





$$5 \times 5 - 1$$



$$8 \times 3 - 1$$



# Introduction



# About Training Content

- Created by: Gene Gendel
- Sourced from: <https://less.works/resources/graphics/index.html>
- Sourced from creative commons repository of Certified LeSS Trainers: <https://less.works/courses/become-less-trainer>
- Sourced from: <https://less.works/resources/learning-resources/books>
- Sourced from: <https://www.keystepstosuccess.com/agile-anti-patterns-with-irony-and-satire/>
- Sourced from: <https://www.keystepstosuccess.com/coach-tools/>



# Taking Care of Key Logistics

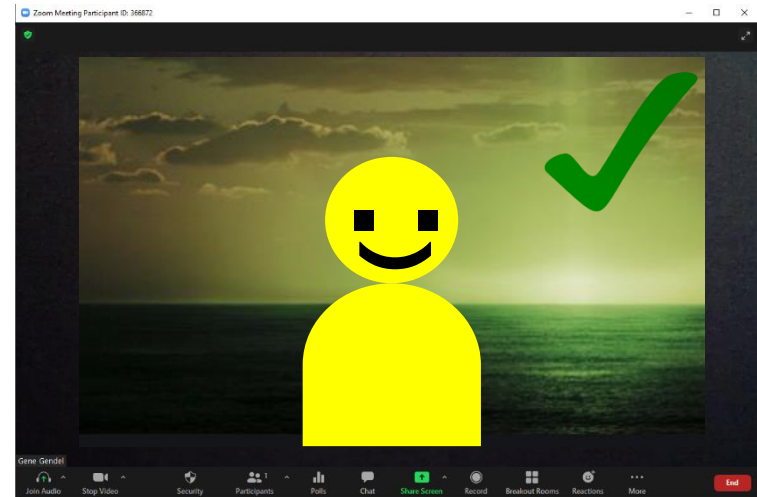
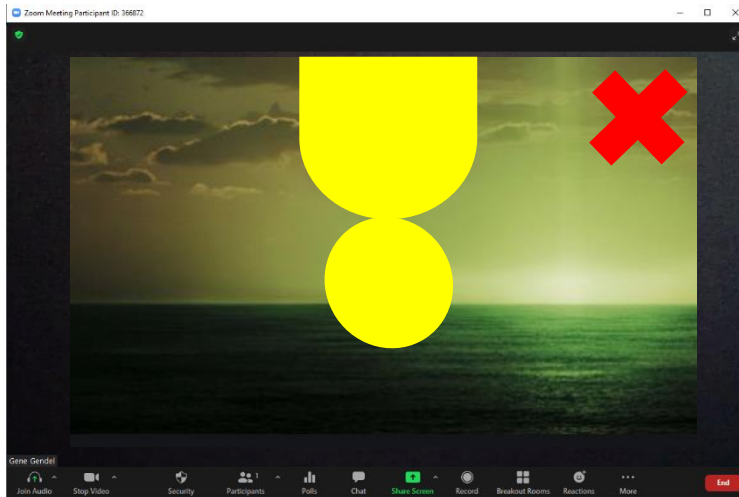
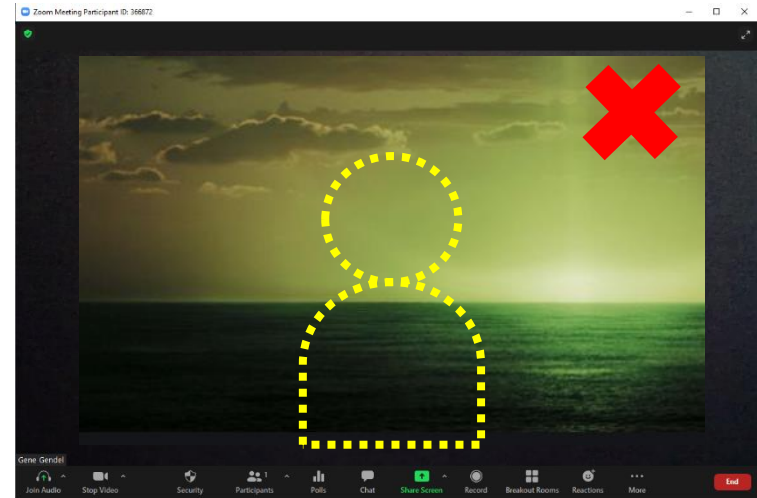


OR





# Taking Care of Key Logistics



Created by: Gene Gendel



# Taking Care of Key Logistics

This film has NOT been modified  
from its original (public) version.  
It has NOT been modified to fit  
this screen.



0:01 / 0:05



Created by: Gene Gendel



# Taking Care of Key Logistics

- *Structured Learning (Zoom or PPT)*
- *Class Activities (Exercises)*
- *Q&A*
- *Frequent Breaks and Lunch Time*



# Training Agenda

- **Introduction, Logistics, Stage Setting**
- **Commonly known omissions with scaling**
- **LeSS, as Organizational Design/De-Scaling Framework**
- **Informed Consent in LeSS**
- **What is System Thinking (& Modelling)?**
- **LeSS Rules**
  - **Structure**
  - **Product**
  - **Sprint**
- **LeSS Guides:**
  - **Three LeSS Adoption Principles**
  - **Organizational Design - 1st Order Factor**
  - **Local Optimization**
  - **Internal Contracts (& HR)**
  - **LeSS Guides: Organizing by Customer Value**
  - **Management**
  - **Scrum Master**
  - **Product**
  - **Product Owner**
  - **Product Backlog**
  - **Definition of Done**
  - **Product Backlog Refinement**
  - **Sprint Planning**
  - **Coordination & Integration**
  - **Review & Retrospective**
- **LeSS (Huge) Rules**
  - **Structure**
  - **Product**
  - **Sprint**
- **LeSS (Huge) Guides:**
  - **Organizing by Customer Value**
  - **Product Owner**
  - **Product Backlog**
- **Extras:**
  - **Product Definition & Exploration Activities in LeSS**
  - **LeSS Case Studies**



# About Instructor

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:

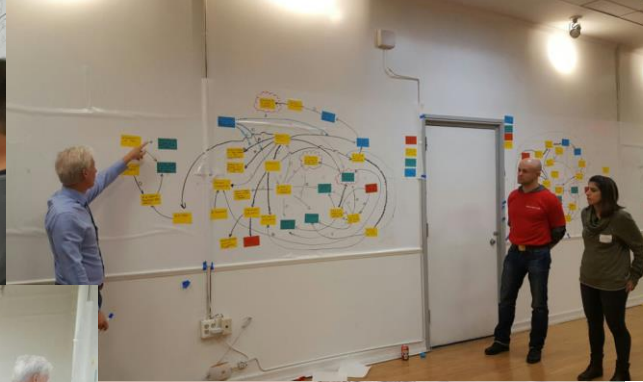
- [Agile Coaching: Wisdom from Practitioners](#) (free pdf).
- Best Agile Articles of [2017](#) and [2018](#)



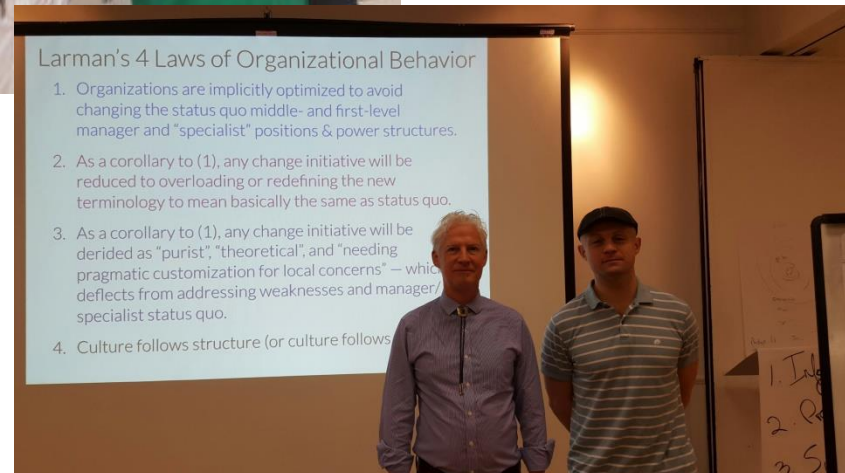
Gene significantly contributes to global and local agile communities, where he influences people by running workshops, webinars, seminars, panel discussions, lunch & learn, coaching retreats, professional gatherings and other events. While teaching and coaching Large Scale Scrum (LeSS) to organizations, 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.



# My Journey



**+ living and coaching  
it for a few years**



## Larman's 4 Laws of Organizational Behavior

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 overloading or redefining 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", and "needing pragmatic customization for local concerns" — which deflects from addressing weaknesses and manager/specialist status quo.
4. Culture follows structure (or culture follows



# Our Class



# Gaining Access & Collaborating



<https://miro.com/welcome/FwXv9enWQ083O8iCjH2g57AxYcw526U6haaxDsDtWkfapZxLUDyNJarnxYgJOcoS>



[https://drive.google.com/drive/folders/1uaXepWWVmIQLlwDHboL1Rwip\\_4rmGM\\_8?usp=sharing](https://drive.google.com/drive/folders/1uaXepWWVmIQLlwDHboL1Rwip_4rmGM_8?usp=sharing)



# Stage Setting

- In this course, we shall be mainly modelling the system and getting deep into the system, going after systemic root causes.
- Solutions and “pragmatic steps” - will come later 😊
- “How does it relate to what we do today?” – later
- The most important pre-requisites:
  - **Understanding SCRUM**
  - **Open (very) mind**

**Temporary PAUSE on:**

*PMI/PMO/PMP  
Six Sigma/SAFe/Spotify  
Internal frameworks, methodologies,  
“operating model”*

**type of experience**



# Stage Setting

*All Questions are Good*

When we say....

*“This is **B.S.** Question” ☺*

we mean

*“This is Basic Scrum (1-team) Question”*

and therefore,

It is addressed in the **SCRUM GUIDE**...



# Recognizing Most Common Problems

## Top-3 Abused Words

- 1) Agile
- 2) Scaling
- 3) Enterprise



When put together, in the same phrase, they become “**Enterprise Agile Scaling**” – what seems to be the most popular and frequently used phrase, nowadays.

Instead of using this phrase obsessively (titles, job descriptions, etc.), please consider the following:

- ❖ *If “Agile” != “Adaptive”, the word is probably misused*
- ❖ *In order to SCALE goodness, you may need to DE-SCALE (remove) badness. Therefore, by obsessively focusing on scaling, you might be increasing (scaling up) levels of badness in the system (organization) ☹*
- ❖ *“Enterprise” – does not always mean “huge”. A 50-person company (HR, Finance, Legal, Vendor Management, Business, Operations, IT) – is an enterprise. A 500-person IT Department alone – is NOT an enterprise.*



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



# Recognizing Most Common Problems

“We considered a bunch of names, and agreed eventually on ‘agile’ as we felt that captured the ***adaptiveness*** and ***response to change*** which we felt was so important to our approach.”

Martin Fowler



Sourced from creative commons repository of Certified LeSS Trainers: <https://less.works/courses/become-less-trainer>

Gene Gendel, Certified Enterprise & Team Coach (CEC-CTC), Certified LeSS Trainer (CLT)



# Recognizing Most Common Problems

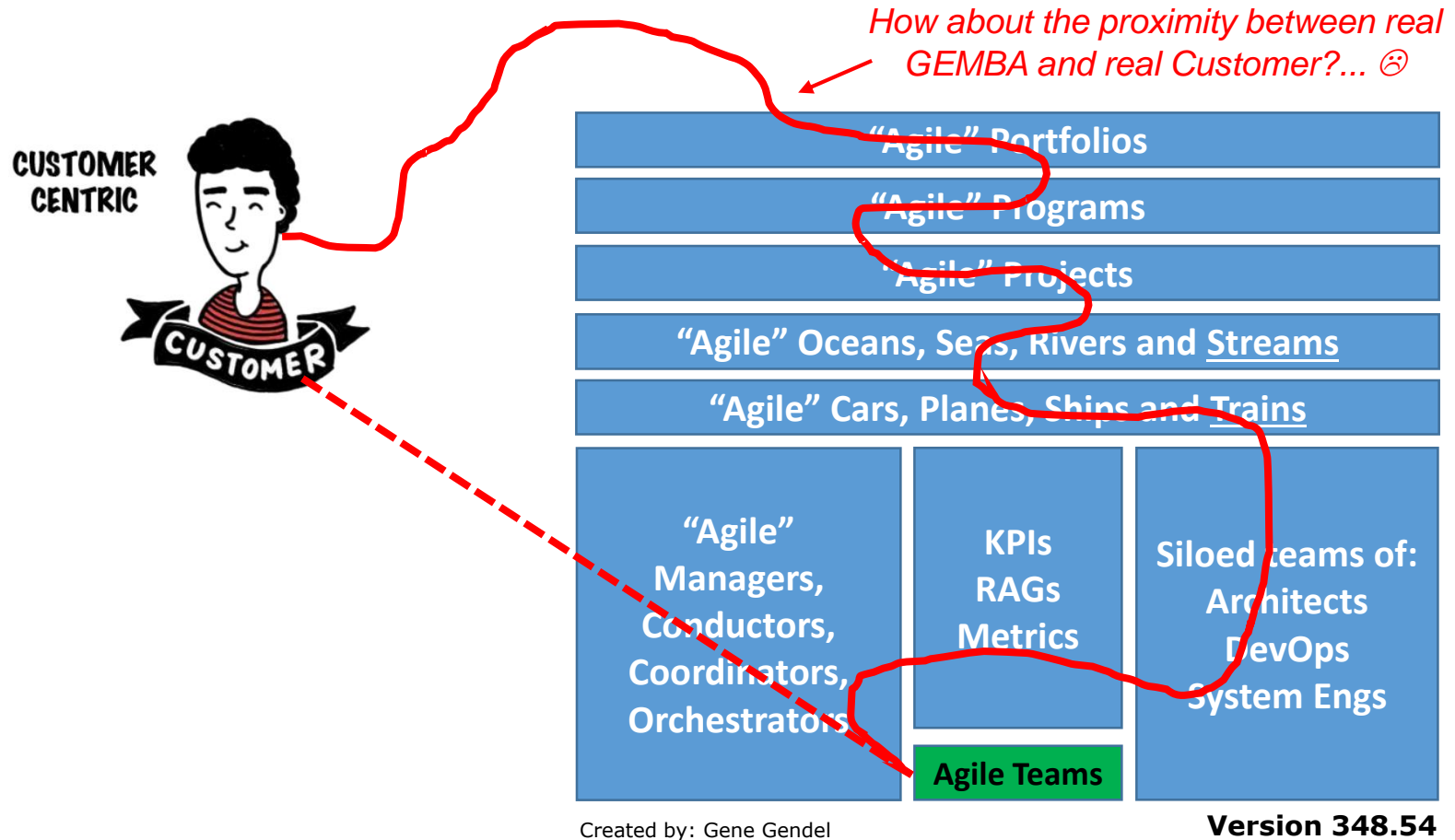
"The ability to move with quick easy grace, to be nimble and adaptable. To embrace change and become masters of change - to compete through adaptability by being able to change faster and cheaper than your competition can. "

Craig Larman





# Recognizing Most Common Problems



- Where is customer – centricity?
- How **loooooong** is a feedback loop between customers and developers?
- How adaptive (agile, fast-reacting) can teams really be, to Customer's requests?
- What has changed from what you saw prior to "installation"?

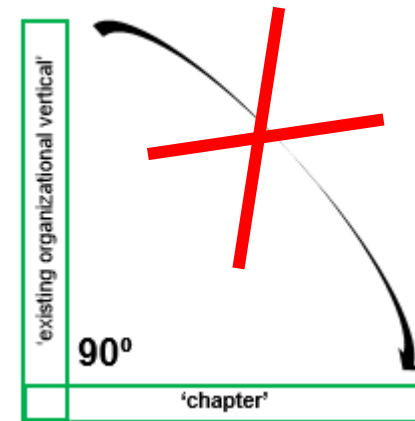
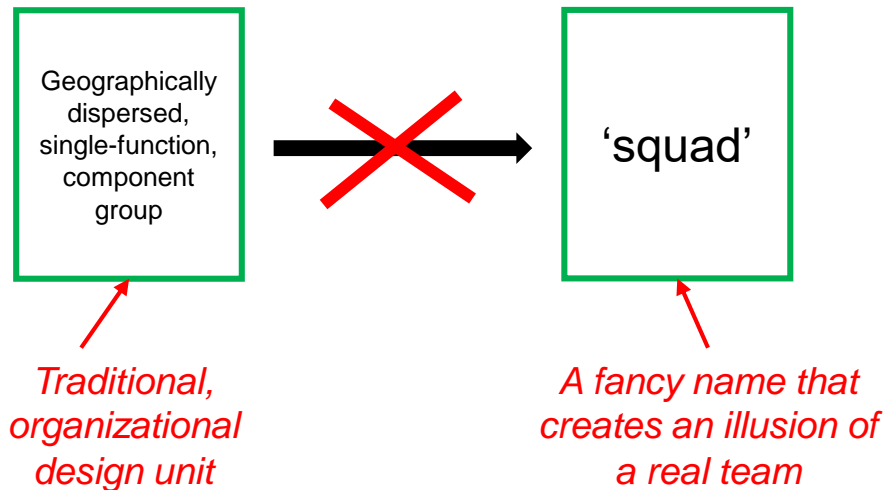
<https://www.keystepstosuccess.com/2020/05/05-05-less-talks-dave-snowden-answering-tough-questions-qa/>



# Recognizing Most Common Problems

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 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**



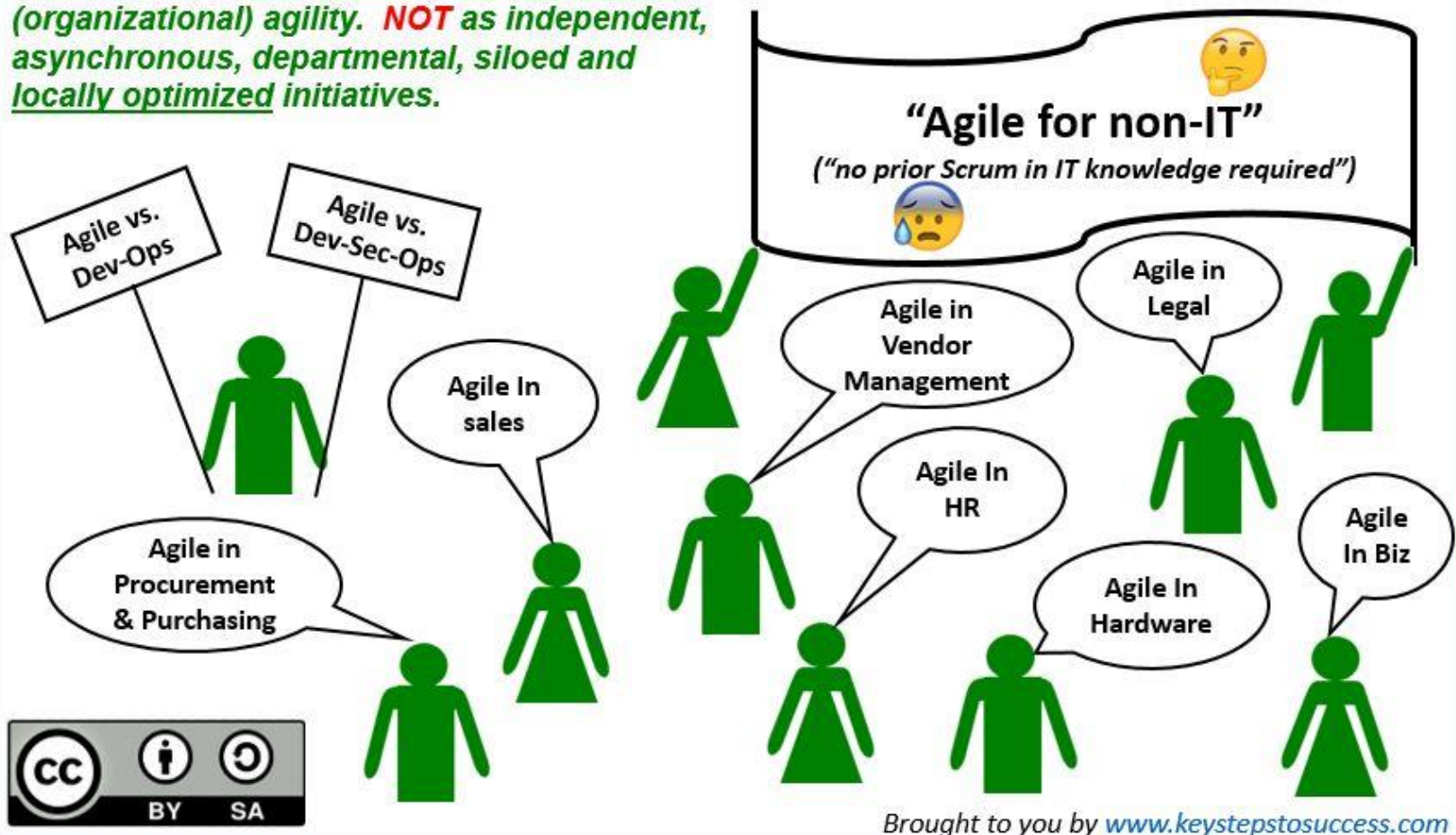
Created by: Gene Gendel



# Recognizing Most Common Problems

## Beware of Fragmented Agility

*The below should be viewed as complimentary dimensions of one, inclusive, eco-systemic (organizational) agility. **NOT** as independent, asynchronous, departmental, siloed and locally optimized initiatives.*




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



# Recognizing Most Common Problems

*Avoid these common anti-patterns...*


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

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

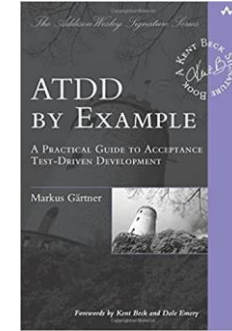
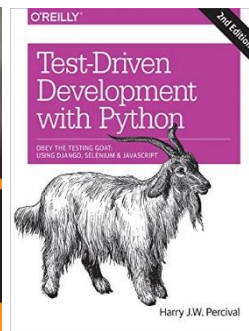
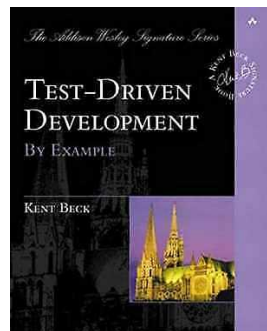


Avoid  
(Tool – Driven Transformations)  
**TDT**

*Success of agile transformations is inversely proportional to reliance on "agile" tooling solutions (in partnership with heavy "scaling" frameworks) and directly proportional willingness to de-scale and simplify;)*

Sourced from: <https://www.keystepstosuccess.com/agile-anti-patterns-with-irony-and-satire/>

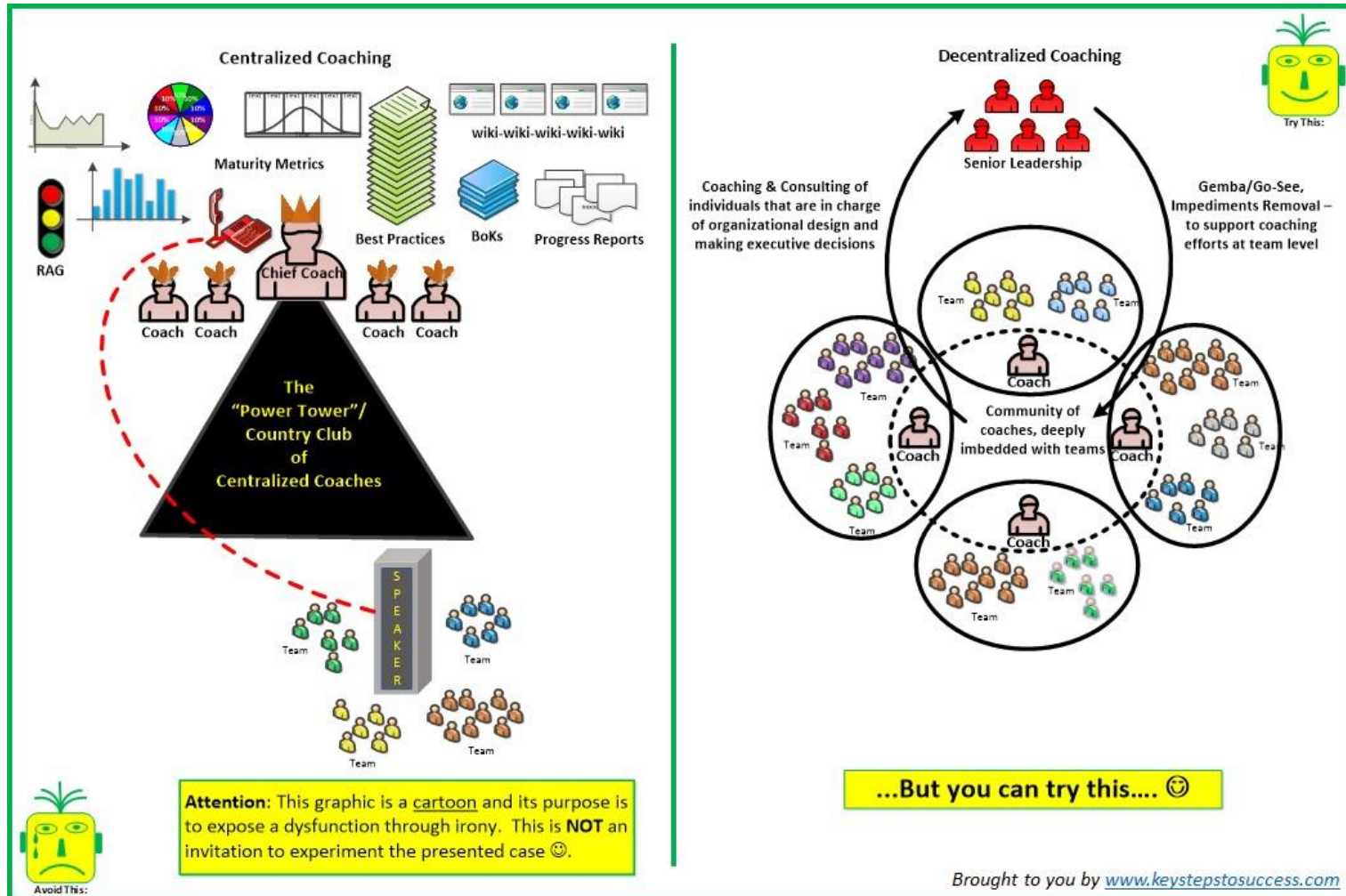
*...that are not to be confused with...*





# Recognizing Most Common Problems

*Avoid these common anti-patterns...*

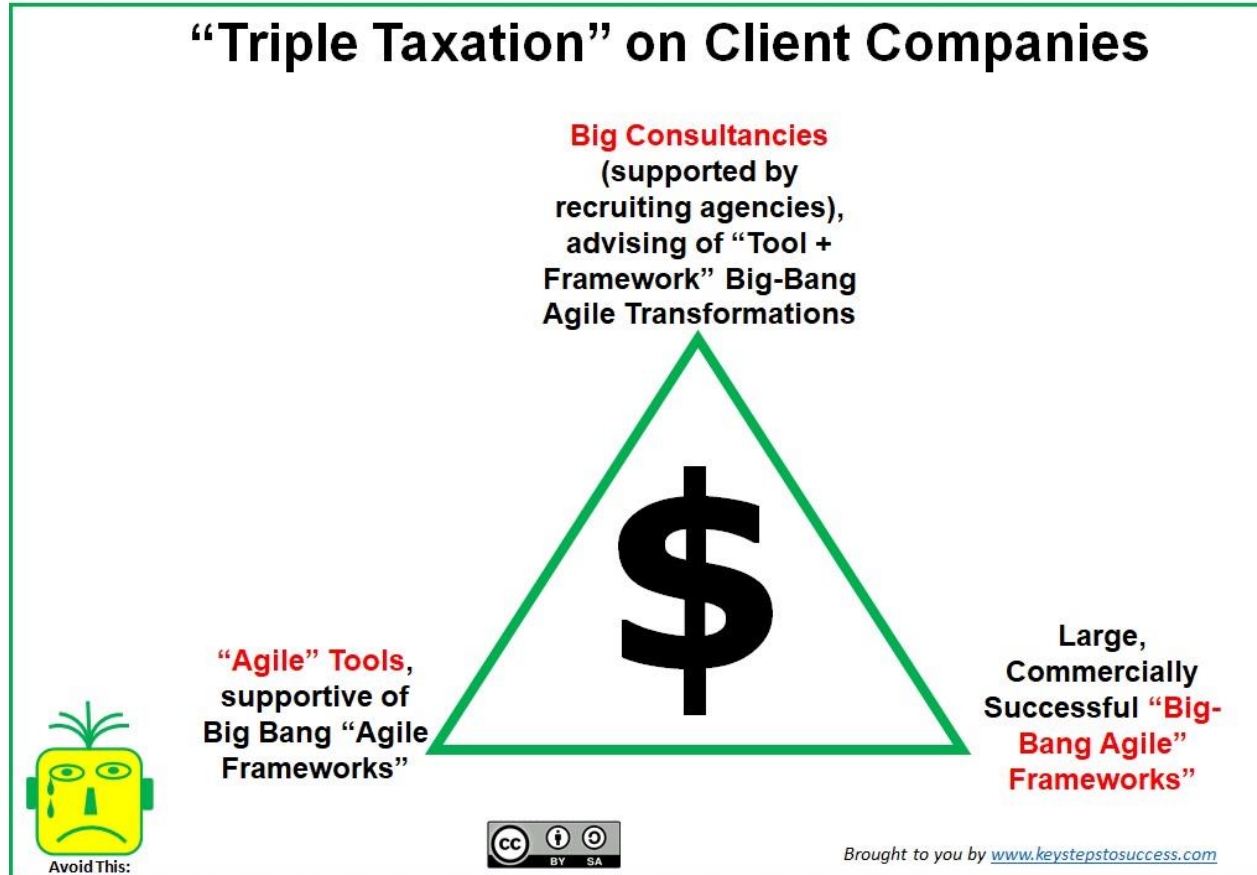


Sourced from: <https://www.keystepstosuccess.com/agile-anti-patterns-with-irony-and-satire/>



# Recognizing Most Common Problems

*Avoid these common anti-patterns...*



Sourced from: <https://www.keystepstosuccess.com/agile-anti-patterns-with-irony-and-satire/>



# Recognizing Most Common Problems

- [05/05 – LESS TALKS: Dave Snowden: Answering Tough Questions](#)
- [Ken Schwaber: unSAFe at any speed](#)
- [Mike Cohn: L.A.F.A.B.L.E \(Large Agile Framework Appropriate for Big, Lumbering Enterprises\), by Mike Cohn](#)
- [S. Fe is not Agile. S. Fe is not even Scrum, by Mike Beedle](#)
- [Issues with SAFe, by Ron Jeffries](#)
- ["SAFe = shitty Agile for Enterprises", by Martin Fowler](#)
- [SAFe: Market Share Increase. Rapid Growth. What Is The Recipe?, by Gene Gendel](#)
- [Dan Snowden: SAFe: the infantilism of management](#)
- [Does SAFe agree with the Agile Manifesto?, by Peter Merel](#)
- [SAFE ≠ AGILE, by Tom Mellor](#)
- [Dependencies, Scrum of Scrums, and SAFe, by Ron Jeffries](#)
- [https://ronjeffries.com/categories/safe/, by Ron Jeffries](https://ronjeffries.com/categories/safe/)
- [GOTO 2015: Agile is Dead, by Dave Thomas](#)
- [Why do so many companies seem to jump straight to SAFe when starting Agile?, by Mark Levison](#)
- [U.S. Air Force Questions about Agile /SAFe Memo? -highly discouraging from using rigid, prescriptive frameworks such as SAFe | \(Slide 12 screenshot\)](#)
- [CSO Memo on Agile – and SAFe, by Nicolas M. Chaillan \(US Air Force Chief Software Officer\)](#)
- ["It's just a toolbox" – essentials and accidents in scaling agile, by Dr. Agilefant](#)
- [Scaling Agility or Bureaucracy, by Ari Tikka and Ran Nyman](#)
- [LeSS SAFe comparison, by Ari Tikka and Ran Nyman](#)
- [The Horror Of The Scaled Agile Framework, by Neil Killick](#)
- [You Don't Need a Complicated Story Hierarchy, by Mike Cohn](#)
- [Let's Acknowledge SAFe for What It Is....And Move On, by Mike Cottmeyer](#)
- [Revenge of the PMO, by Marty Cagan](#)
- [Why SAFe Is Not The Scaled Agile Approach You Need, by Renee Thoughton](#)
- [Remove References To Scrum From SAFe!, by Den Sunny](#)
- [Beware SAFe \(the Scaled Agile Framework for Enterprise\), an Unholy Incarnation of Darkness, by Sean Dexter](#)
- [10 Common Mistakes when Implementing SAFe, by Michael Küsters](#)
- [Steal "Agile": Despicable Mission, Jacques Morali, Victor Willis, Henri Belolo](#)
- [How is SAFe different from Scrum/Agile project management?, by Peter Stevens](#)
- [Failed #SquadGoals, by Jeremiah Lee](#)
- [Case Study : When emulating Scaling Agile at Spotify went awry at Refinery29, by Andy Park](#)
- [The Spotify "Model": Don't Simply Copy-Paste, by Evan Campbell](#)
- [Spotify didn't implement the Spotify model by copying Spotify...by Kent Beck](#)
- ["Spotify Model" – 10 lessons in transplantology, by Kate Hobler \(Terlecka\)](#)
- [Scaling Agile @ Spotify, with Tribes, Squads, Chapters & Guilds, by Henrik Kniberg & Anders Ivarsson](#)
- [Don't Copy the Spotify Model, by Ben Linders](#)
- [Over-engineered "Agile Operating Model" by Gene Gendel](#)
- [Accelerated career path by "!@#%^ Coach", by Gene Gendel](#)

<https://www.keystepstosuccess.com/2016/05/safe-market-share-increase-rapid-growth-what-is-the-recipe/>



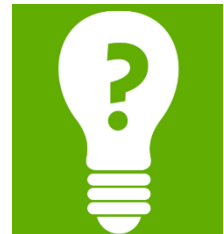
# Class Activity

## Class:

- Whole Class: Split up into teams. Discuss most commonly seen scaling *pitfalls* and *omissions*.
- For top-3 (vote) and suggest a few root causes per each.
- Capture in Class Log.

**Instructor:** Catalyze. Reflect.

**Duration:** 10 min





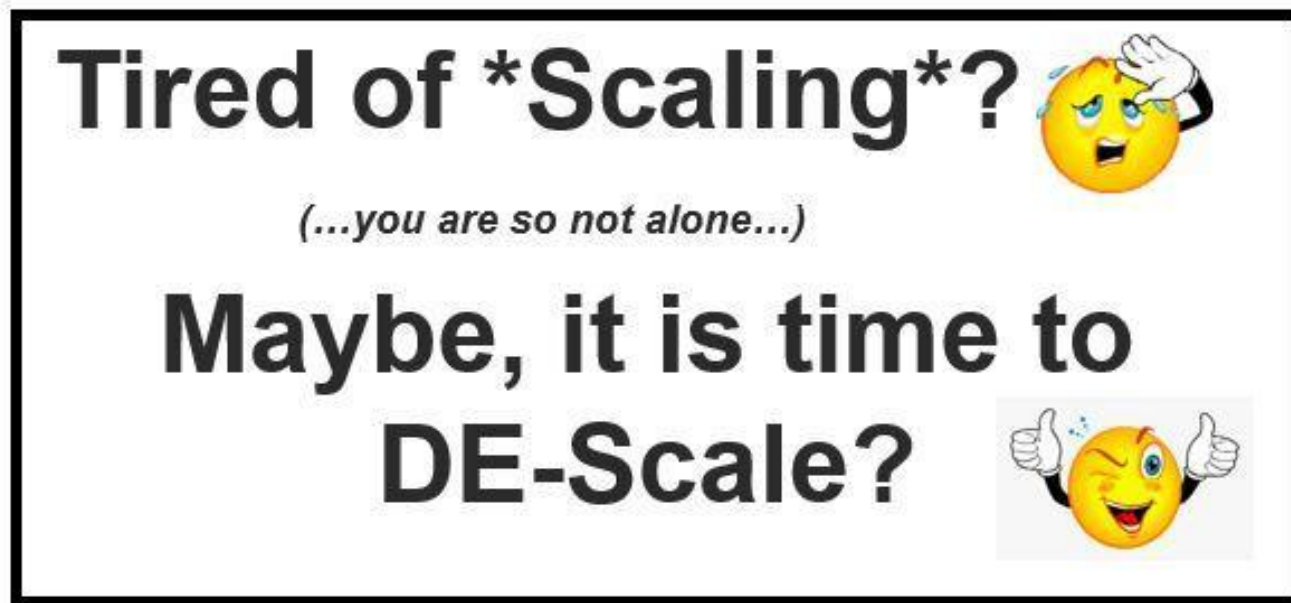




# Introduction

## Less is More: *De-Scaling with LeSS*

LeSS is was actually “*mis-labeled*” 😊.... And this requires some explanation...LeSS is about Scaling Scrum ..... by means of **Organizational De-scaling**...

