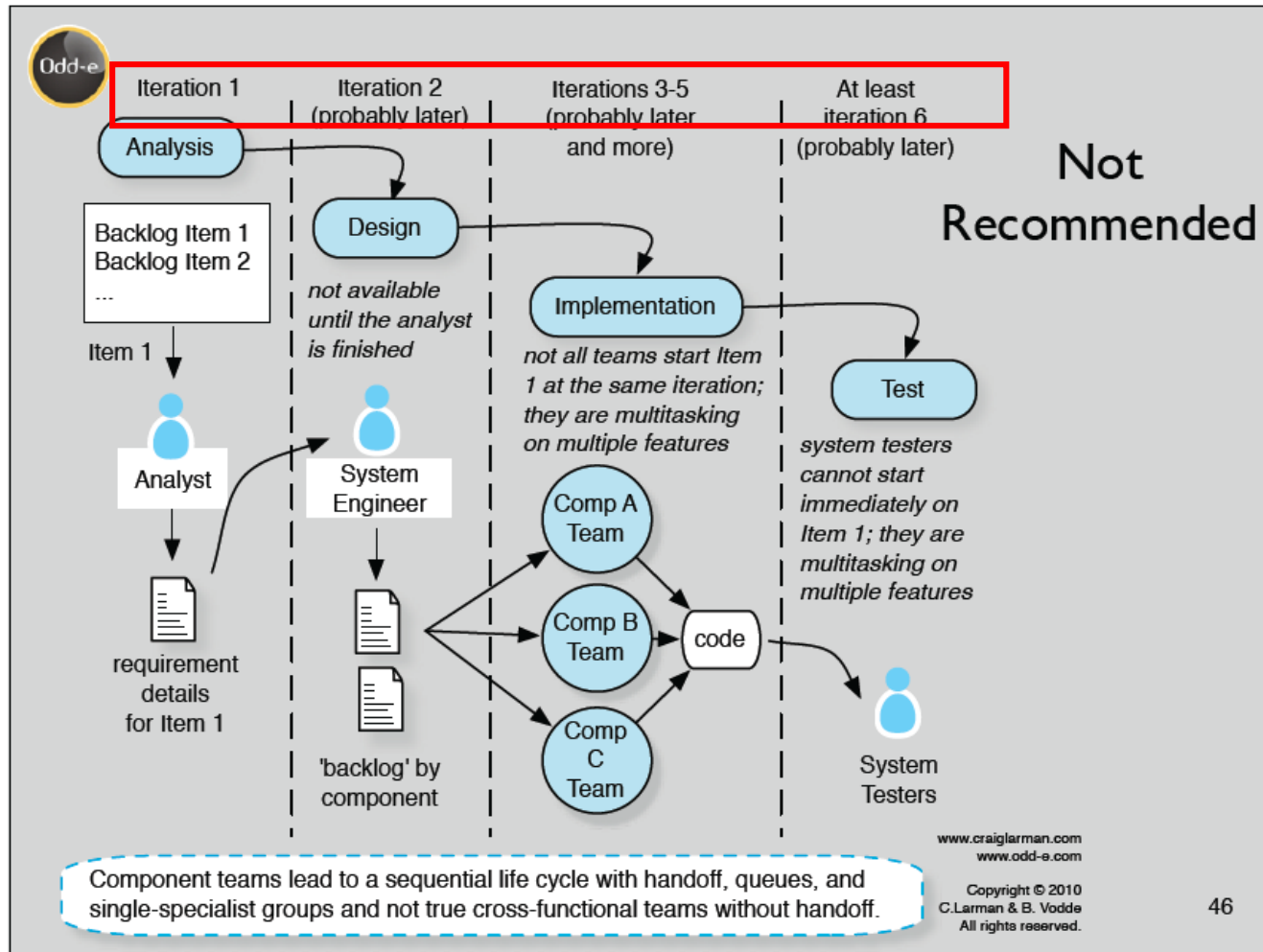
Copyright © 2010
C. Larman & B. Vodde
All rights reserved.

Component & Feature Teams: Side by Side



Sourced from LeSS training materials of C. Larman and B. Vodde

Beware of Mini-Waterfall in Scrum Clothes

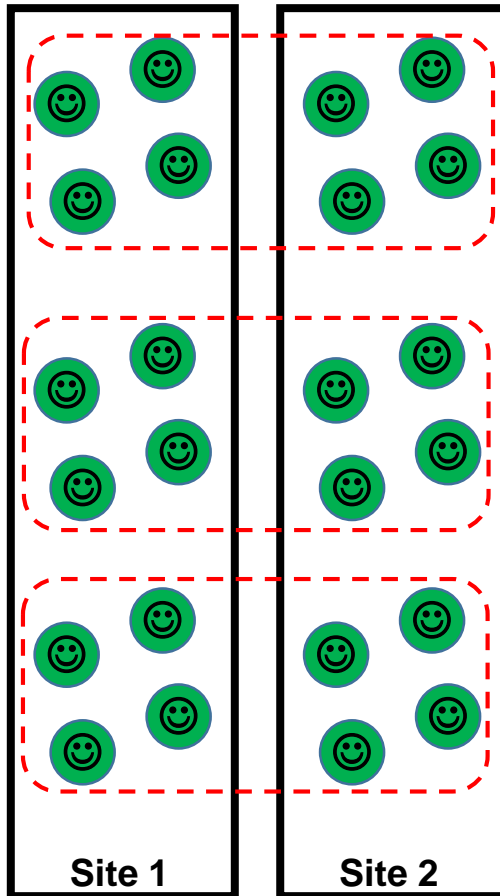


- Fake Scrum
- “Wagile”
- Mini-waterfall

Sourced from LeSS training materials of C. Larman and B. Vodde

Team Location in LeSS

Distribution inside team

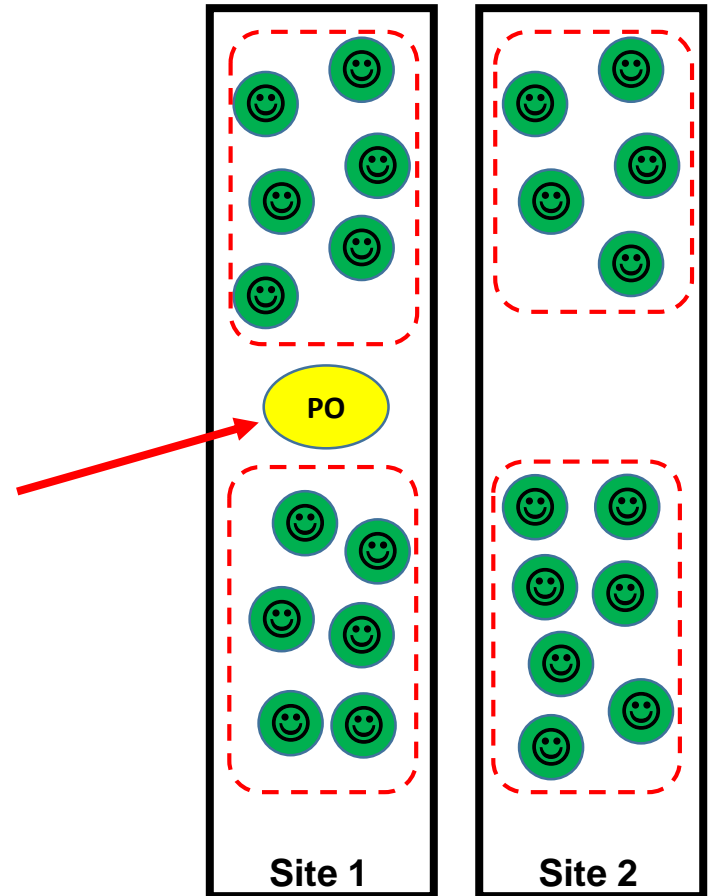


“All good Java developers are in Boston”...“The best SQL experts are in Bangalore”

Really? ☺

AVOID

Team colocation within site



TRY



5 min Break

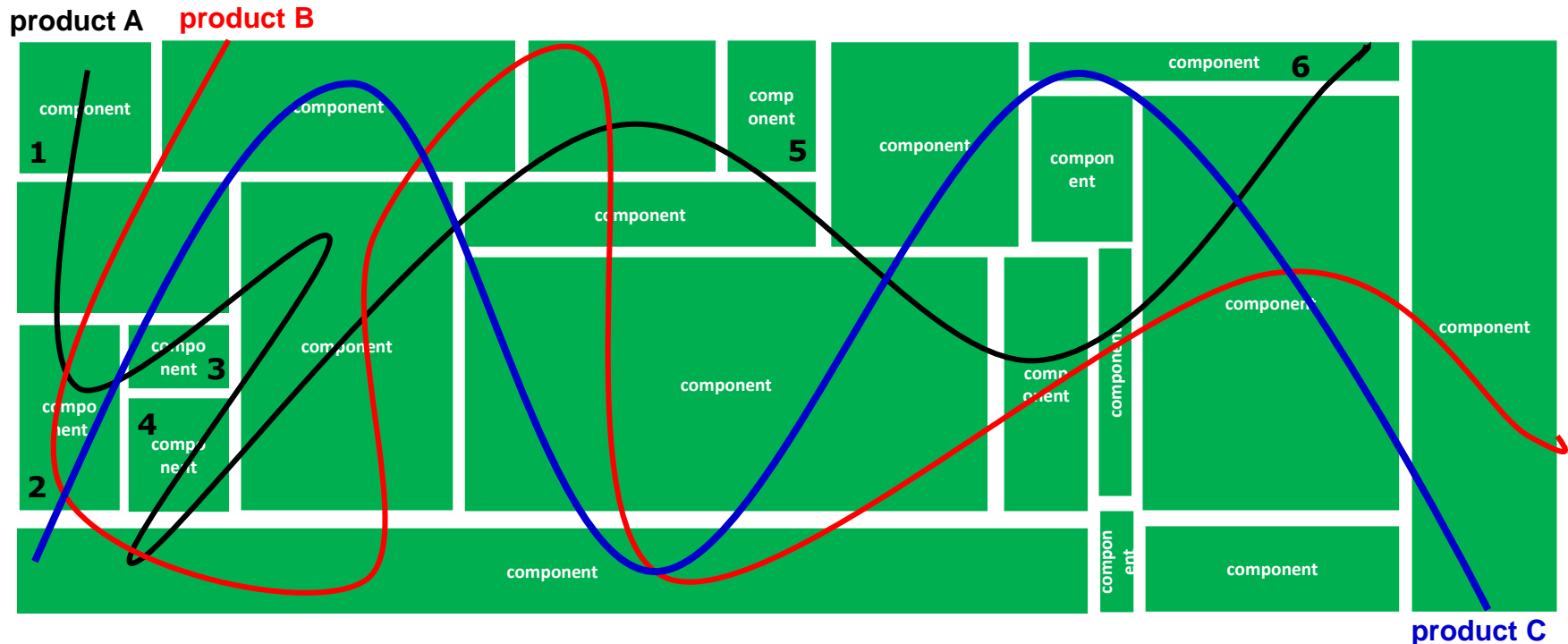
LeSS Product vs Project/Program/Portfolio

Which Product “Dimension” Holds Real Value?

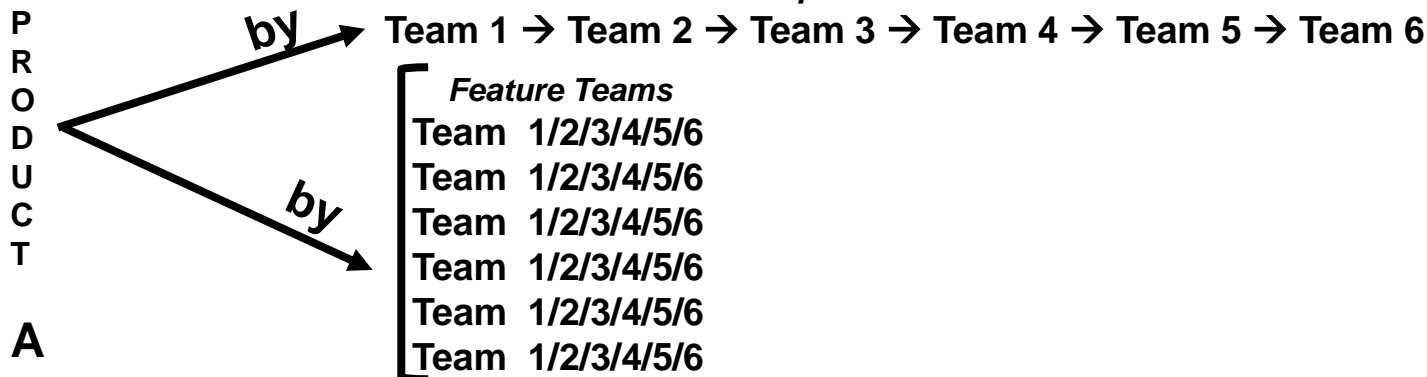
								Component										IOU
								Component										IOU
F	F	F	F	F	F	F	F	Component	F	F	F	F	F	F	F	F	F	IOU
E	E	E	E	E	E	E	E	Component	E	E	E	E	E	E	E	E	E	IOU
A	A	A	A	A	A	A	A	Component	A	A	A	A	A	A	A	A	A	IOU
T	T	T	T	T	T	T	T	Component	T	T	T	T	T	T	T	T	T	IOU
U	U	U	U	U	U	U	U	Component	U	U	U	U	U	U	U	U	U	IOU
R	R	R	R	R	R	R	R	Component	R	R	R	R	R	R	R	R	R	IOU
E	E	E	E	E	E	E	E	Component	E	E	E	E	E	E	E	E	E	IOU
								Component										IOU
								Component										IOU
\$	\$	\$	\$	\$	\$	\$	\$		\$	\$	\$	\$	\$	\$	\$	\$	\$	

How Do you Define Your Product?