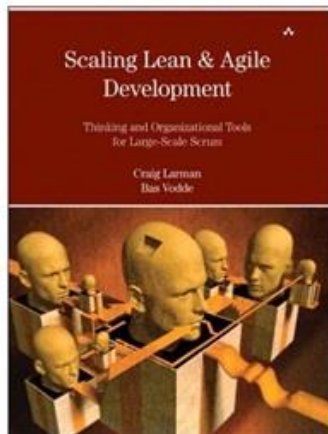


Created by: Gene Gendel

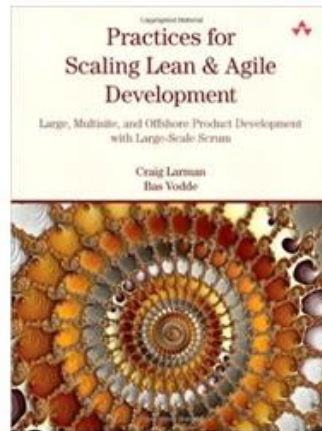


# Introduction

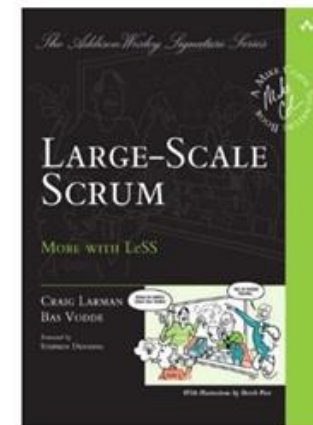
Large Scale Scrum (LeSS) - is Scrum. It is a DE-SCALING, organizational design framework for large-scale, multi-site agile product development. LeSS is based on close to 20 years of extensive research and deep learning. In LeSS, a team is an organizational building block. Understanding LeSS requires learning the whole system, not just some of its parts. LeSS adoptions are **DEEP & NARROW**.



2008



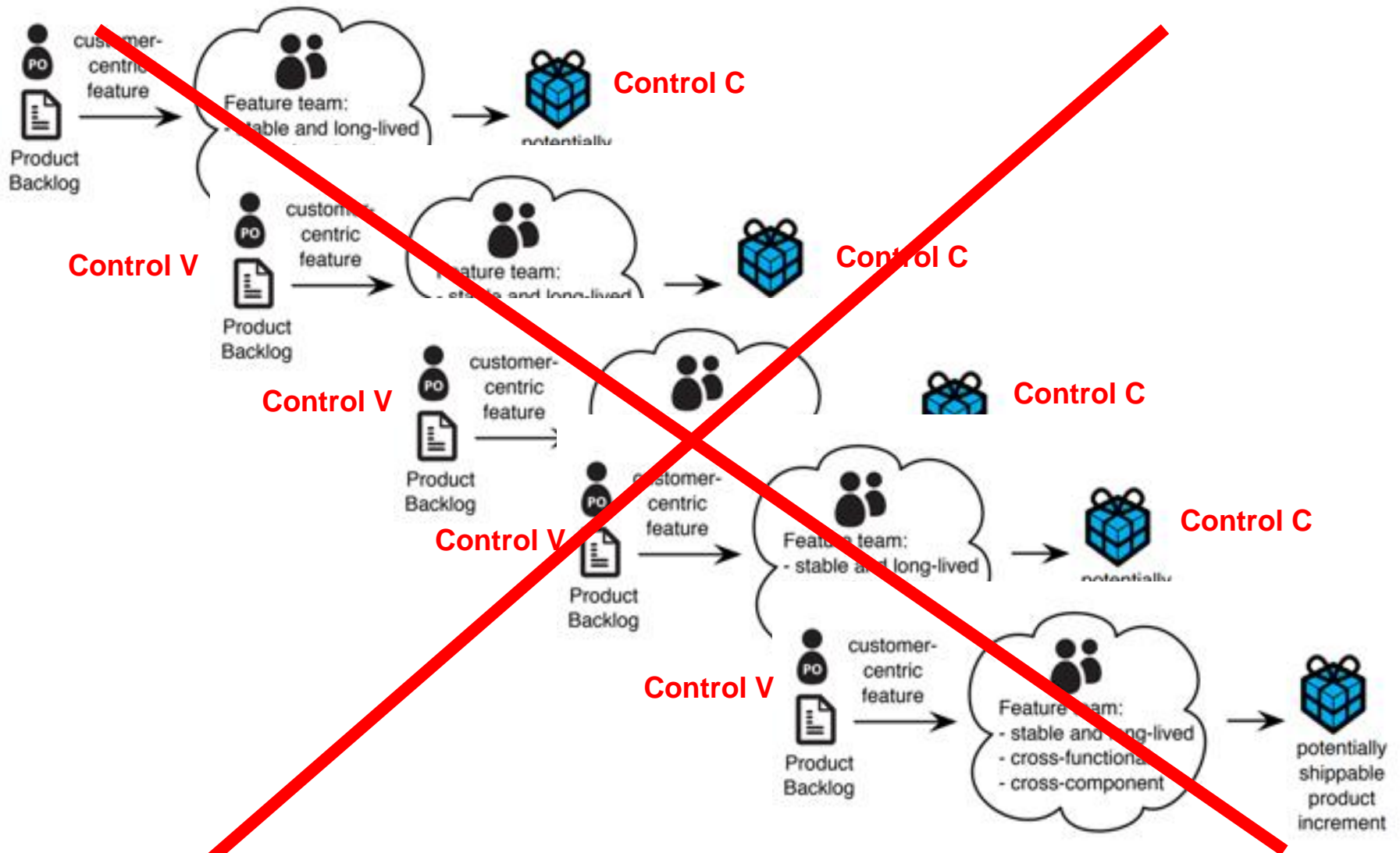
2010



2016



# Introduction



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

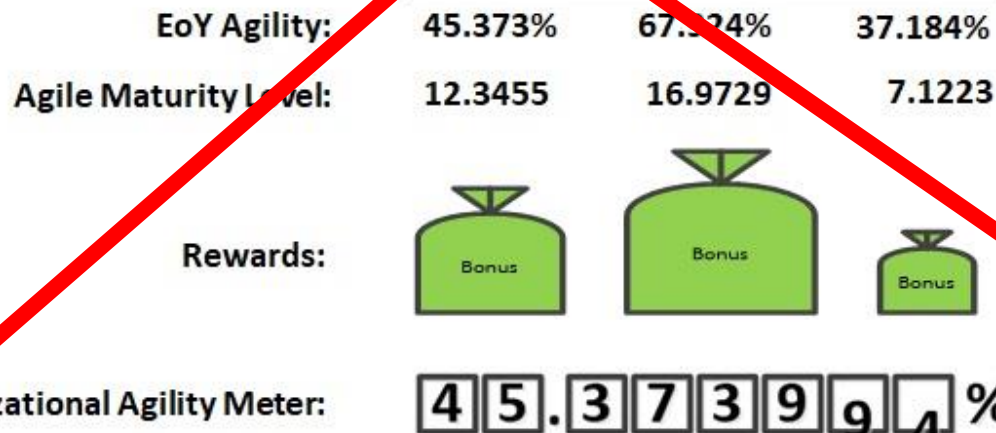


# Introduction

**Attention:** This graphic is a cartoon and its purpose is to expose a dysfunction through irony. This is **NOT** an invitation to experiment with this approach 😊.



## Agile “Semi-Finals” Race



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



# Introduction

## LeSS is not:

- Many teams doing their own so-called “Scrum”, and
- Working on different so-called “products”
- Delivering components that need further integration
- Working for many so-called “Product Owners” (“TOO”)
- Maintaining private, silo-ed and dependent backlogs

*Many people with different strategy, vision, mission. Competing with each other.*



**NOT LeSS**

## LeSS is:

- 2-8 teams, working together in real Scrum, and:
- Working on the same, widely defined product
- Delivering fully integrated, end-to-end features
- Working for only one REAL Product Owner
- Maintaining only one, shared backlog

*Only one person with shared strategy, vision, mission. No internal competition.*



**LeSS**

Sourced from creative commons repository of Certified LeSS Trainers: <https://less.works/courses/become-less-trainer>

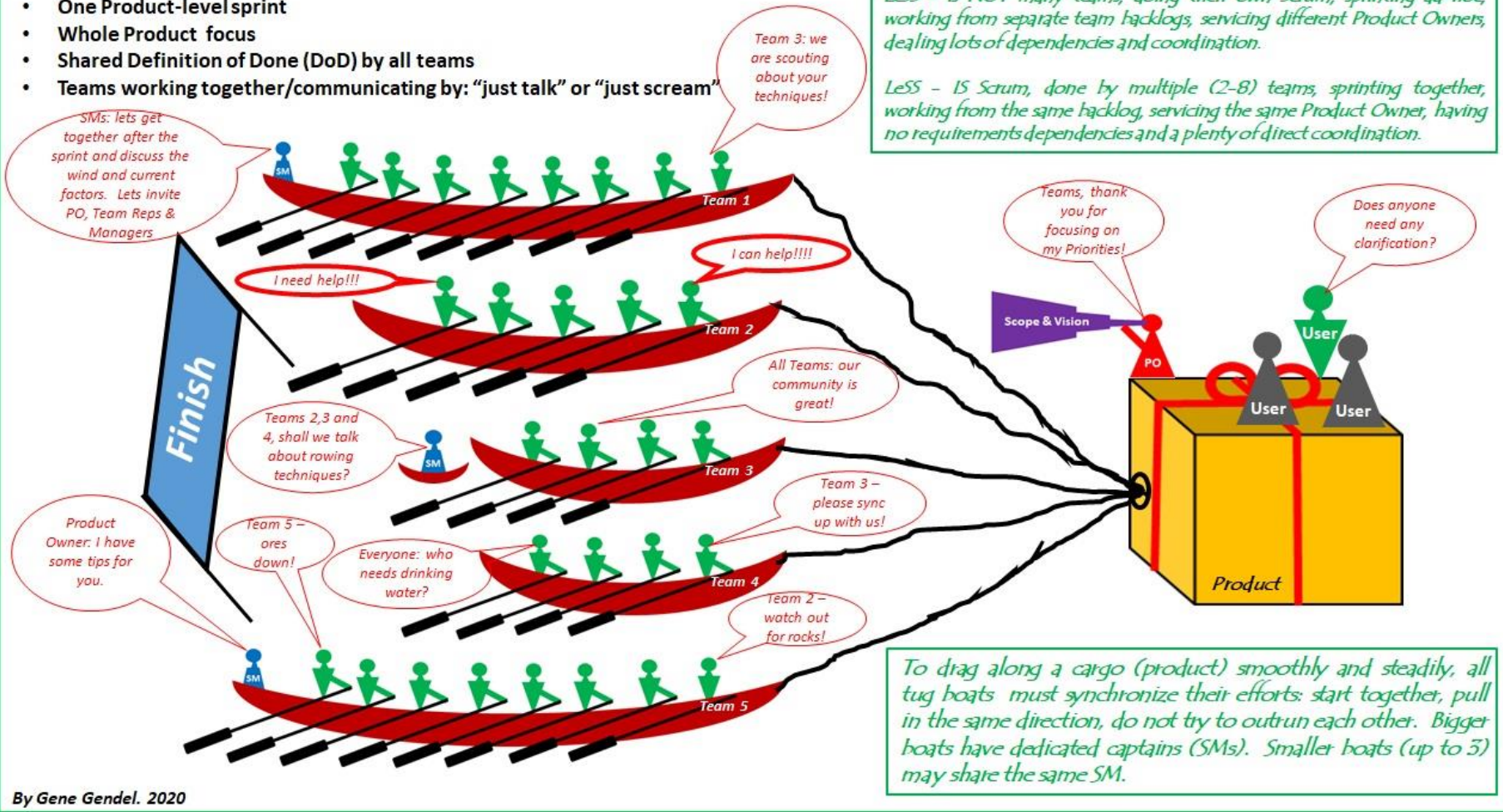


# Introduction

- 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.*

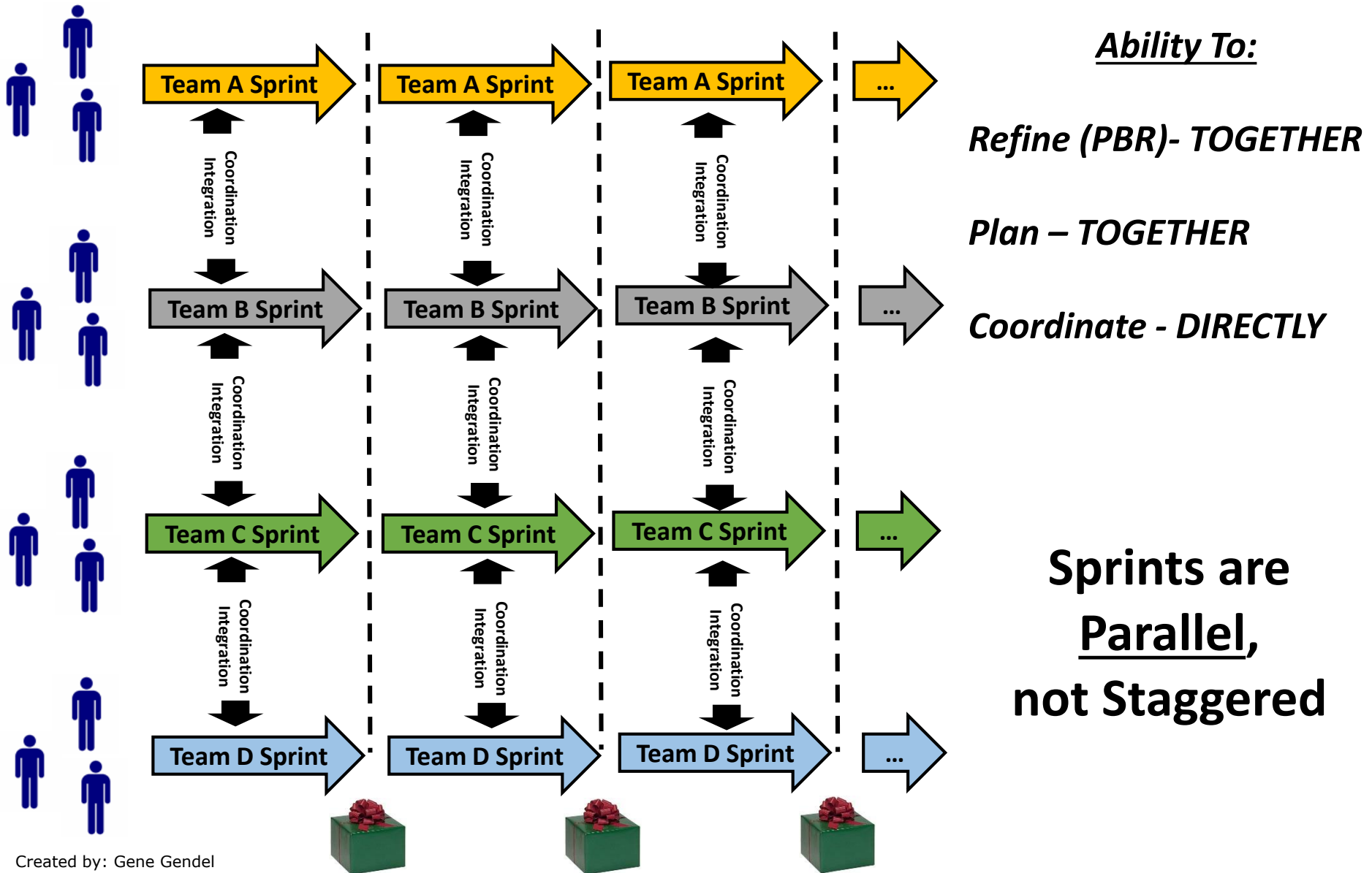
*LeSS - IS Scrum, done by multiple (2-8) teams, sprinting together, working from the same backlog, servicing the same Product Owner, having no requirements dependencies and a plenty of direct coordination.*



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



# Introduction



Created by: Gene Gendel



# Introduction

## 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/>



# LeSS Management



WIKIPEDIA  
The Free Encyclopedia

[Main page](#)  
[Contents](#)  
[Current events](#)  
[Random article](#)  
[About Wikipedia](#)  
[Contact us](#)  
[Donate](#)

[Contribute](#)  
[Help](#)  
[Learn to edit](#)  
[Community portal](#)  
[Recent changes](#)  
[Upload file](#)

Not logged in

Article

Talk

Read

Edit

View history

## Informed consent

From Wikipedia, the free encyclopedia

*This article is about consent to medical or research procedures. For consent in other contexts, see [Consent](#). For the [House of Commons](#), see [House of Commons](#).*

**Informed consent** is a process for getting permission before conducting a healthcare intervention on a person, for conducting some form of research on a person, or for disclosing a person's information. A [health care provider](#) may ask a patient to [consent](#) to receive [therapy](#) before providing it, a [clinical researcher](#) may ask a [research participant](#) before enrolling that person into a [clinical trial](#), and a researcher may ask a research participant before starting some form of [controlled experiment](#). Informed consent is collected according to guidelines from the fields of [medical ethics](#) and [research ethics](#).

An informed consent can be said to have been given based upon a clear appreciation and understanding of the facts, implications, and consequences of an action. Adequate informed consent is rooted in respecting a person's dignity.<sup>[1]</sup> To give informed consent, the individual concerned must have adequate reasoning faculties and be in possession of all relevant facts. Impairments to reasoning and judgment that may prevent informed consent include basic intellectual or emotional immaturity, high levels of stress such as [posttraumatic stress disorder](#) (PTSD) or a severe [intellectual disability](#), severe [mental disorder](#), [intoxication](#), severe [sleep deprivation](#), [Alzheimer's disease](#), or being in a [coma](#).

Sourced from: [https://en.wikipedia.org/wiki/Informed\\_consent](https://en.wikipedia.org/wiki/Informed_consent)

Degree of sr.  
management  
support (real  
gembu, not "in  
spirit only")

5



# Class Activity

## Class:

- Split up into teams. Each team – discuss the biggest ‘aha’ moments about LeSS and the most prevalent *‘this is gonna be tough for us to do’* reasons.
- *Teams converge and share findings.*
- Capture in Class Log

**Instructor:** Catalyze. Reflect.

**Duration:** 10 min









## ***Causation*** **VS** ***Correlation***



# System Modelling & Thinking

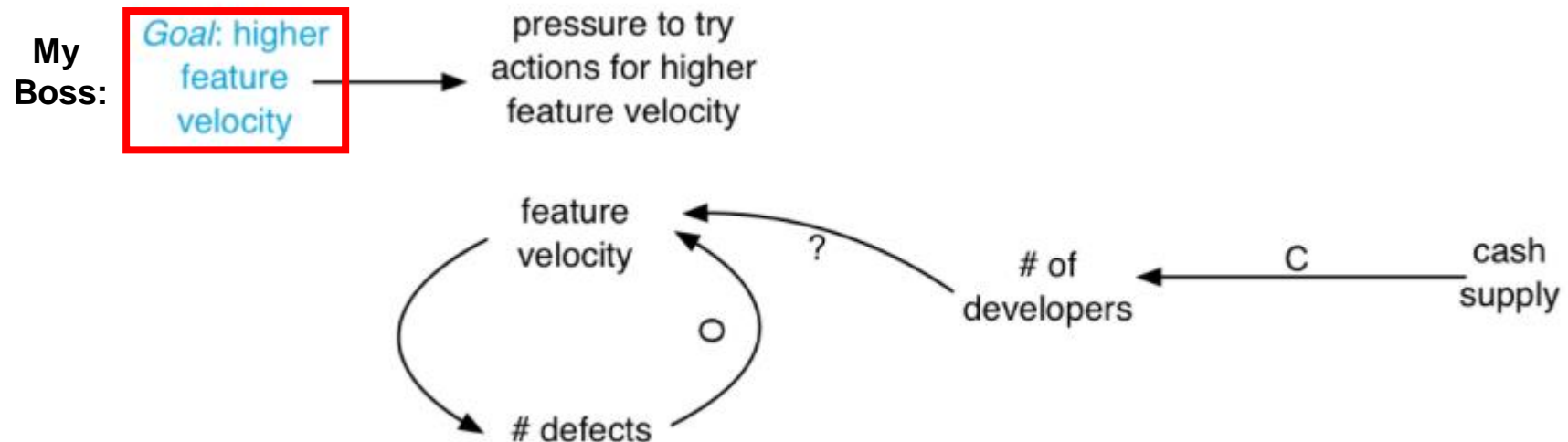


“ In the most basic sense, a system is any group of interacting, interrelated, or interdependent parts that form a complex or unified whole that has a specific purpose. ”

**Daniel Kim** [thesystemsthinker.com](http://thesystemsthinker.com)



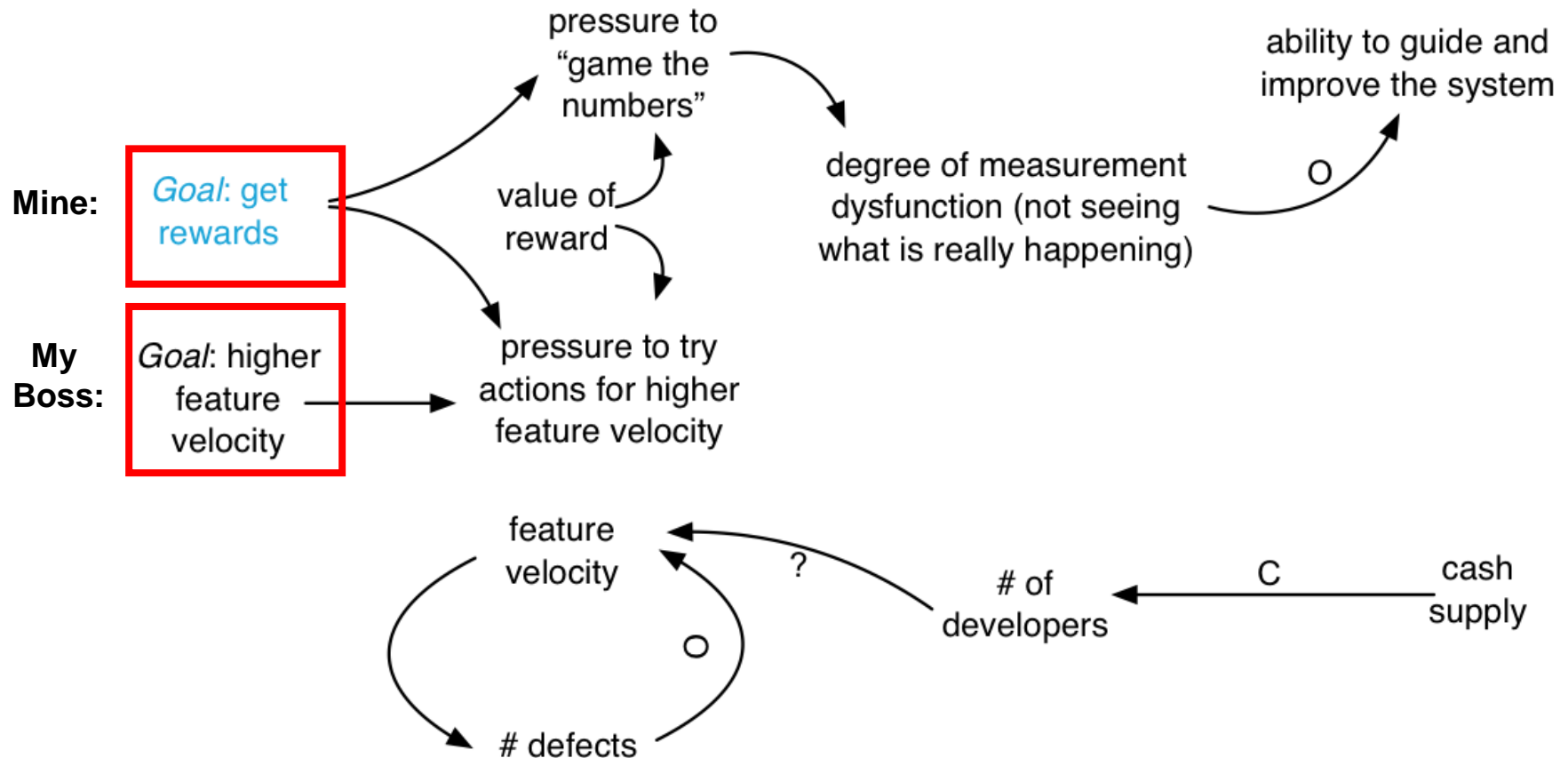
# Seeing, Thinking & Modelling The System



<https://less.works/less/principles/systems-thinking>



# Seeing, Thinking & Modelling The System



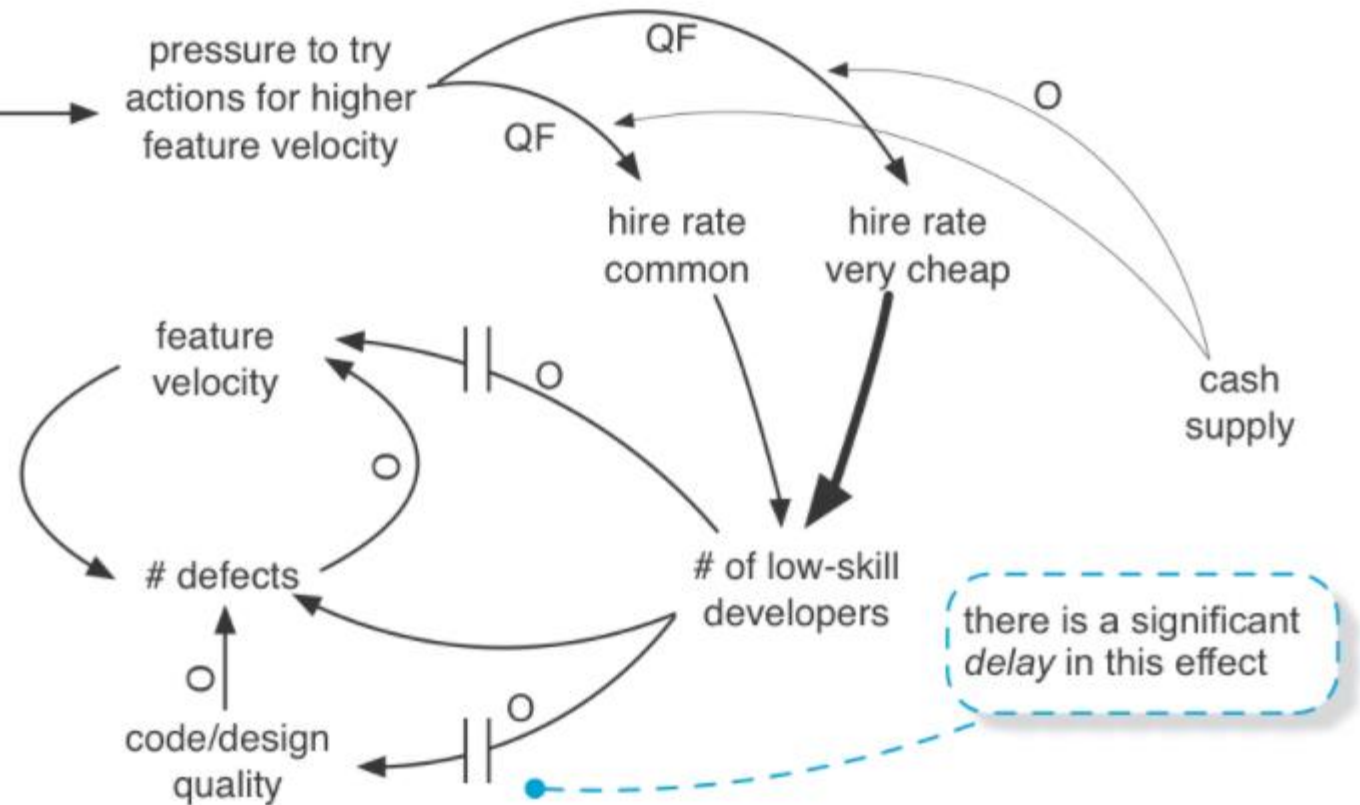
<https://less.works/less/principles/systems-thinking>



# Seeing, Thinking & Modelling The System

My  
Boss:

Goal: higher  
feature  
velocity



<https://less.works/less/principles/systems-thinking>



# Seeing, Thinking & Modelling The System



Brooks's Law: Adding manpower to a late software project makes it later.

(Fred Brooks)

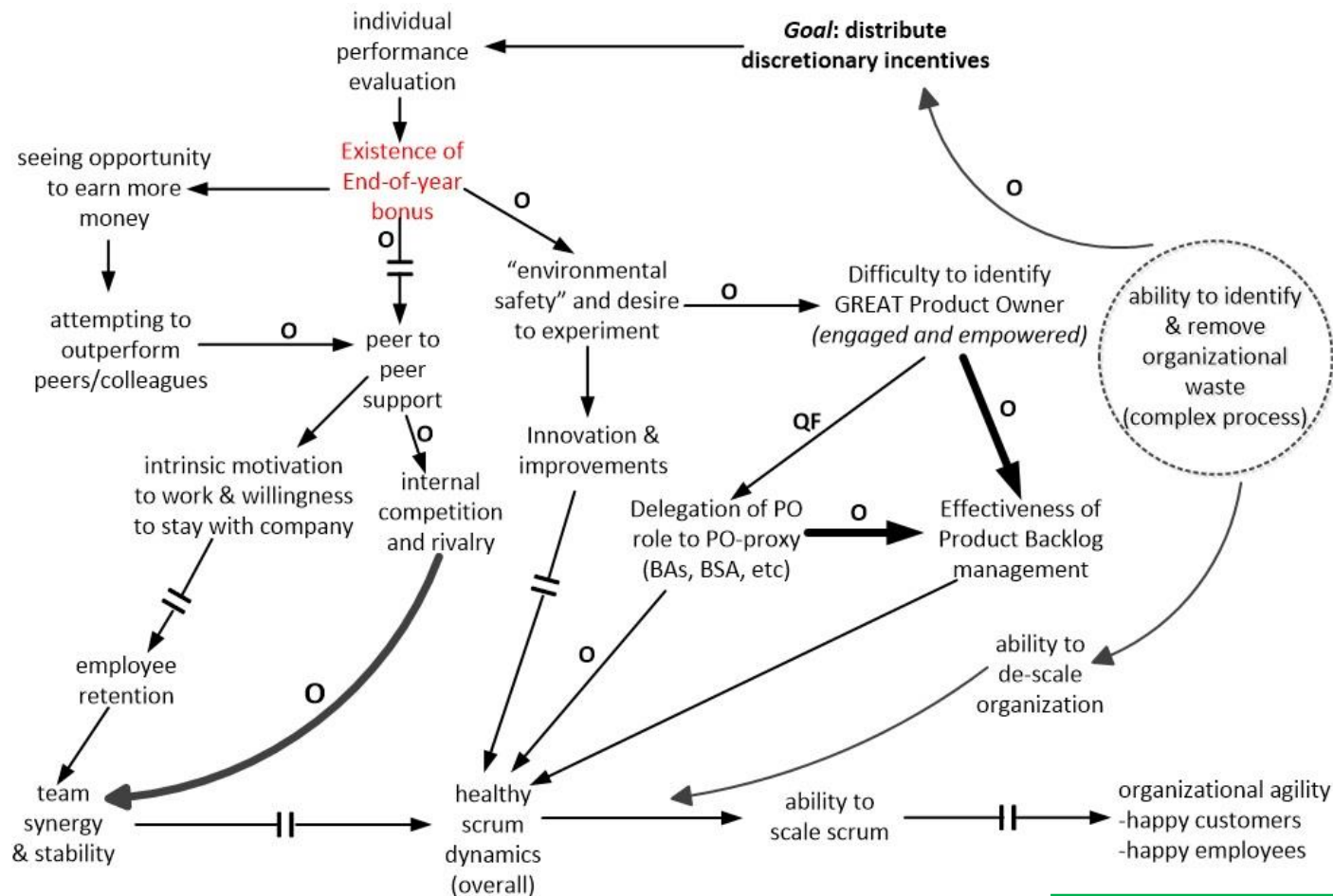
## CONWAY'S LAW

*"Any organization that designs a system will inevitably produce a design whose structure is a copy of the organization's communication structure."*  
Melvin E. Conway





# Seeing, Thinking & Modelling The System



Sourced from: <https://www.keystepstosuccess.com/coach-tools/>

**NB:** variables that strongly relate to **system optimizing goals** can be highlighted



# Seeing, Thinking & Modelling The System

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



# Seeing, Thinking & Modelling The System

**Linear Thinking Does Not  
Really Help Much...  
In Complex  
Organizational Settings**



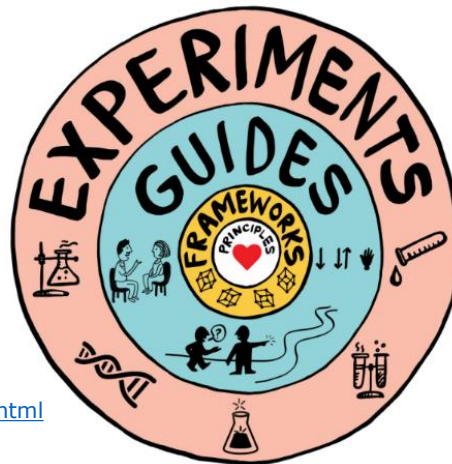






# LeSS

## ***Organizational Design Framework***

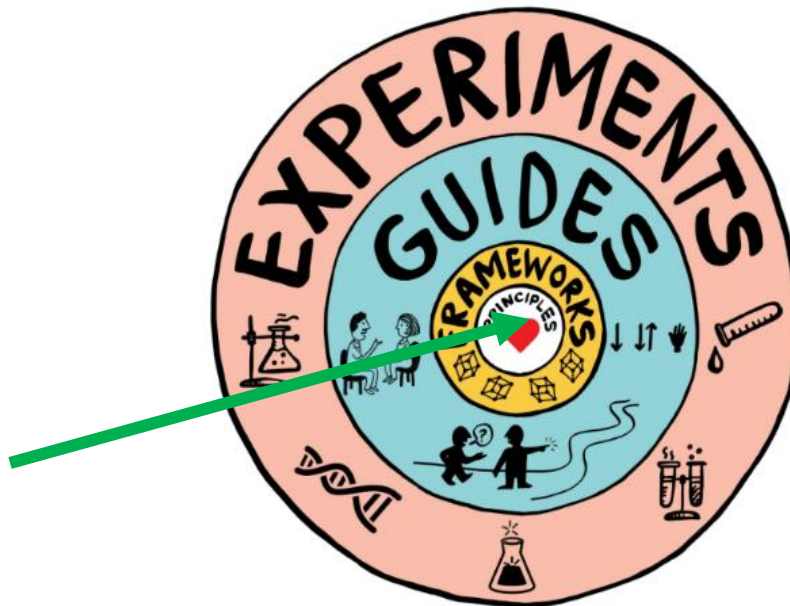


Sourced from: <https://less.works/resources/graphics/index.html>

***LeSS 'Onion'***



# LeSS Principles



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



# LeSS Principles

Understand how systems with queues behave in the R&D domain, and apply those insights to managing queue sizes, work-in-progress limits, multitasking, work packages, and variability.

It is not "new and improved Scrum." LeSS is about applying the principles, elements, and purpose of Scrum in a large-scale context. Multiple-team Scrum, not multiple Scrum teams.

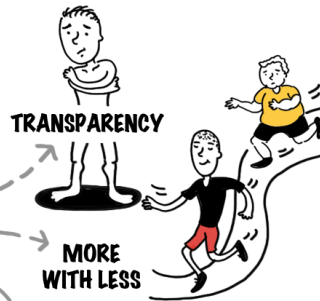
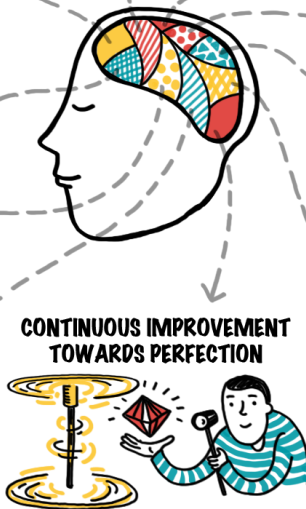
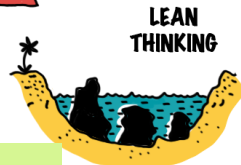
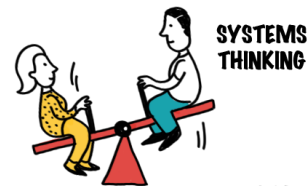
Based on tangible 'done' items, short cycles, working together, common definitions, and driving out fear in the workplace.

Inspection and adaptation of the product, processes, organizational design, and practices to craft a situational appropriate organization based on Scrum, rather than following a detailed formula. And empirical process control requires and creates transparency.

See, understand, and optimize the whole system (not parts), and explore system dynamics. Avoid the local and sub-optimizations of focusing on the 'efficiency' or 'productivity' of individuals and individual teams. Customers care about the overall concept-to-cash cycle time and flow, not individual steps.

Create an organizational system whose foundation is managers-as-teachers who apply and teach systems thinking and lean thinking, manage to improve, and who practice Go See at gemba. Add the two pillars of respect for people and continuous improvement. All towards the goal of perfection.

**LARGE-SCALE SCRUM IS SCRUM**



(1) In empirical process control: more learning with less defined processes. (2) In lean thinking: more value with less waste and overhead. (3) In scaling, more ownership, purpose, and joy with less roles, artifacts, and special groups

One Product Backlog, one Product Owner, one potentially shippable product increment, one Sprint—regardless if there are 3 or 33 teams. Customers want the product, not a part.

Identify value and waste in the eyes of the paying customer. Reduce the cycle time from their perspective. Increase feedback loops with real customers. Everyone understands how their work today directly relates to paying customers.

Create and deliver a product all the time, without defects, that utterly delights customers, improves the environment, and makes lives better. Do humble and radical improvement experiments each Sprint towards that.

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

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



# Class Activity

## Class:

- Whole Class: discuss which mentioned principles are currently followed in your organization. Which ones are not? How to do you see/know this?
- Capture in Class Log

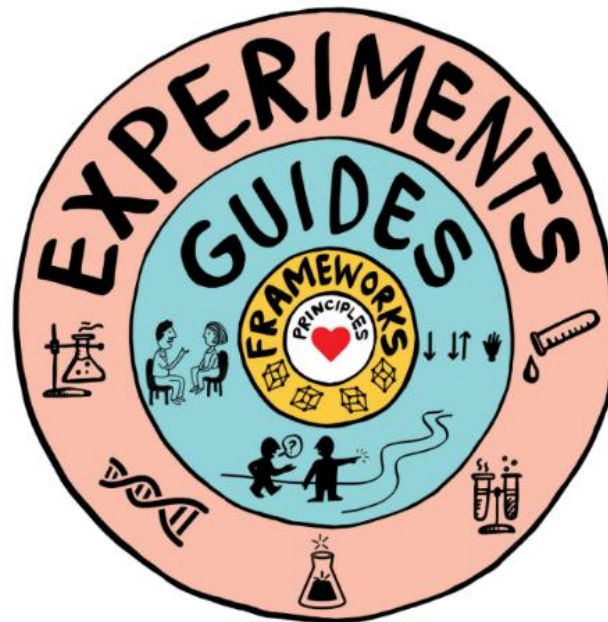
**Instructor:** Catalyze. Reflect.

**Duration:** 10 min



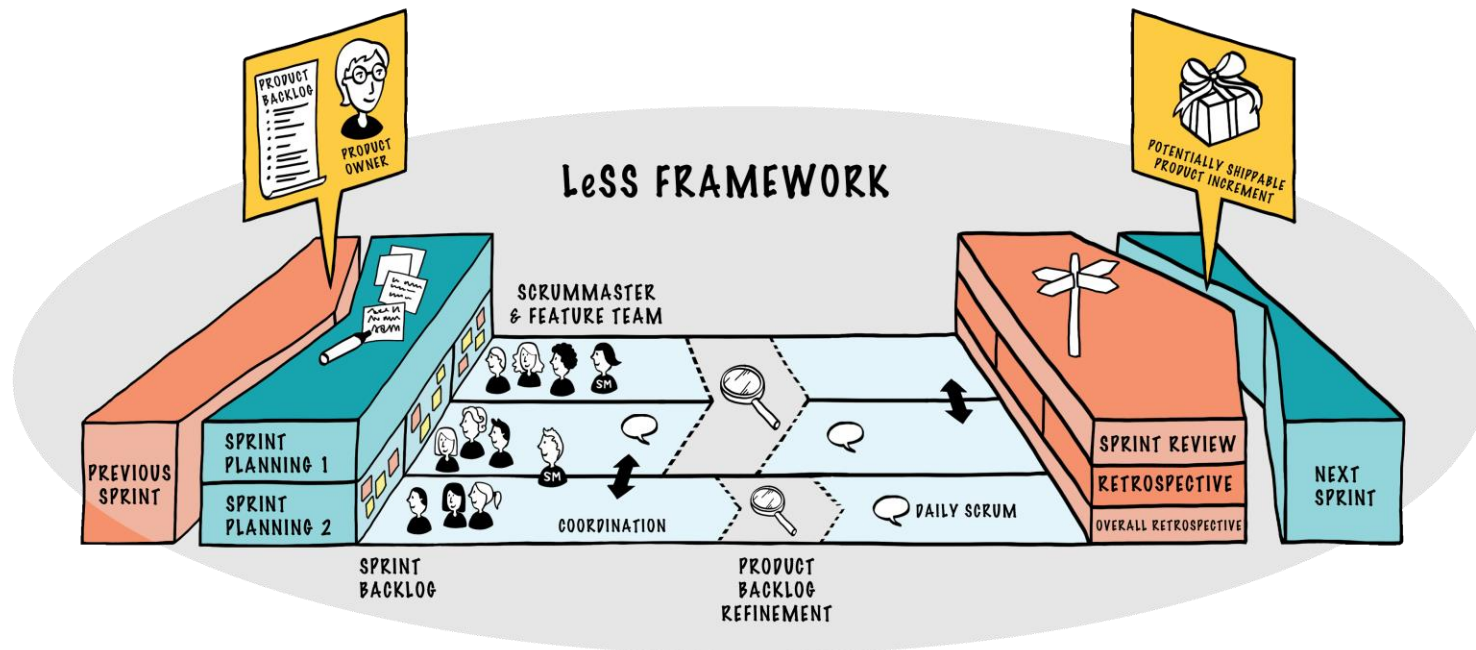


# LeSS Rules



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





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

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





# LeSS Rules: *LeSS Structure*

- Structure the organization using **real teams** as the basic organizational building block.
- Each team is (1) **self-managing**, (2) **cross-functional**, (3) **co-located**, and (4) **long-lived**.
- The majority of the teams are **customer-focused feature teams**.
- Scrum Masters are responsible for a **well-working LeSS adoption**. Their focus is towards the Teams, Product Owner, organization, and development practices. A Scrum Master does not focus on just one team but on the **overall organizational system**.
- A Scrum Master is a dedicated **full-time role**.
- One Scrum Master can serve **1-3 teams**.
- In LeSS, **managers are optional**, but if managers do exist their role is likely to change. Their focus shifts from managing the day-to-day product work to improving the value-delivering capability of the product development system.
- Managers' role is to improve the product development system by practicing **Go See**, encouraging Stop & Fix, and "experiments over conformance".
- For the product group, establish the **complete LeSS structure "at the start"**; this is vital for a LeSS adoption.
- For the larger organization beyond the product group, adopt LeSS **evolutionarily** using Go and See to create an organization where experimentation and improvement is the norm.





# LeSS Rules: *LeSS Product*

- There is **one Product Owner** and **one Product Backlog** for the complete shippable product.
- The Product Owner shouldn't work alone on Product Backlog refinement; he is supported by the **multiple Teams** working directly with customers/users and other stakeholders.
- All **prioritization** goes through the Product Owner, but **clarification** is as much as possible directly between the Teams and customer/users and other stakeholders.
- The definition of product should be as **broad and end-user/customer centric as is practical**. Over time, the definition of product might expand. Broader definitions are preferred.
- One **Definition of Done** for the whole product common for all teams.
- Each **team can have their own stronger** Definition of Done by expanding the common one.
- The perfection goal is to **improve the Definition of Done** so that it results in a shippable product each Sprint (or even more frequently).





# LeSS Rules: *LeSS Sprint*

- There is **one product-level Sprint**, not a different Sprint for each Team. Each Team starts and ends the Sprint at the same time. Each Sprint results in an **integrated whole product**.
- Sprint Planning consists of **two parts**: Sprint Planning One is common for all teams while Sprint Planning Two is usually done separately for each team. Do multi-team Sprint Planning Two in a shared space for closely related items.
- Sprint Planning **One** is attended by the Product Owner and Teams or Team representatives. They together tentatively select the items that each team will work on that Sprint. The Teams identify opportunities to work together and final questions are clarified.
- Each Team has their own **Sprint Backlog**.
- Sprint Planning **Two** is for Teams to decide **how** they will do the selected items. This usually involves design and the creation of their Sprint Backlogs.
- Each Team has their own Daily Scrum.
- Cross-team coordination is decided by the teams. **Prefer decentralized and informal coordination over centralized coordination**. Emphasize **Just Talk** and informal networks via communicate in code, cross-team meetings, component mentors, travelers, scouts, and open spaces.
- Product Backlog Refinement (**PBR**) is **preferably done with multiple teams** to increase shared learning and to exploit coordination opportunities.
- There is one product **Sprint Review; it is common** for all teams. Ensure that suitable **stakeholders** join to contribute the information needed for effective inspection and adaptation.
- Each Team has their **own Sprint Retrospective**.
- An **Overall Retrospective** is held after the Team Retrospectives to discuss cross-team and system-wide issues, and create improvement experiments. This is attended by Product Owner, Scrum Masters, Team representatives, and managers (if any).



# Class Activity

## Class:

- Identify and capture some (all?) LeSS rules that seem to be impossible to meet under given organizational conditions.
- Capture in Class Log

**Instructor:** Catalyze. Reflect.

**Duration:** 15 min









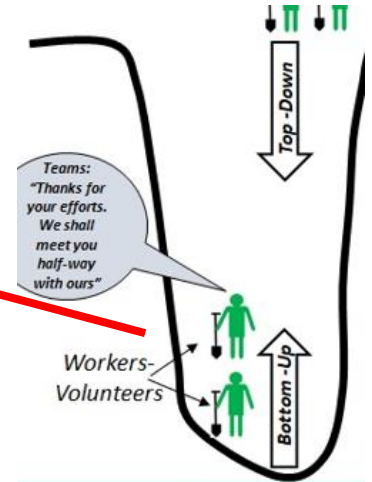
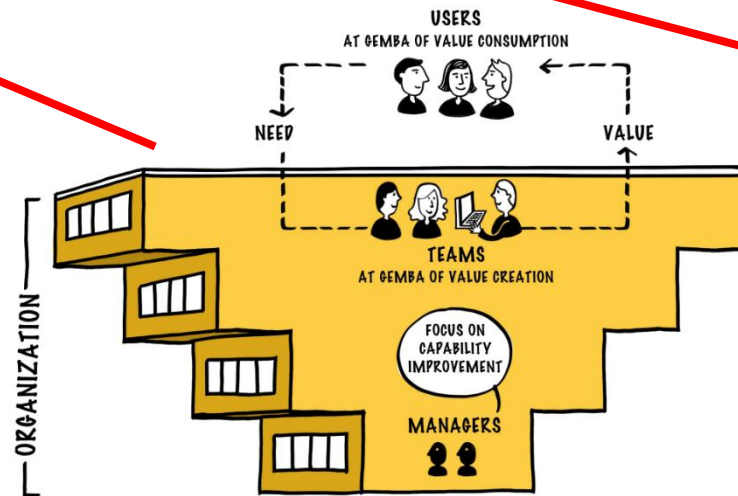
# **3 (Three) LeSS Adoption Principles**



# LeSS Guides: *Adoption*

## Three Adoption Principles

- deep and narrow over broad and shallow
- top-down and bottom-up
- use volunteering



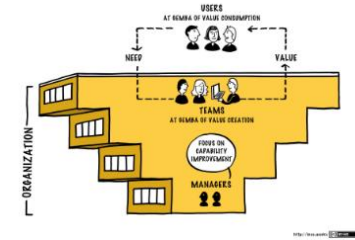
<http://less.works> BY-ND

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

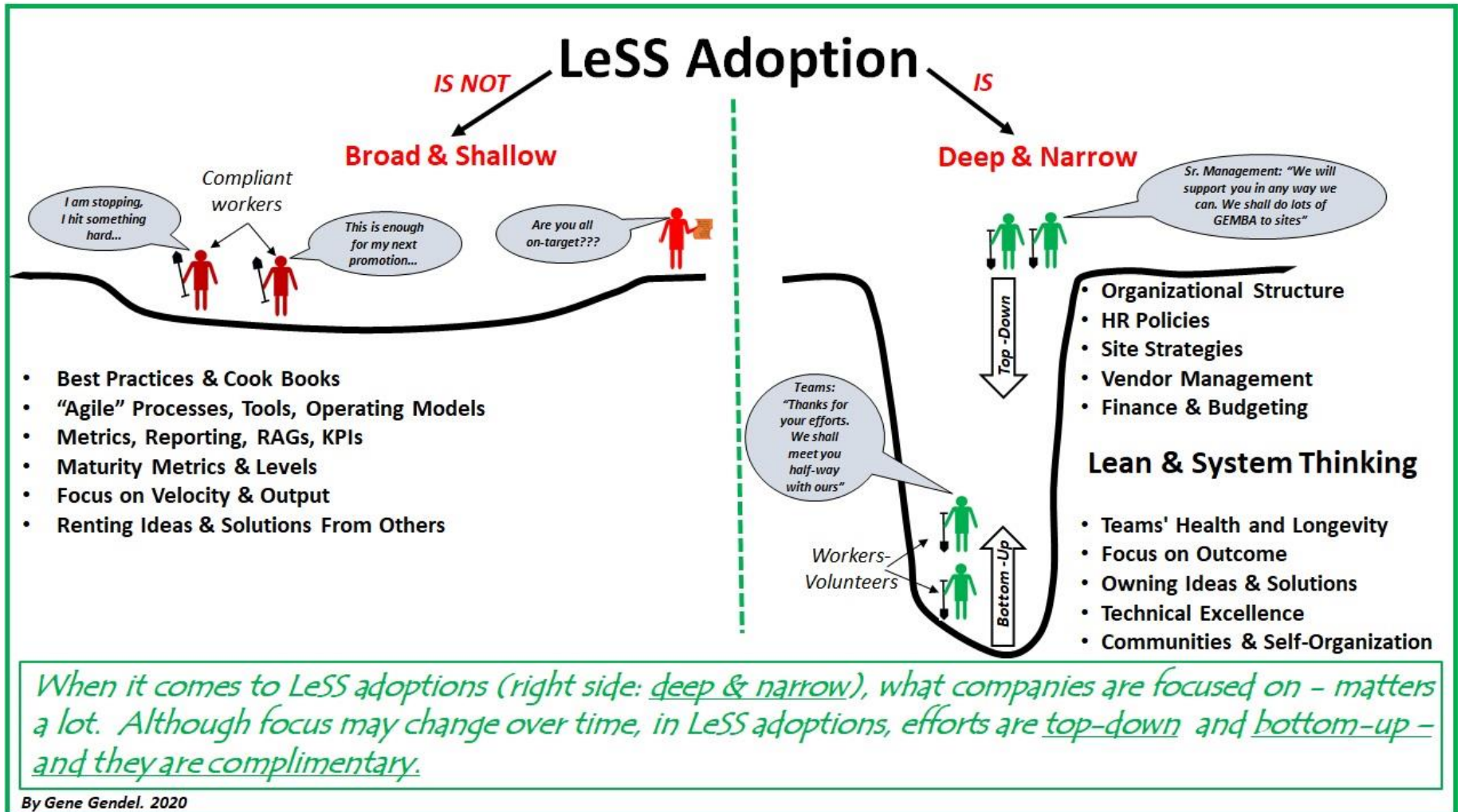
<http://less.works> BY-ND



# LeSS Guides: Adoption



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





# Class Activity

## Class:

- Reflect on LeSS adoption principles.
- Reflect on current agile transformation principles of your organization (goals, targets, focal points)
- Capture in Class Log

**Instructor:** Catalyze. Reflect.

**Duration:** 10 min









$$8 \times 3 - 2$$



# The First Order Factor

*(for organizational agility)*



# Organizational Design

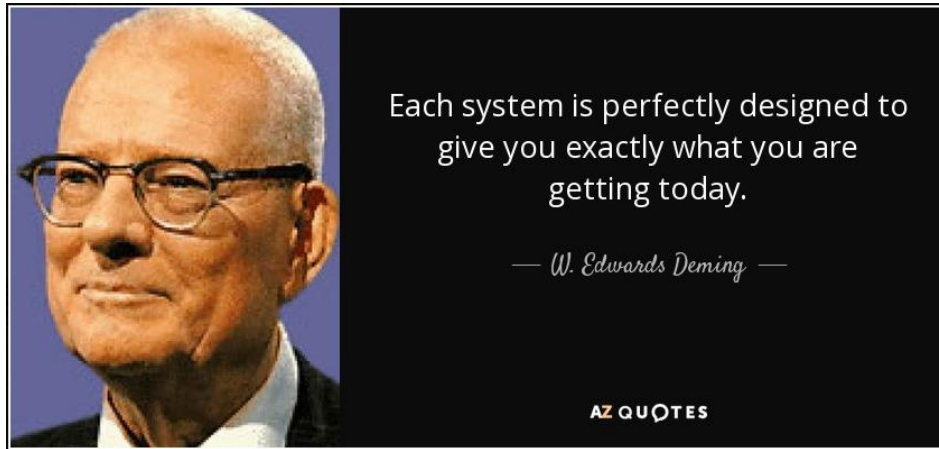
*...this requires some clarification...*

**Organizational STRUCTURE –  
is the 1<sup>st</sup> Order Factor (Variable) that  
has impact on everything else in an  
ECOSYSTEM: behaviors, norms,  
values, principles, policies**



# Organizational Design

*...and therefore, it would be reasonable to expect...*



Source from: <https://www.azquotes.com/quote/1411417>

**Organizational Design elements  
should be consistent with the  
System Optimizing goals  
(...passing “consistency test”...)**



# Organizational Design

Traditional large groups are complicated — not because they must, but because **their organizational designs create an illusion of ‘necessary’ complexity**

Craig Larman





# Local Optimization

## Larman's Laws of Organizational Behavior

([Spanish translation](#))([Russian translation](#)) ([Japanese translation](#))([Persian/Farsi translation](#))([Polish translation](#))([Portuguese translation](#))

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. (in large established orgs) *Culture follows structure*. And in tiny young orgs, structure follows culture.

Elaboration:

Sourced from: [https://www.craiglarman.com/wiki/index.php?title=Larman%27s\\_Laws\\_of\\_Organizational\\_Behavior](https://www.craiglarman.com/wiki/index.php?title=Larman%27s_Laws_of_Organizational_Behavior)



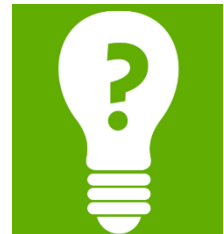
# Class Activity

## Class:

- Whole Class. Discuss, what is being emphasized mostly in your organizational settings? What areas/domains get a lion's share of attention in your organizational transformation? What is not given enough attention?
- Capture in Class Log

**Instructor:** Catalyze. Reflect.

**Duration:** 10 min









# Local Optimization



# Local Optimization

*(“This Is Not My Job”)*



Sourced from: <http://www.tsmplug.com/games/pes-2014-download/>



# Local Optimization

...whereas, **Global optimization** refers to finding the optimal value of a given function among all possible solution...

... **Local optimization** finds the optimal value within the neighboring set of candidate solution...

<https://www.lqi-global.com/dictionary/from-optimization-to-clustering/45858>

[Pentagon Wars – Bradley Fighting Vehicle Evolution](#)





# Local Optimization

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

How could that be?



Sourced from: <https://www.youtube.com/watch?v=5unMIXg6WL4>



# Local Optimization

## Frequently Heard Justifications:

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





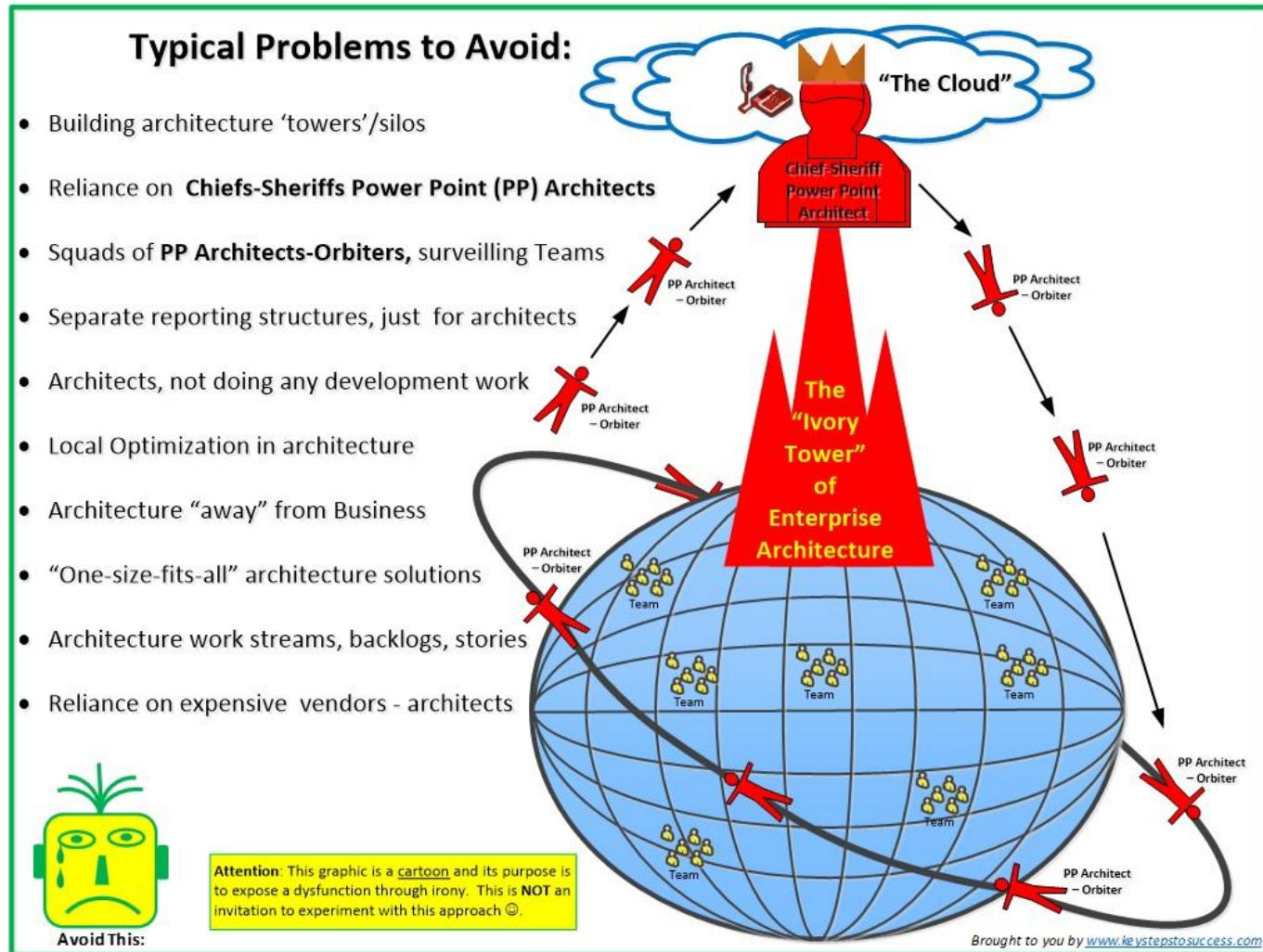
# Local Optimization In UI/UX Design



Sourced from: <https://www.meetup.com/Design-Sprint-NYC/>



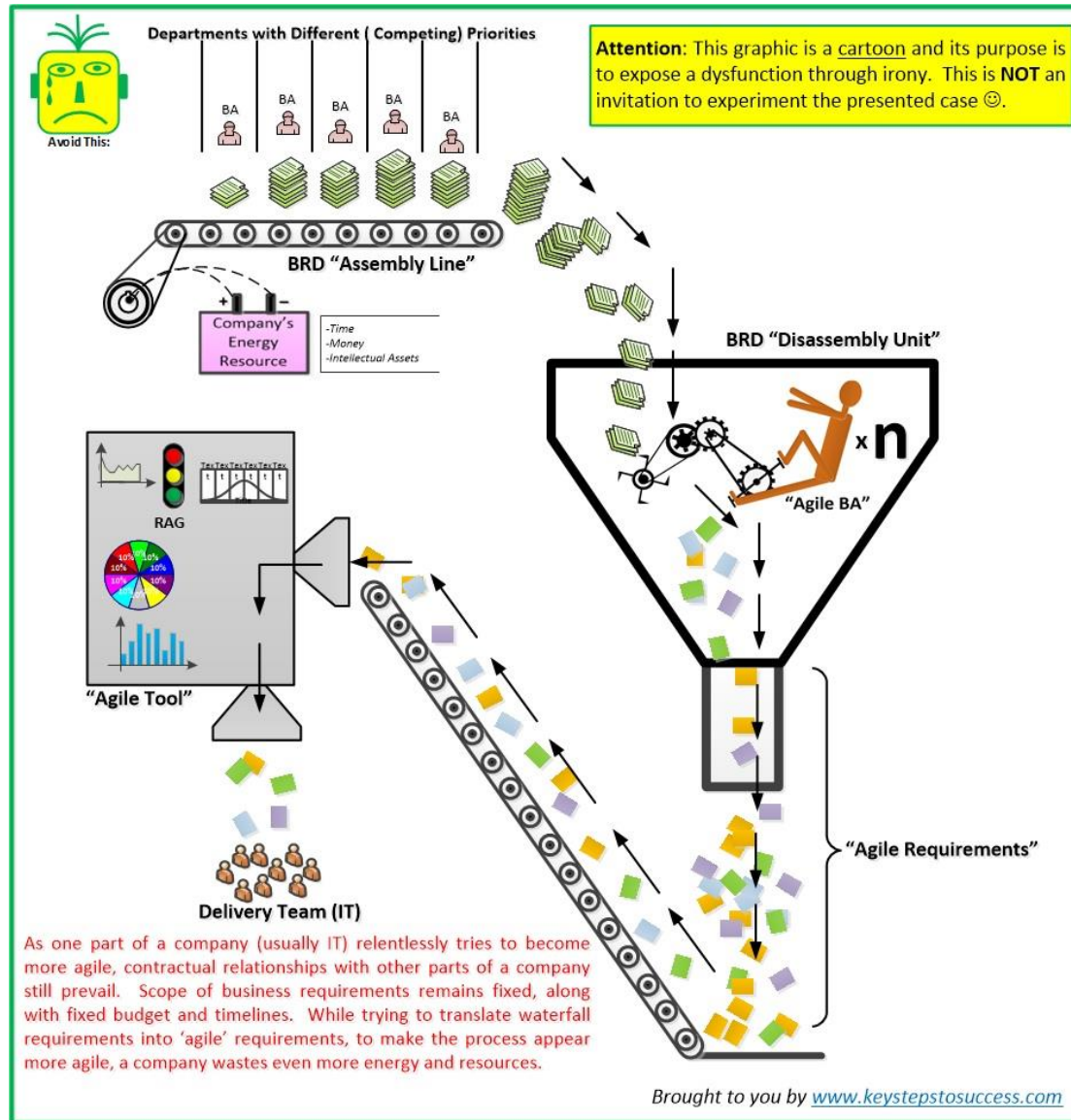
# Local Optimization



Sourced from: <https://www.keystepstosuccess.com/agile-anti-patterns-with-irony-and-satire/>



# Local Optimization



Sourced from: <https://www.keystepstosuccess.com/agile-anti-patterns-with-irony-and-satire/>



# Class Activity

**Class:** Review instructions of a system modelling exercise on the next slide. Split up into teams.

**Instructor:** Provide instructions.

**Duration:** next page





## Exercise

**Duration:** 10 min

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

**Method/Tool:** System Modeling with CLD

**Instructor:** review with Class



# Local Optimization in **Tool-Driven WBS** - Exercise

Likelihood that WBS will mimic a hierarchy of issue type schema, codified in a tool

Dependency on heavy, commercially successful frameworks (e.g. S@#%)

Degree of emphasis an organization makes on metrics and reporting

Reliance on workflow management tools, as means of 'agile project management'

Degree of freedom teams have, when refining and splitting work

Number of ex-managers, looking for ways to stay busy

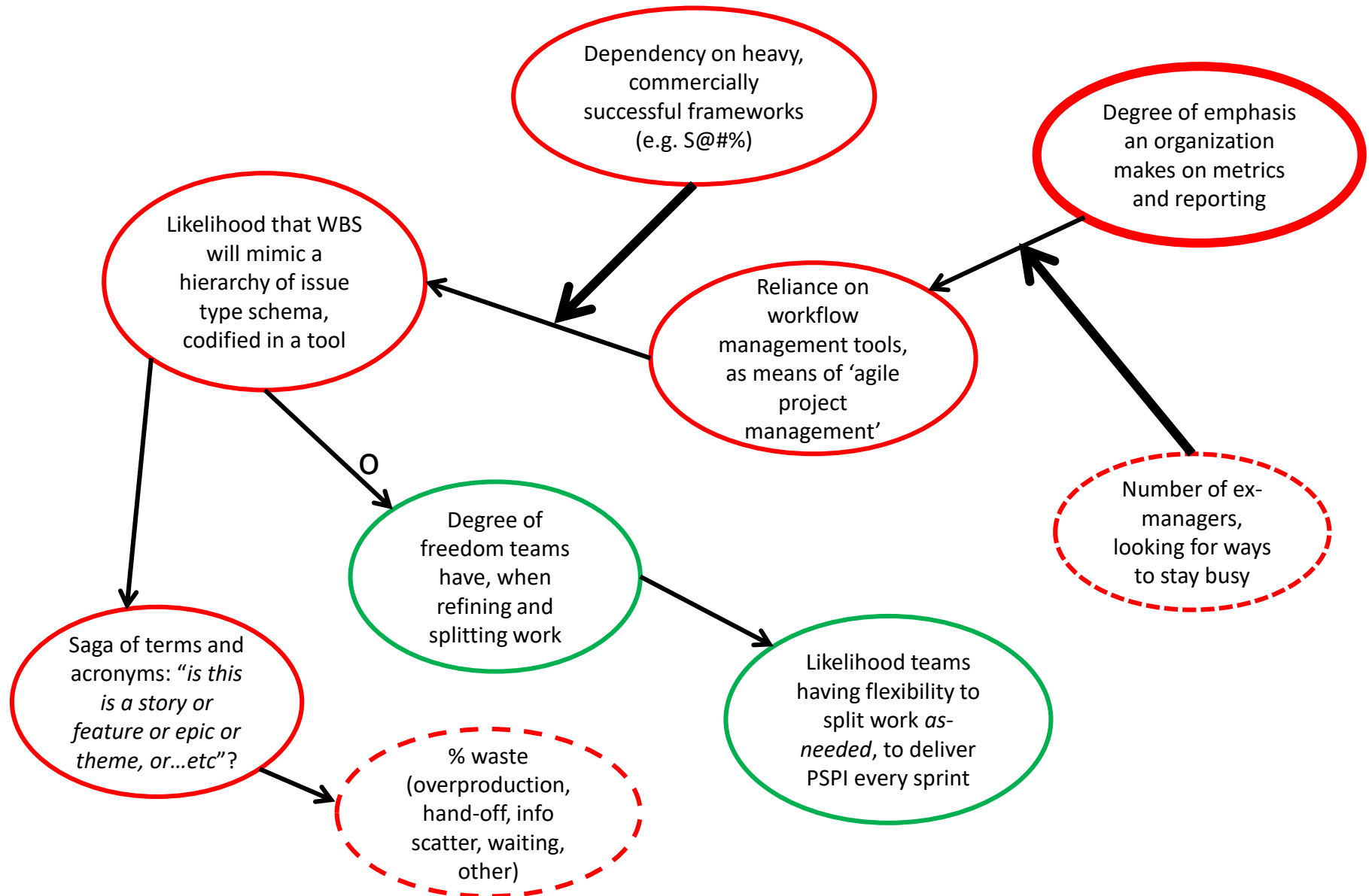
Saga of terms and acronyms: *"is this is a story or feature or epic or theme, or...etc"*?

Likelihood teams having flexibility to split work *as-needed*, to deliver PSPI every sprint

% waste (overproduction, hand-off, info scatter, waiting, other)



# Local Optimization in **Tool-Driven WBS** - Exercise









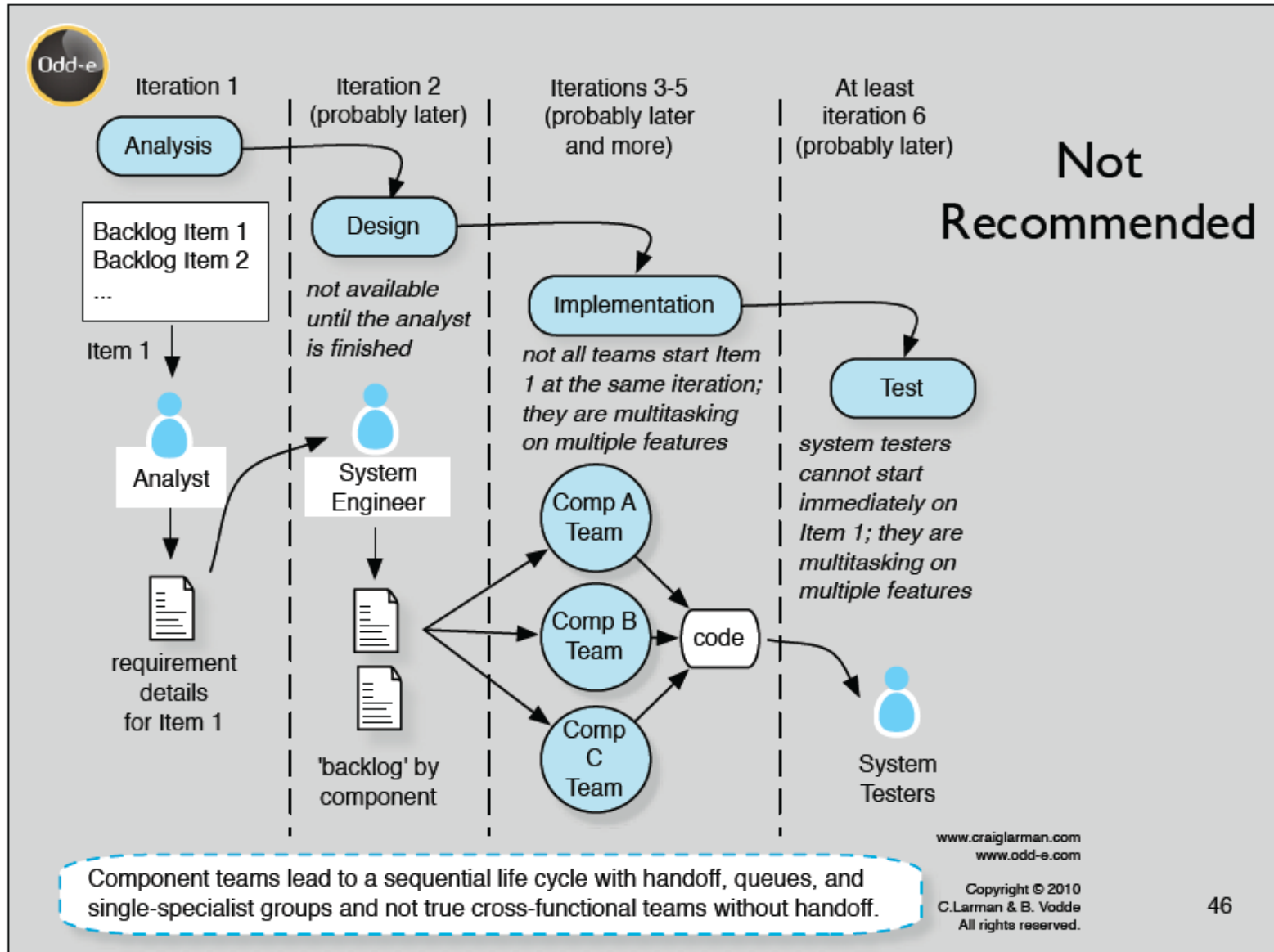
# Local Optimization



<https://www.youtube.com/watch?v=hNuu9CpdjIo>



# Local Optimization



Sourced from creative commons repository of Certified LeSS Trainers: <https://less.works/courses/become-less-trainer>



# Local Optimization

## Seeing (and Hearing) Local Optimization

“Everyone is doing their best yet overall systems throughput is degrading. How can that be?” This is the paradox of **local optimization** —when a person or departmental decision maker optimizes for the local view or self-interest. The party making the decision frequently *believes they are making the best decision*, but because ‘best’ is a local optimization, in fact it sub-optimizes overall system throughput. This is a result of “silo mentality,” misunderstanding, fear, limited information, delayed feedback, ignorance, careerism, avarice, and other common *organizational learning disorders*.