Exercise: Bricks and Snakes



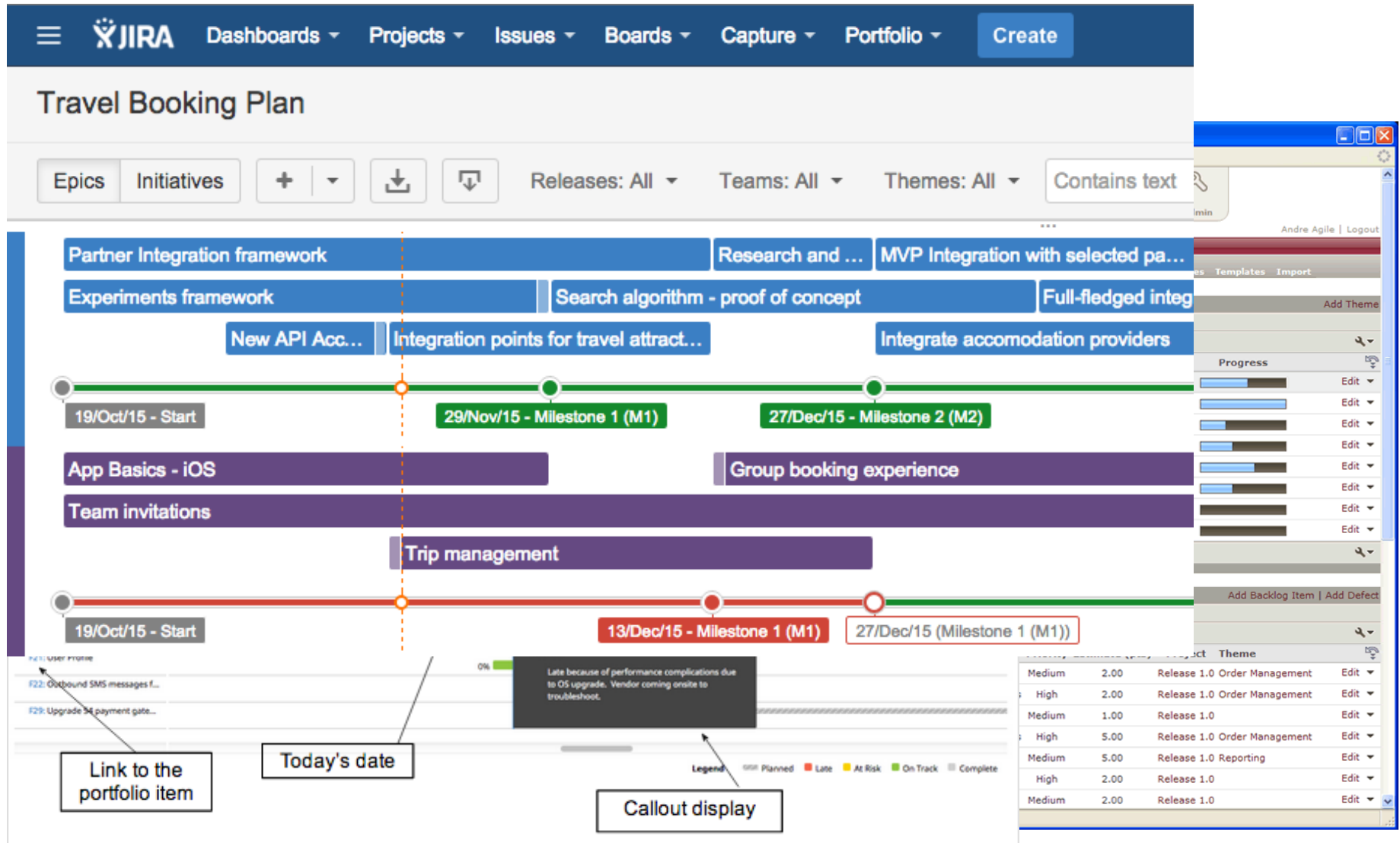
Component Teams



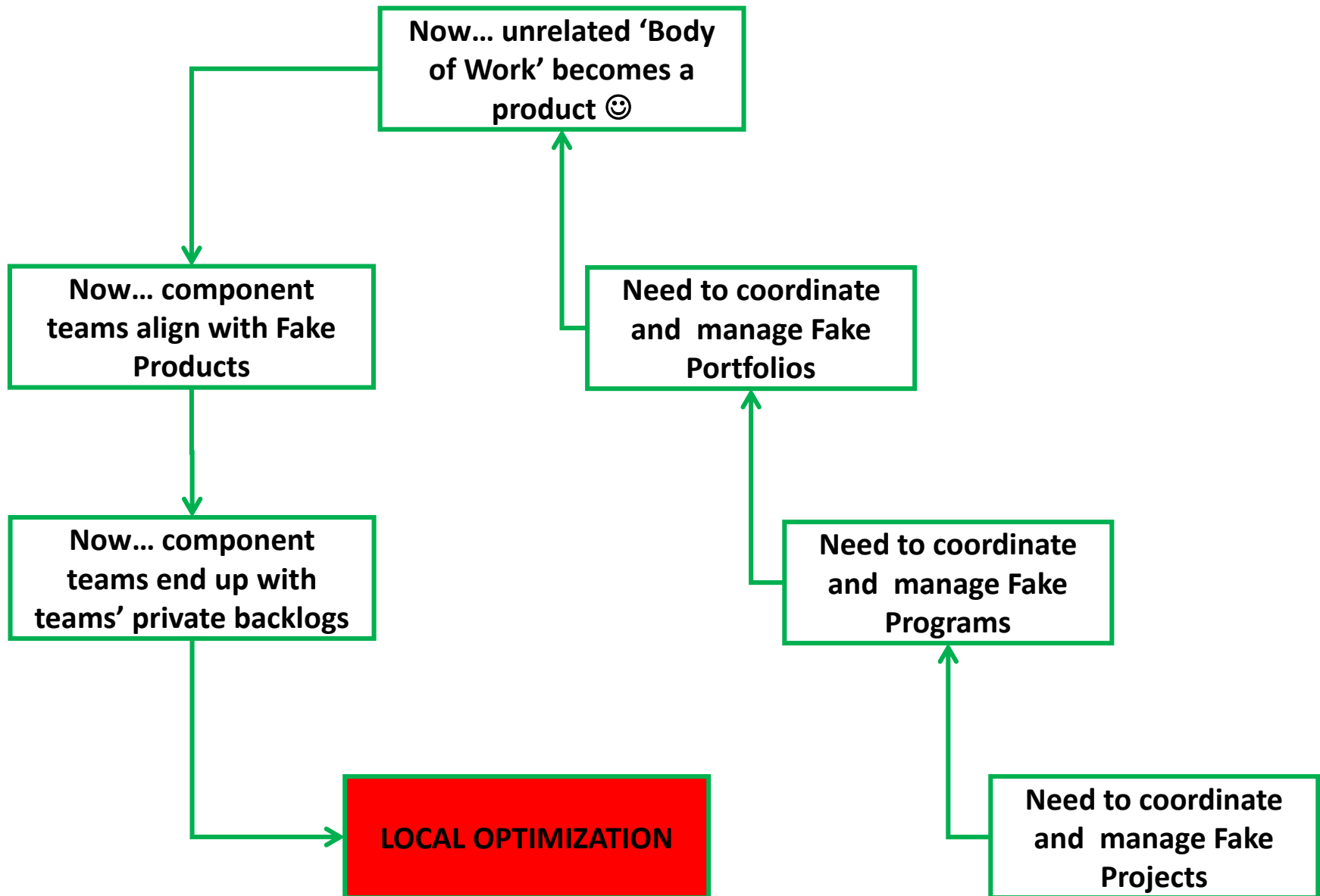
Fake Product Portfolio



Fake Product Portfolio



Fake Product Portfolio



Self-Inflicted Wound: Local Optimization



Requires complex management and reporting structure with multiple translation layers of management bureaucracy

fake portfolio

fake program

fake program

fake project

fake project

fake project

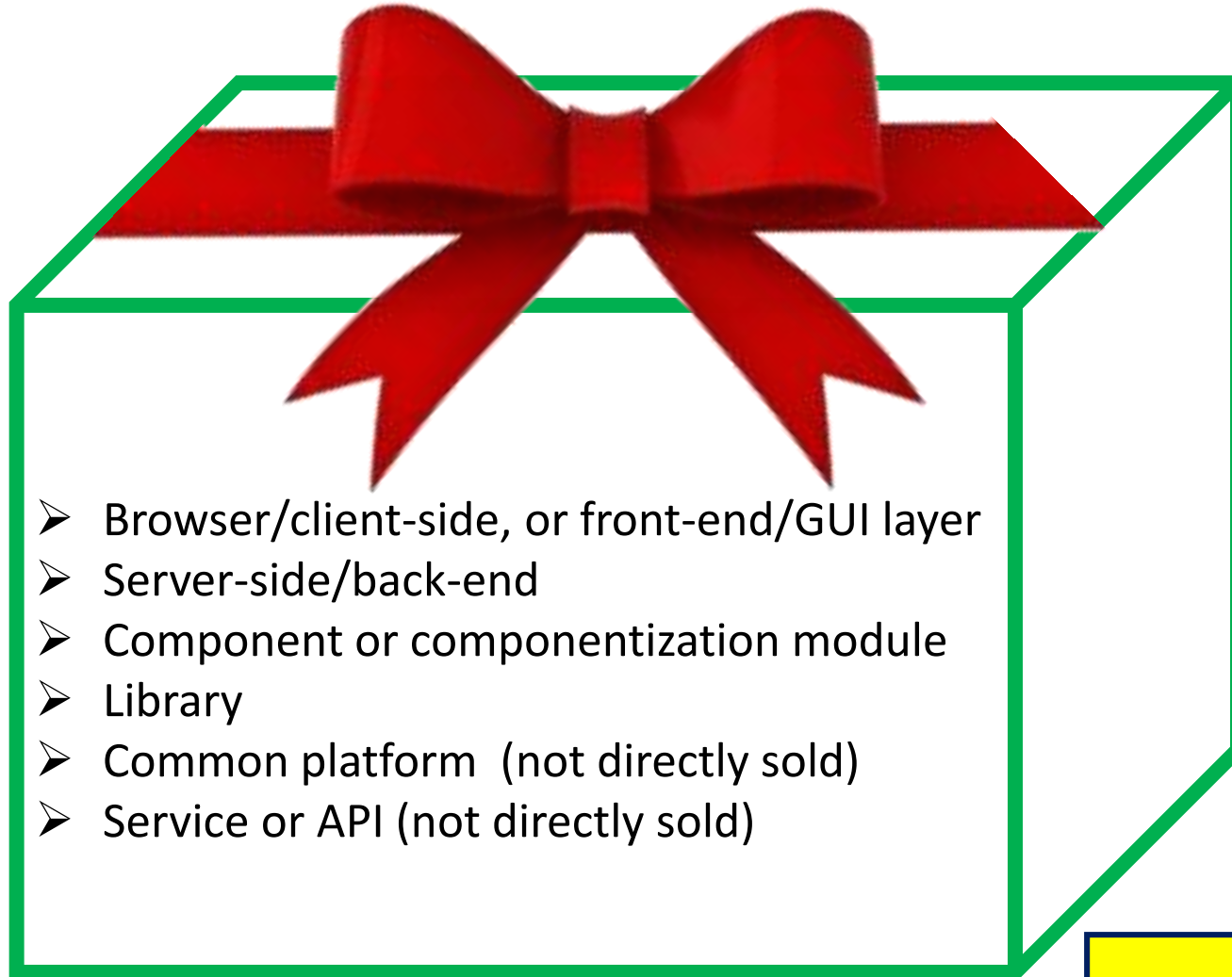
fake project

fake project

fake project

fake project

Moment of Self-Assessment



- Browser/client-side, or front-end/GUI layer
- Server-side/back-end
- Component or componentization module
- Library
- Common platform (not directly sold)
- Service or API (not directly sold)

Q: Do you consider these as your 'products' or 'projects'?

Product Definition in LeSS

- Organizing by Business Value and Customer Centricity
- **Expanding** Product as wide as possible
- **Constraining** Product to keep it reasonably manageable
- Maintaining single Product Backlog (not multiple, private team backlogs)



PRODUCT

Group Activity (Product Definition)

Exercise

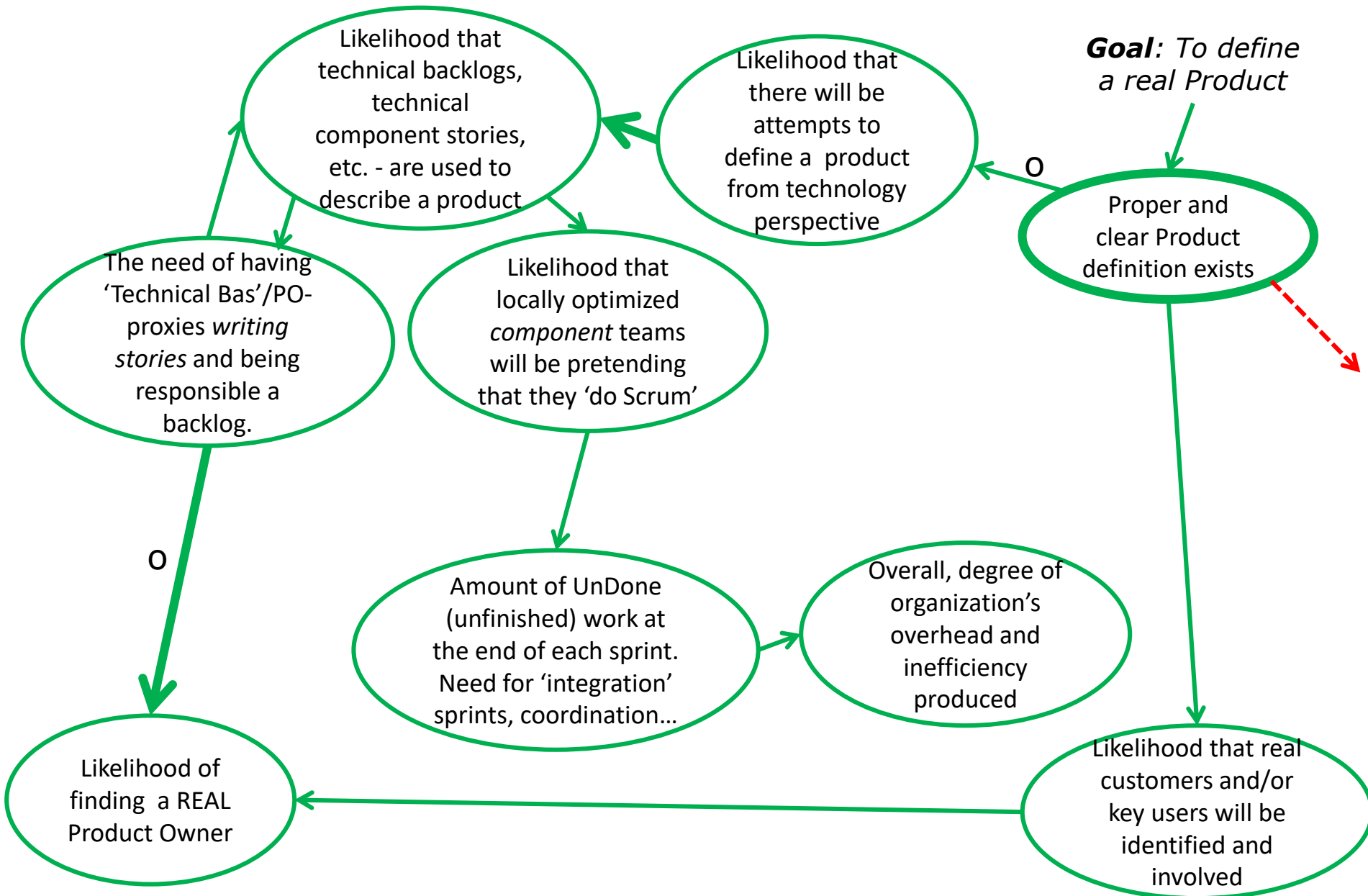
Duration: 10 min

Class: in-groups, brainstorm some of the most common pitfalls in defining a **product**. Work with provided system variables to create a model.