**Team Structures**

**Org. Structures**

**Documentation**

**Definition of Done**

**Backlogs**

**Role Definitions**

**Product Design**

**Goals & Metrics**



# Local Optimization

*...by DE-scaling with LeSS, we...*

Reduce Local Optimization in	In favor of
In narrow definition (due to narrow products, programs, portfolios, value streams and their org. design elements)	Broader product definition
Private backlogs (due to team-specific backlogs and their org. design elements)	One product backlog
Fixed-everything Planning (due to Contract Game)	Adaptive planning by a business-side Product Owner
Component development (single-specialty groups)	Feature teams that code cross-components with shared code
Analysis & design (due to separate analysts, designers, team-“PO” and “proxies”)	Multi-skillset feature teams, clarifying directly with users



$$5 \times 5 - 2$$



# Class Activity

**Class:** Review instructions of a system modelling exercise on the next slide.

**Instructor:** Provide instructions. Split up into teams.

**Duration:** next page





## Exercise

**Duration:** 10 min

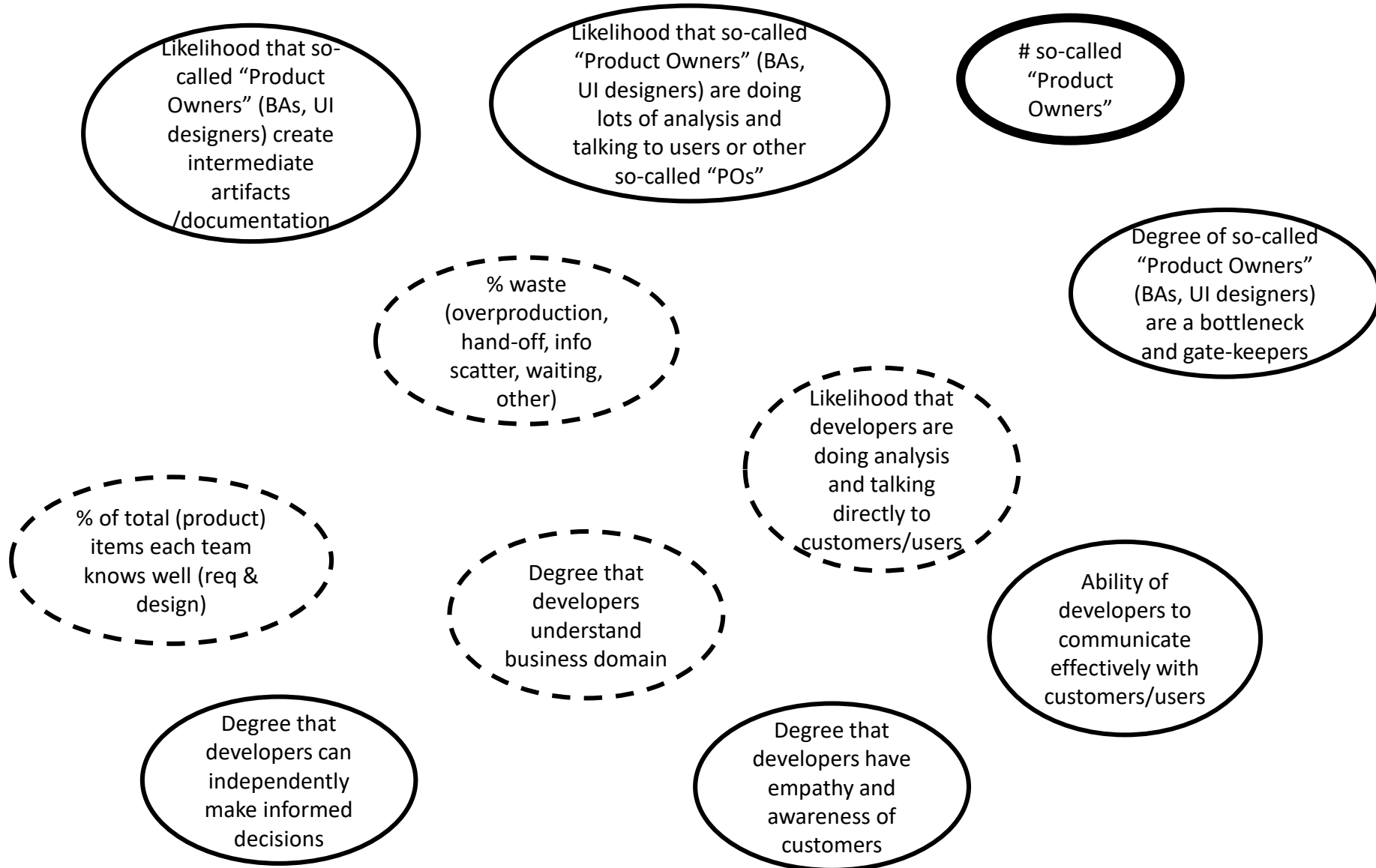
**Class:** in-groups, brainstorm some of the most common examples of ***Local Optimization in Analysis and Design*** in your respective organizations. Work with provided system variables to create a model.

**Method/Tool:** System Modeling with CLD

**Instructor:** review with Class

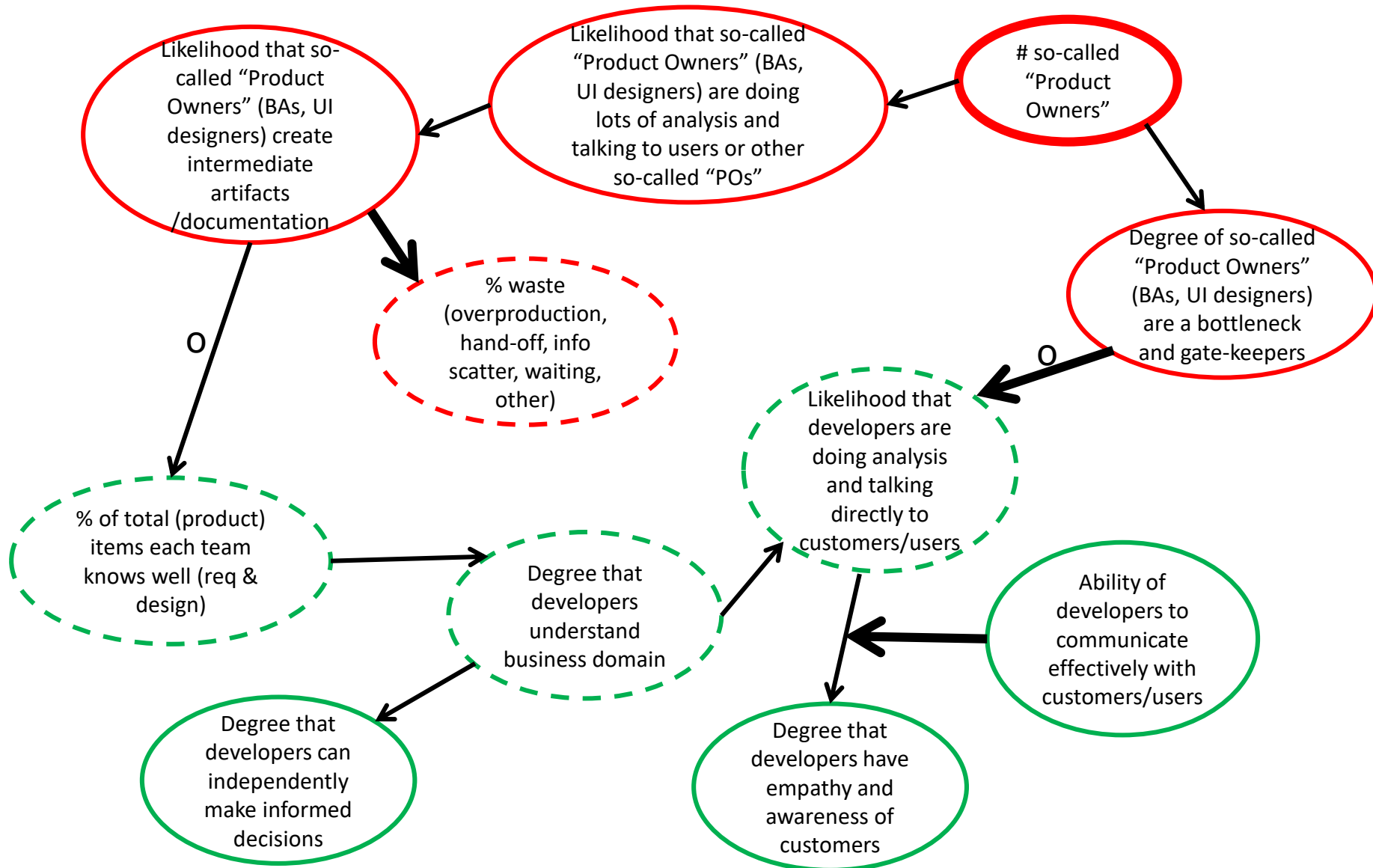


# Local Optimization in **Analysis & Design**- Exercise





# Local Optimization in **Analysis & Design**- Exercise

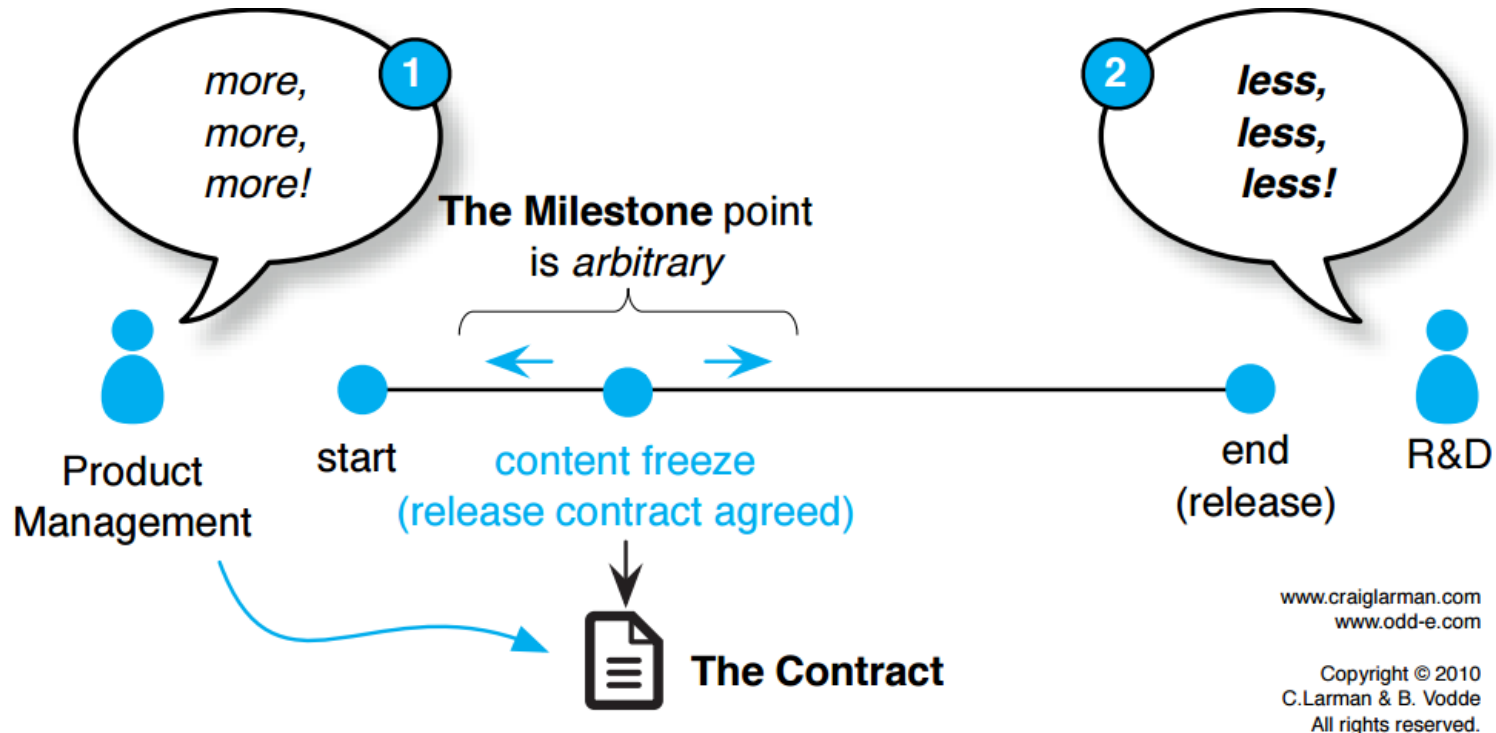








# Internal Contracts



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

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



# Internal Contracts

## AGILE CONTRACTS PRIMER

Derived from the book...

*Practices for Scaling Lean & Agile Development:*

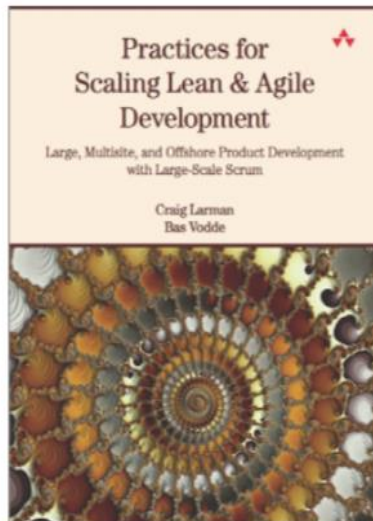
*Large, Multisite, & Offshore Product Development with Large-Scale Scrum*

by Tom Arbogast, Craig Larman, and Bas Vodde

Version 5

Please send us comments for future versions, at [www.agilecontracts.org](http://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
<b>Action Tools</b>		
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

## Survival Sheet to Vendor Selection on Agile Projects

### How to choose Vendor?

- **Vendor Management System (VMS)** – The easiest thing to do, would be to refer to and pick from VMS, as long as a vendor card-rate is in a ballpark of what you wish to pay. **But please, don't do that.** Do not let costs become the most important determining factor in your selection. There is a chance that a vendor you are about to choose ended up in VMS, based on old selection criterion, very different from what you might be looking for, for agile work. Do not automatically assume that old relationships will seamlessly work under new conditions, while operating in new ways.
- **Case Studies / Use Cases / References** – They could be great ways to understand if a vendor is really capable to do what they say they can do. As a client, be always skeptical about heavy, well-formatted power point decks, with lots of fine-print, when they are used by a vendor during initial presentations. Ask for practical demonstrations, working solutions and engage a vendor in extensive Q&A – and please make sure that real hands-on doers present/answer, NOT engagement/sales managers that are specifically trained to make a great first impression. Whenever possible, ask for references from other clients of the same vendor, to provide feedback about similar work that was performed for them.
- **Interviewing Vendor workers** – Make sure that you interview every person from the vendor-side, who will be involved in performing work. Be on a lookout for workers that were just hired by a vendor or *swapped* (for other workers) last minute, just before work commenced, or were being asked to split their time with other projects/clients.

### How to structure your ongoing relationship with Vendor?

- **SOW** – Regarding SOW types (e.g. Design/Detail, T&M, Performance Based), try avoiding 'fixed everything' (time, scope, budget) agreements. When all three corners of a 'management triangle' are locked, work becomes very non-adaptive/non-agile/rigid. It will increase risk aversion and decrease interest to experiment and innovate. Try building in some contingencies and flexibility into one of the three above variables. Don't fix-plan work by using waterfall tooling (project plans, Gantt charts, etc.)
- **Location of Vendor people** – Ideally, bring vendor workers on-site (client) and fully integrate them (physical space, daily interaction) with your internal workers. Once engaging vendor people, don't treat them as 'second-class citizens'. Engage them in team-bonding and other social activities to minimize polarization and other adverse behaviors that are typical in contractual relationships. If geographic distribution is inevitable, at least, try to engage with a vendor in the same time zone.
- **Strong Partnership at Senior Leadership level** – It is imperative to establish close working ties between senior leadership of a client company and senior leadership of a vendor company, not just at the time of SOW creation but beyond it. A relationship must be genuine, multi-dimensional and long-lasting. Client leadership must keep vendor leadership well informed of long-term company strategy, vision and expected future service needs. Vendor leadership must keep client leadership well informed of its internal dynamics, such as staffing limitations, plans for expansion to another geographic area, etc. If client-vendor relationships at senior leadership level are superficial and contractual only, it will likely lead to disjuncture and miss-alignment at team (workers) level down the road. Periodic retrospectives between leaderships of both sides, facilitated by a third, impartial party - are strongly recommended.
- **Communication with Vendor Workers** – Communicate directly with doers, not with their line/engagement managers or alike proxies/conduits. Make sure that intra-team (e.g. Scrum, Kanban) relationships between vendor people and client people prevail over reporting relationships on a vendor side.
- **Investing in Vendor learning** – Invest in education and training of vendor people *if you think this will strengthen your relationship and there will be a notable ROI*, while they work for you (client). Be also wise about who you invest into and to what extent. Make sure you don't (over)invest in what a vendor was expected to know from start.
- **Multiple Vendor Involvement** – Be on a look out for any signs of potential rivalry or competition between multiple, concurrently engaged, vendors - this will jeopardize a healthy working environment. Avoid assigning activities to different vendors in ways that increase hand-overs and lead to additional contractual relationships and sequential work (e.g. vendor A - design/development, vendor B - architecture, vendor C - testing).

### How to track progress of your relationship with Vendor?

- **Progress Tracking & Communication media** – Select a single "source of truth" to capture, track and visualize work by agile teams. If a physical board is not sufficient, leverage an electronic tool but make sure that there are no multiple versions information (metrics, reporting, statuses, RAGs etc). Try basing all communications to senior leadership and stakeholders on raw metrics and empirical data that comes directly from teams, without passing through multiple layers of messaging, refinement and interpretation. Avoid having a vendor using one set of work management tools and you (client) - another set.

### How to position Product Ownership in fully outsourced (Vendor) solutions?

- **Client-Vendor interaction** – Make sure that product ownership represents you (client), clearly and unambiguously. A product owner should be positioned organizationally in a way that he/she faces *externally*, and communicates with/sets priorities to doers/team members (vendor) directly (not through engagement managers, BAs or other translation layers), as well as *internally* - by closely interacting with SMEs, stakeholders and other internal customers, with the latter providing clarifications but not setting priorities.

<https://www.keystepstosuccess.com/2019/10/about-contracts-that-support-agile-ways-of-working/>

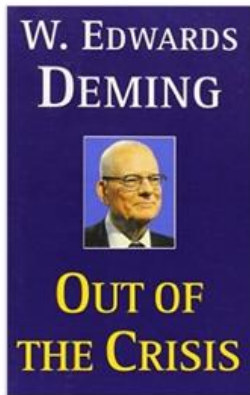


# Internal Contracts

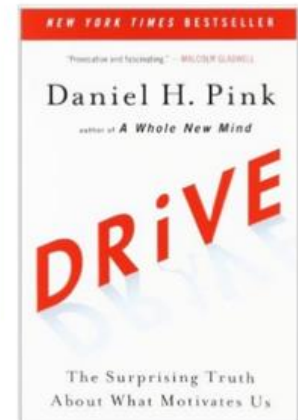
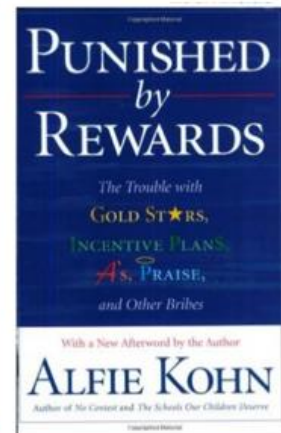
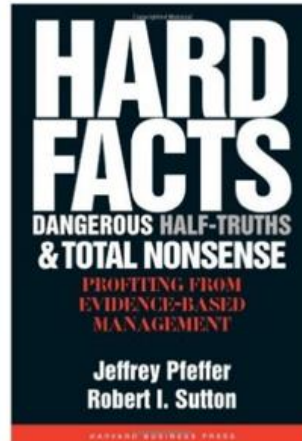
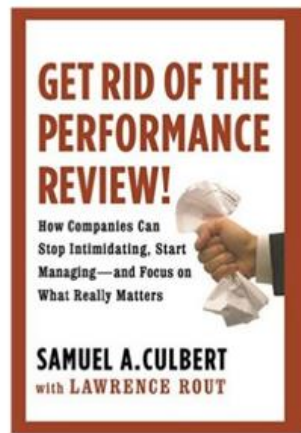
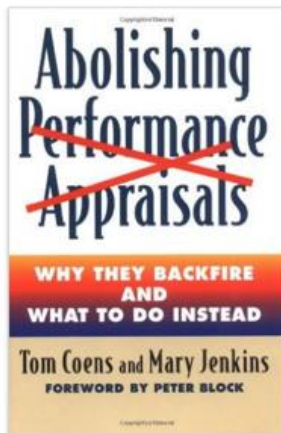
## *HR Involvement is Huge*

### RSA ANIMATE: Drive: The surprising truth about what motivates us (Daniel Pink)

#### *Gap Between Science and Business*



"The idea of a merit rating is alluring. The sound of the words captivates the imagination: pay for what you get; get what you pay for; motivate people to do their best, for their own good. The effect is exactly the opposite of what the words promise."



Created by: Gene Gendel



# Internal Contracts

*HR Involvement is Huge*

structural change:  
formally in HR titles:  
only one **broad** job title

e.g. **Product Developer**

## Big Ideas For Biz!!!

## Big Ideas For R&D !!!



Book-1 Experiments

- *Avoid...Job titles 276*
- *Try...Create only one job title 277*
- *Try...Generic title with levels 277*

Sourced from creative commons repository of Certified LeSS Trainers: <https://less.works/courses/become-less-trainer>



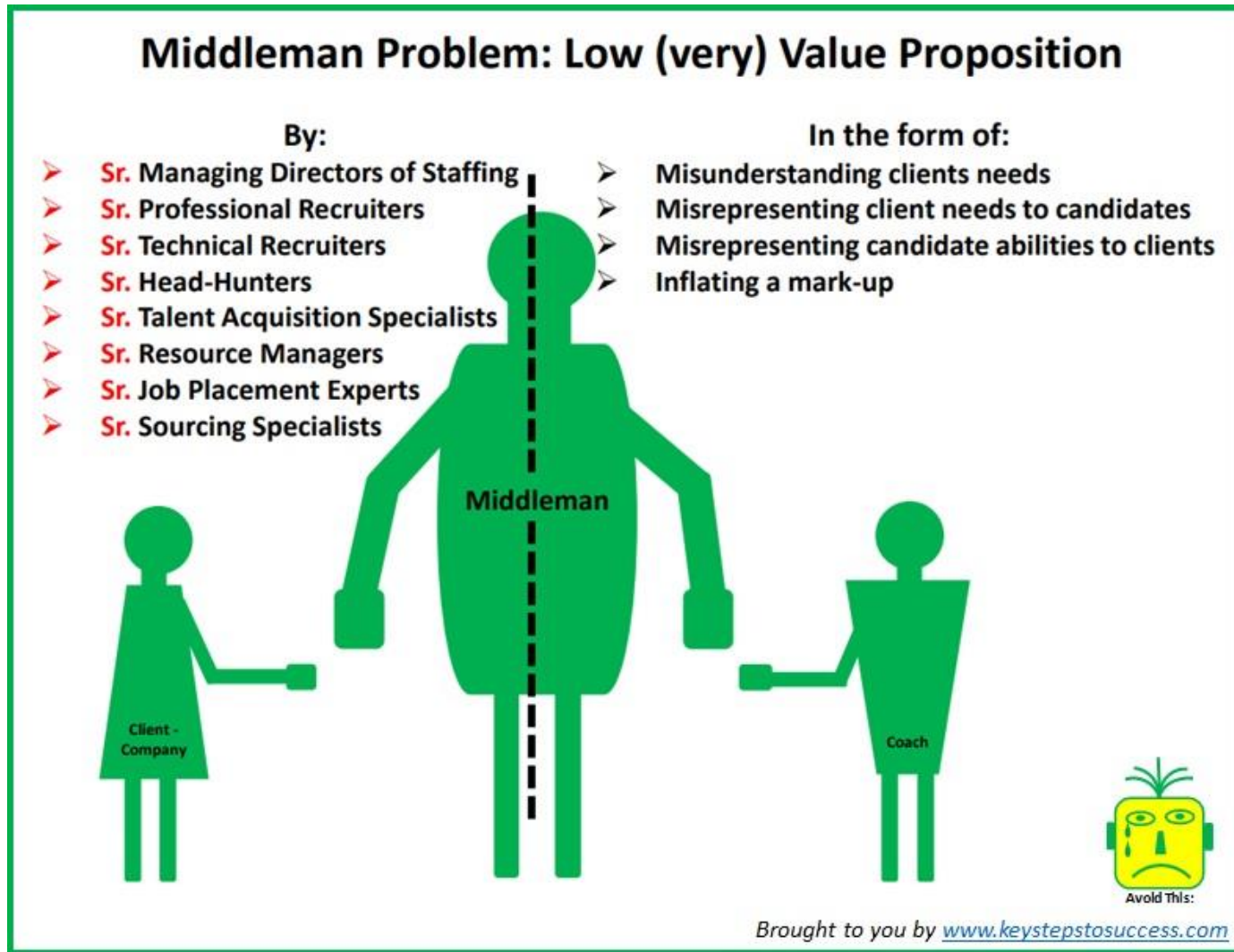
# Internal Contracts

## *HR Involvement is Huge*

- [How HR Can Save or Destroy Agile, by Frank Leong](#)
- [What HR Doesn't Know about Scrum, by Michael James](#)
- [Hitting the target but missing the point – myths about target setting, by Bjarte Bogsnes](#)
- [The "Sandwich Approach" Undermines Your Feedback, by Roger Schwarz](#)
- [Workers feel trapped by "rigid" workplaces, ILM finds, by Jo Faragher](#)
- [Agile Contracts, by Tom Arbogast, Craig Larman, and Bas Vodde](#)
- [Agile HR with SAFE, by Fabiola Eyholzer with Dean Leffingwell](#)
- [The impact of human resource and operational management practices on company productivity: A longitudinal study, on <https://www.researchgate.net>](#)
- [A big Dutch bank is replacing 5,800 people with machines, at a cost of \\$2 billion, by Eshe Nelson](#)
- [The Truth about Performance Reviews and Other Corporate Bullsh\\*t by Liz Ryan](#)
- [Why Your Best Employees Are Leaving For Your Competition, by Chad Halvorson](#)
- [You Could Lose Your Best Employees Because of These 7 HR Blunders, by Sujan Patel](#)
- [Performance Management for Agile People, by TechWell Contributor](#)
- [Rethinking the Review, by Julie Cook Ramirez](#)
- [Unjust Deserts?, by Mary Poppendieck](#)
- [Performance without Appraisal, by Esther Derby](#)
- [Unjust Deserts, by Mary Poppendieck](#)



# Word Of Caution About External Contracts

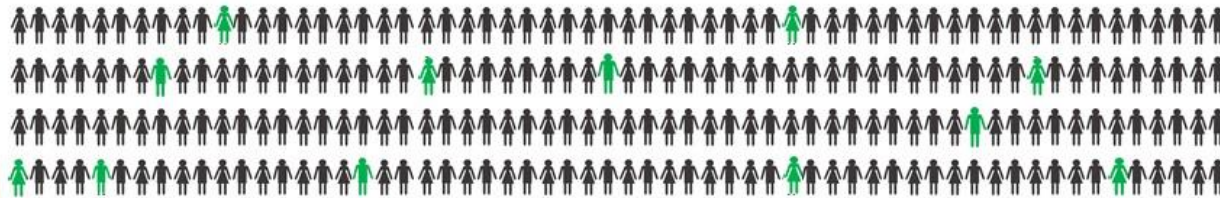


Sourced from: <https://www.keystepstosuccess.com/agile-anti-patterns-with-irony-and-satire/>

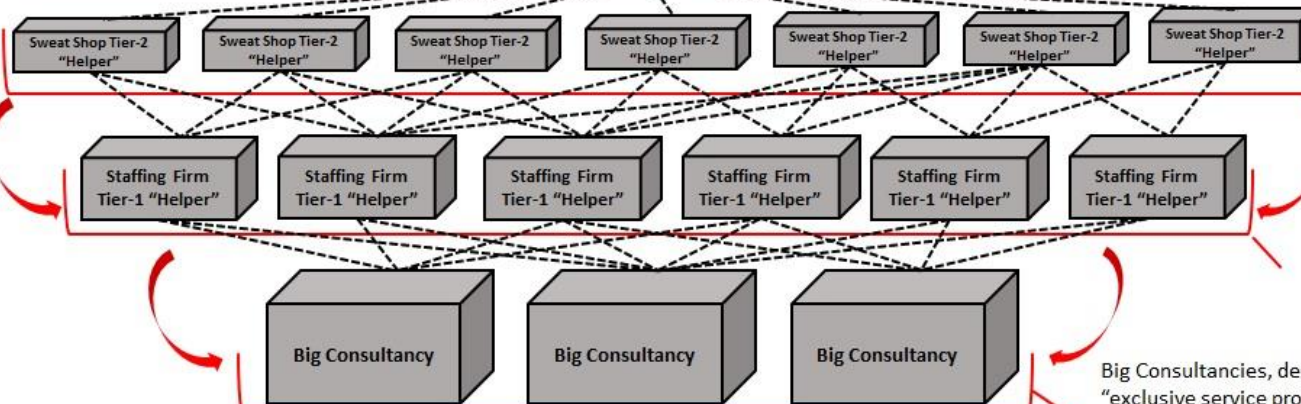


# Word Of Caution About External Contracts

## "Resource-Processing Food Chain"



A huge pool of workers, with the word "agile", in their public profile. Majority of these people are at risk of being displaced from their traditional roles, now wanting to become agile coaches, as per Larman's Law #4. Amongst them, there is a tiny fraction of **really experienced coaches** that are very hard to find in this huge mix.



Remote site-based, "boiler room"-"sweat-shop" style companies, full of VOIP cold-callers that call, text and email every person, whose public profile contains the word "agile" - this is an "agile leads" generation factory.  
**People work for an extremely low pay.**

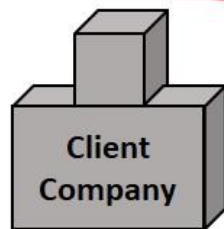
Staffing firms, aggressively trying to hunt for resources, while using primitive (border-line unethical) screening techniques, to submit a heavy volume of profiles to Big Consultancies.  
**People work for a very small mark-up fee.**

Big Consultancies, desperately fighting for an opportunity to become an "exclusive service provider" to Client. While not having their own internal expertise (usually, their consultants are *not* coaches), to meet Client's needs, they now have to hunt for additional resources to fill a gap.  
**Big Consultancies charge Client Companies a lot of money.**

Client Companies that have a shortage of their own internal agile expertise to meet demand for quality training, coaching and organizational design advisory. They naively rely on "best practices" and thick power point decks, presented by Large Consultancies - long time preferred vendors that are great at traditional staffing and cookie-cutting "installations" of heavy frameworks.



Avoid This:



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

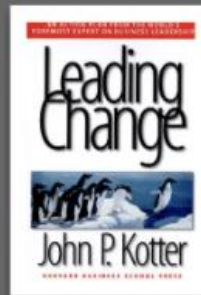
Sourced from: <https://www.keystepstosuccess.com/agile-anti-patterns-with-irony-and-satire/>



# Word Of Caution About External Contracts

"I know what I wouldn't have done, and that is outsourcing the problem to McKinsey... and have them find a solution. The research results are clear. The solutions must come from within the organization."

**Dr. J. Kotter**





# Class Activity

**Class:** Review instructions of a system modelling exercise on the next slide.

**Instructor:** Provide instructions. Split up into teams.

**Duration:** next page





## Exercise

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

**Instructor:** Give short feedback. Offer a recommended solution.

**Duration:** 10 min

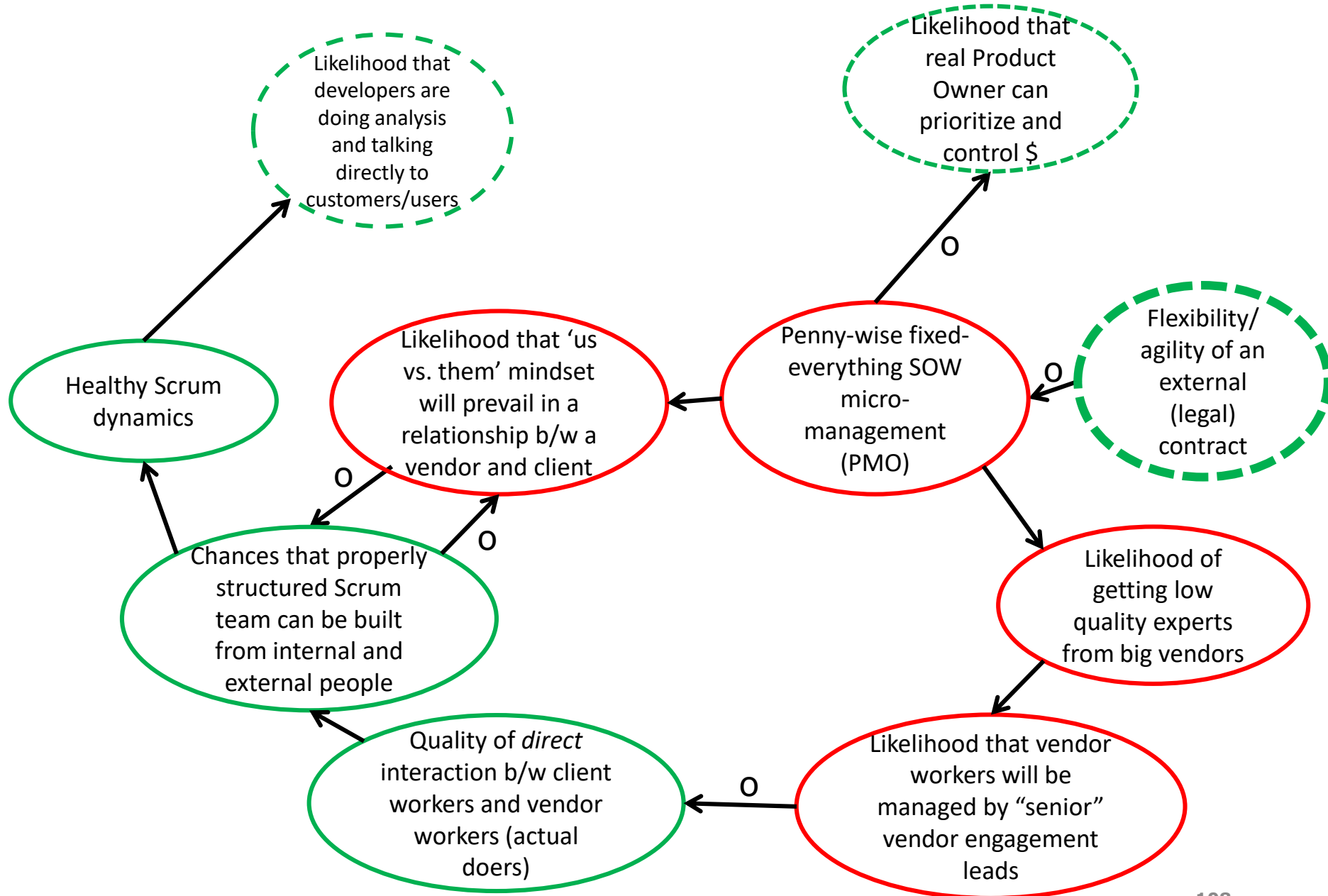


# Local Optimization in **Internal Contracts** - Exercise





# Local Optimization in **Internal Contracts** - Exercise







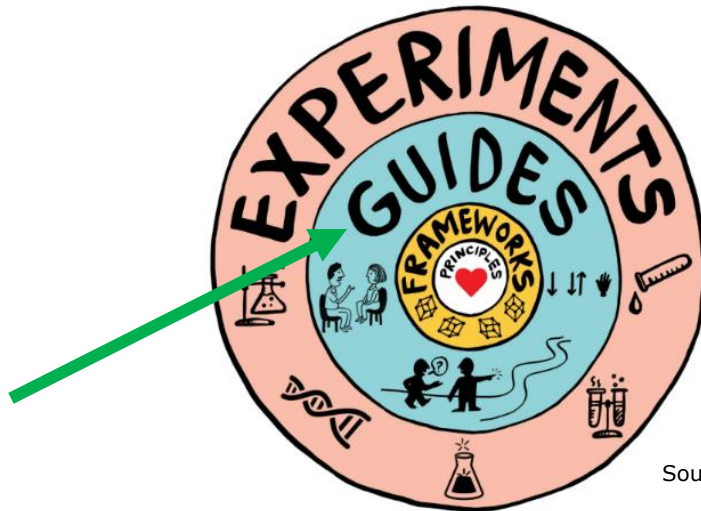


$$8 \times 3 - 3$$



# LeSS Guides

## (Organize by Customer Value)



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



# LeSS Guides: *Organize by Customer Value*

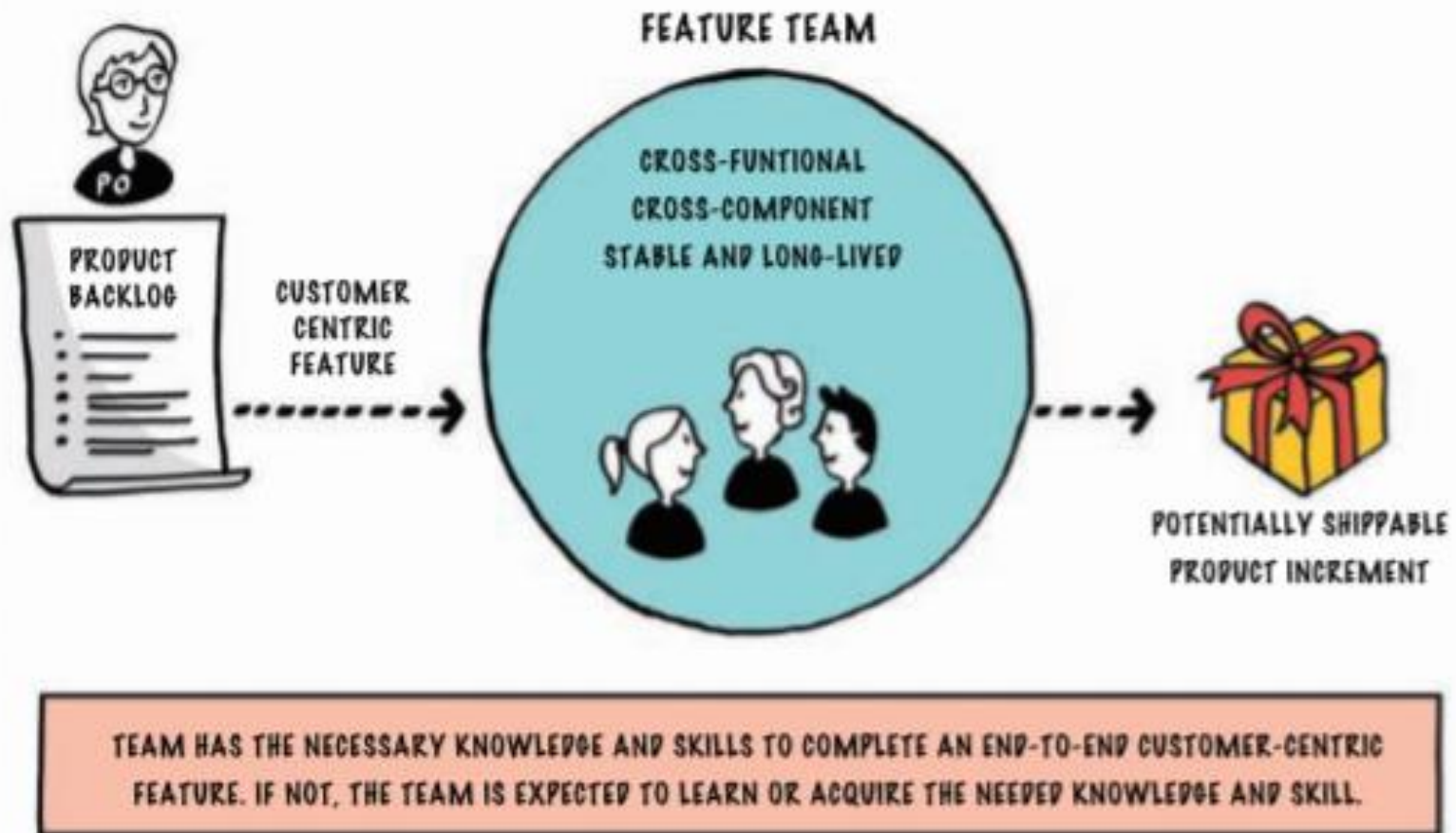


Sourced from: <https://less.works/resources/graphics/index.html>

<http://less.works> 



# LeSS Guides: *Organize by Customer Value*



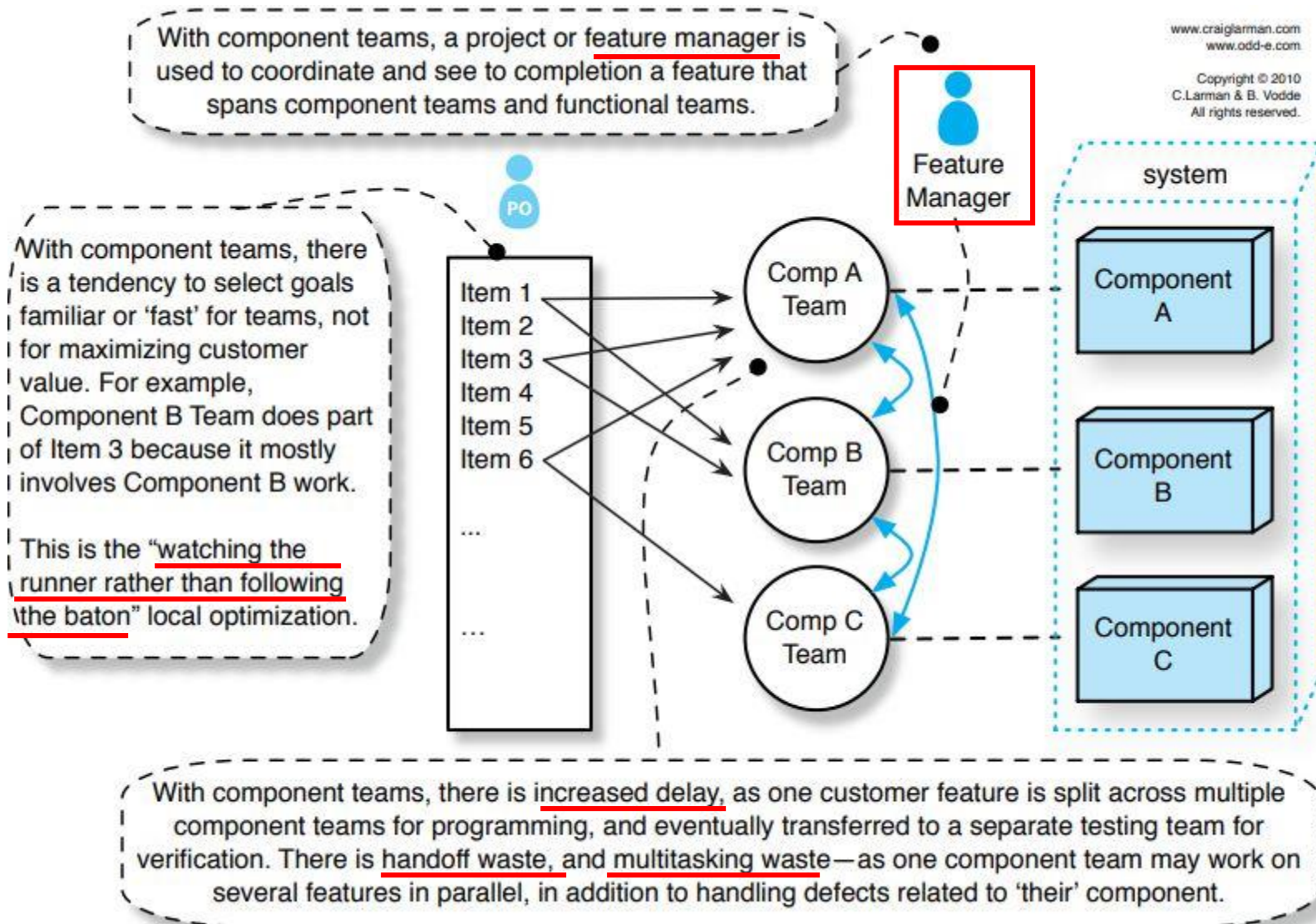
Sourced from: <https://less.works/resources/graphics/index.html>