Method/Tool: System Modeling with CLD

Instructor: review with Class

Suggested Solution (Product Definition)



Day 3

Group Activity (Fake Projects/gams)

Exercise

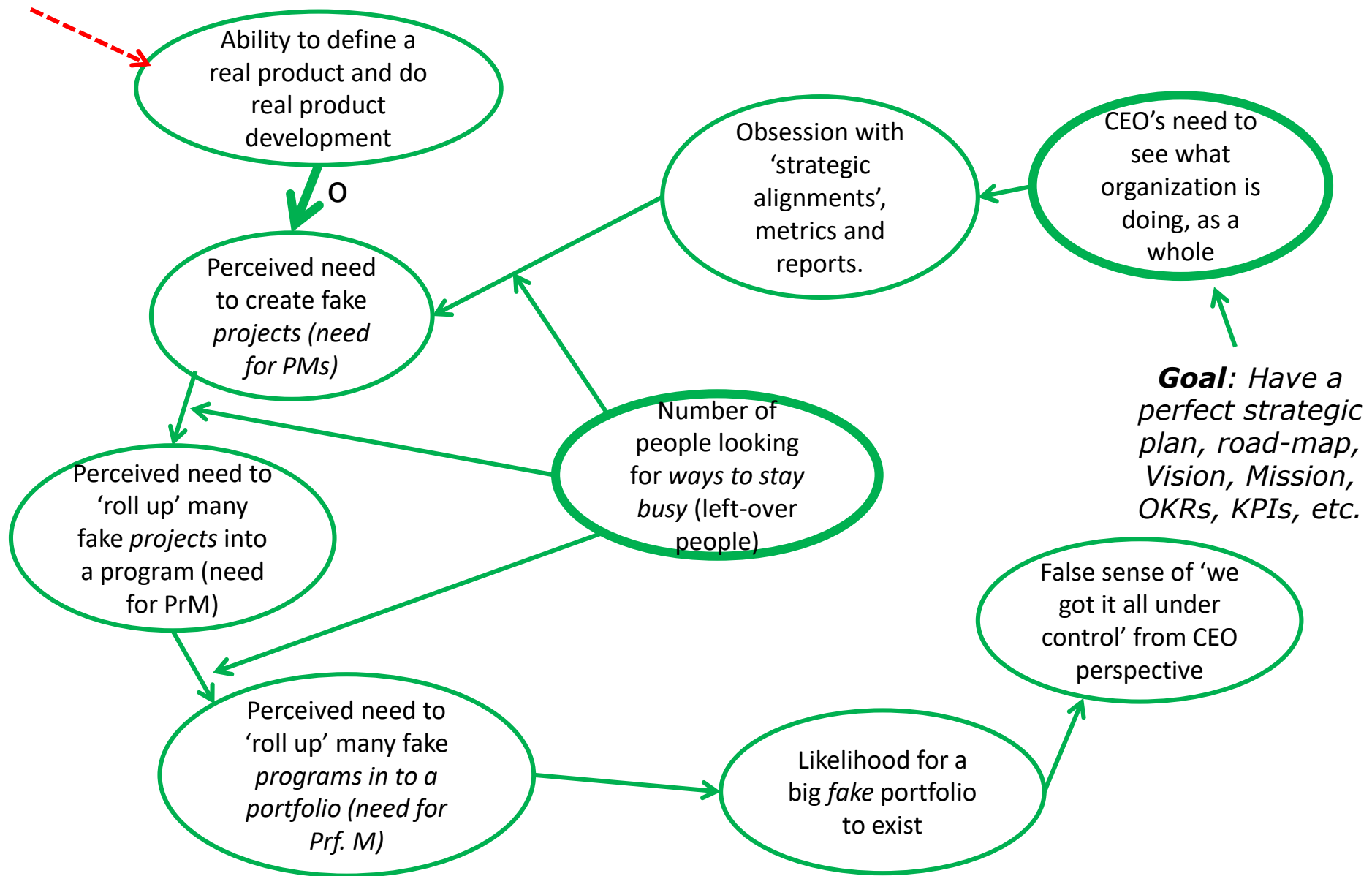
Duration: 10 min

Class: in-groups, brainstorm some problems that could be caused by ***fake projects, programs and portfolios***. Work with provided system variables to create a model.

Method/Tool: System Modeling with CLD

Instructor: review with Class

Suggested Solution (Fake Projects/gams)



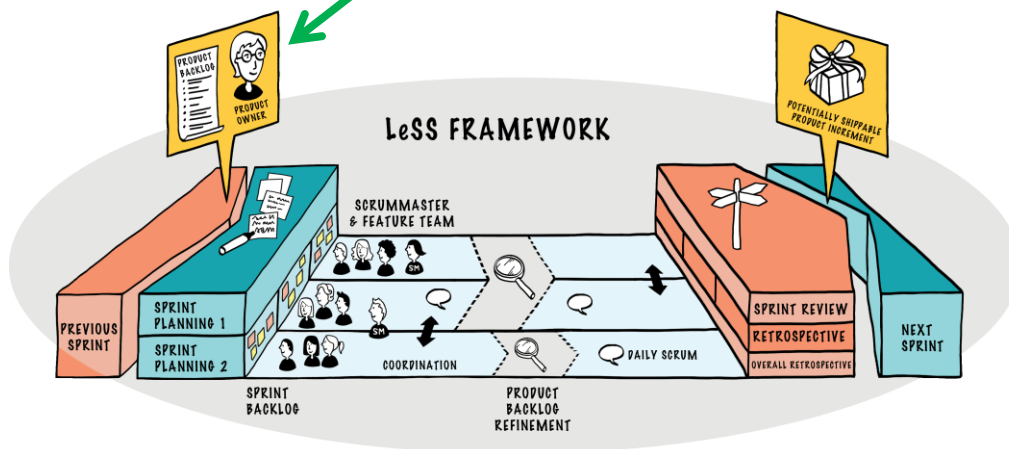
LeSS Roles

LeSS Responsibilities

LeSS Relationships

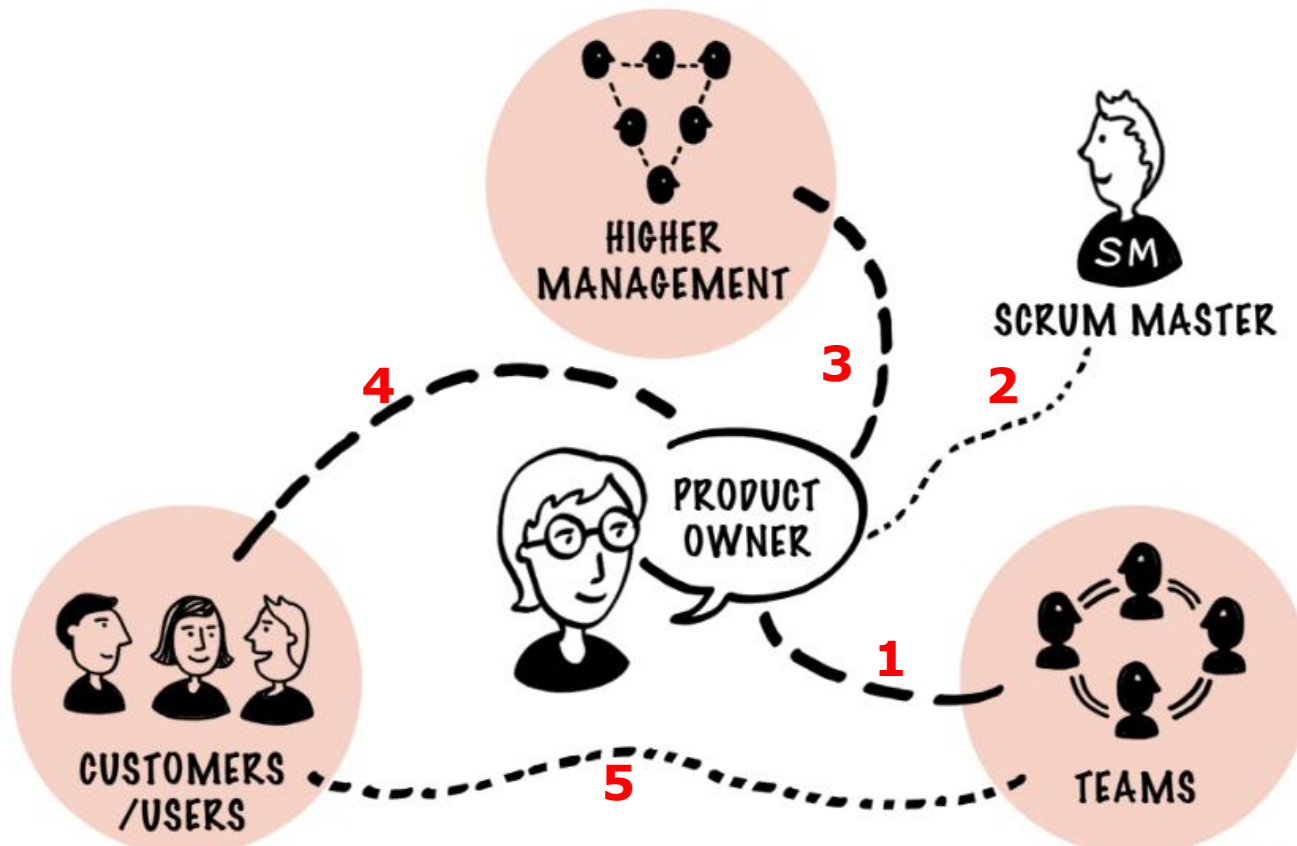
Product Owner: Who Is The Best Candidate?