# LeSS Guides: *Organize by Customer Value*

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

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



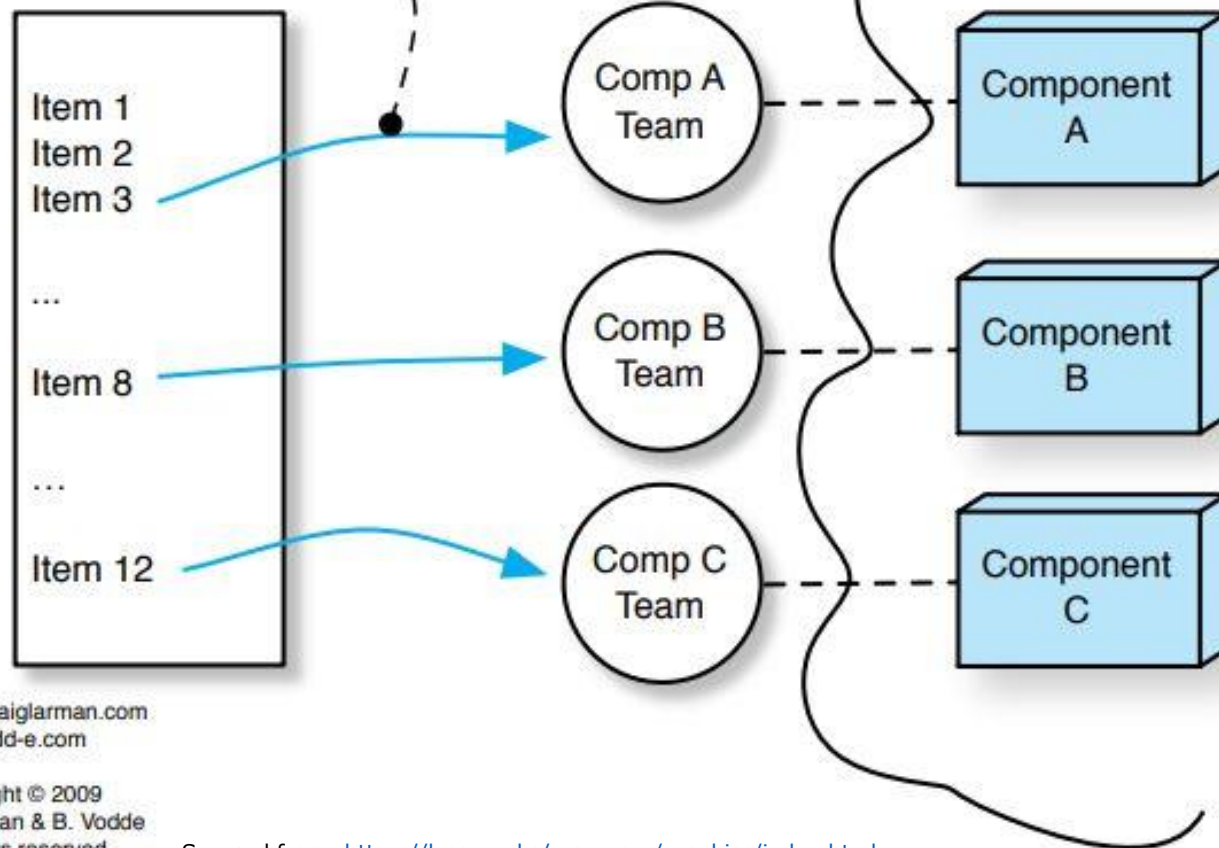
Sourced from: <https://less.works/resources/graphics/index.html>



# LeSS Guides: *Organize by Customer Value*

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.


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



# LeSS Guides: *Organize by Customer Value*

*Synonyms for “Component Team”*

Local Optimization

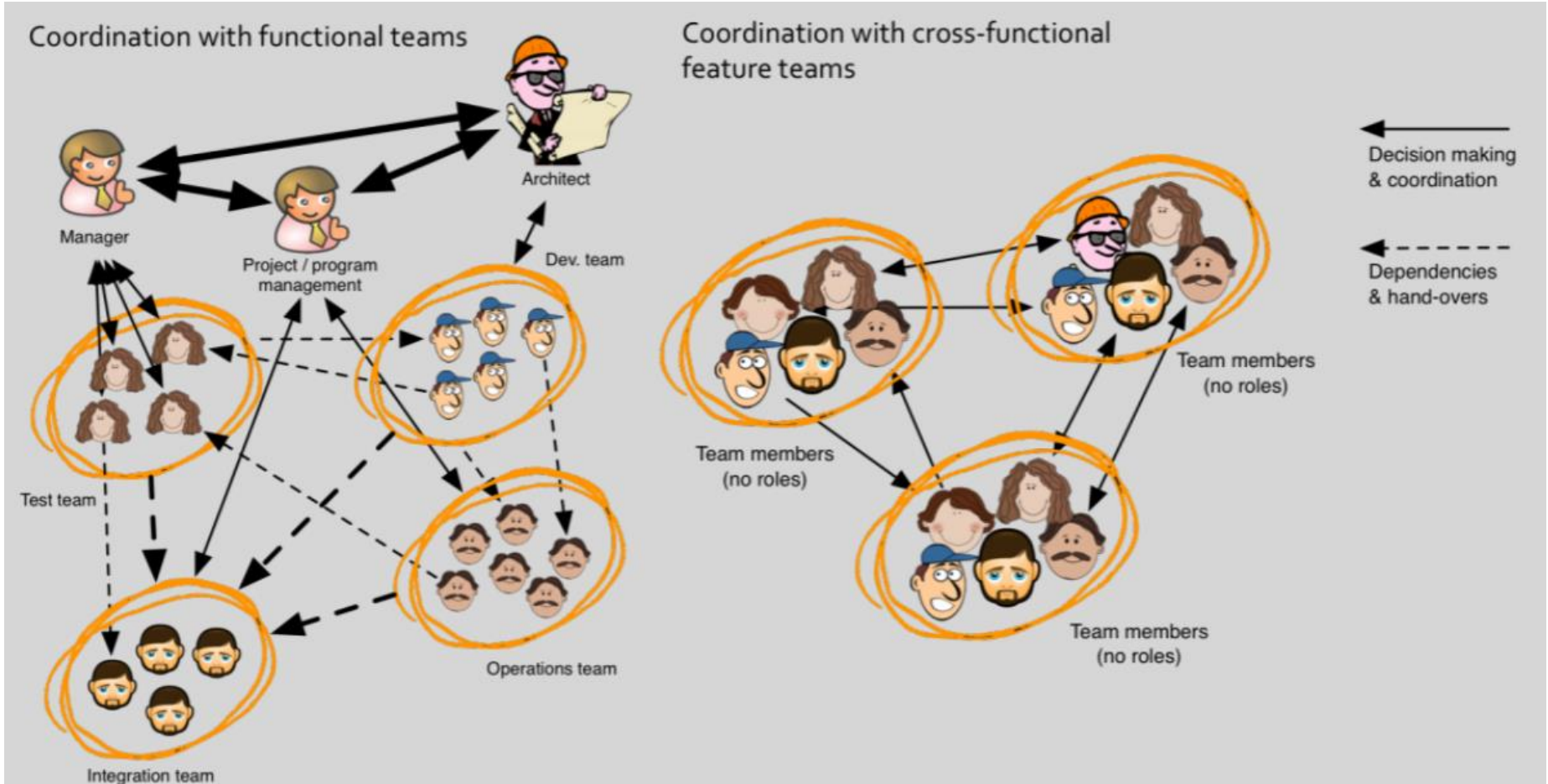


platform team  
core or shared services team  
micro-service team  
application team  
subsystem team  
library team  
service team  
API team  
front-end (or back-end) team  
DB team  
module team  
framework team  
DDD bounded-context team

Sourced from creative commons repository of Certified LeSS Trainers: <https://less.works/courses/become-less-trainer>



# LeSS Guides: *Organize by Customer Value*



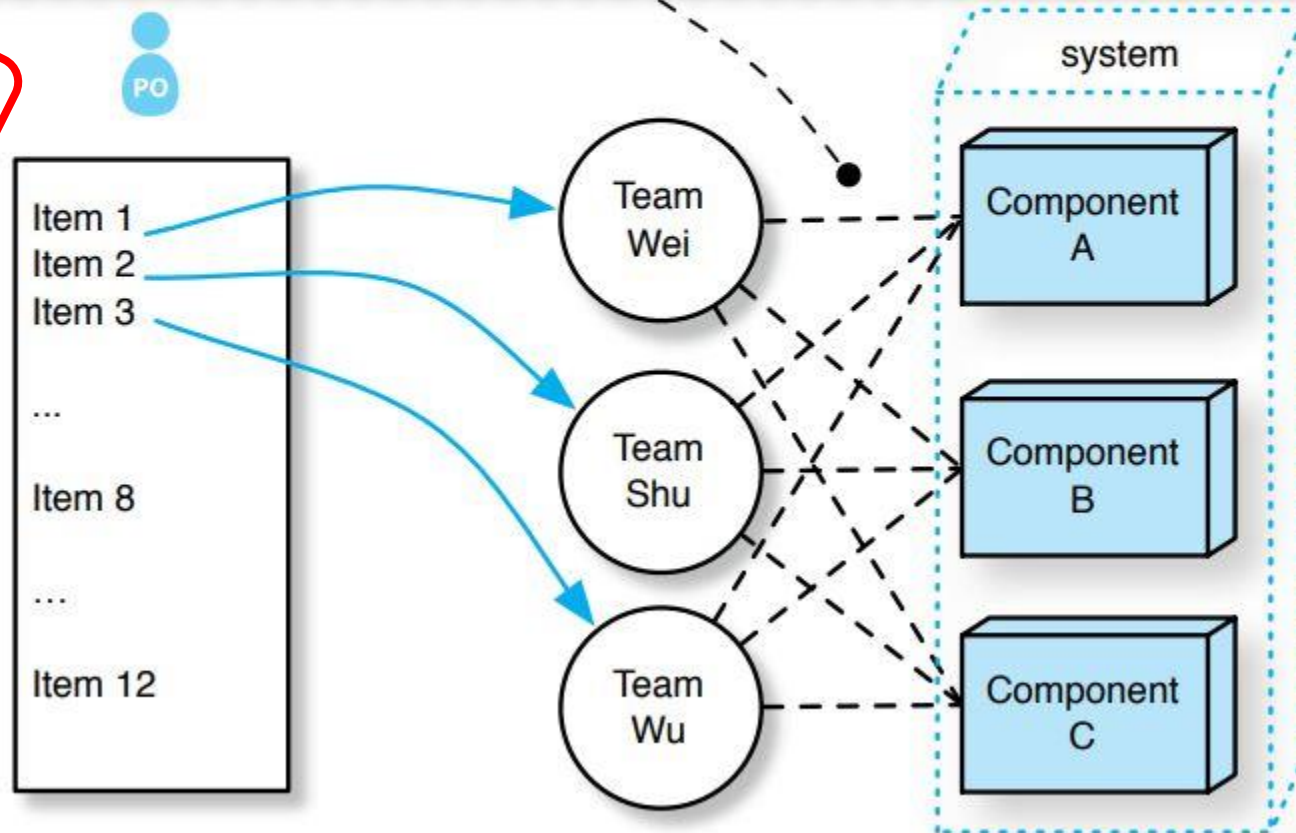
Sourced from creative commons repository of Certified LeSS Trainers: <https://less.works/courses/become-less-trainer>



# LeSS Guides: *Organize by Customer Value*

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**



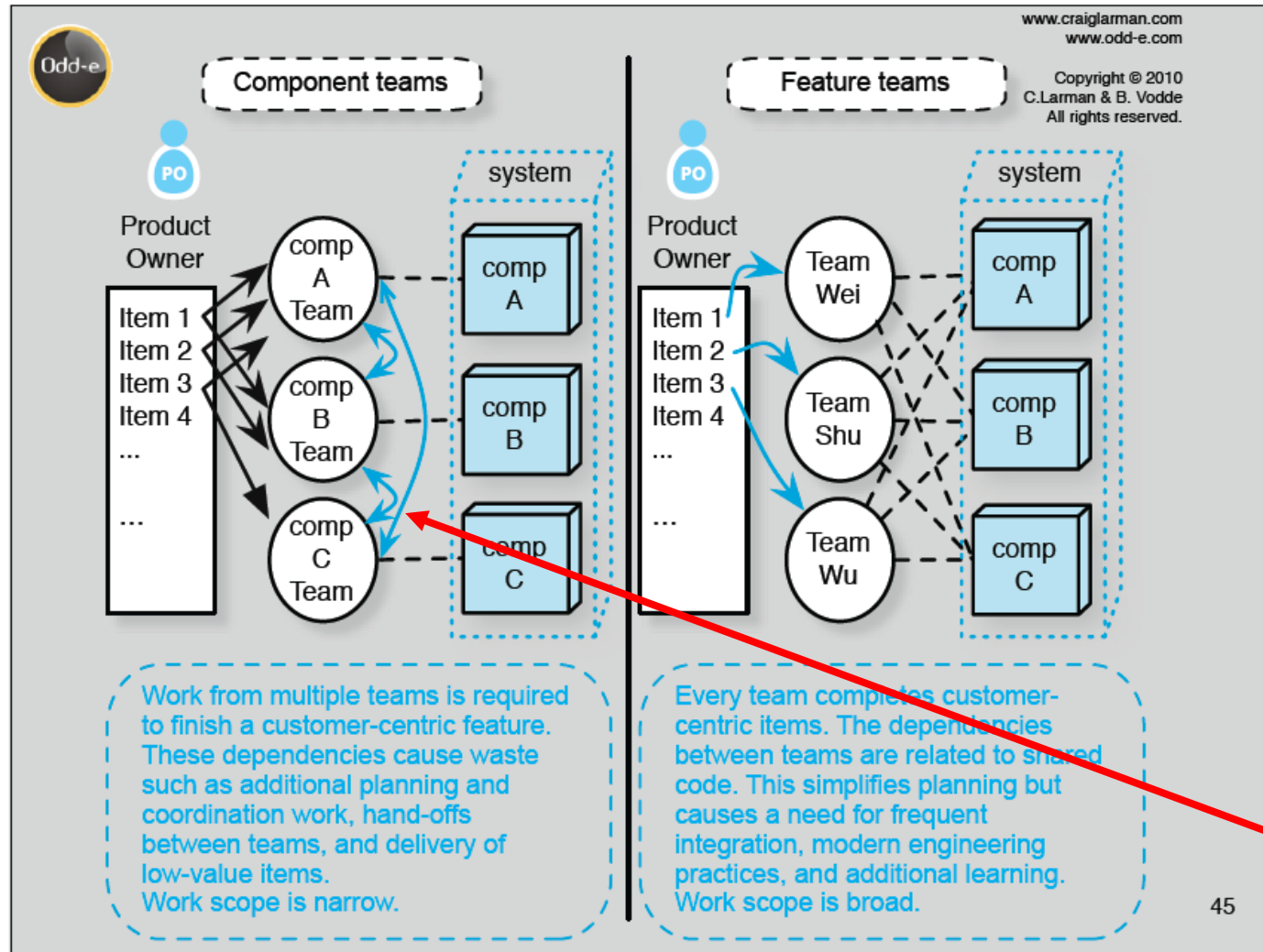
www.craiglarman.com  
www.odd-e.com

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

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



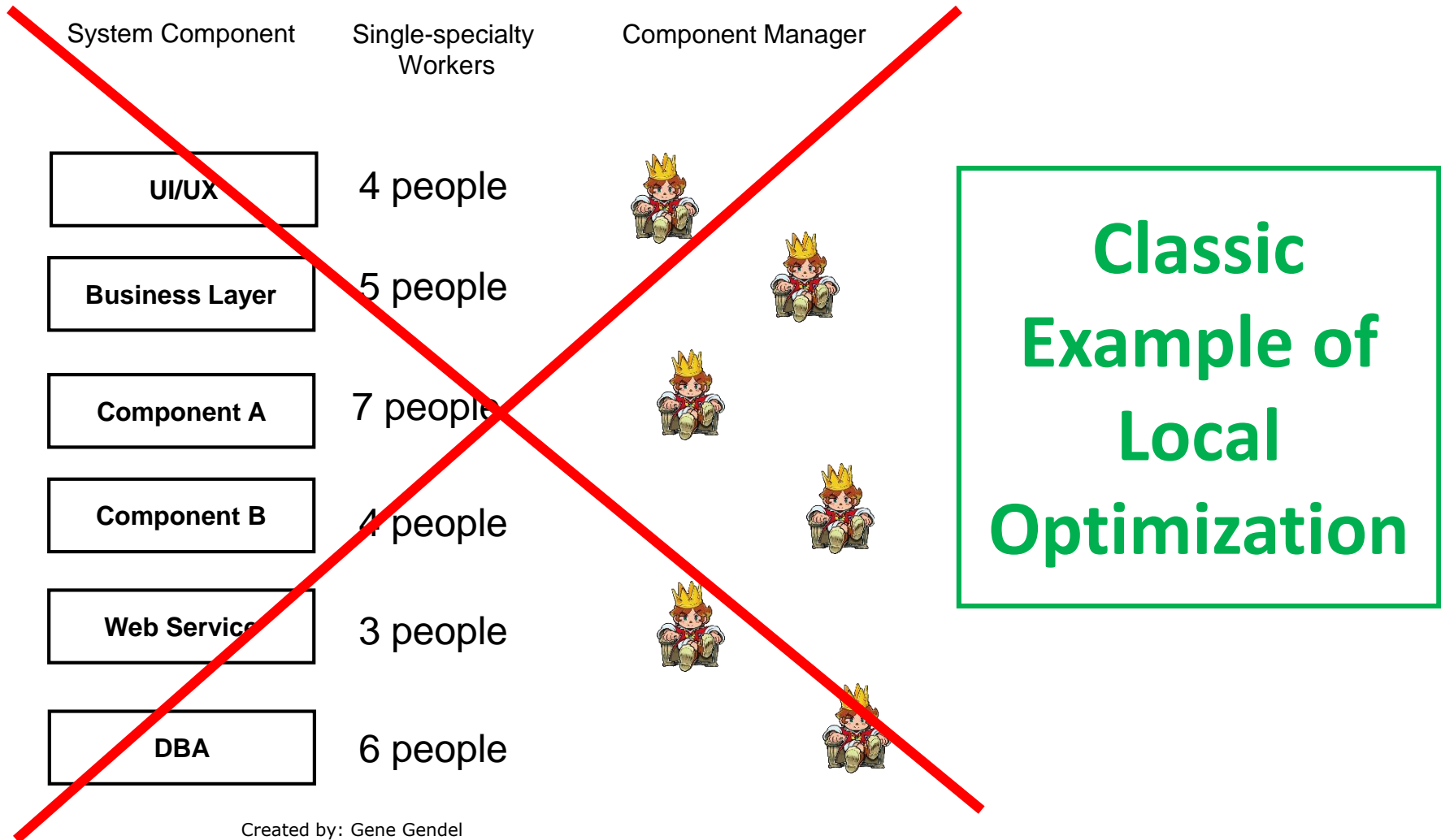
# LeSS Guides: *Organize by Customer Value*





# LeSS Guides: *Organize by Customer Value*

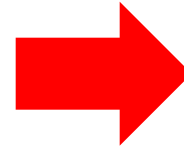
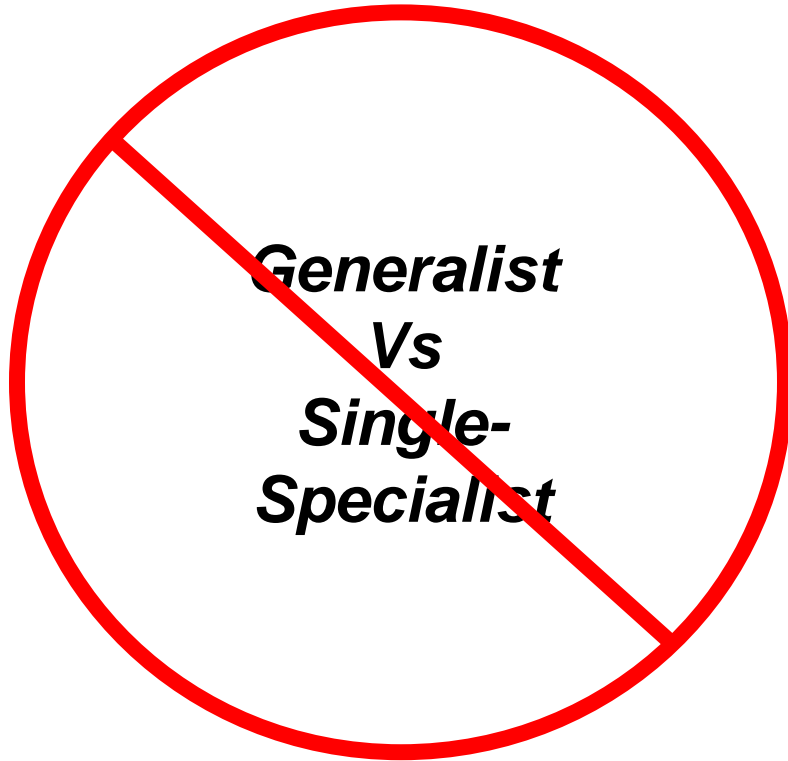
## *"Back story" of Component Teams*





# LeSS Guides: *Organize by Customer Value*

*Avoiding False Dichotomies*



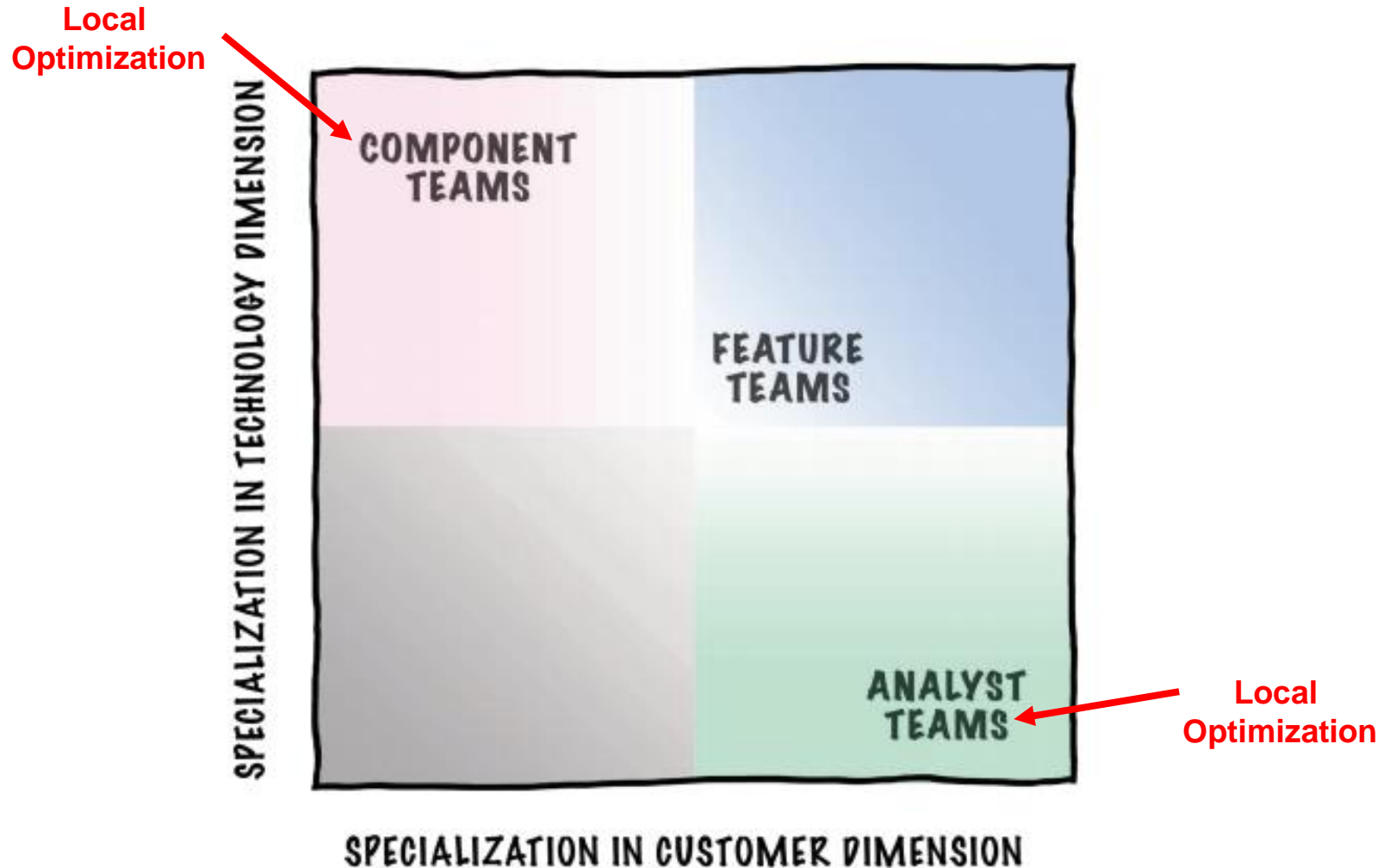
***Primary Skill***  
***Secondary Skill***  
***Tertiary Skill***  
***.....***

Created by: Gene Gendel



# LeSS Guides: *Organize by Customer Value*

*Prefer Specialization in Customer Domain/Dimension*



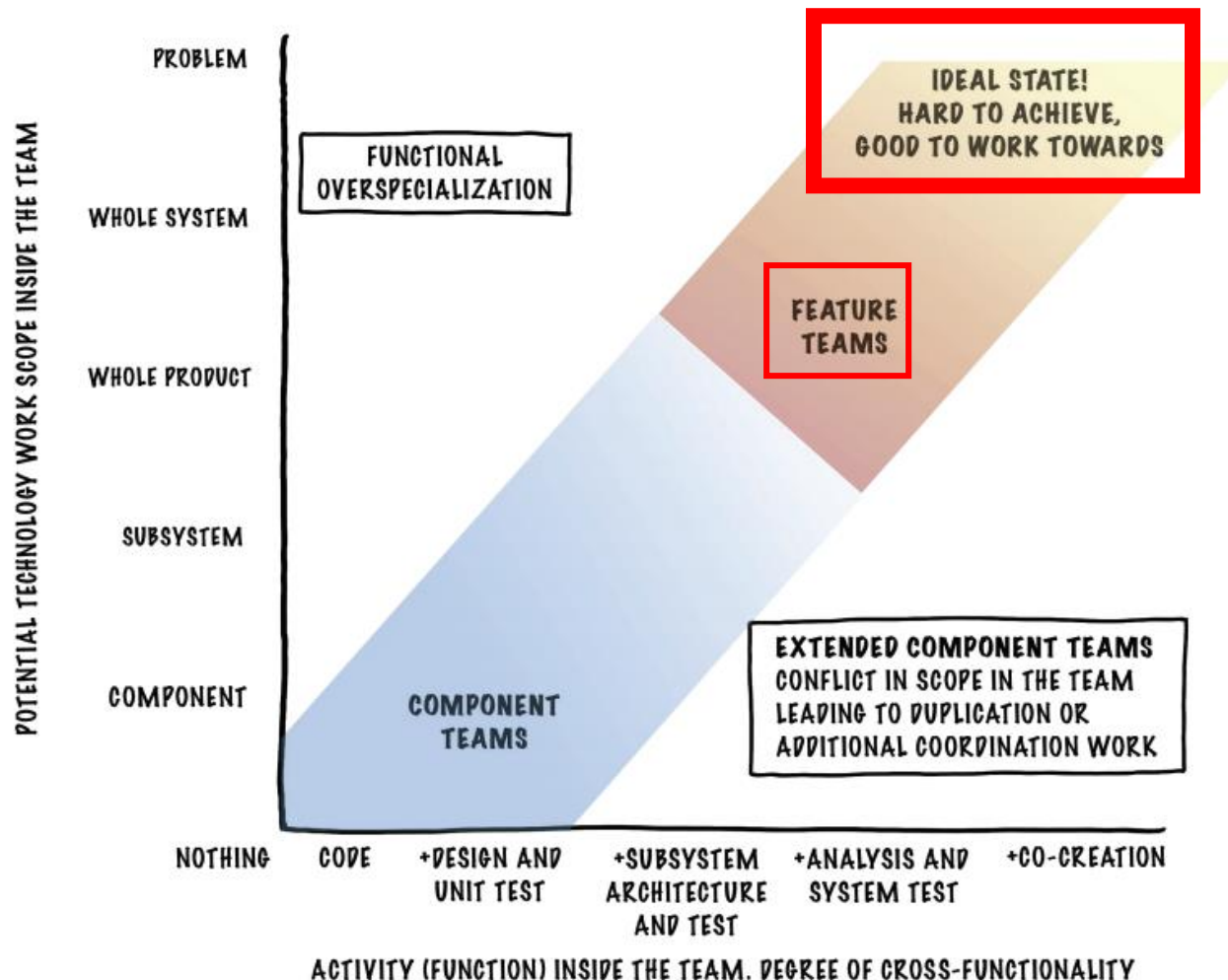
Sourced from: <https://less.works/resources/graphics/index.html>

https://less.works 



# LeSS Guides: *Organize by Customer Value*

## Feature-Team Adoption Maps



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


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

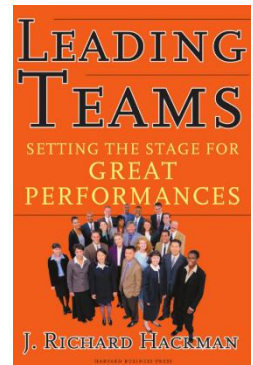


# LeSS Guides: *Organize by Customer Value*

## Feature-Team Adoption Maps (Cont.)

### 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



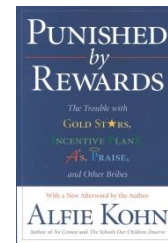
<https://less.works/resources/graphics/index.html>

Figure 1. Types of Teams.