- External: Product Manager
- Internal: Manager of Business Function



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

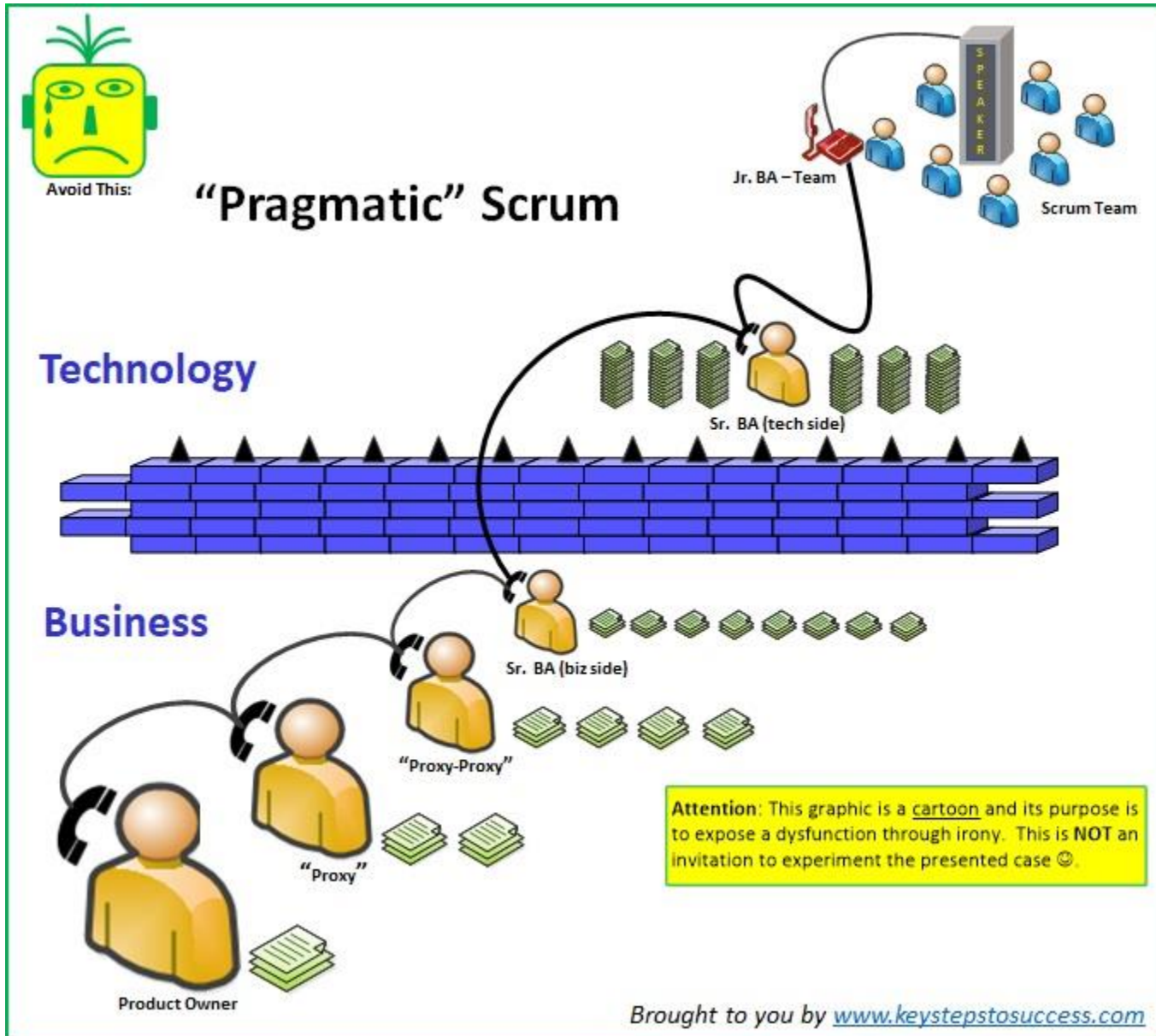
Product Owner: 5 Relationships



Sourced from LeSS training materials of C. Larman and B. Vodde

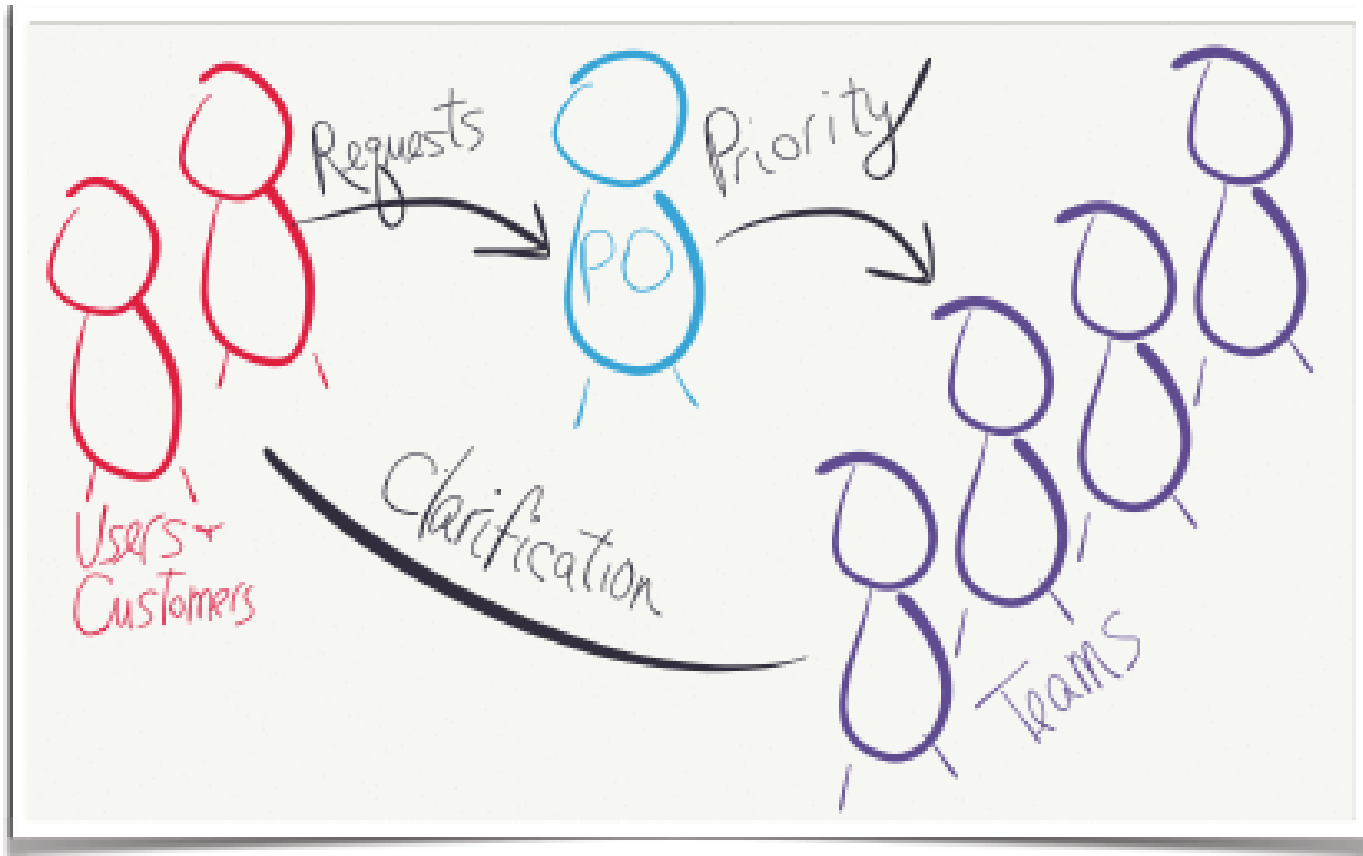
<http://less.works> 

Product Owner: Avoid This Dysfunction



Product Owner: Prioritization vs. Clarification

Clarification vs Prioritization



Sourced from LeSS training materials of C. Larman and B. Vodde

159

159

No Team Output Owner (“TOO”)



0:05 / 9:42



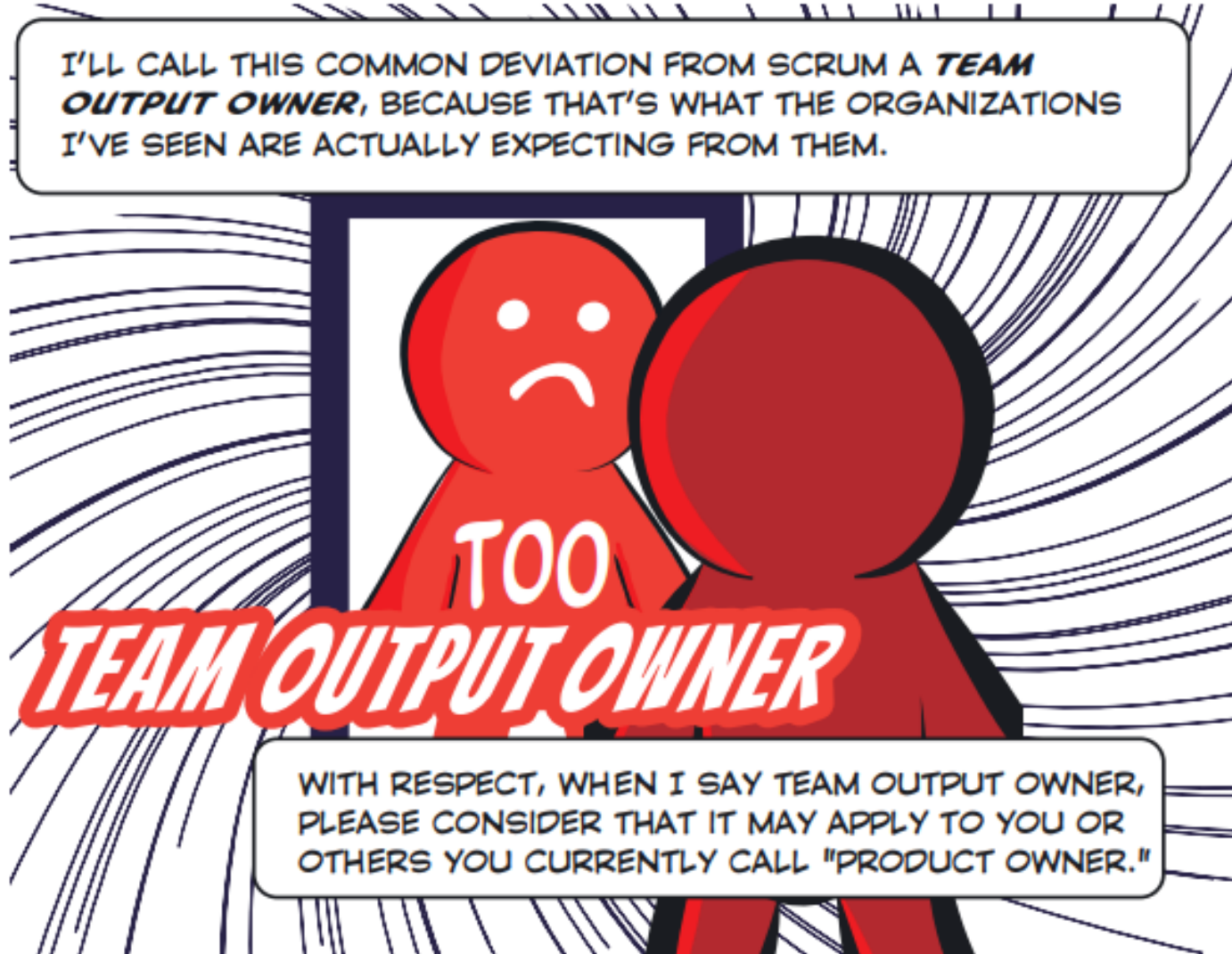
Why "Scrum" Isn't Making Your Organization Agile: Harmful Misconceptions About Product Owner Role



MJ (MICHAEL JAMES)
has written many things,
including the Scrum Master Checklist at
<http://ScrumMasterChecklist.org>
Email him at mj@seattlescrum.com
or tweet him at [@michaeldotjames](https://twitter.com/michaeldotjames)

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

Team Output Owner (TOO): “More-More-More”

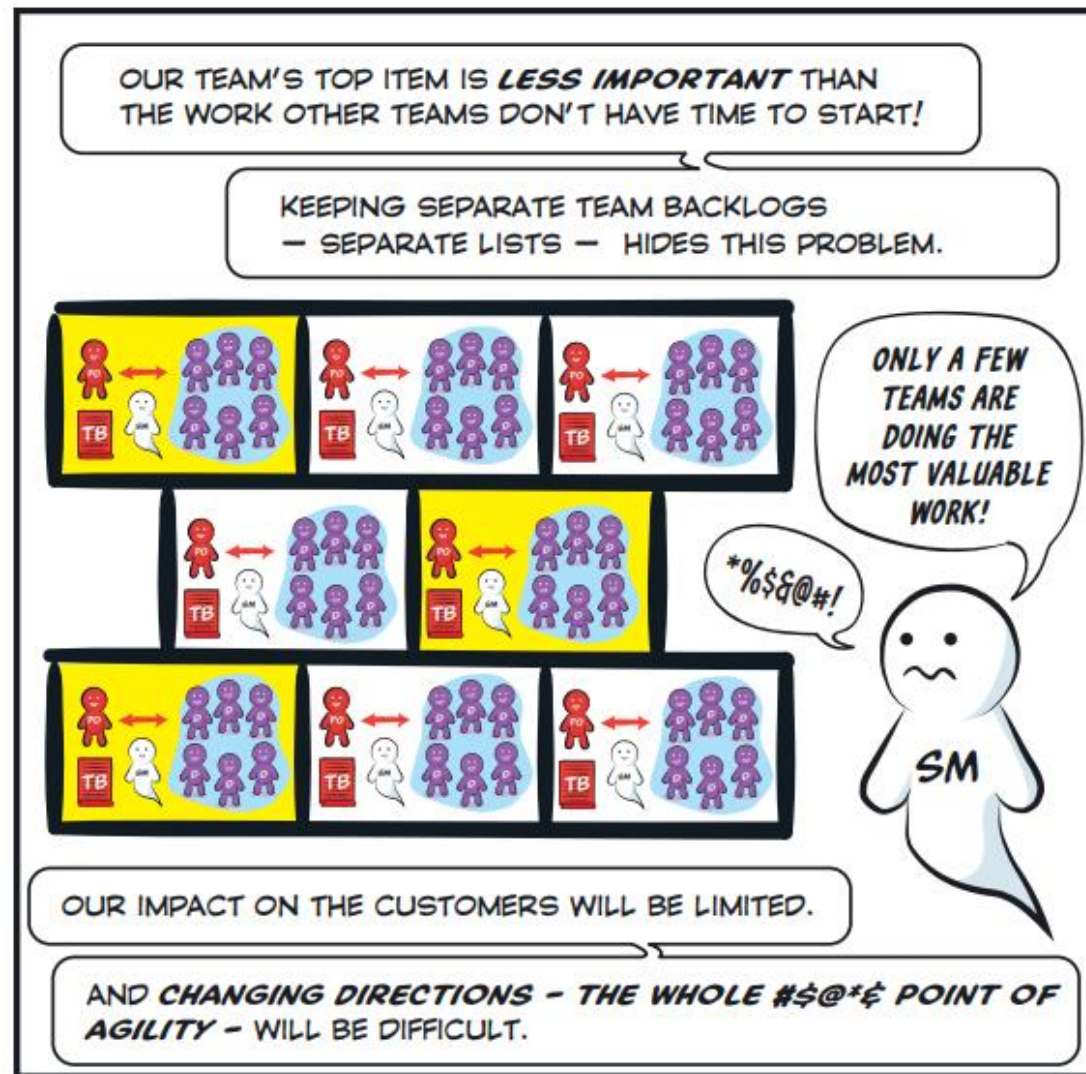


(C) 2019 Michael James (mj). All rights reserved.

4

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

Team Output Owner (TOO): Local Optimization

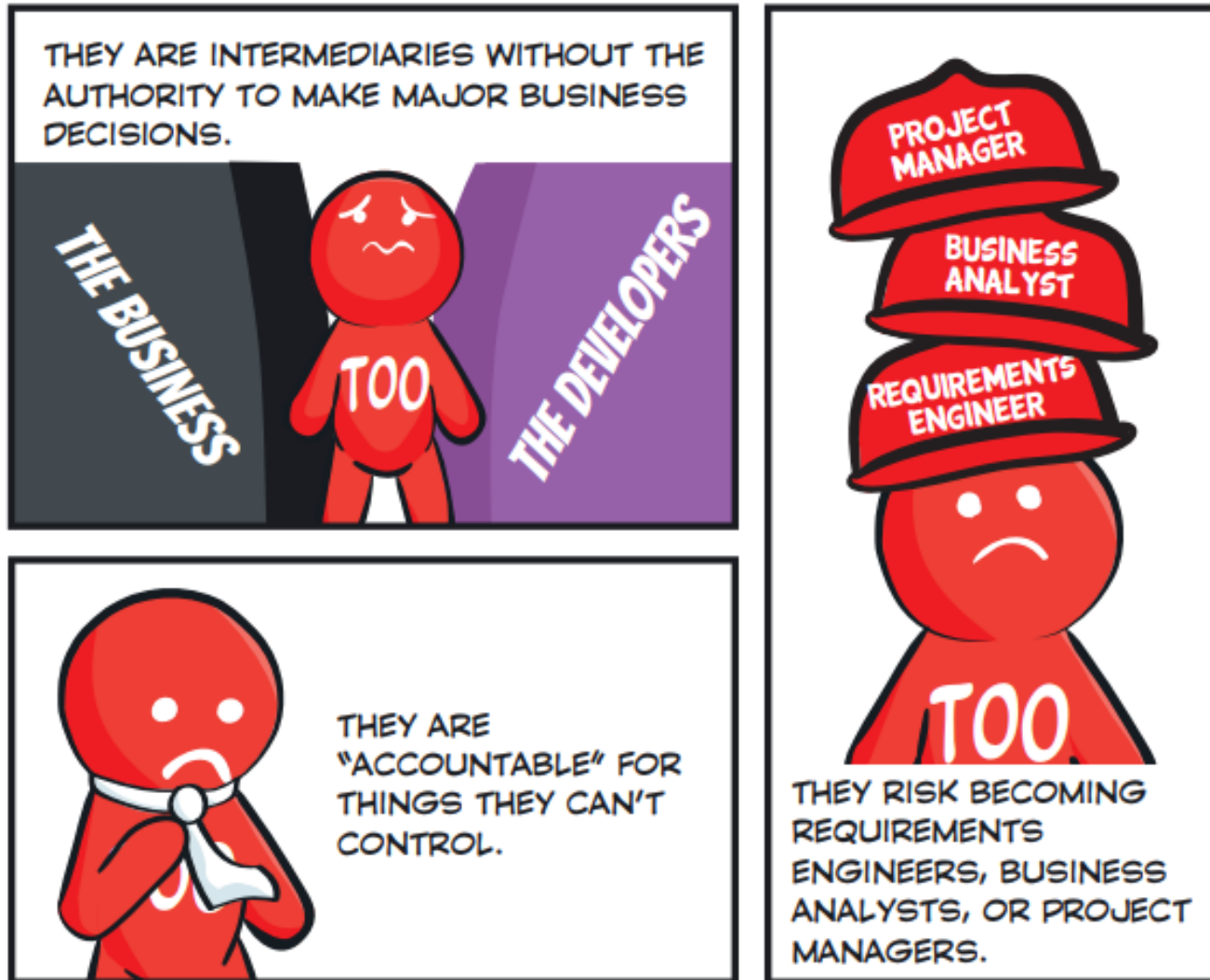


17

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

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

Team Output Owner (TOO): “Accountability”