# LeSS Guides: *Organize by Customer Value*

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"





# LeSS Guides: *Organize by Customer Value*

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



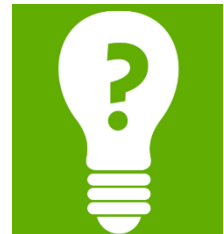
# Class Activity

## Class:

- Whole Class. Give a few examples of team design that is *not* optimized for delivering maximum customer value.
- Explain your POV from as stand-point of *local optimization* and *sub-optimal organizational design*.
- Capture in Class Log

**Instructor:** Catalyze. Reflect.

**Duration:** 10 min

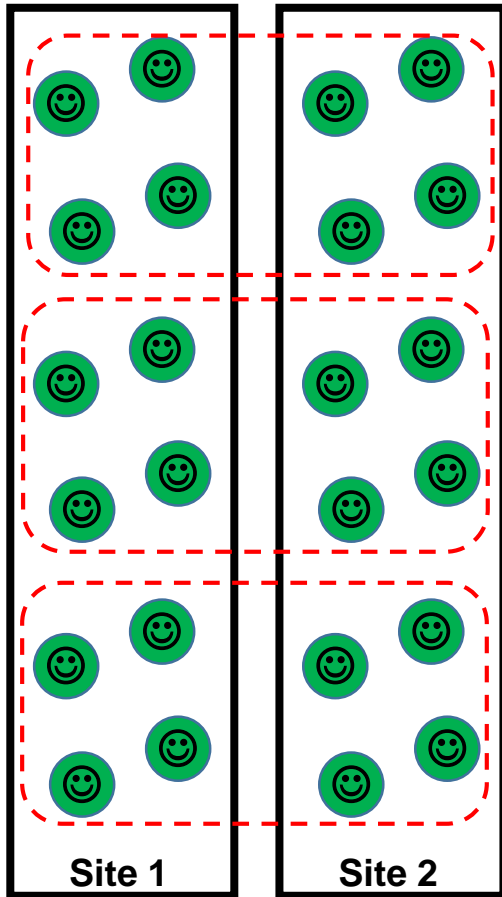




# LeSS Guides: *Organize by Customer Value*

## Team Location in LeSS

Distribution inside team

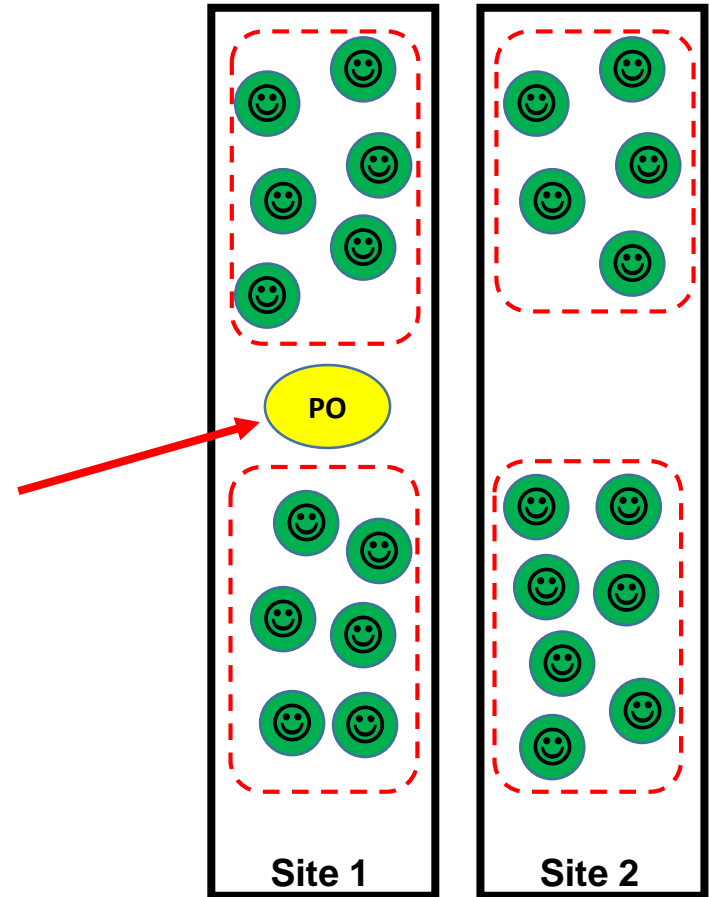


*"All good Java developers are in Boston"... "The best SQL experts are in Bangalore"*

*Really? 😊*

# AVOID

Team colocation within site



# TRY

Created by: Gene Gendel



# LeSS Guides: *Organize by Customer Value*

## *LeSS Organizational Structure*



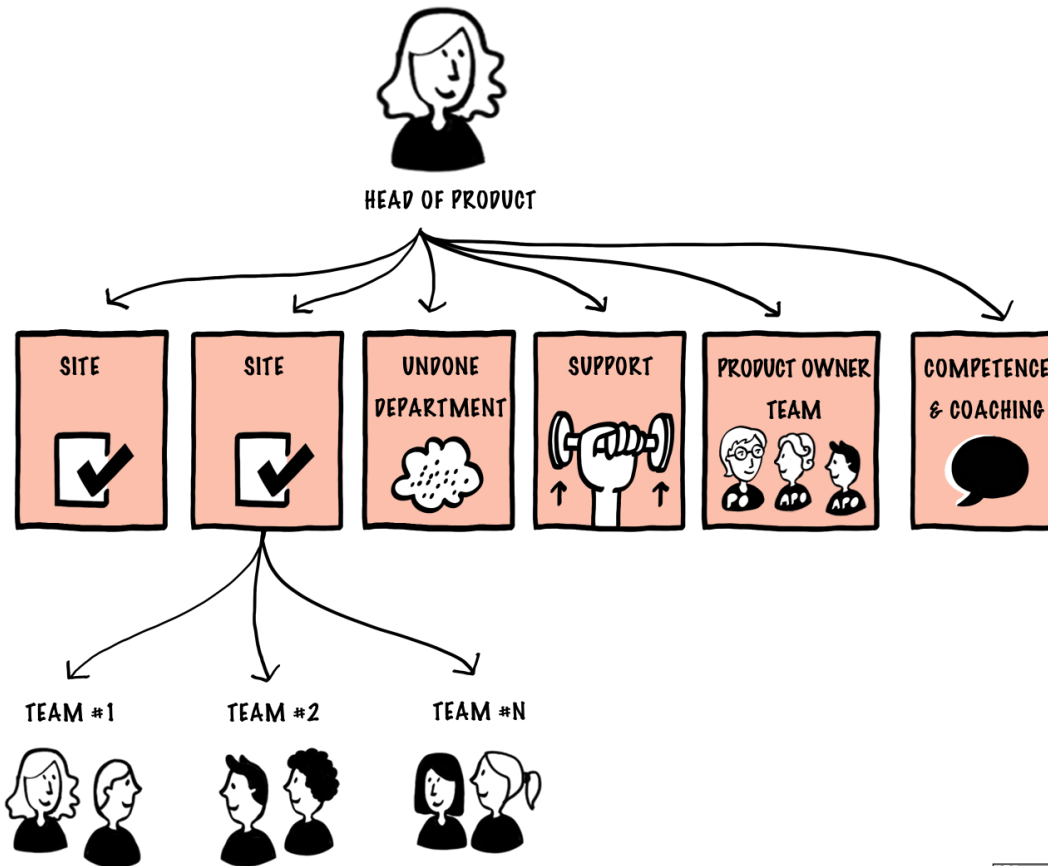
**Reminder: LeSS is Organizational Design Framework**

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



# LeSS Guides: *Organize by Customer Value*

## LeSS (Huge) Organizational Structure



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

**Reminder: LeSS is Organizational Design Framework**

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



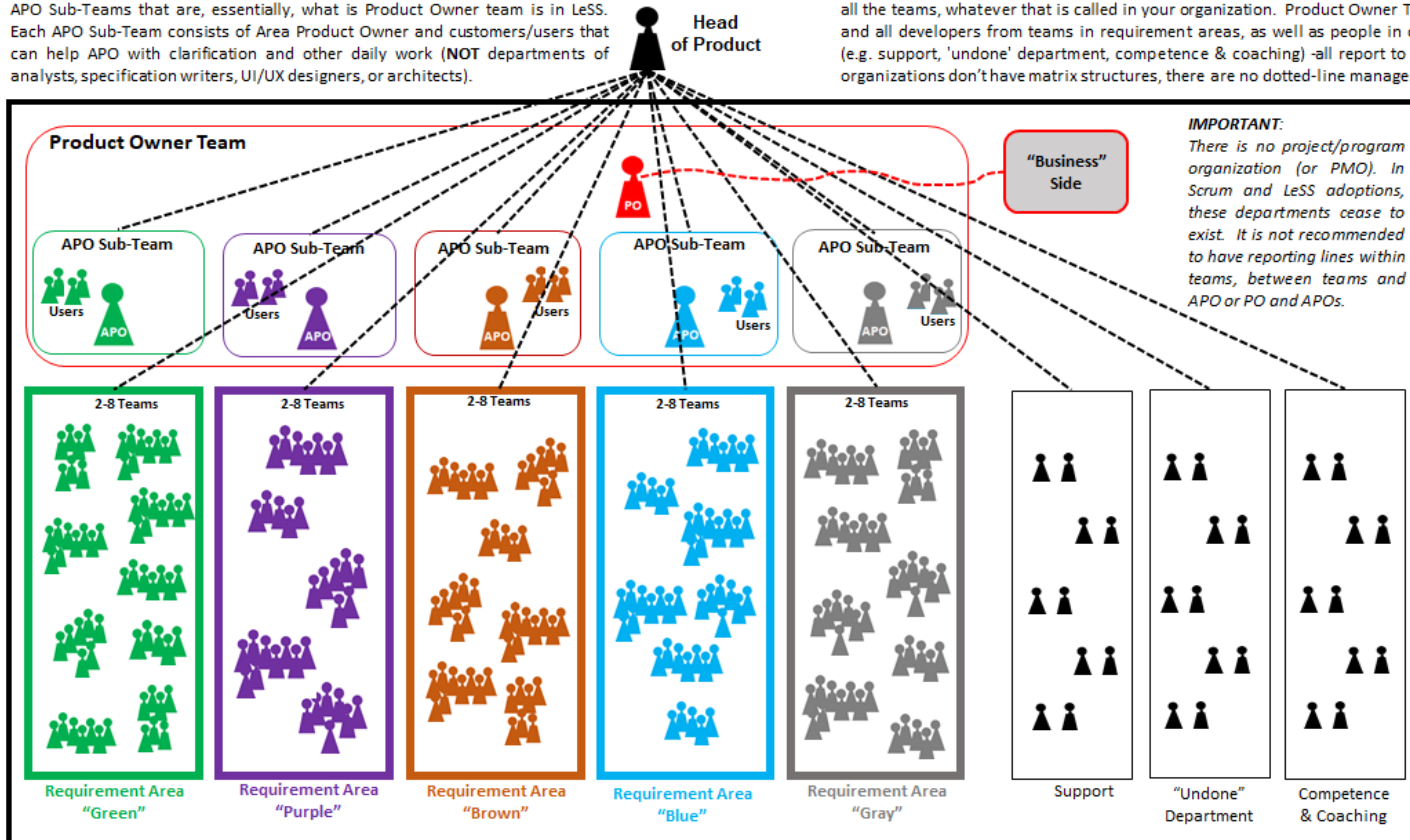
# LeSS Guides: Organize by Customer Value

## LeSS (Huge) Organizational Structure

### 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.



By Gene Gendel. 2020

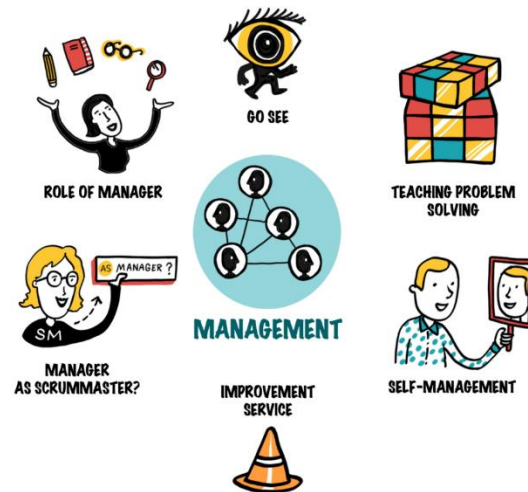
Sourced from: <https://less.works/resources/graphics/index.html>







# LeSS Guides (Management)



<http://less.works> 

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



# LeSS Guides: *Management*



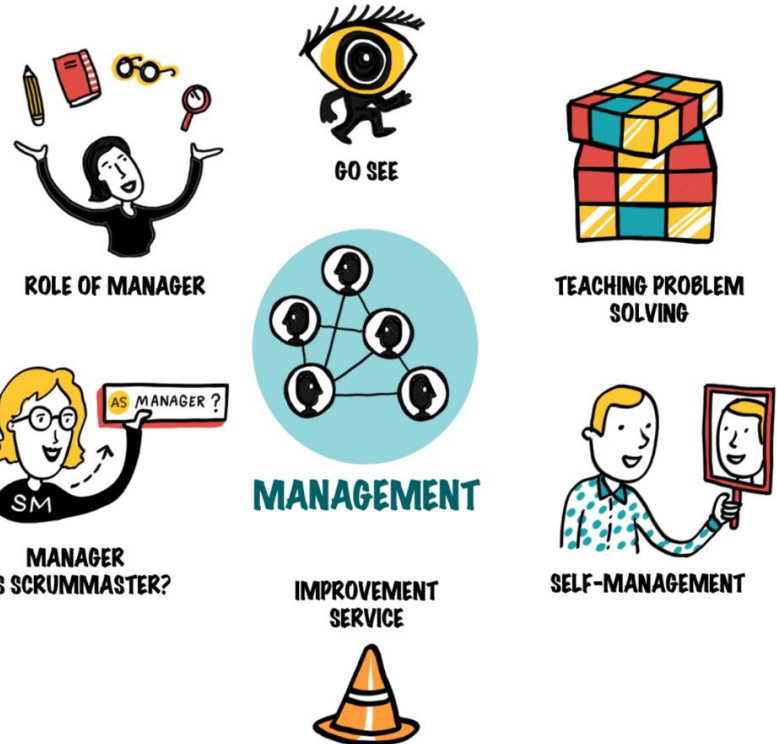
## Konosuke Matsushita (1)

"We will win and you will lose. You cannot do anything about it because your failure is an internal disease. Your companies are based on Taylor's principles. Worse, your heads are Taylorized, too. You firmly believe that sound management means executives on one side and workers on the other, on one side men who think and on the other side men who can only work. For you, management is the art of smoothly transferring the executives' ideas to the workers' hands."



Panasonic Founder

9



<http://less.works>

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

Sourced from: <https://www.keystepstosuccess.com/agile-anti-patterns-with-irony-and-satire/>



# LeSS Guides: *Management*



" More software projects have gone awry from management's taking action based on incorrect system models than for all other causes combined. "

**Weinberg-Brooks Law**



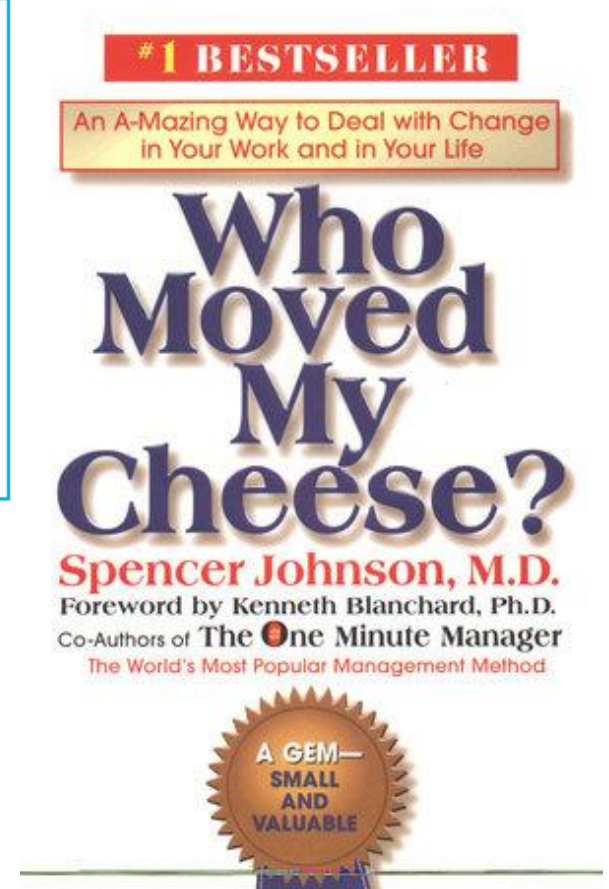
# LeSS Guides: *Management*

## ***Managers Are Optional***

In LeSS, managers are optional. but if managers do exist, their role is likely to change. Their focus shifts from managing the day-to-day product work to improving the value-delivering capability of the product development system.

Managers' role is to improve the product development system by practicing Go See, encouraging Stop & Fix, and "experiments over conformance."

Sourced from: <https://less.works/resources/learning-resources/books>





# Class Activity

**Class:** Whole class, identify all possible activities that a single Scrum team needs to accomplish. Discuss, what activities stay with a team, what goes to Product Owner and what to Scrum Master?  
*What remains unassigned?*

**Instructor:** Catalyze. Reflect.

**Duration:** 10 min





# Class Activity

All possible activities that a single Scrum team needs to accomplish

Team	Product Owner	Scrum Master	other





# LeSS Guides: *Management*

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.

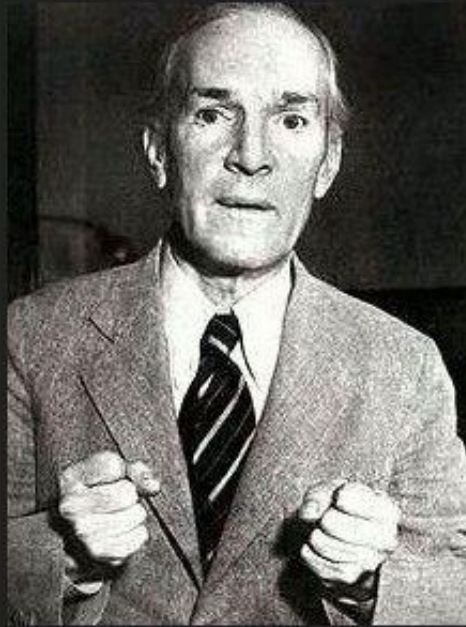
Sourced from: <https://less.works/resources/learning-resources/books>



**Job & Salary Safety**  
but not  
**Role Safety**



# LeSS Guides: *Management*



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

(Upton Sinclair)

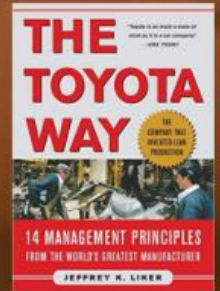
izquotes.com

Sourced from: [https://www.reddit.com/r/anticapitalism/comments/eivy6c/it\\_is\\_difficult\\_to\\_get\\_a\\_man\\_to\\_understand/](https://www.reddit.com/r/anticapitalism/comments/eivy6c/it_is_difficult_to_get_a_man_to_understand/)



# LeSS Guides: *Management*

„ When I first began learning about the Toyota Production System, I was enamoured by the power of [one-piece flow, kanban, and other lean tools]. But along the way, experienced leaders within Toyota kept telling me that these tools and techniques were not the key. Rather the power behind TPS is a company's **management commitment to continuously invest in its people** and **promote a culture of continuous improvement**. After studying for almost 20 years.....this is finally sinking in.“



Jeffrey Liker

Sourced from creative commons repository of Certified LeSS Trainers: <https://less.works/courses/become-less-trainer>



# LeSS Guides: *Management*

## Managers Are Optional (Cont.)

**John Kotter** 



Professor

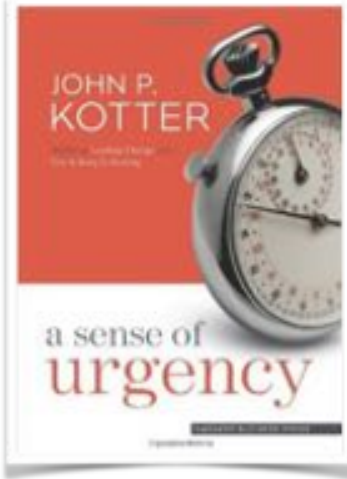
Dr. John Paul Kotter is the Konosuke Matsushita Professor of Leadership, Emeritus, at the Harvard Business School, a New York Times best-selling author, the founder of Kotter International, and a ... [Wikipedia](#)

**Born:** February 25, 1947 (age 69 years), [San Diego, CA](#)

**Nationality:** [American](#)

**Education:** [Harvard University](#), [Harvard Business School](#), [Massachusetts Institute of Technology](#)

Dr. Kotter...



a sense of **urgency** or **existential crisis** needs to be felt by the senior management, to introduce meaningful change, else it unlikely to succeed

468

### **John Kotter - Resistance to Change**

Sourced from: <https://www.keystepstosuccess.com/agile-anti-patterns-with-irony-and-satire/>




# LeSS Management

## Larman's Laws of Organizational Behavior

(Spanish translation)(Russian translation) (Japanese translation)(Persian/Farsi translation)(Polish translation)(Portuguese translation)

**NOT to be  
confused with  
executive  
management**



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. (in large established orgs) *Culture follows structure*. And in tiny young orgs, structure follows culture.

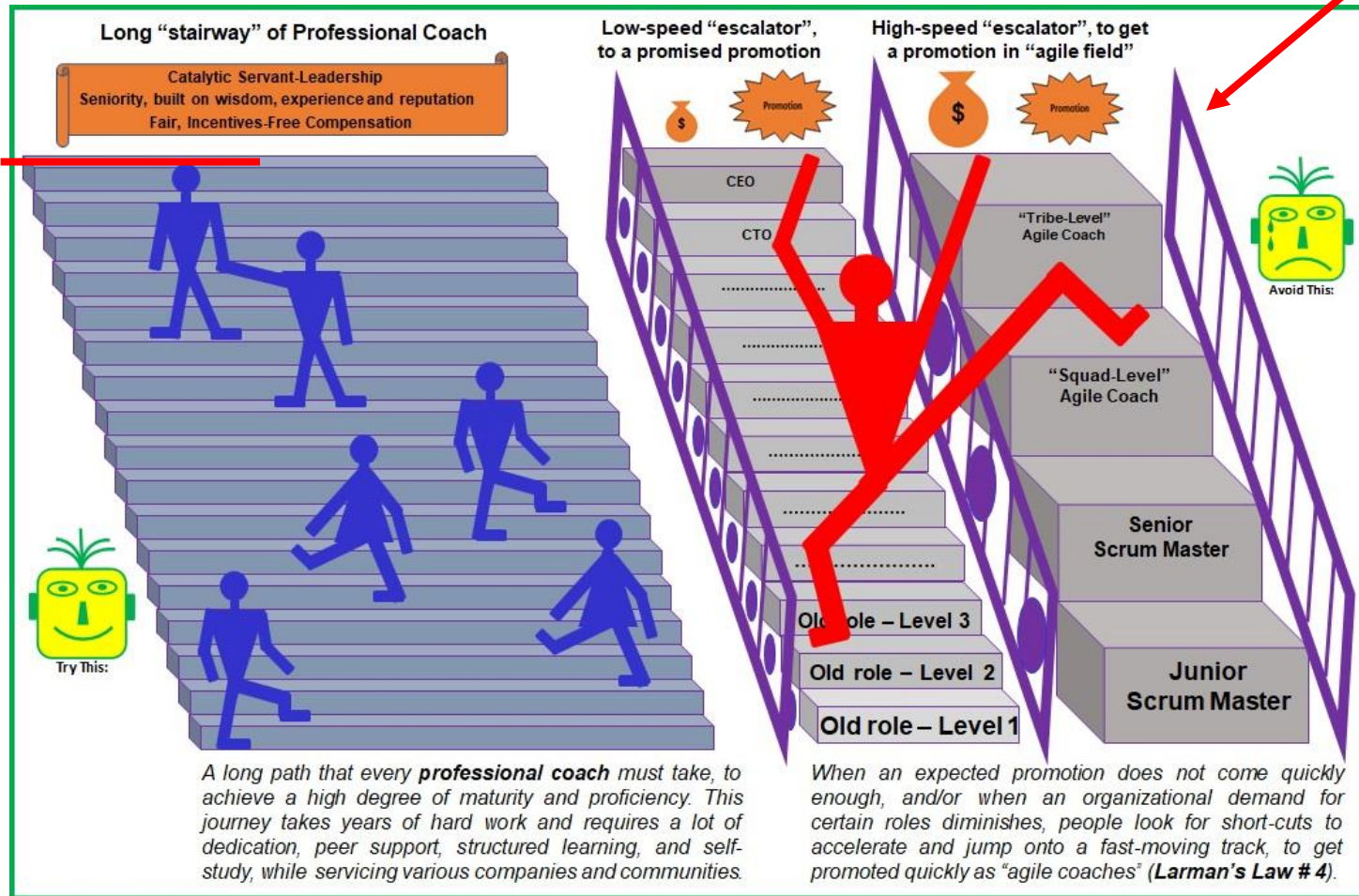
Elaboration:

Sourced from: [https://www.craiglarman.com/wiki/index.php?title=Larman%27s\\_Laws\\_of\\_Organizational\\_Behavior](https://www.craiglarman.com/wiki/index.php?title=Larman%27s_Laws_of_Organizational_Behavior)



# LeSS Guides: *Management*

NOT a  
good  
solution



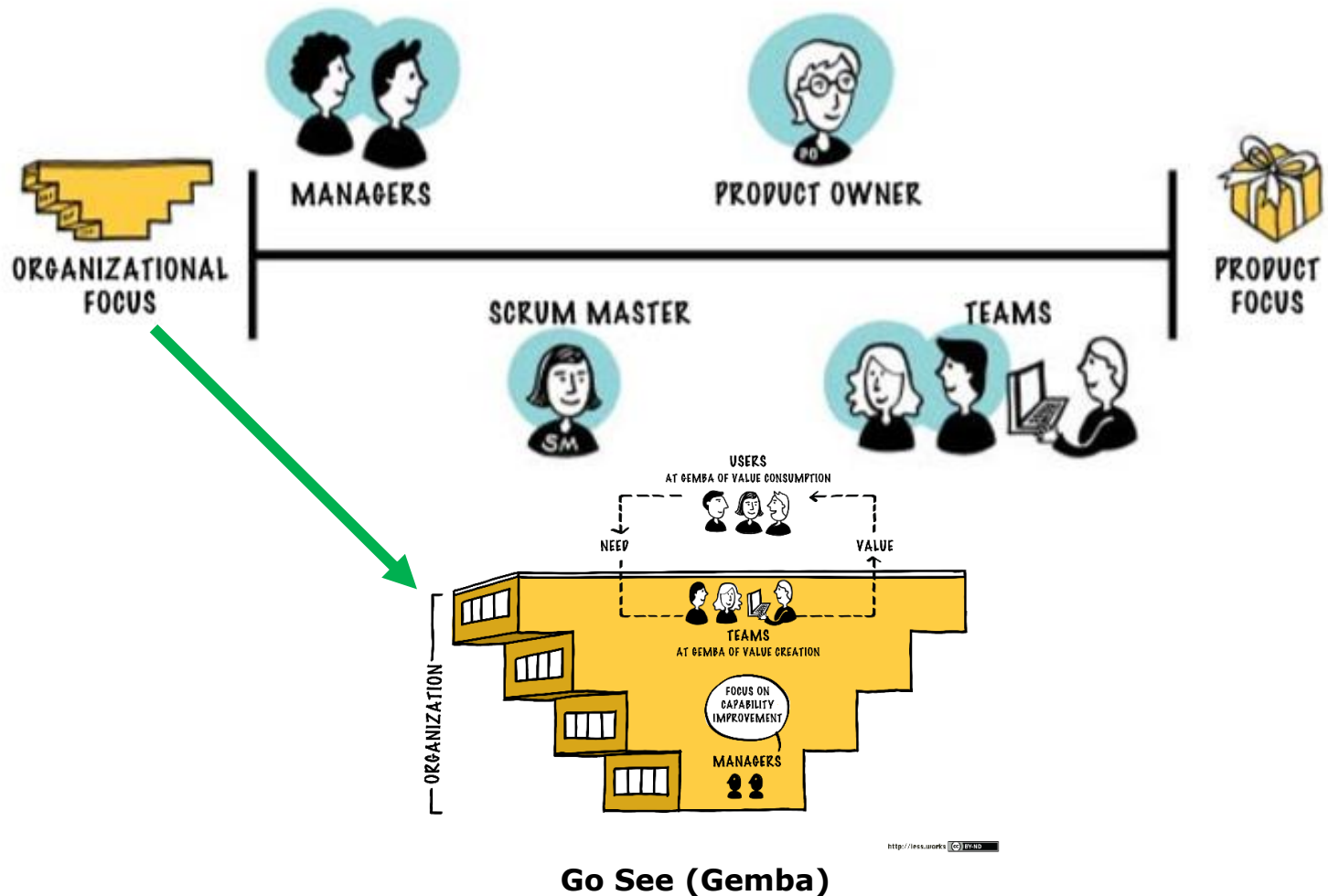
Brought to you by [www.keystepstosuccess.com](http://www.keystepstosuccess.com)

Sourced from: <https://www.keystepstosuccess.com/agile-anti-patterns-with-irony-and-satire/>



# LeSS Guides: *Management*

## The LeSS Organization

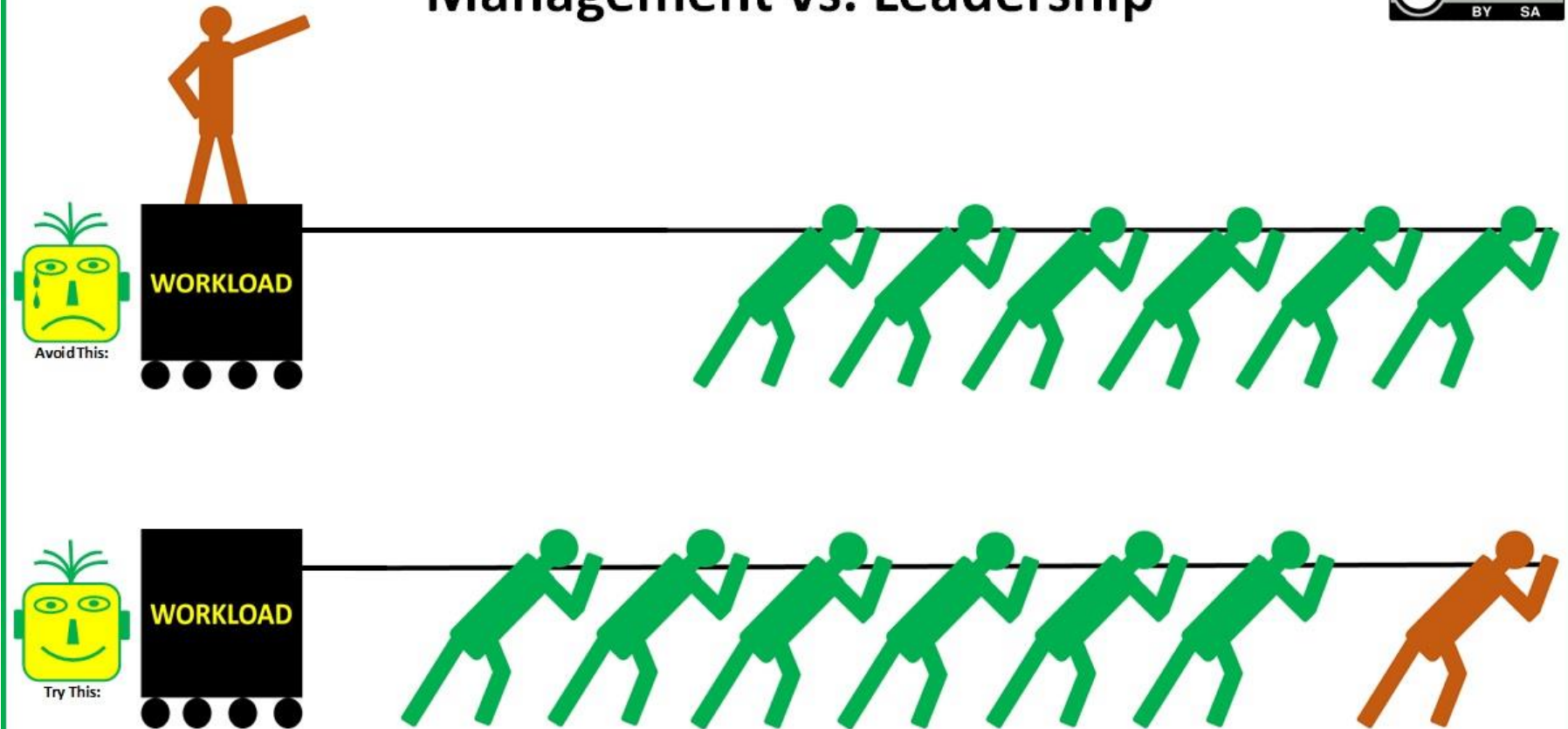


Sourced from: <https://less.works/resources/graphics/index.html>



# LeSS Guides: *Management*

## Management vs. Leadership



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

Sourced from: <https://www.keystepstosuccess.com/agile-anti-patterns-with-irony-and-satire/>



# Class Activity

**Class:** Review instructions of a system modelling exercise on the next slide. Split up into teams.

**Instructor:** Provide instructions

**Duration:** next page





## Exercise

**Duration:** 10 min

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

**Method/Tool:** System Modeling with CLD

**Instructor:** review with Class



# Local Optimization in **Agile Leadership** - Exercise

Title switchers ('agile managers' and Laman's Law # 4-type-of people)

Organizational enablement, to grow internal quality, agile expertise as per highest standards

Building internal Agile "Power Towers" that are away from GEMBA

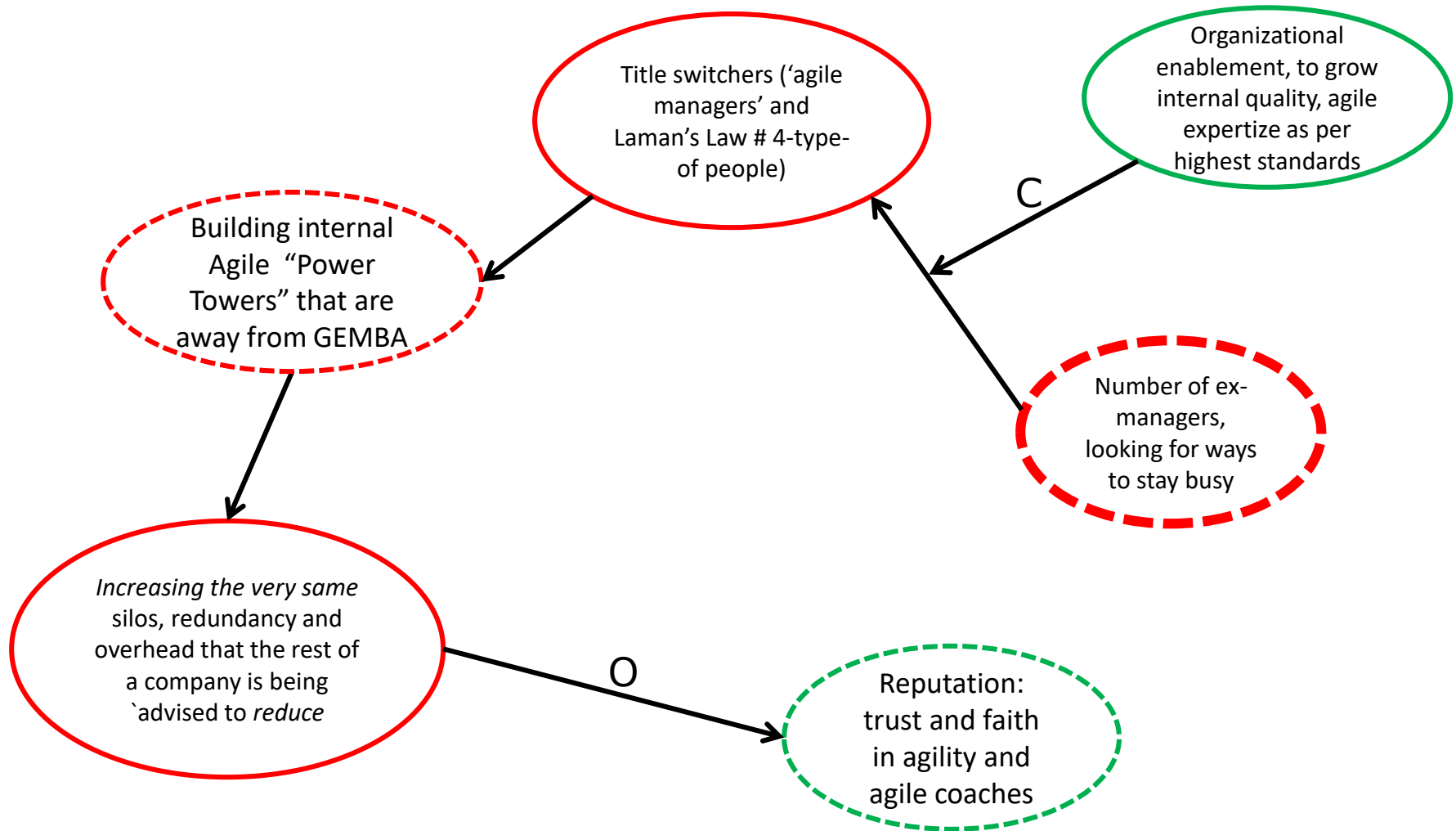
Number of ex-managers, looking for ways to stay busy

*Increasing the very same silos, redundancy and overhead that the rest of a company is being advised to reduce*

Reputation: trust and faith in agility and agile coaches



# Local Optimization in Agile Leadership - Exercise









# LeSS Guides

## (Scrum Master)



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



# LeSS Guides: Scrum Masters



## 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)

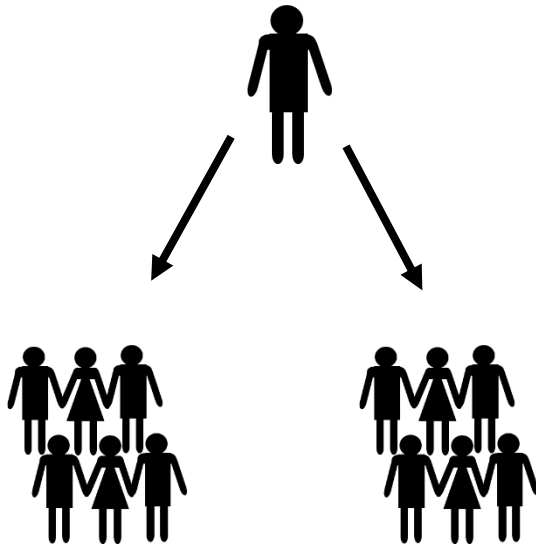
Sourced from: <https://www.keystepstosuccess.com/agile-anti-patterns-with-irony-and-satire/>



# LeSS Guides: *Scrum Masters*

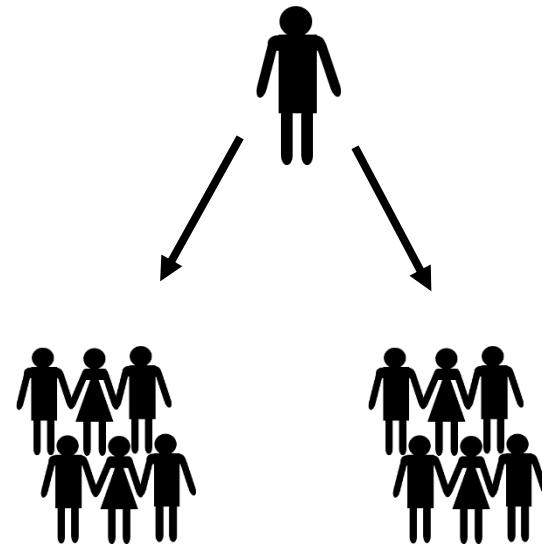
## *Coordination vs. Facilitation*

**Project Manager**



*Coordination*

**Scrum Master**



*Facilitation*



Created by: Gene Gendel



# LeSS Guides: *Scrum Masters*

## 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

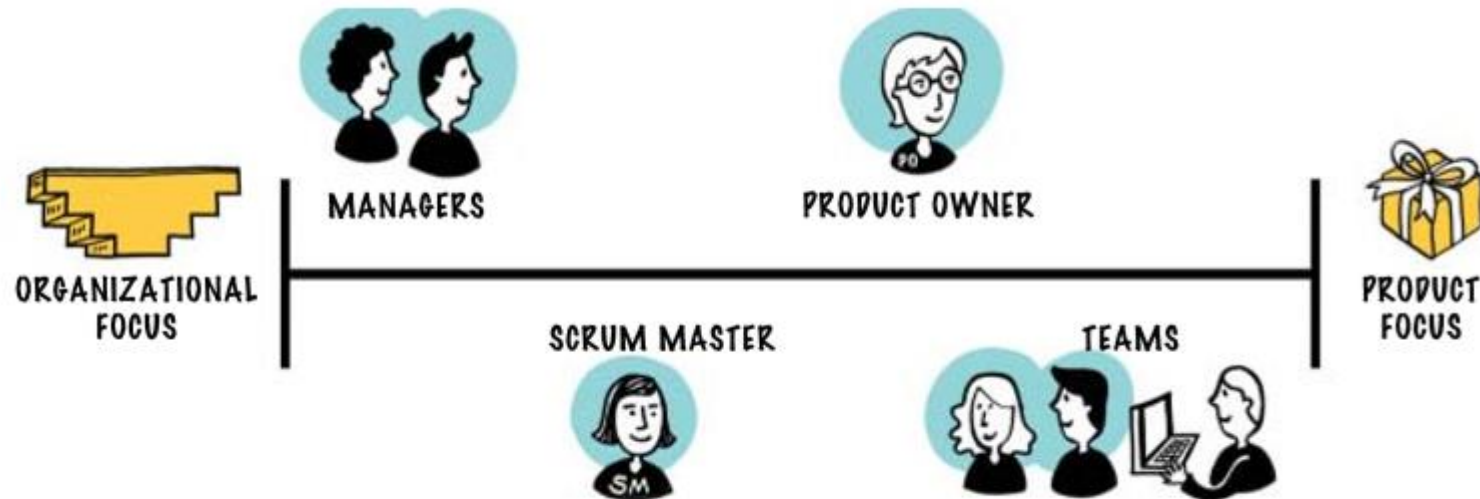
In LeSS, the ScrumMaster role is vital. We’ve seen many organizations try part-time ScrumMasters, which usually leads to no ScrumMasters at all. This then affects the LeSS adoption enormously. In LeSS the ScrumMaster is a dedicated, full-time role in the same way that being a Scrum Team member is a dedicated, 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*



# LeSS Guides: *Scrum Masters*

## *Scrum Master Focus*



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



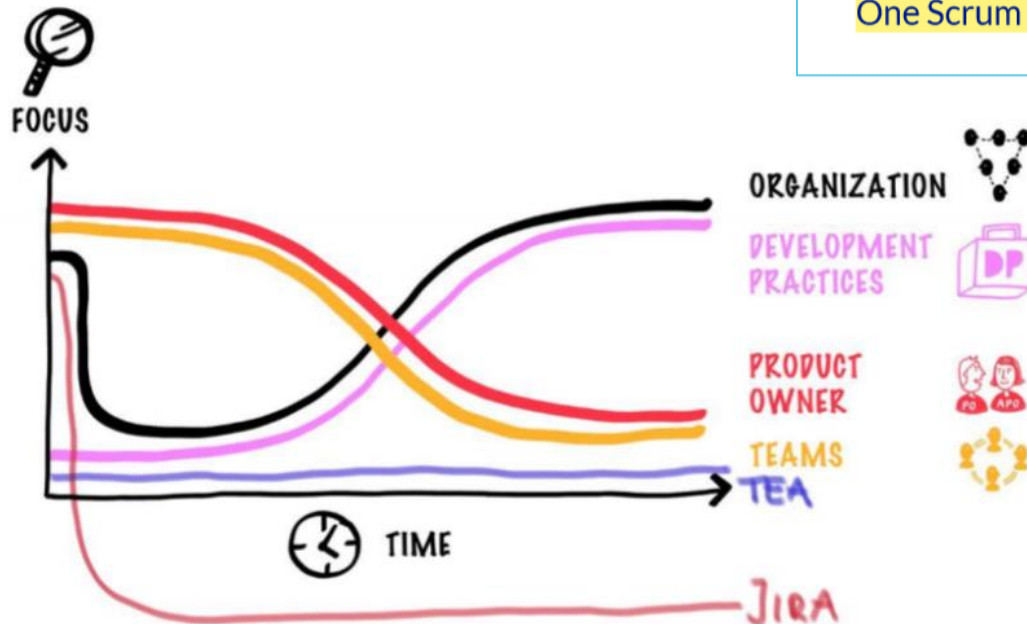
# LeSS Guides: *Scrum Masters*

## Focus Change Over Time

Scrum Masters are responsible for a well-working LeSS adoption. Their focus is on the Teams, Product Owner, organization, and development practices. A Scrum Master doesn't only focus on a team but also on the overall organizational system.