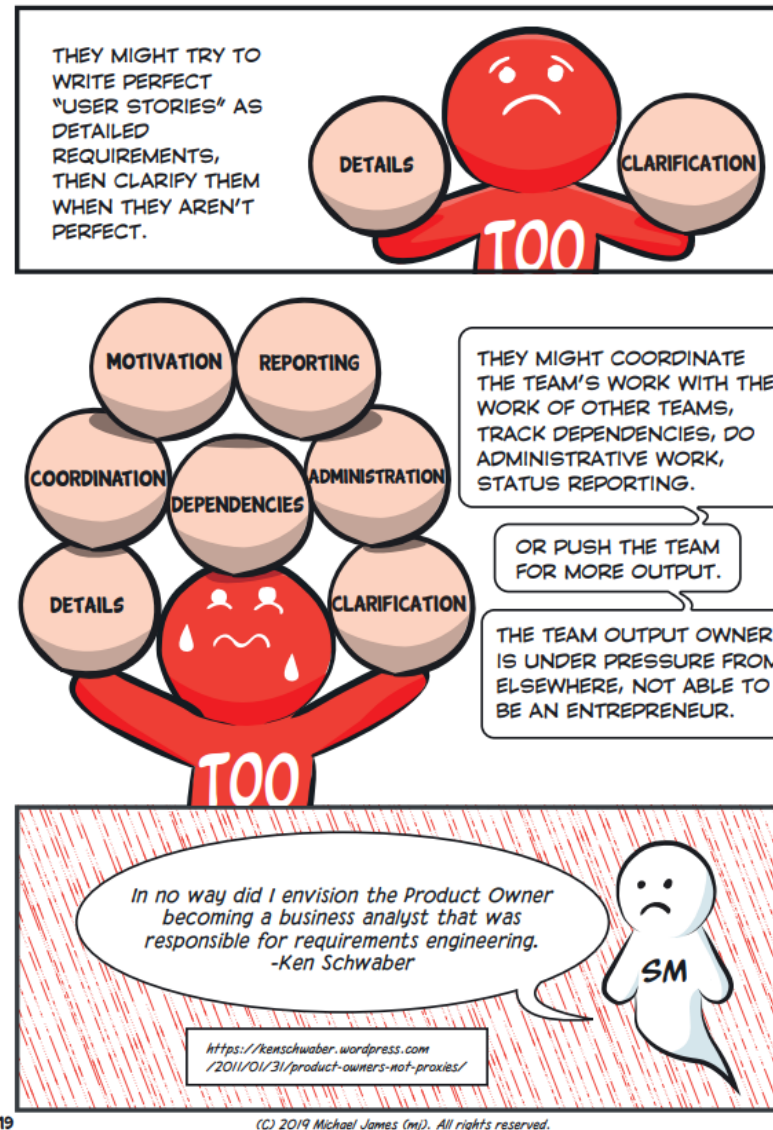


(C) 2019 Michael James (mj). All rights reserved.

18

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

Team Output Owner (TOO) Local Optimization



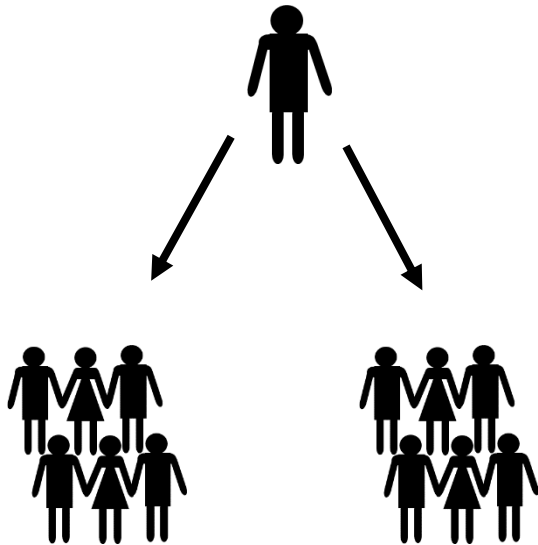
<https://seattlescrum.com/downloads/Why-Scrum-Isnt-Making-Your-Company-Very-Agile-v2.pdf>

ScrumMaster



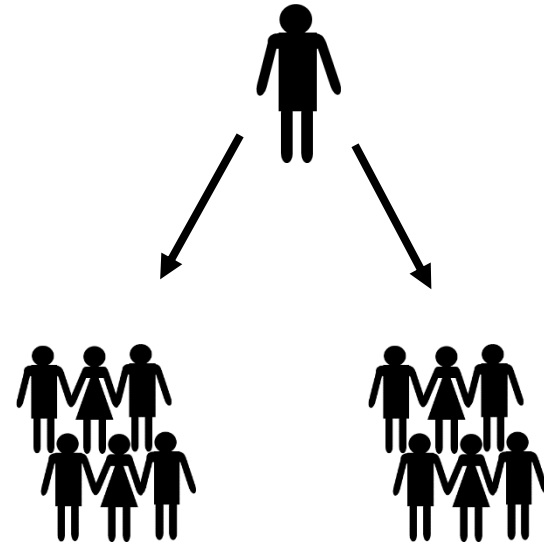
Scrum Master: Style of Engagement

Project Manager



Coordination

Scrum Master



Facilitation



Scrum Master: What is Misunderstood?

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.

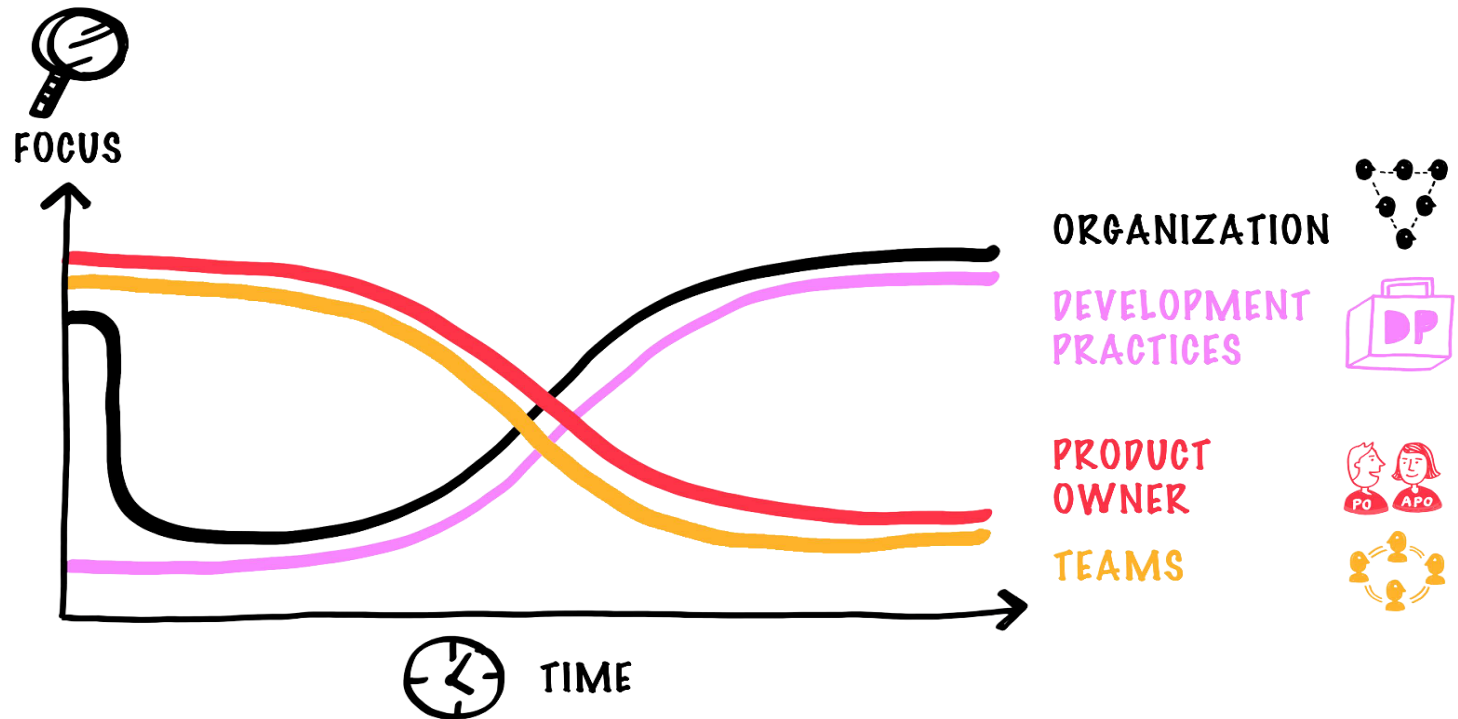
Q: Is there HR Support for Scrum Master role?

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

Scrum Master: Focus



<http://less.works> 

Group Activity (Scrum Master Role)

Exercise

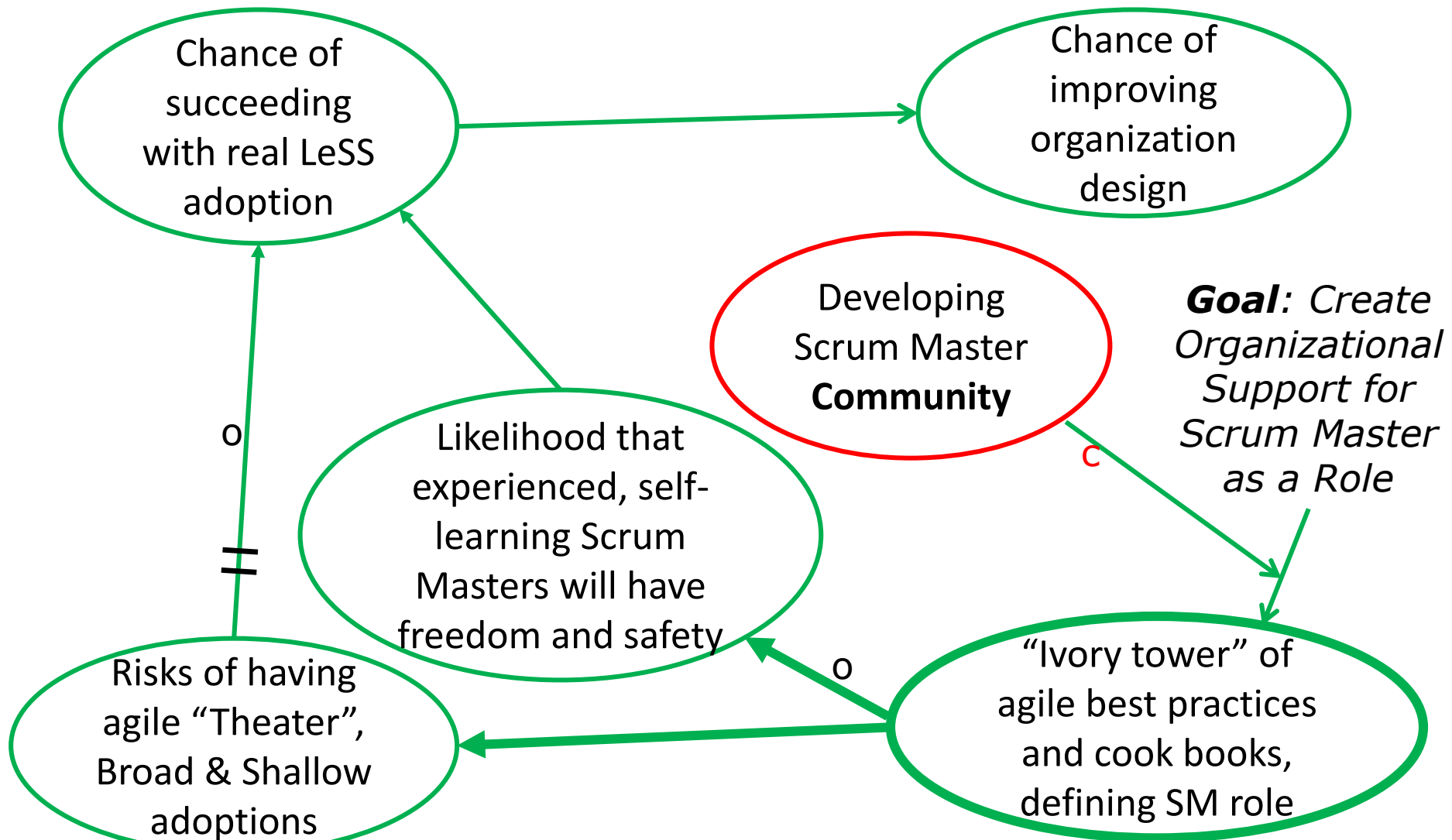
Duration: 10 min

Class: in-groups, brainstorm typical anti-patterns associated with misunderstanding **Scrum Master** profession ; use post-it notes to capture discoveries

Method/Tool: System Modeling with CLD

Instructor: review with Class

Suggested Solution (Scrum Master Role)



Note: This is a suggested solution. Not one-and-only solution.

Team



<http://less.works> 

Team Maturation

Types of teams


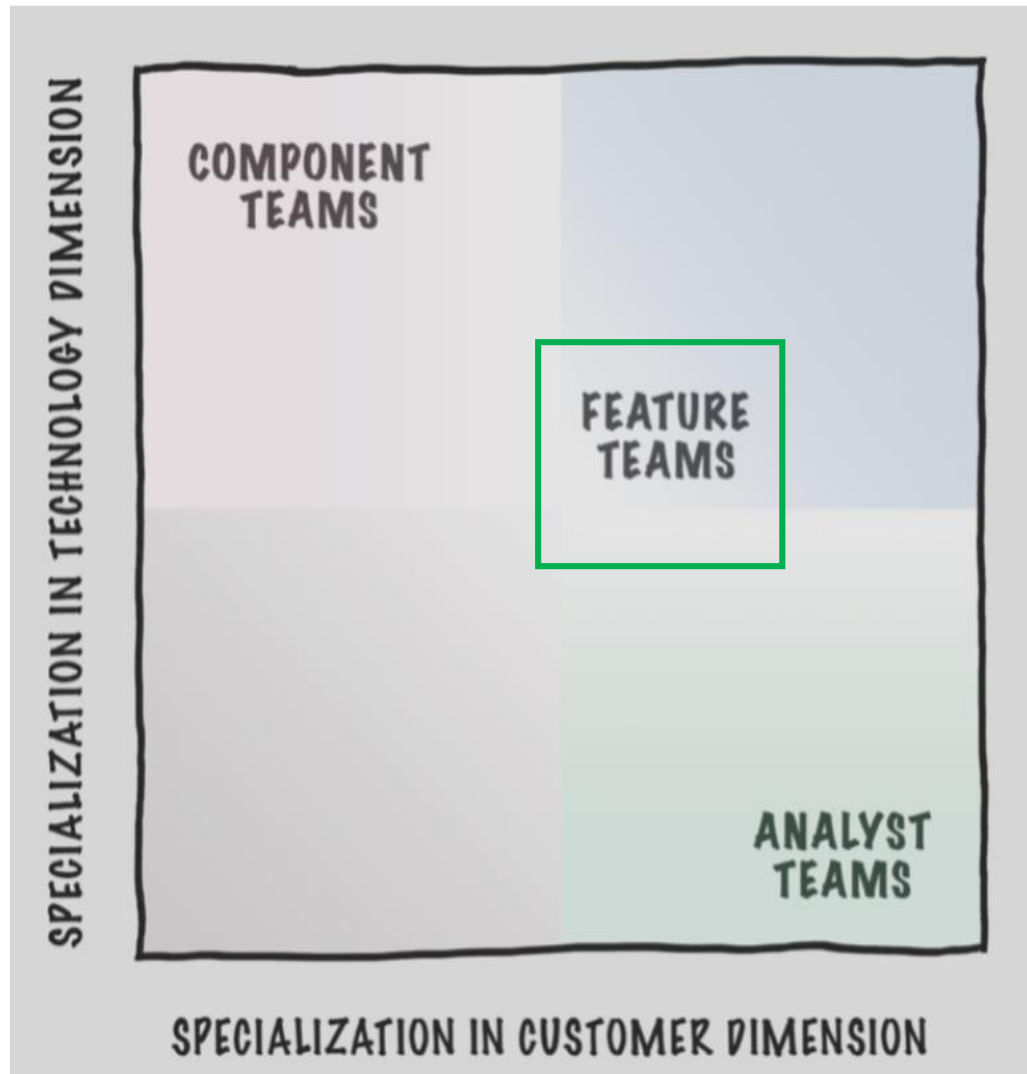
Setting overall direction				
Designing the team and its organizational context	Management Responsibility			
Monitoring and managing work process and progress		Team's Own Responsibility		
Executing the team task				
	Manager-led teams	Self-Managing teams	Self-Designing teams	Self-Governing teams

Figure 1. Types of Teams.

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

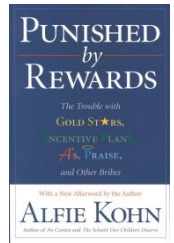
Feature Team Specialization



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

Real Team vs. Fake Team

Want MORE of	Want LESS of
Stable teams that stay together and work on features for a long time	Temporary groups, created on a feature or project
Share all responsibility, as a team	Individual responsibility only for 'their' part based on specialty
Self-management and self-direction	PM-controlled work
Simple, flattened organizations. No matrix	Matrix organizational with resource pools
Full dedication a team	Resource sharing and part time "membership"



Real Team vs. Fake Team

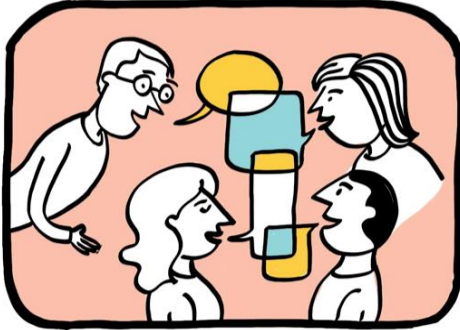
Want MORE of	Want LESS of
Responsible Teams	Single-Function Roles
Focus on Customer	Documentation and “Contracts”
Teams that understand Requirements	Dedicated Analysts
Team ownership, inspection and adaptation	“Best practices” and prescriptive manuals
“Owning”	“Renting”
Well understood existing/simple roles and processes	Additional, vaguely defined roles and processes



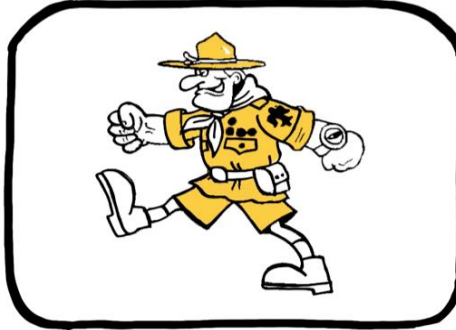
5 min Break

Coordination in LeSS (Outside of LeSS events)

Relationships in LeSS



JUST TALK



SCOUT



OPEN SPACE



TRAVELER



COMMUNITIES



COMPONENT MENTOR

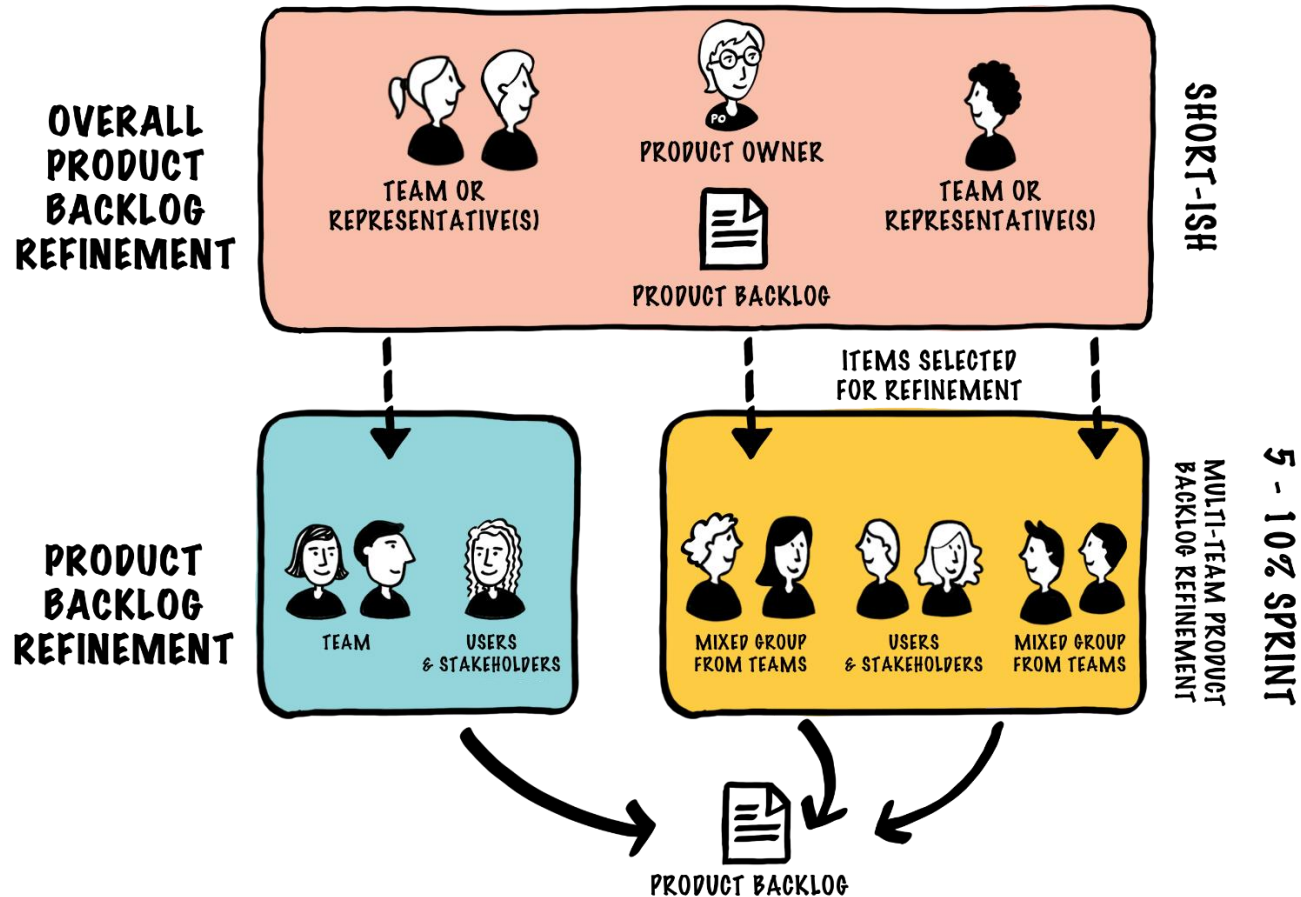
<https://less.works/resources/graphics/index.html>

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

LeSS Events

LeSS Events

LeSS PRODUCT BACKLOG REFINEMENT

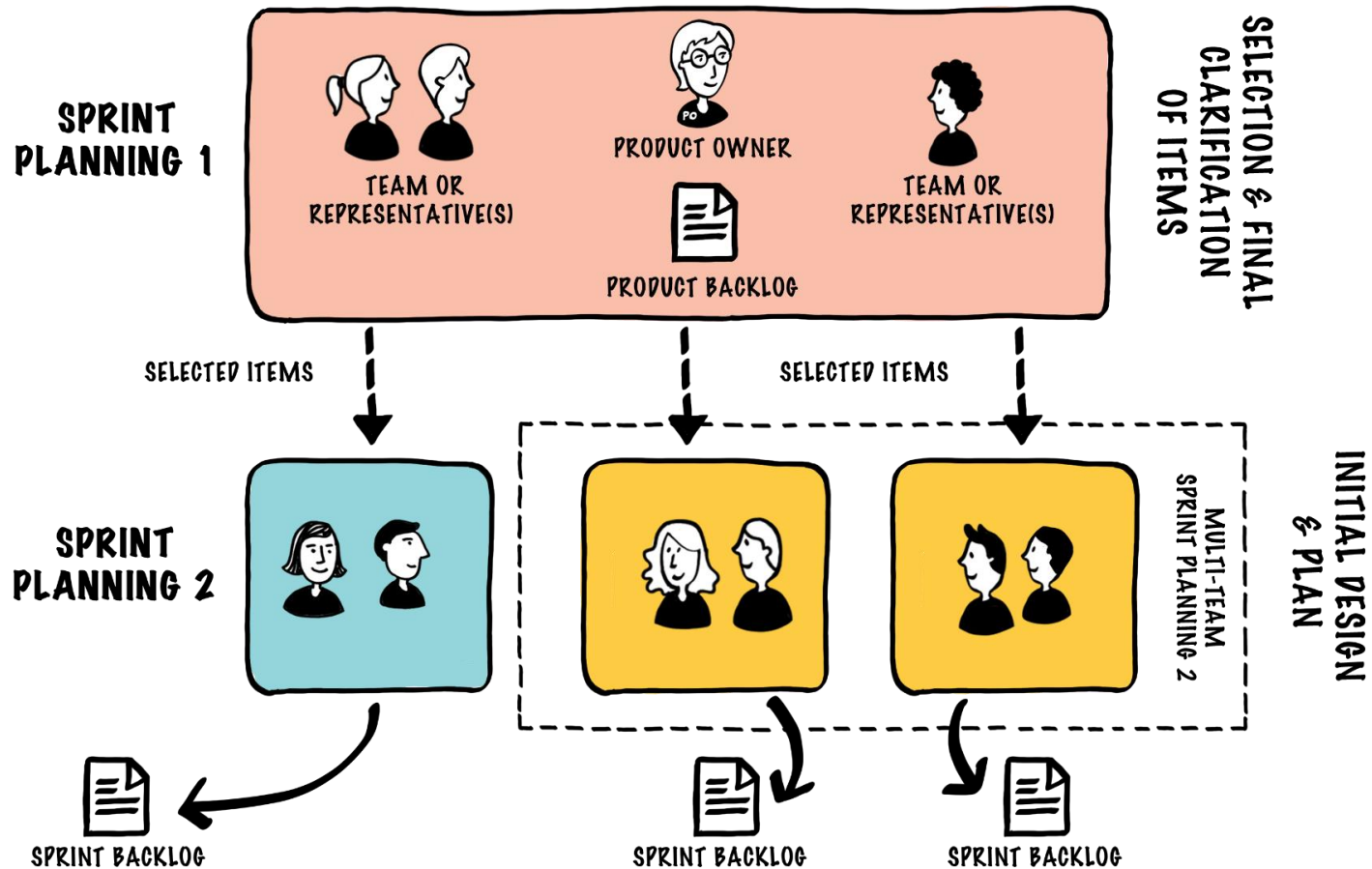


Q: Why Scrum Masters are NOT Representatives in Overall PBR?

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

LeSS Events

LeSS SPRINT PLANNING



Q: Why Scrum Masters are NOT Representatives Sprint Planning 1?

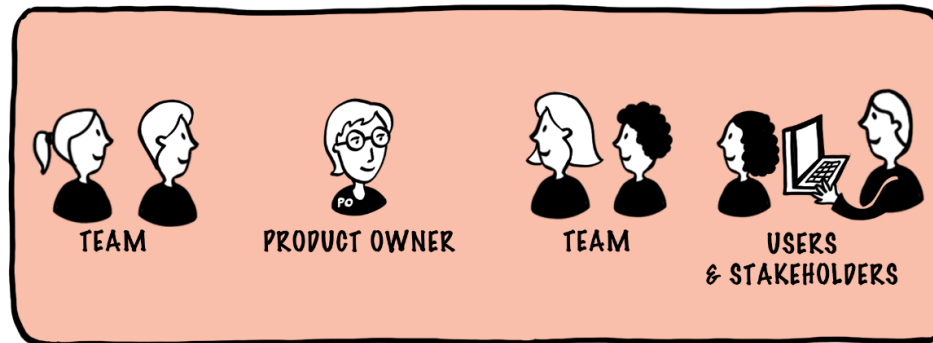
Q: Why SP2 does not require PO? Why multi-team SP2?

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

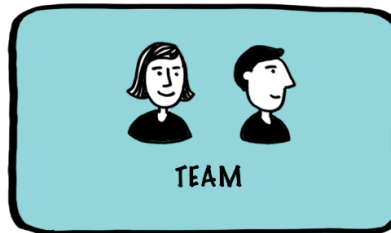
LeSS Events

LeSS SPRINT REVIEW & RETROSPECTIVE

SPRINT REVIEW



TEAM RETROSPECTIVE



OVERALL RETROSPECTIVE



Q: Why Scrum Masters are a part of Overall Retrospective?