A Scrum Master is a dedicated full-time role.

One Scrum Master can serve 1–3 teams.

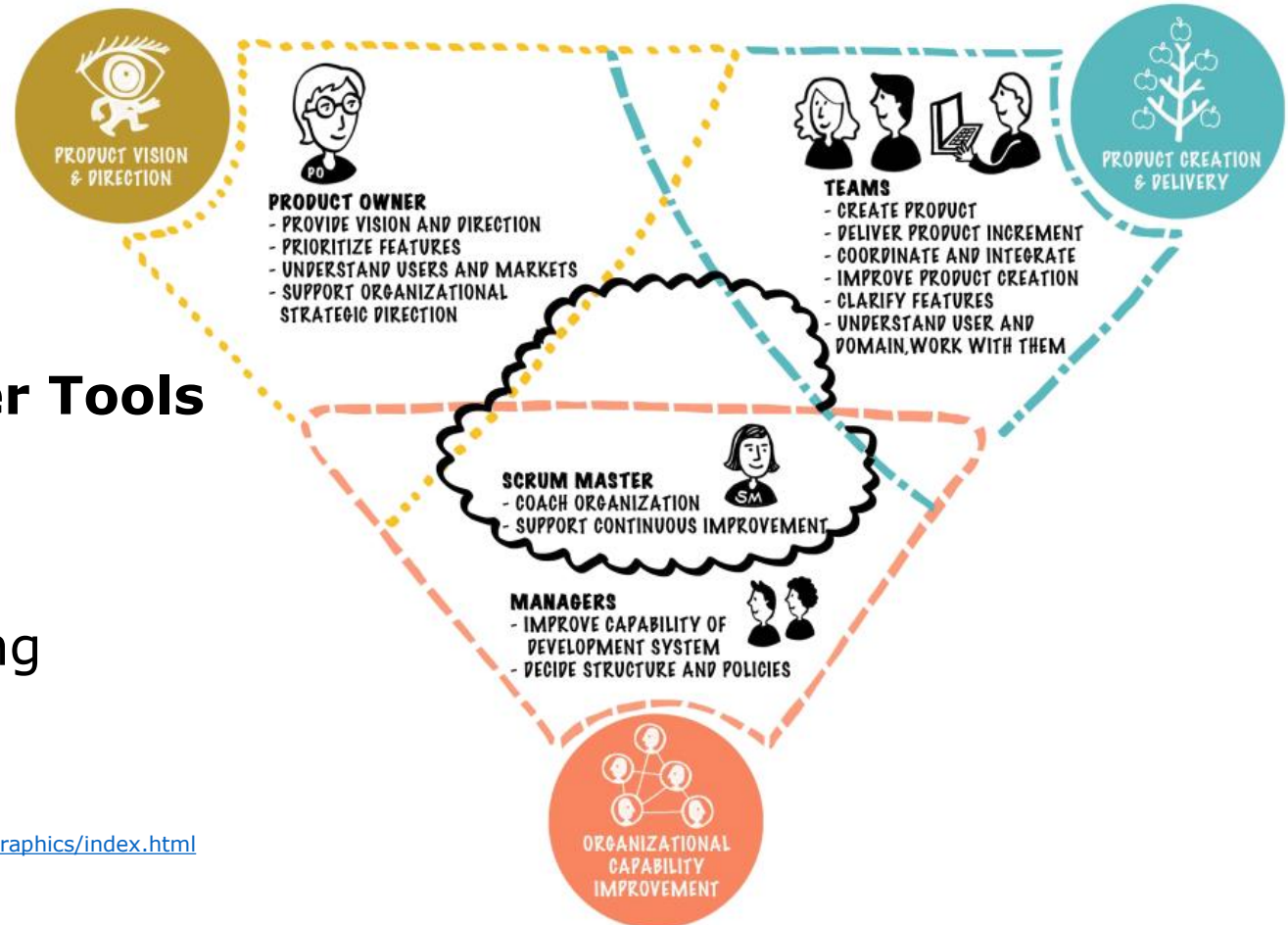


Sourced from: <https://less.works/resources/graphics/index.html>



# LeSS Guides: *Scrum Masters*

## Overlap of Responsibilities



## Five Scrum Master Tools

- Question
- Educate
- Facilitate
- Actively Do Noting
- Interrupt

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



# LeSS Guides: *Scrum Masters*

## *Large-Group Facilitation*



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



# LeSS Guides: *Scrum Masters*

***Promote Learning. Coaching. Mentoring.***



## COMMUNITIES

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



# LeSS Guides: *Scrum Masters*

*Especially, Pay Attention To...*

## Scrum Blind Date [with Mask Show]

**Avoid This:**

**Scrum Team**

**Project Manager**

**Scrum Master**

**Smoke Screen**

**Product Owner**

**Business Analyst**

...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)

Sourced from: <https://www.keystepstosuccess.com/agile-anti-patterns-with-irony-and-satire/>



# LeSS Guides: *Scrum Masters*

## *Beware of Dysfunctions*

- Dysfunctional Product Owner – Teams Relationship
- Dysfunctional Teams – Product Owner relationship
- Us vs. them
- Adopting Scrum without change
- Scrum Master as team assistant or anti-Scrum
- Remote Scrum Master
- Scrum Master as project manager



# LeSS Guides: Scrum Masters

*Especially Pay Attention To...*



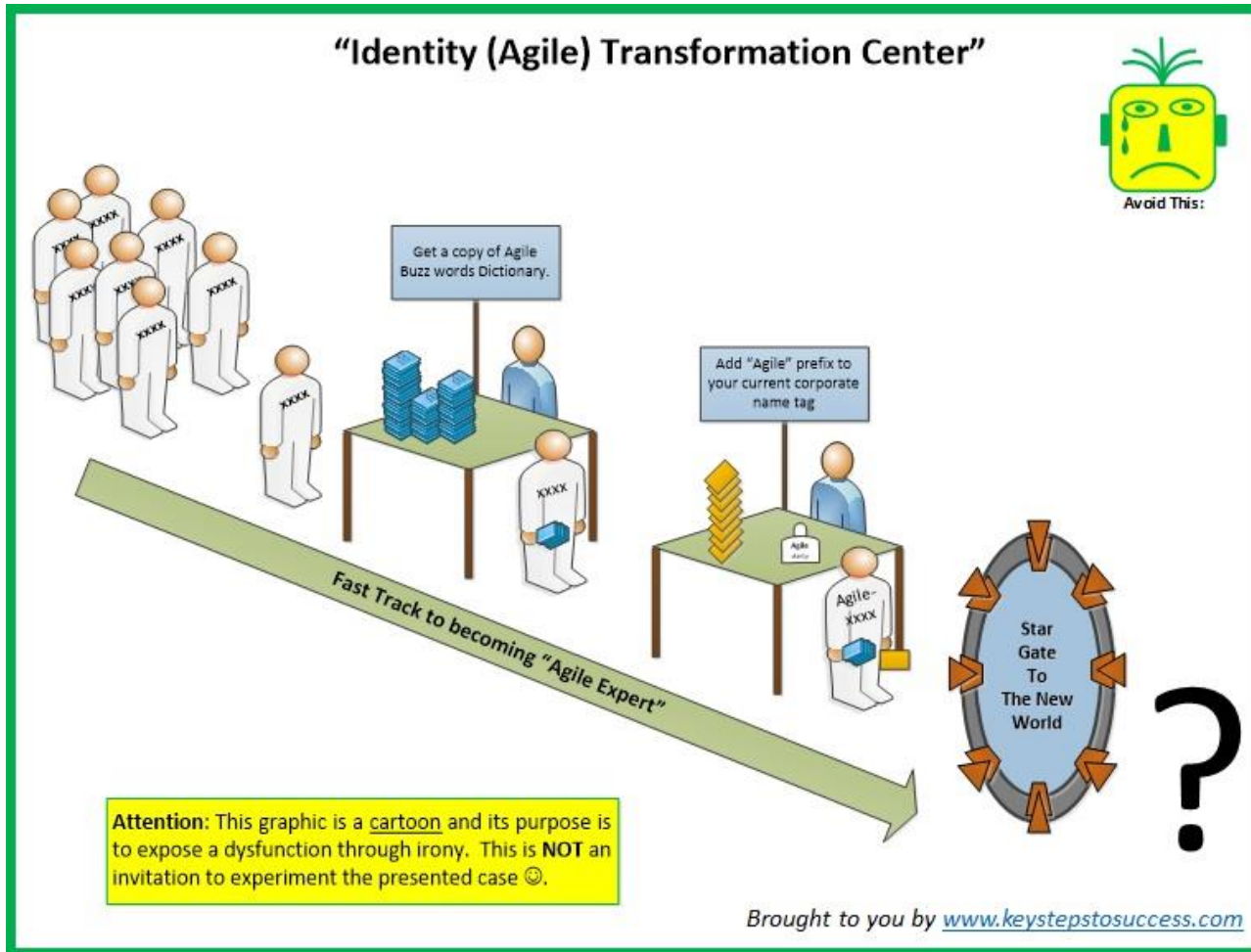
Brought to you by [www.keystepstosuccess.com](http://www.keystepstosuccess.com)

Sourced from: <https://www.keystepstosuccess.com/agile-anti-patterns-with-irony-and-satire/>



# LeSS Guides: *Scrum Masters*

*Especially Pay Attention To...*



Sourced from: <https://www.keystepstosuccess.com/agile-anti-patterns-with-irony-and-satire/>



# LeSS Guides: *Scrum Masters*

## *Big Ideas For Scrum Master...*

### Tips:

LeSS  
Huge

- > Pick one Scrum Master who helps the Product Owner Team and gives them improvement feedback.
- > Have a Scrum Master for two teams, each belonging to different Requirement Areas.
- > Organize the previously mentioned LeSS community events (such as internal gatherings) across Requirement Areas.
- > Organize a multi-area Retrospective and/or Review spanning at least two areas.

Sourced from: <https://less.works/resources/learning-resources/books>



# Class Activity

**Class:** Review instructions of a system modelling exercise on the next slide. Split up into teams.

**Instructor:** Provide instructions

**Duration:** next page





## 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



# Local Optimization in **Scrum Master Role** - Exercise

Chances of  
seeing real,  
long-lasting,  
organizational  
improvements

Allowing to  
develop!!!  
Scrum Master  
**Community**

Company's  
desire to have  
more Scrum  
Masters

Likelihood that  
experienced, self-  
driven Scrum  
Masters will have  
freedom and  
safety

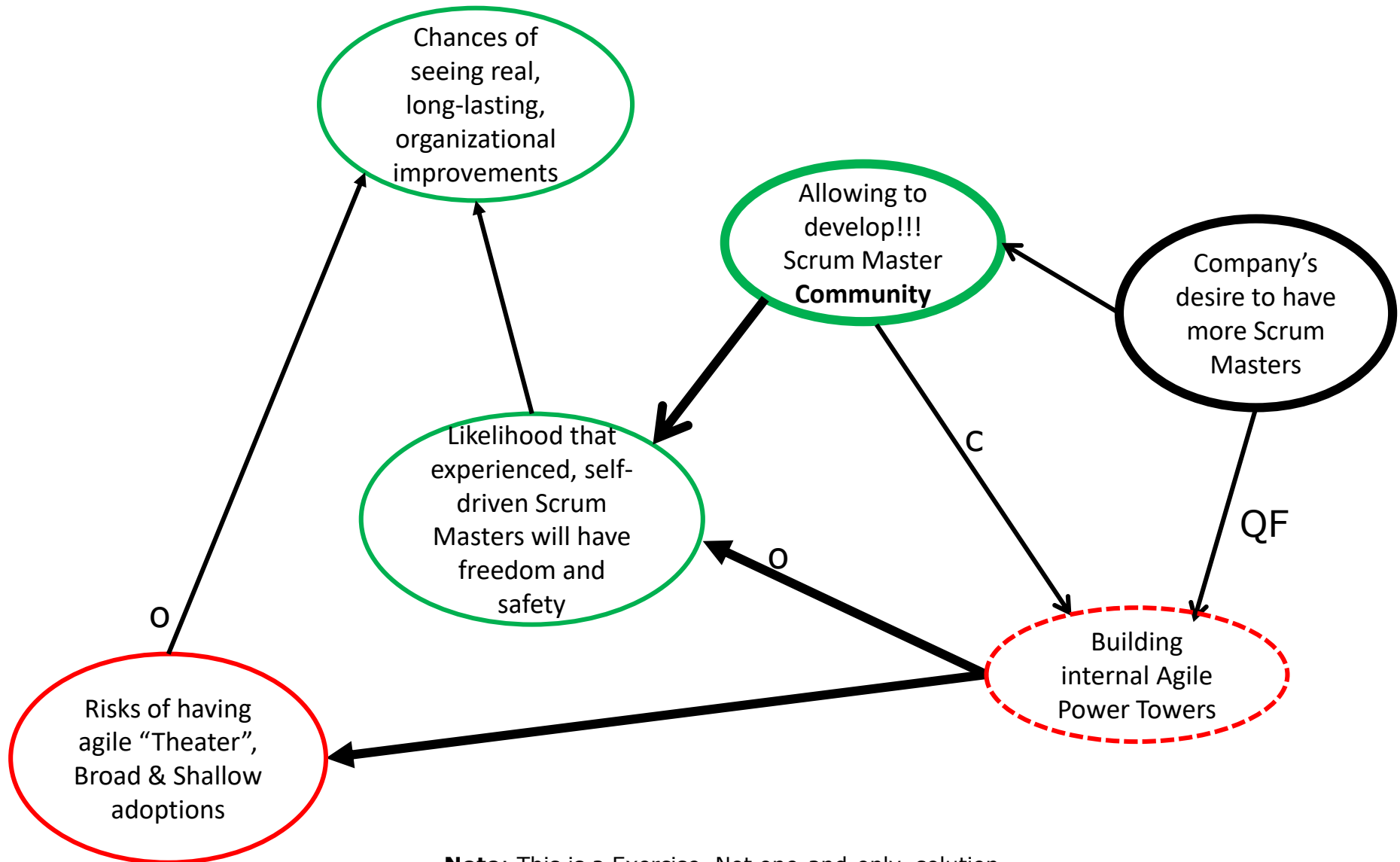
Risks of having  
agile "Theater",  
Broad & Shallow  
adoptions

Building  
internal Agile  
Power Towers

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



# Local Optimization in **Scrum Master Role** - Exercise



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







$$5 \times 5 - 3$$



$$8 \times 3 - 4$$

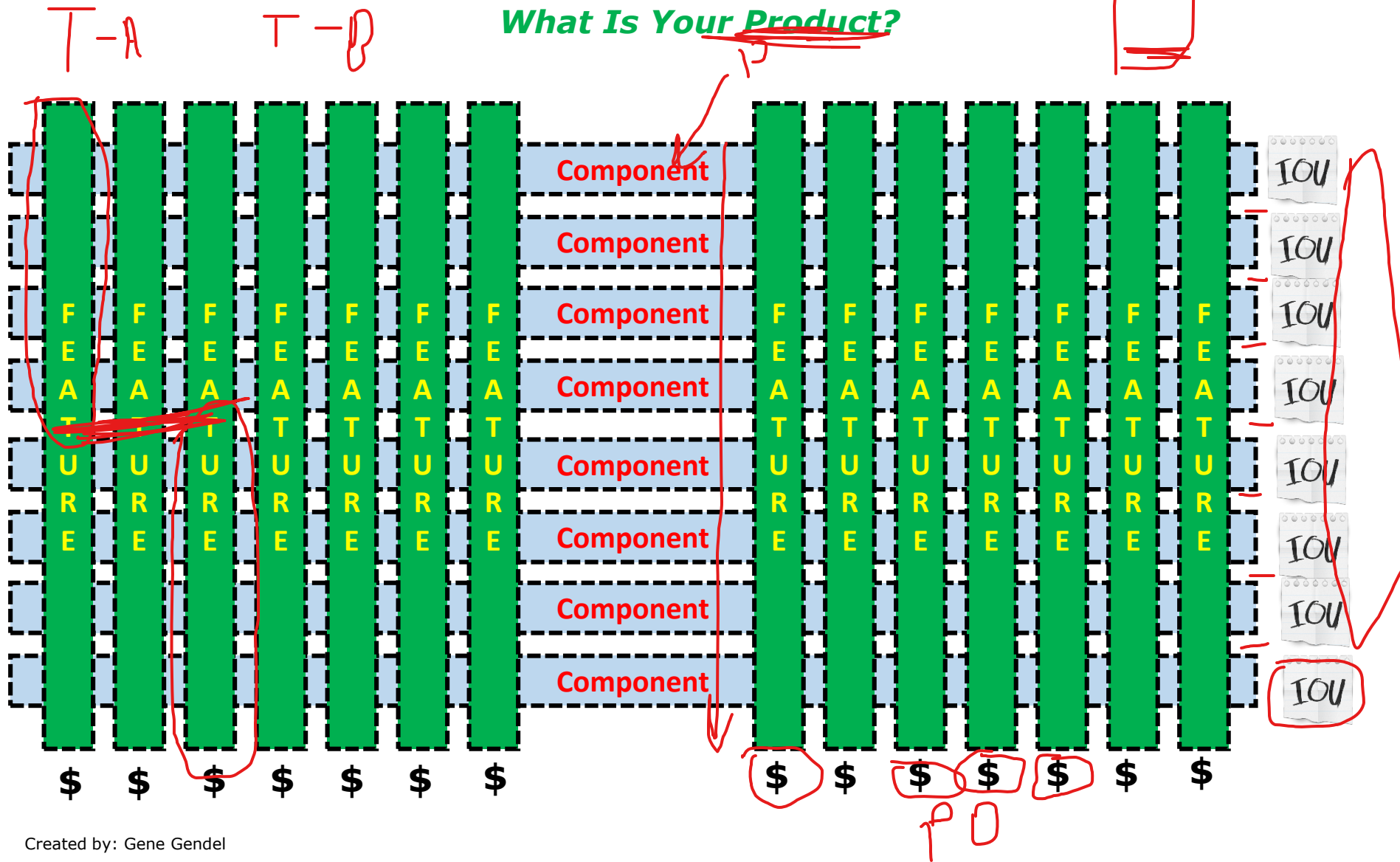


# LeSS Guides

## (Product)



# LeSS Guides: *Product*





# LeSS Guides: *Product*

## *What Is Your Product?*

The definition of product should be as broad and end-user/customer centric as is practical. Over time, the definition of product might increase. Broader definitions are preferred.

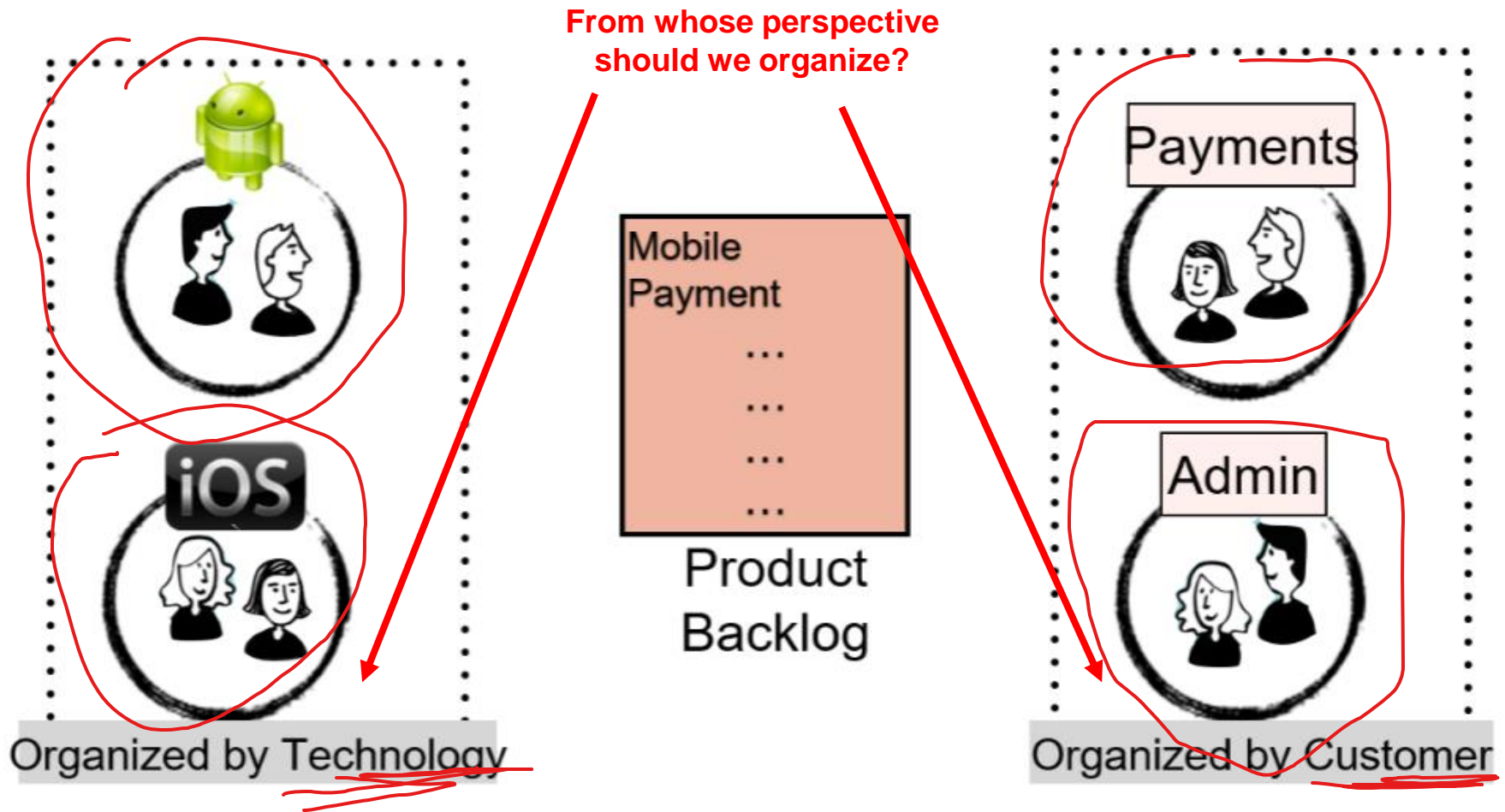
LeSS de-scales organizational complexity by broader product definitions, dissolving unnecessary complex organizational solutions and solving them in simpler ways.

Sourced from: <https://less.works/resources/learning-resources/books>



# LeSS Guides: *Product*

## *What Is Your Product?*



Sourced from creative commons repository of Certified LeSS Trainers: <https://less.works/courses/become-less-trainer>



# LeSS Guides: *Product*

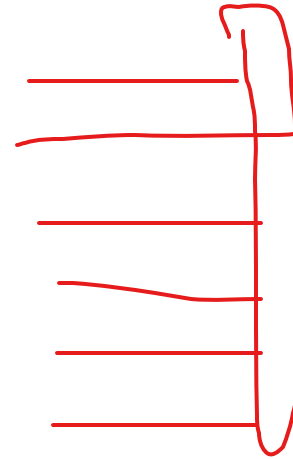
A “Broad” Product & “Implicit Backlogs”

“Oh yes, we have only **one broad product**, and...

Team-iOS = iOS items

Team-Android = Android items

Team-Backend = Backend items”



“IO Channels” & Product Definition

**Google Maps?**

iOS Google Maps?

Android Google Maps?

Web Google Maps?

**Faux “Scrum”?**





# LeSS Guides: *Product*

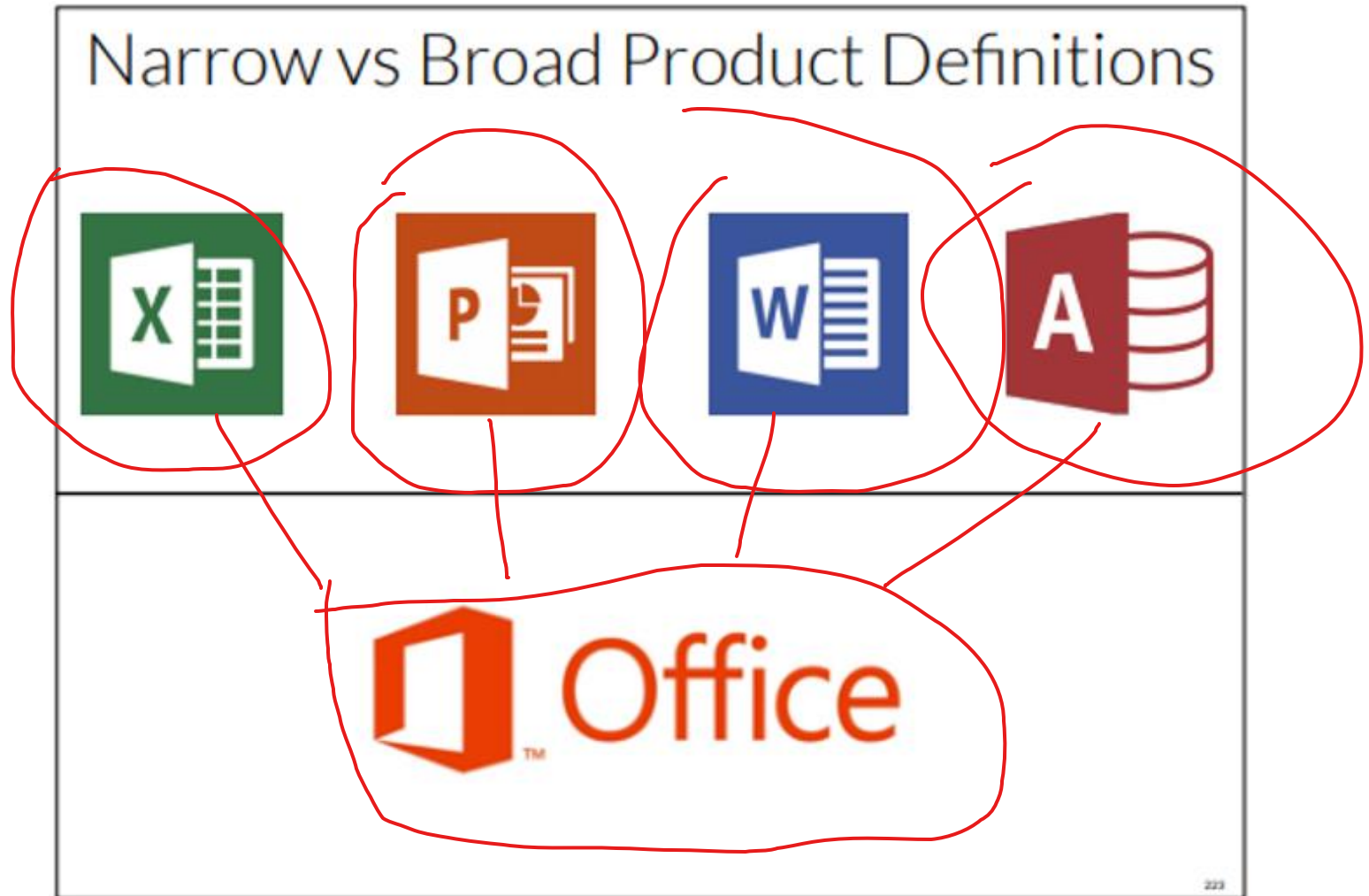
## Synonyms for “Component Team”





# LeSS Guides: *Product*

## *What Is Your Product?*

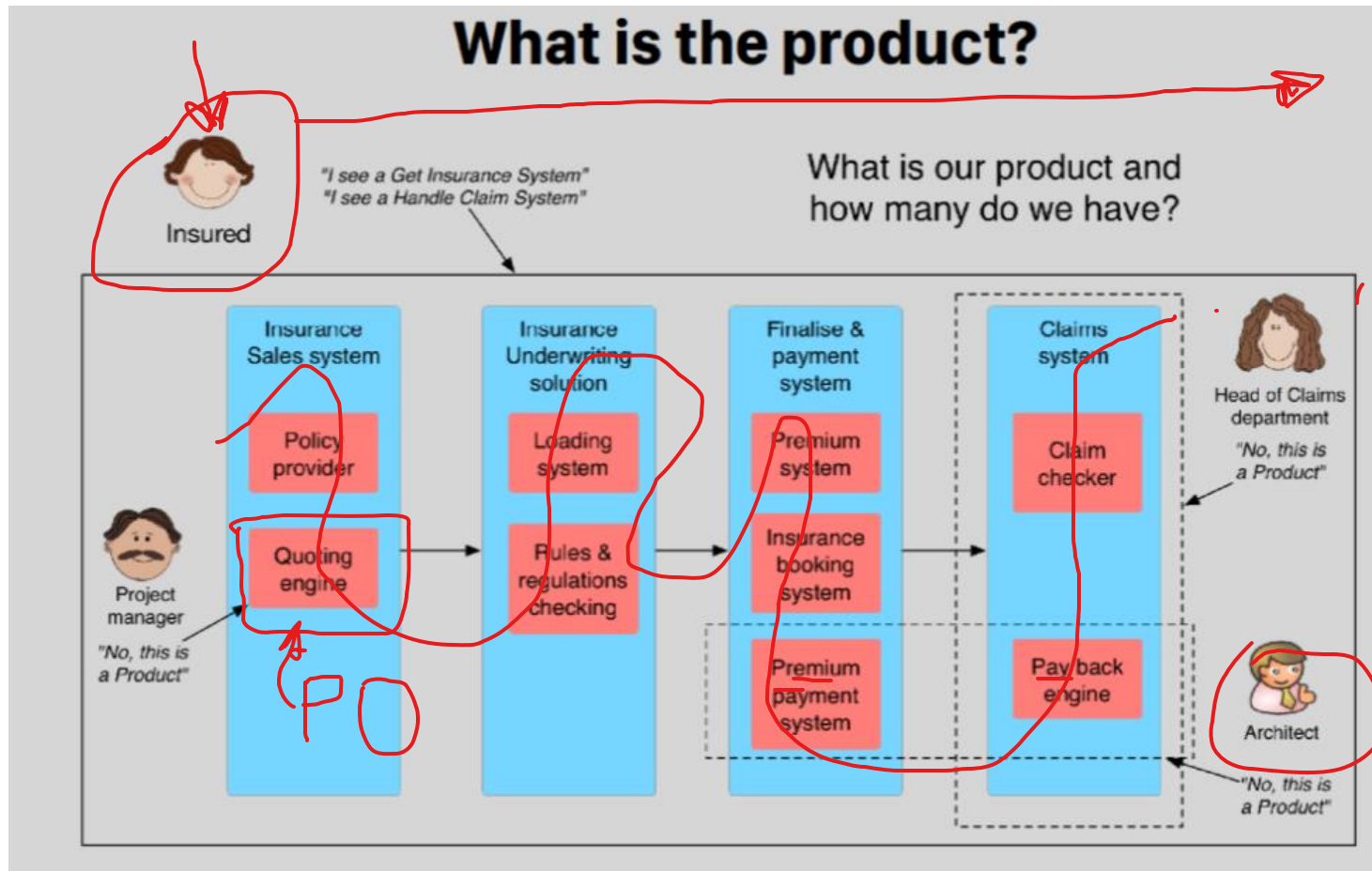


Sourced from creative commons repository of Certified LeSS Trainers: <https://less.works/courses/become-less-trainer>



# LeSS Guides: *Product*

## *What Is Your Product?*



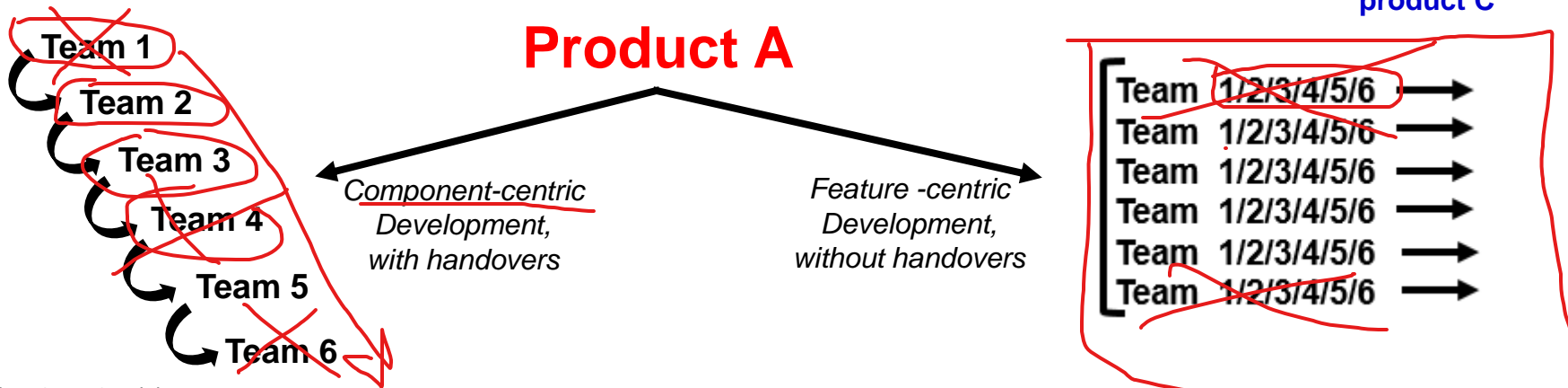
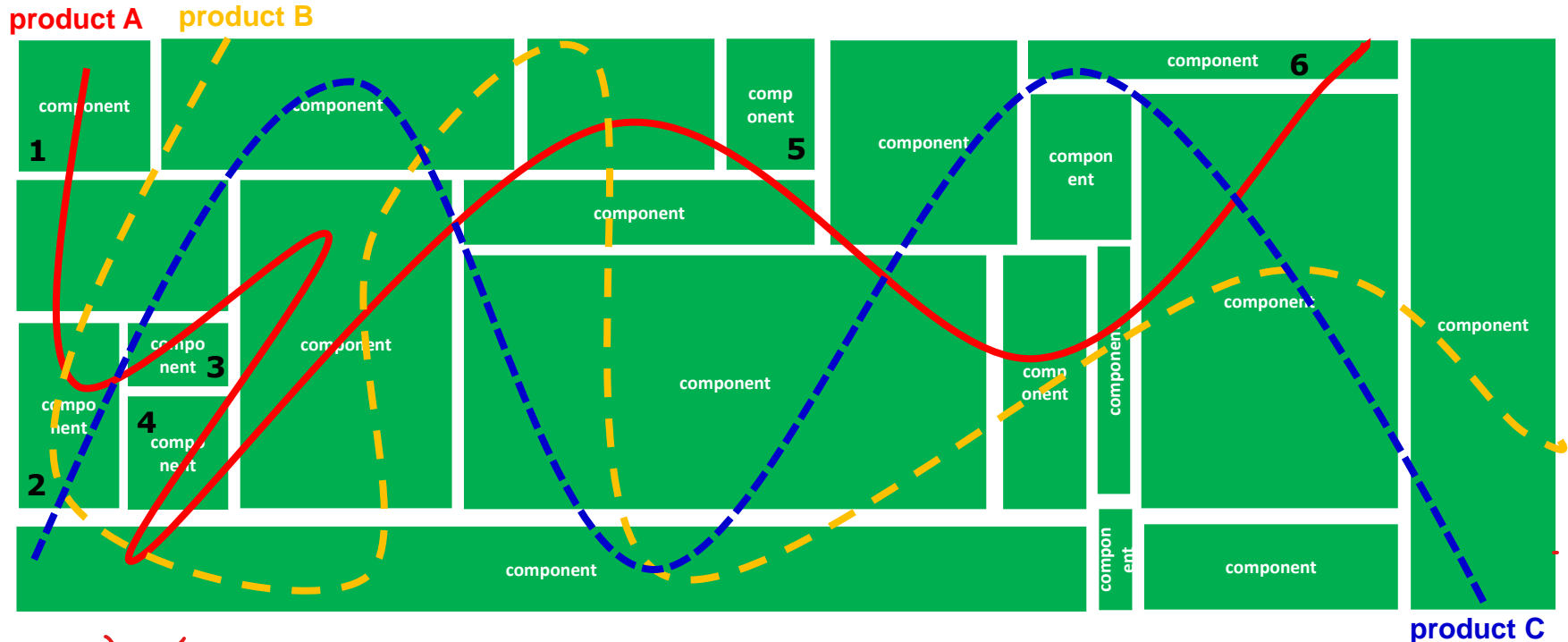
Sourced from creative commons repository of Certified LeSS Trainers: <https://less.works/courses/become-less-trainer>



# LeSS Guides: *Product*

PO

## Exercise: Bricks and Snakes





# LeSS Guides: *Product*

$$2-8+>>>\approx<70$$

Handwritten red annotations: A horizontal line connects the '8' and the first '>' symbol. Below this line, there is a handwritten '3-9' with a '\*' symbol above the '3'.