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

Technical Excellence LeSS

Technical Excellence

DoD



POTENTIALLY SHIPPABLE

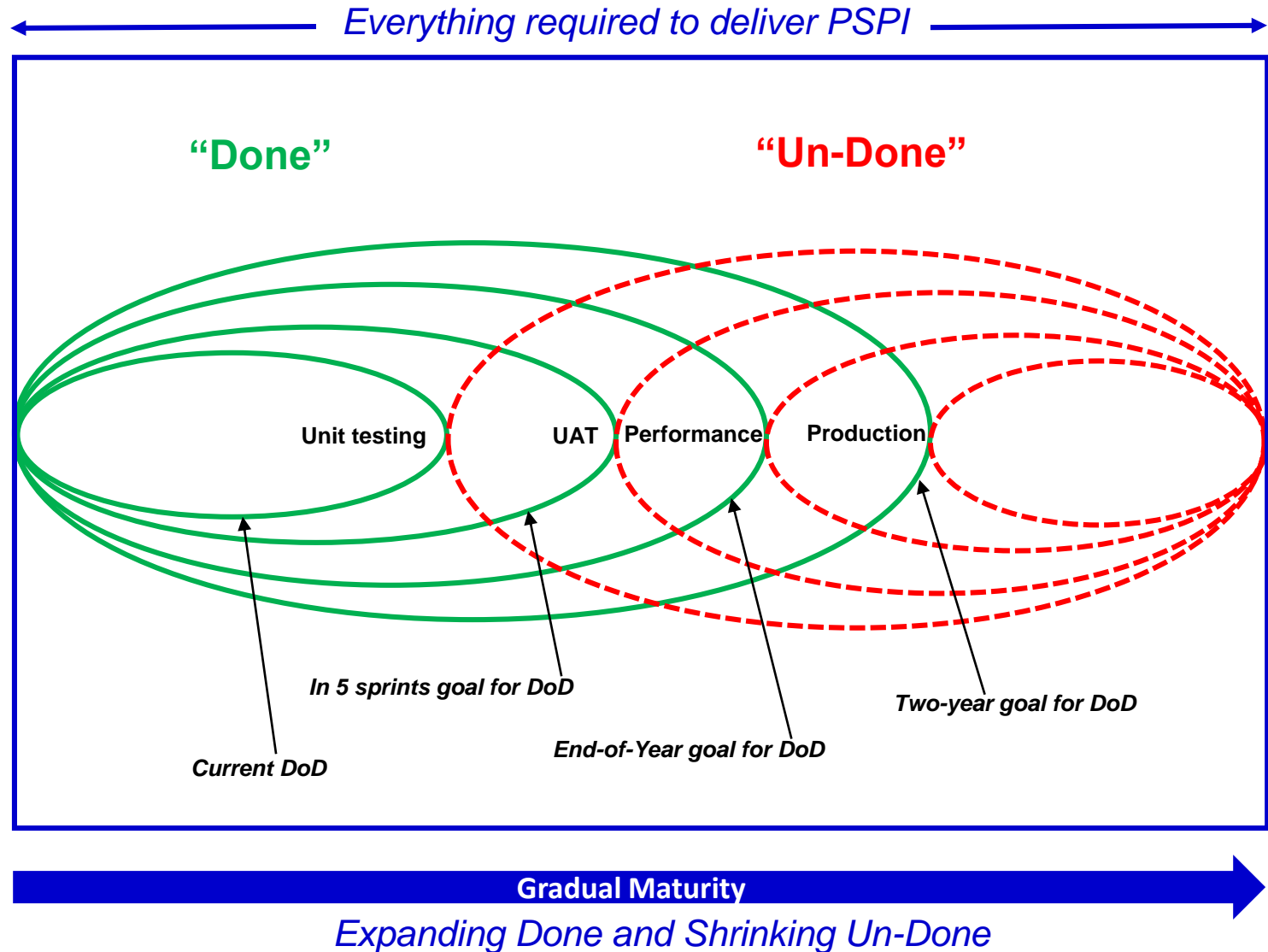
- + TEST
- + STATIC ANALYSIS
- + INTEGRATION
- + PACKAGING
- + STAGING
- + CUSTOMER DOCUMENTATION
- + RISK EVALUATION
- + CODE
- + APPROVAL
- + UPDATE MARKETING MATERIAL
- + REGULATION
- + PREPARE FOR CUSTOMER FEEDBACK

UNIT
INTEGRATION
USER ACCEPTANCE
ACCEPTANCE
SYSTEM
PERFORMANCE
STABILITY
USABILITY
STRESS
MONKEY
SMOKE

...This is not specific to
LeSS.....This is just good-
old simple Scrum...

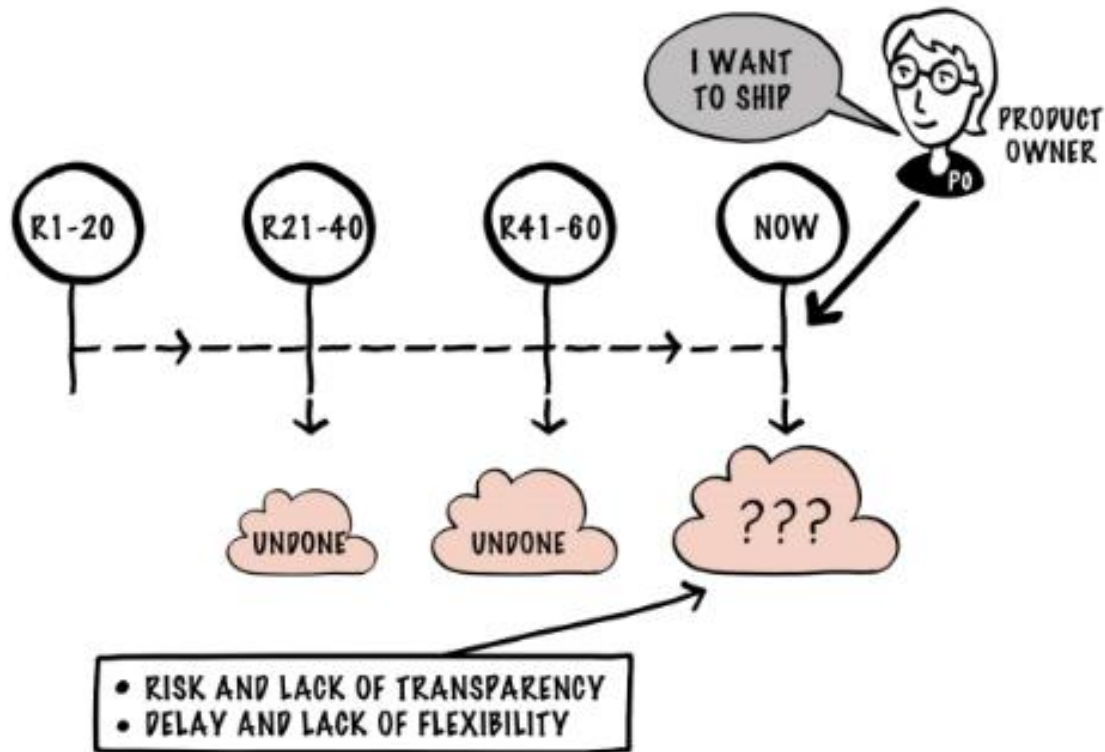
UNDERLINED ONES ARE THE DEFINITION OF DONE

Technical Excellence



Technical Excellence

Dealing with Un-Done Work

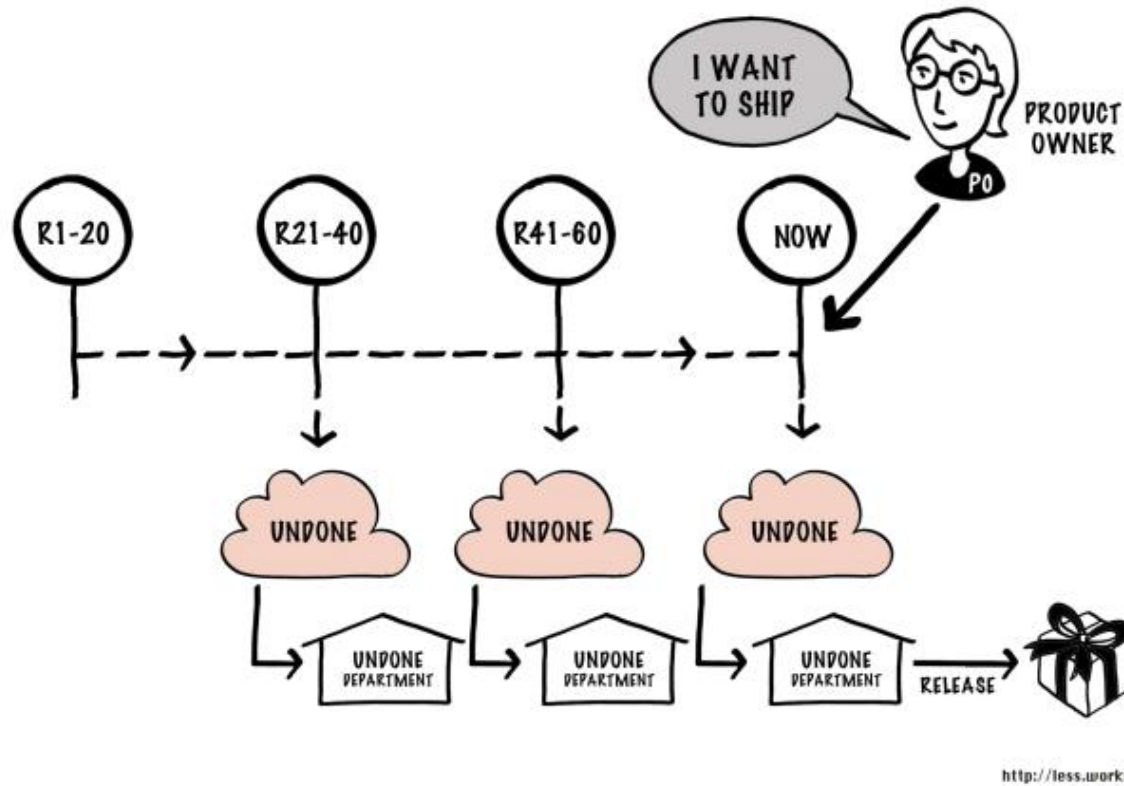


<https://less.works> 

Release Sprints

Technical Excellence

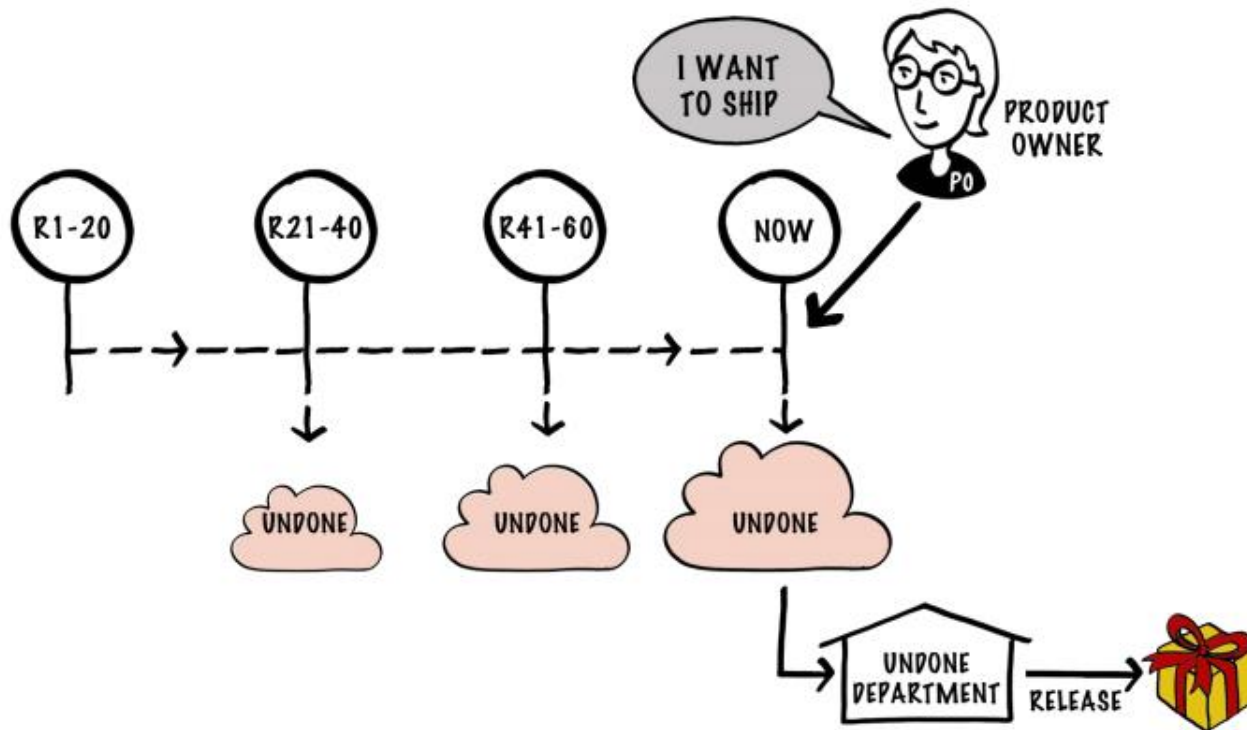
Dealing with Un-Done Work



Pipelining to Undone Department

Technical Excellence

Dealing with Un-Done Work



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

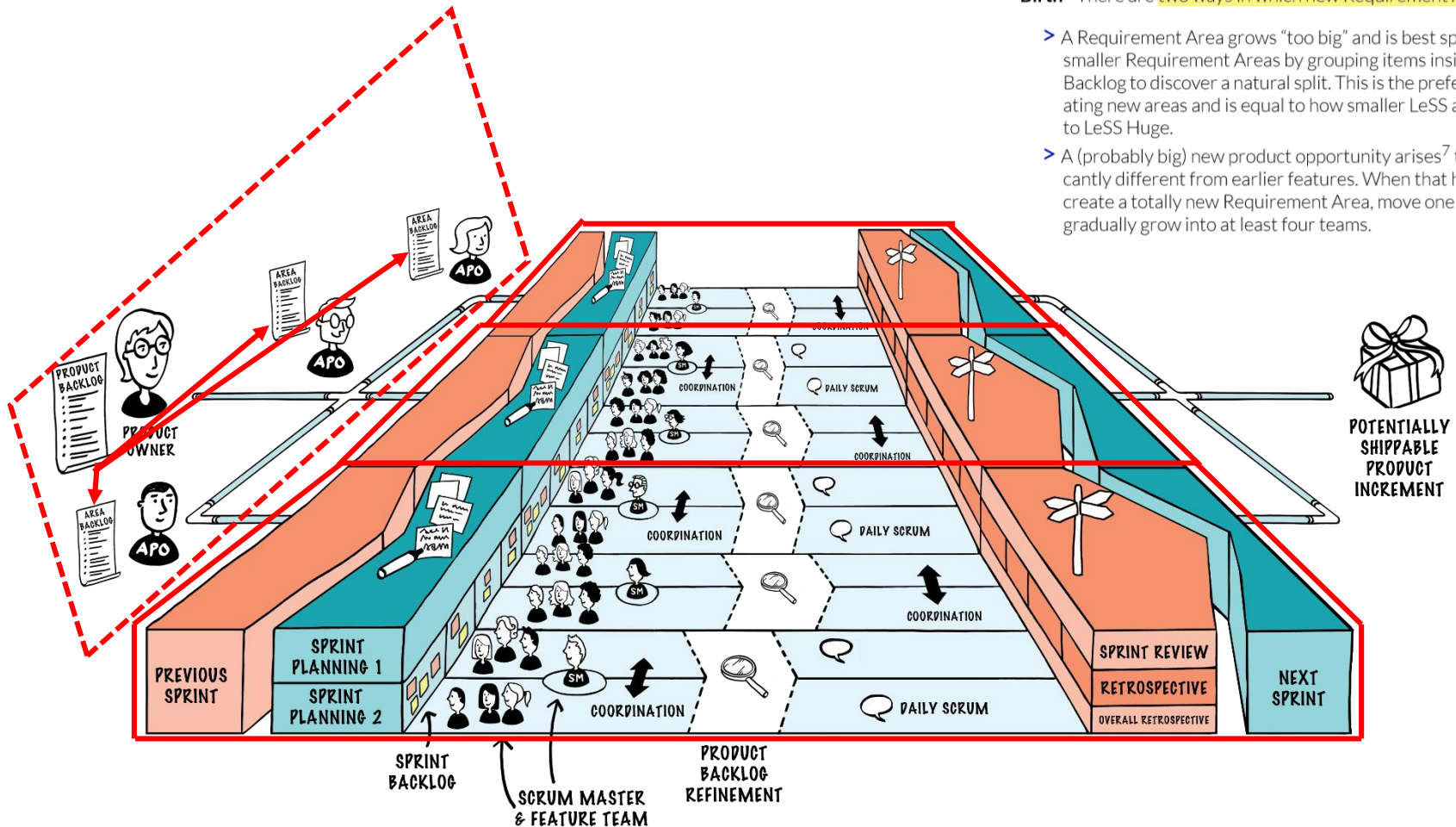
Undone Department Finalizes

LeSS Huge

LeSS Huge

Birth—There are two ways in which new Requirement Areas are born:

- A Requirement Area grows “too big” and is best split into two smaller Requirement Areas by grouping items inside the Area Backlog to discover a natural split. This is the preferred way of creating new areas and is equal to how smaller LeSS adoptions grow to LeSS Huge.
- A (probably big) new product opportunity arises⁷ that is significantly different from earlier features. When that happens, we can create a totally new Requirement Area, move one team into it and gradually grow into at least four teams.



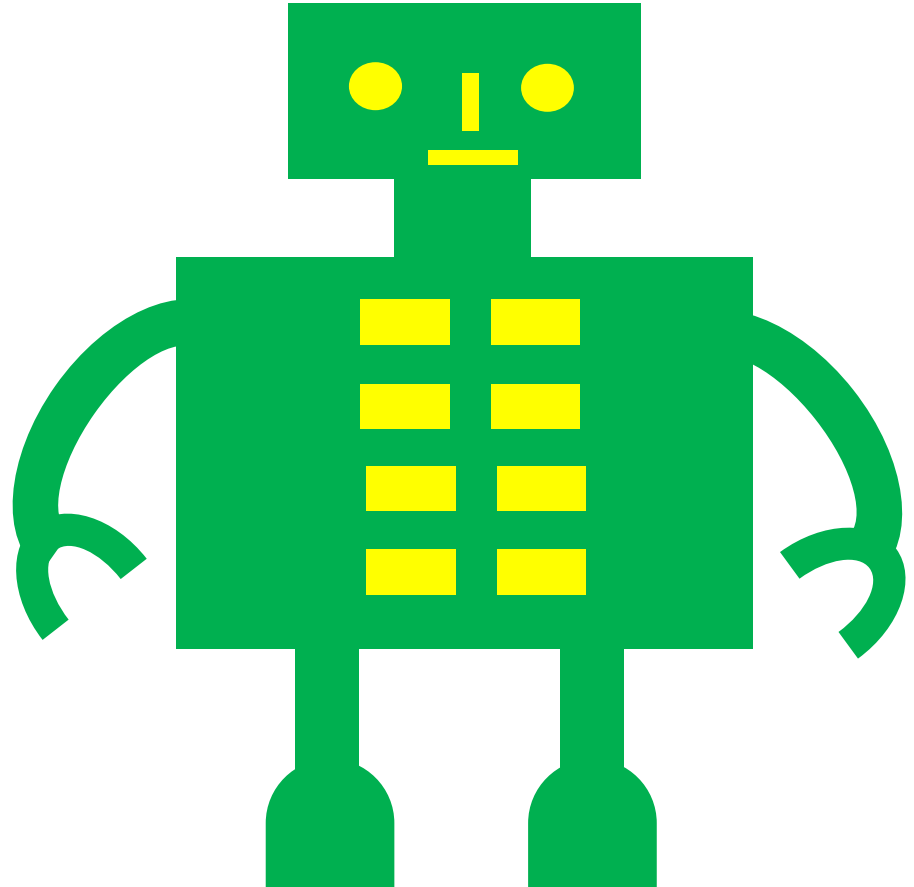
<http://less.works> 

- Avoid having multiple Requirement Areas for as long as possible
- Any product backlog item belongs to one Product Area only

Q & A

**What
Is
Next?**

What is Next?



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”.



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**.

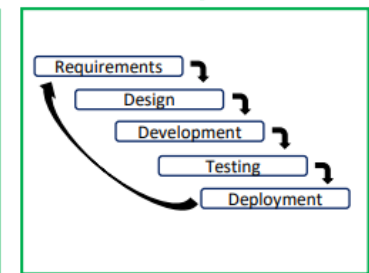
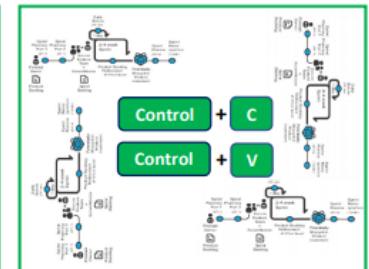
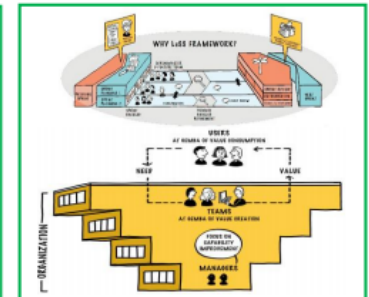
Transition from independent basic Scrum, performed by multiple teams to LeSS, is a desirable approach, when a product is widely defined and real Product Owner is identified

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>

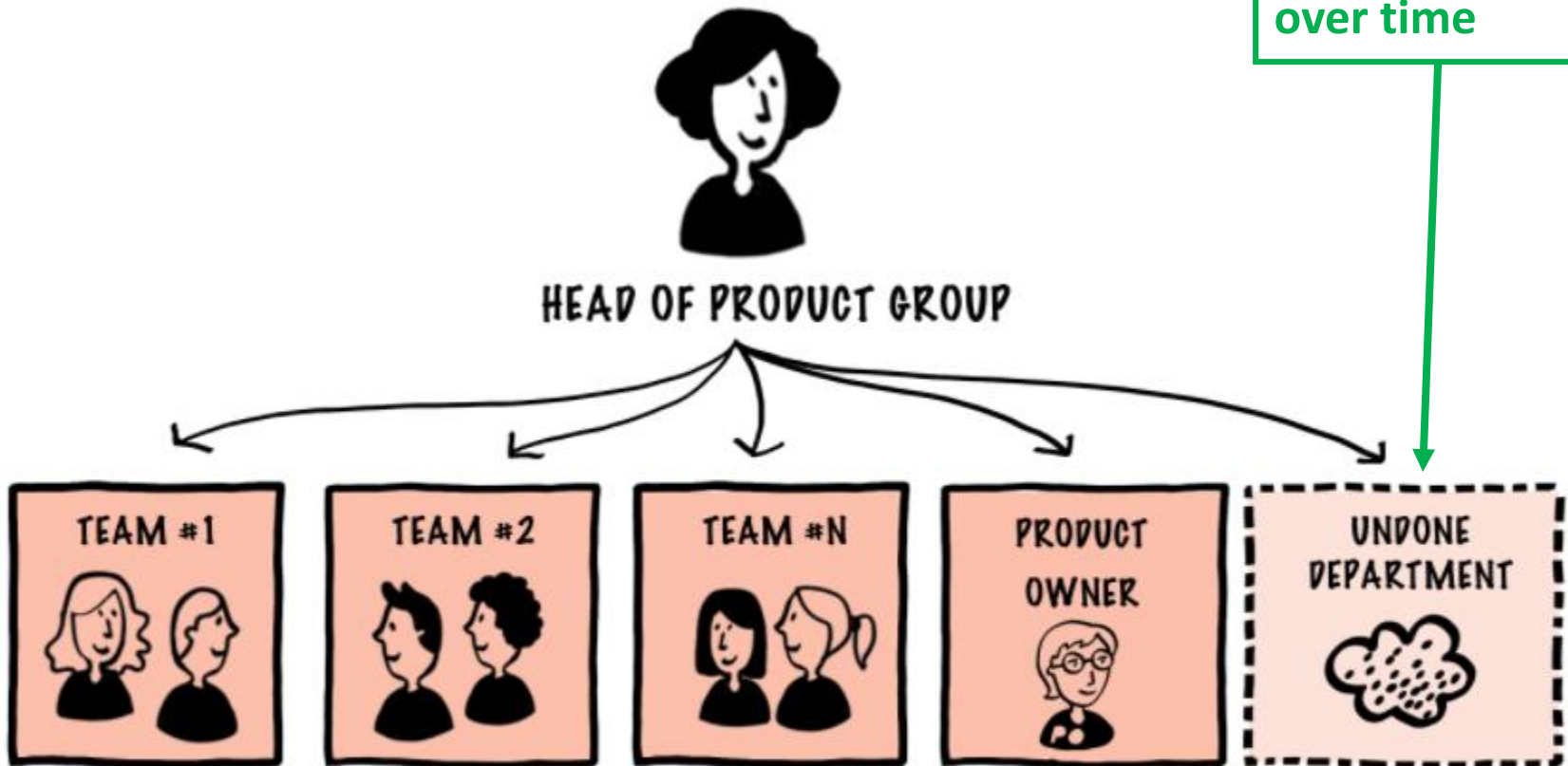
Scaling Scrum

Waterfall

Source: http://www.keystepstosuccess.com/wp-content/uploads/2018/02/scrum_scaling_org_descaling-1.pdf

Relationships in LeSS

To be dissolved
over time



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

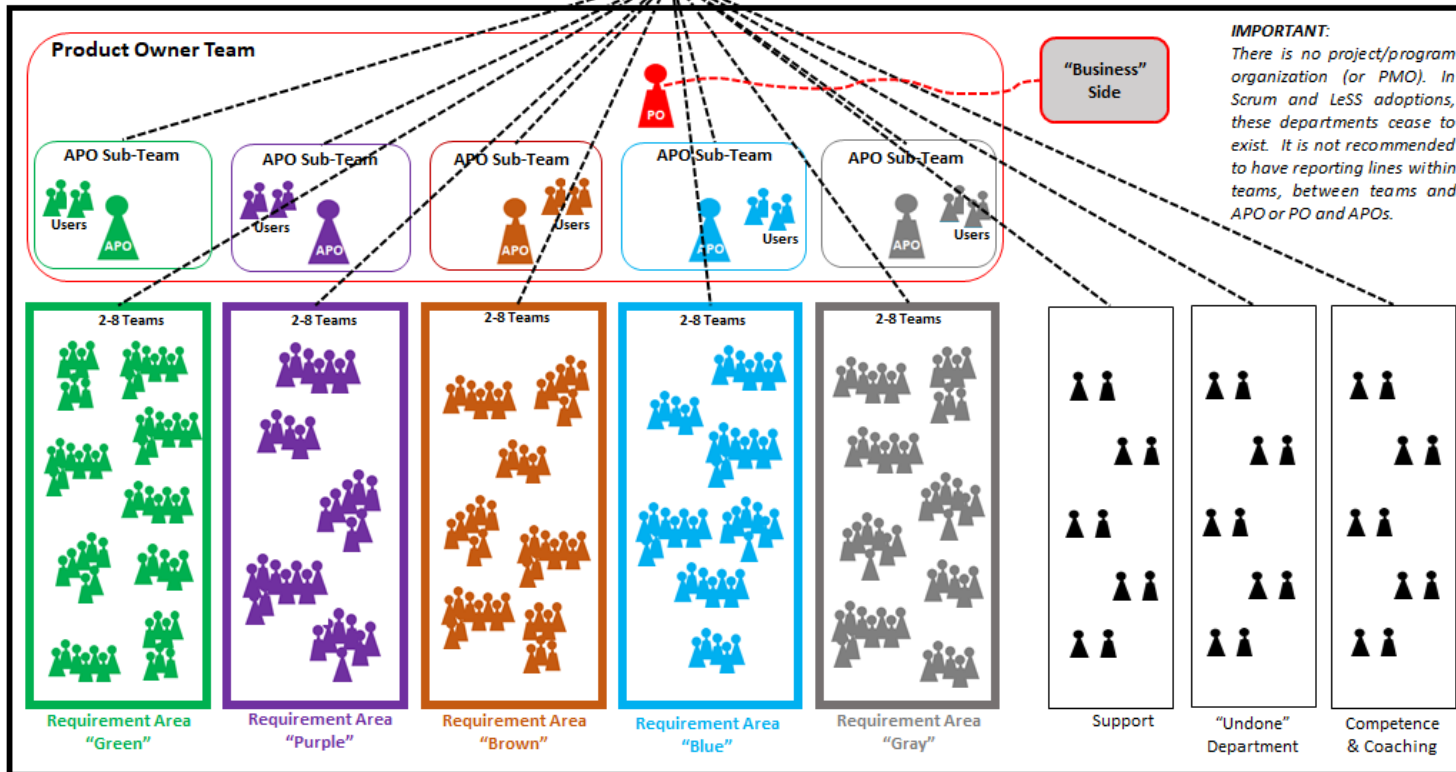
<http://less.works> CC BY-ND

Relationships in LeSS

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

Product Owner Team— Product Owner plus all Area Product Owners. Conceptually, the same as in a LeSS structure. In huge LeSS Huge, 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