- 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)

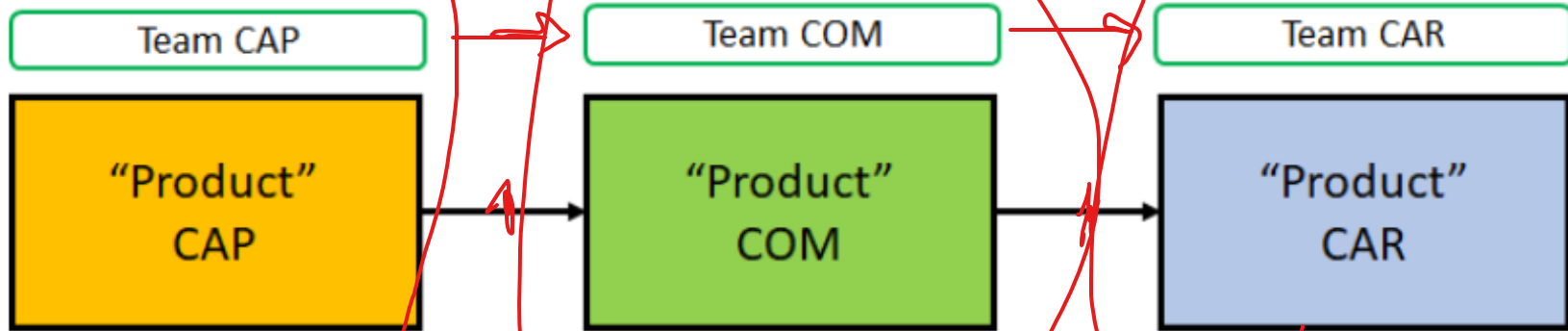
  


**PRODUCT**



# LeSS Guides: *Product*

## *Initial "Product" Definition*

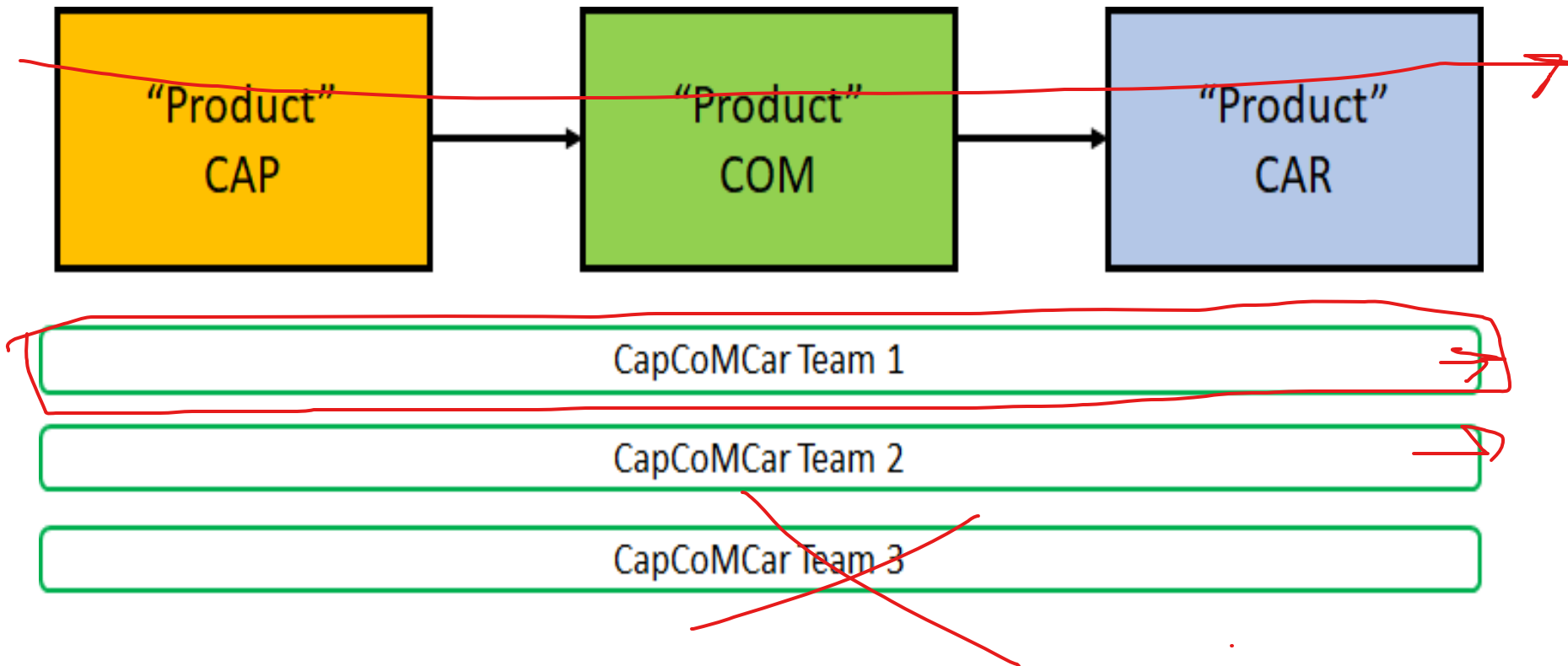


Created by: Gene Gendel



# LeSS Guides: *Product*

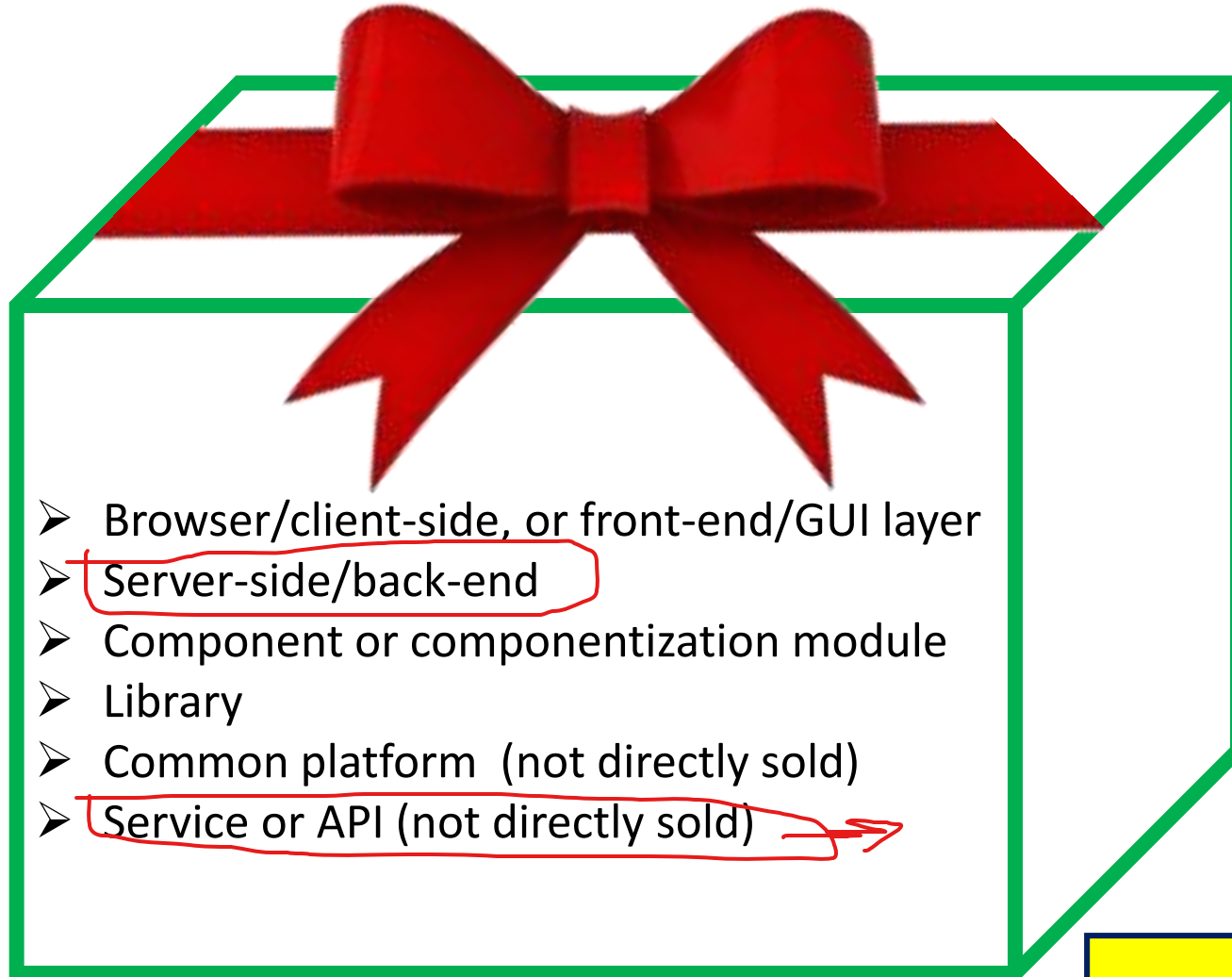
## *Expanded Product Definition*



Created by: Gene Gendel



# LeSS Guides: *Product*



- 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'?**

Created by: Gene Gendel



# Class Activity

**Class:** Review instructions of a system modelling exercise on the next slide. Split up into teams.

**Instructor:** Provide instructions.

**Duration:** next page





## Exercise

**Duration:** 10 min

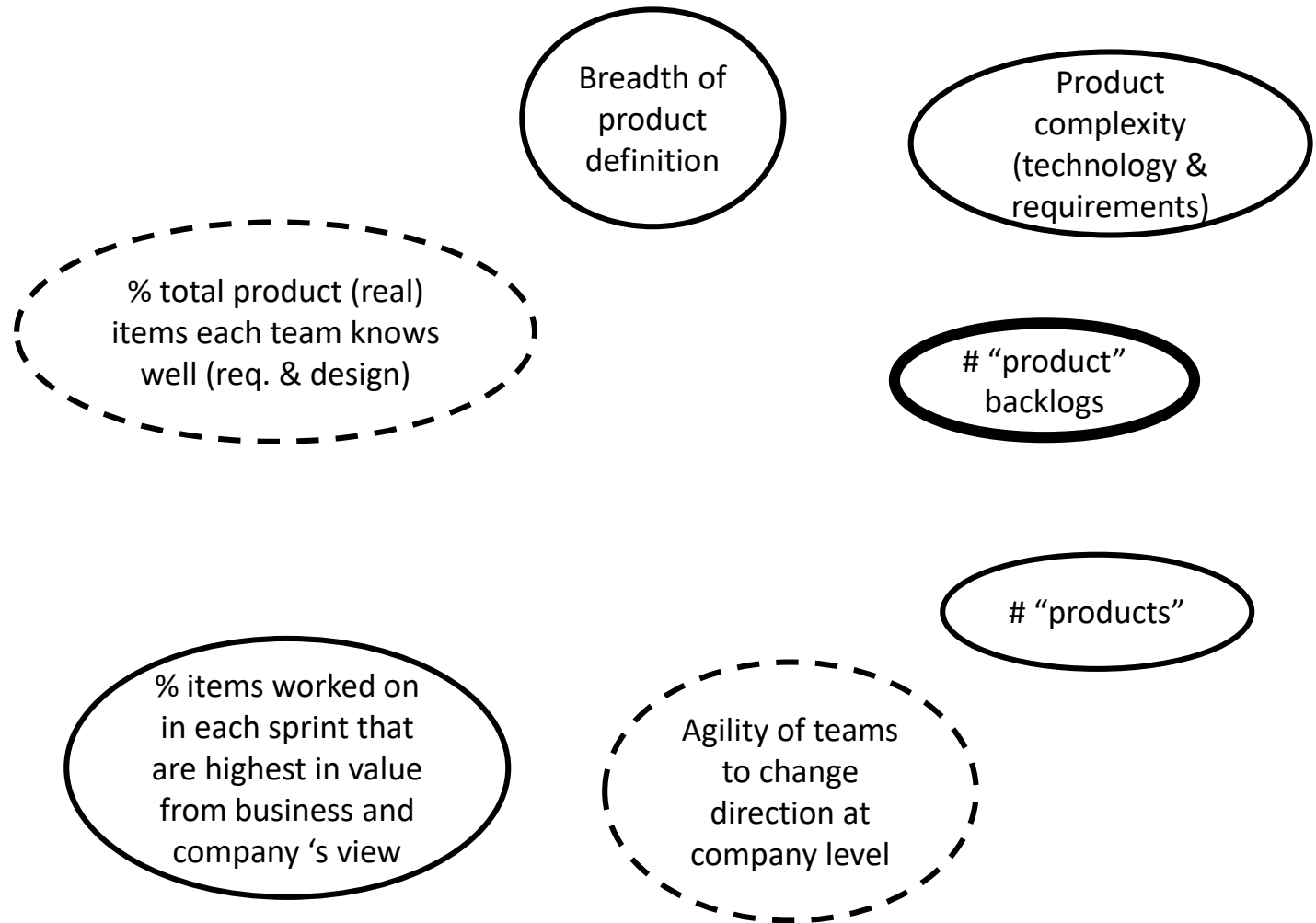
**Class:** in-groups, brainstorm some of the most common examples of ***Local Optimization in Product Definition*** in your respective organizations. Work with provided system variables to create a model.

**Method/Tool:** System Modeling with CLD

**Instructor:** review with Class

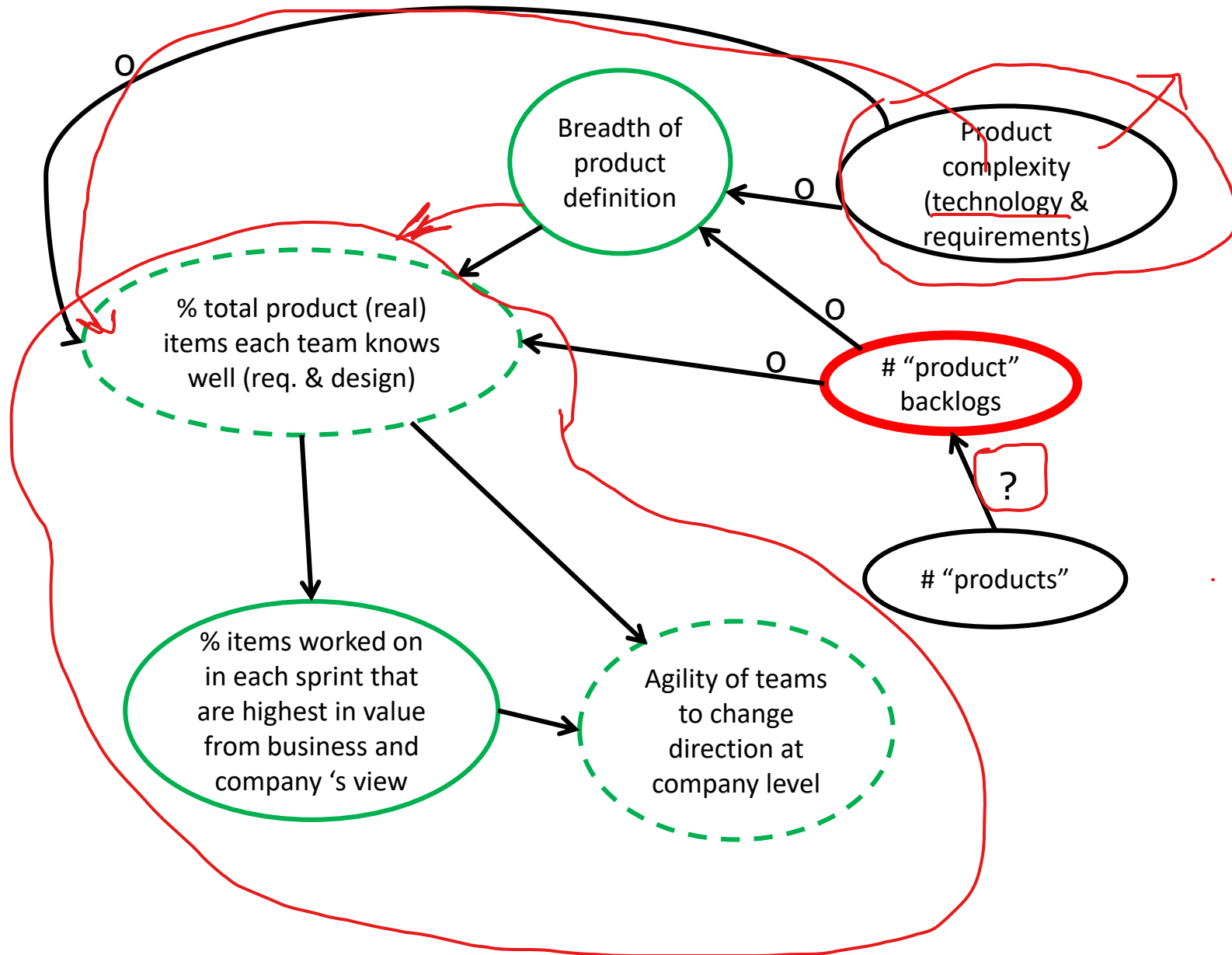


# Local Optimization in **Product Definition** - Exercise





# Local Optimization in **Product Definition** - Exercise



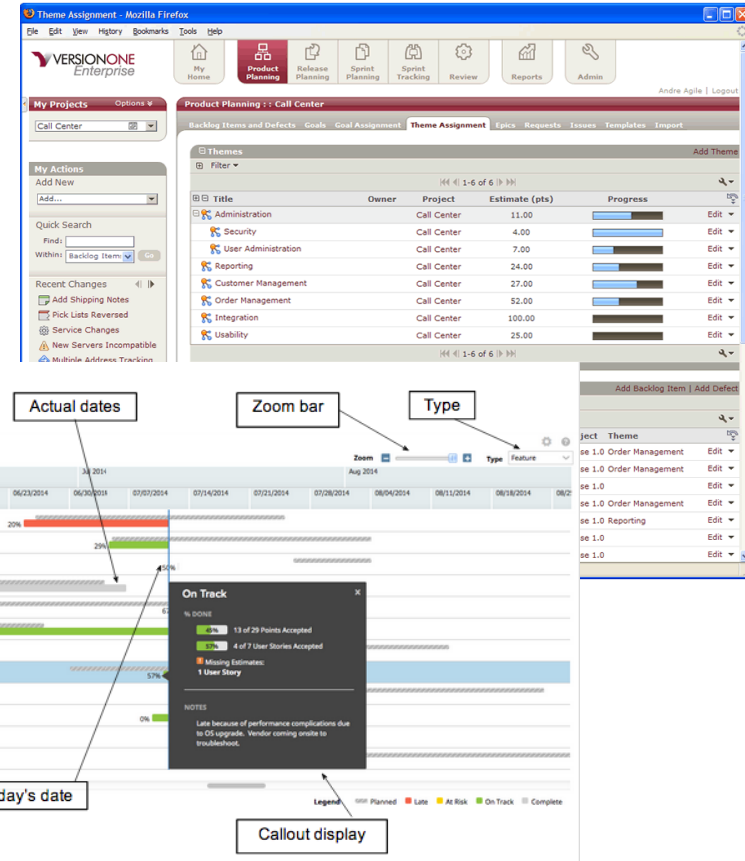
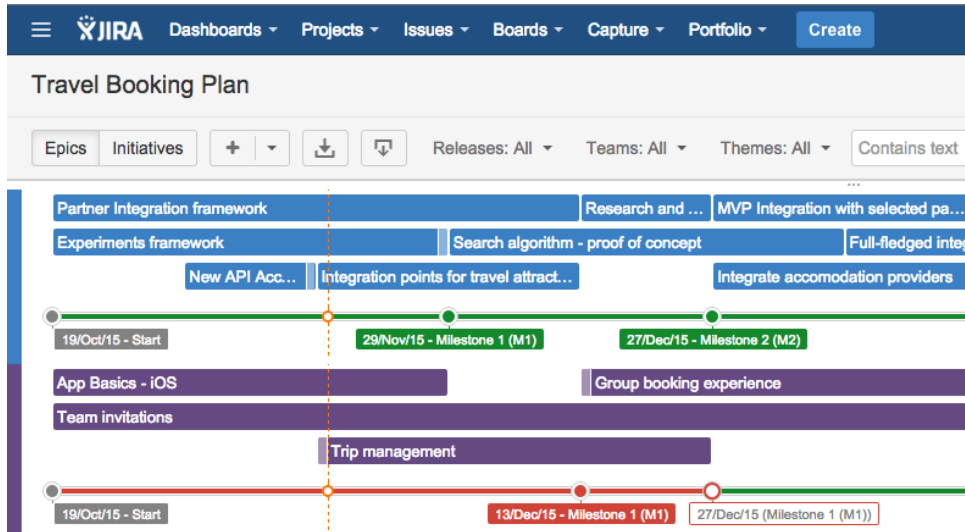


.



# LeSS Guides: *Product*

## *Product over Project or Program*

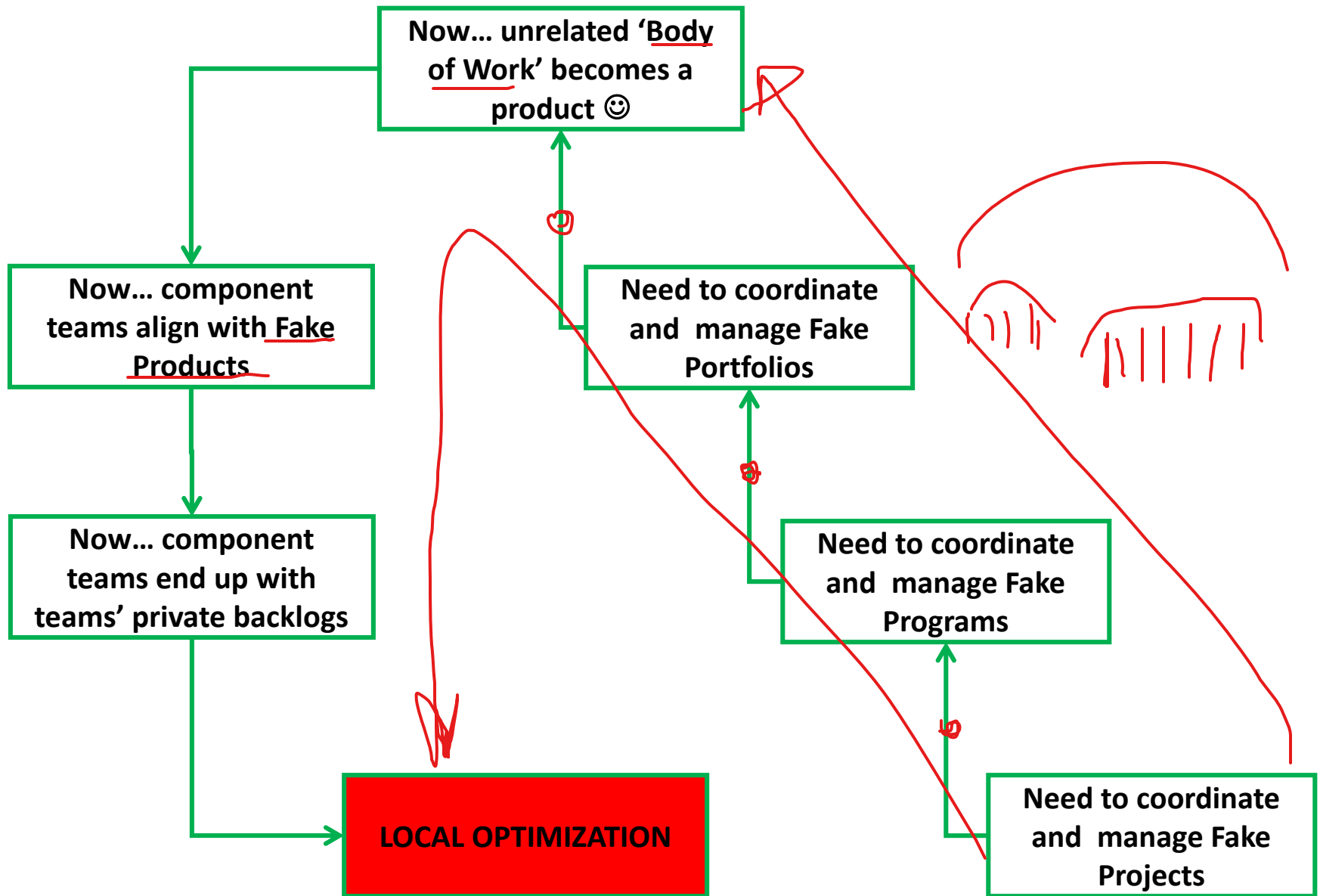


Don't manage products using projects or programs!

Sourced from: <https://www.keystepstosuccess.com/agile-anti-patterns-with-irony-and-satire/>



# LeSS Guides: *Product*

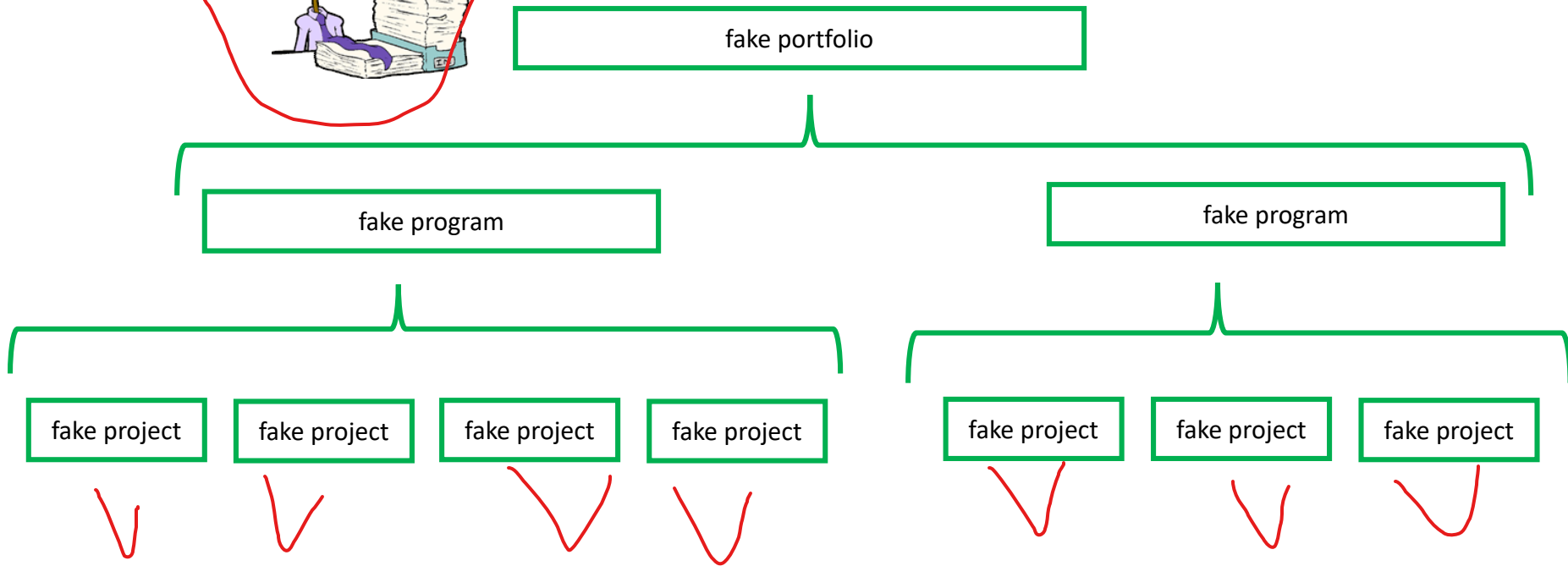




# LeSS Guides: *Product*



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





# Class Activity

**Class:** Review instructions of a system modelling exercise on the next slide. Split up into teams.

**Instructor:** Provide instructions.

**Duration:** next page





## Exercise

**Duration:** 10 min

**Class:** in-groups, brainstorm some of the most common pitfalls in defining a **product and problems with complex WBS (projects, programs and portfolios) and redundant roles**. Work with provided system variables to create a model.

**Method/Tool:** System Modeling with CLD

**Instructor:** review with Class

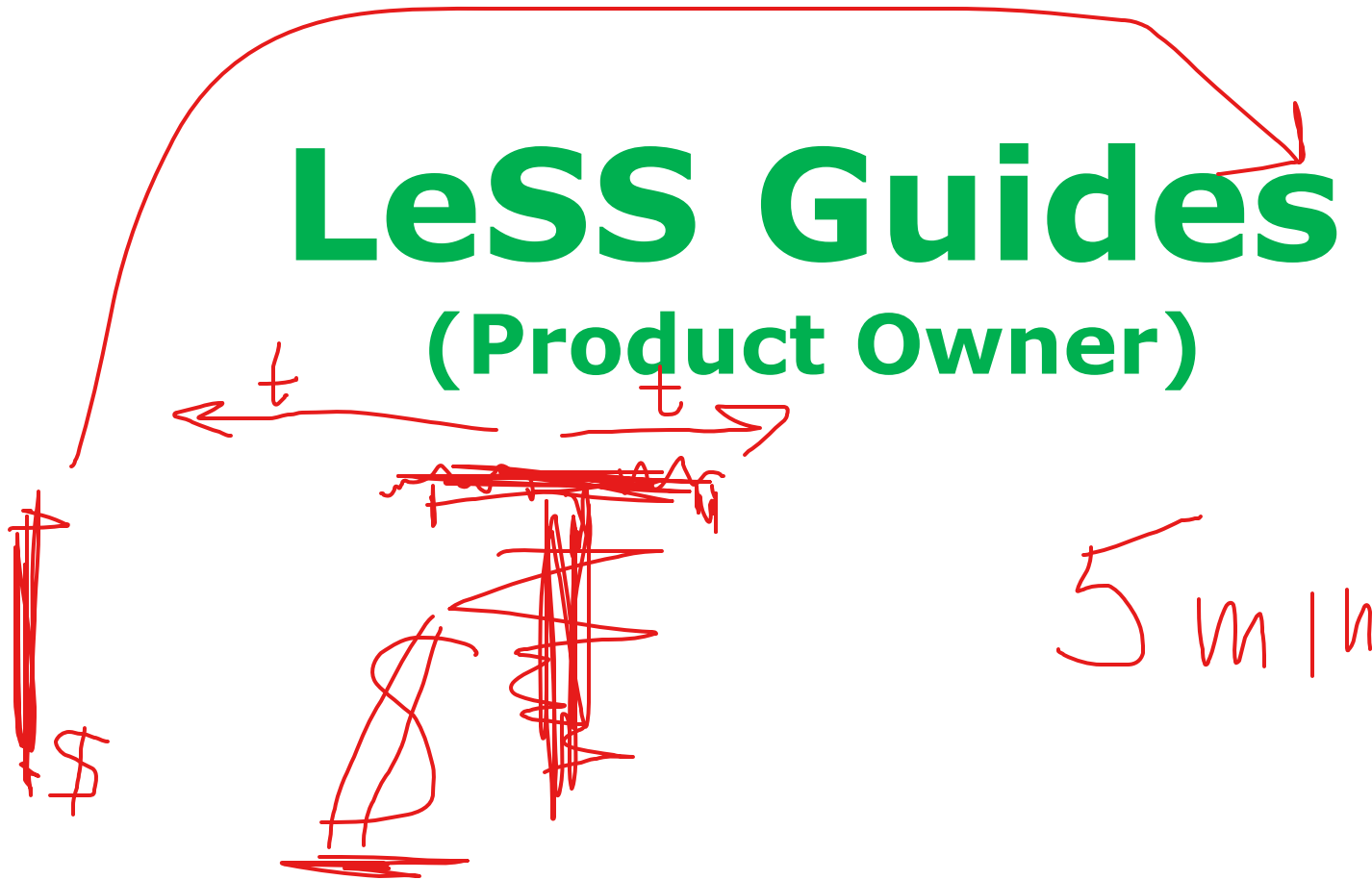






# LeSS Guides

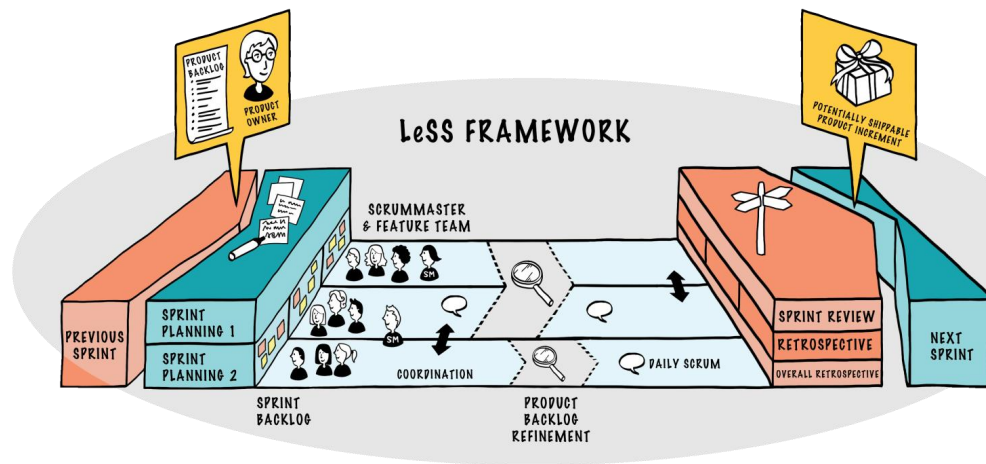
## (Product Owner)





# LeSS Guides: *Product Owner*

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



<http://less.works> 

<http://less.works> 



# LeSS Guides: *Product Owner*

## Why So Many Scrum Implementations Fail?



### 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'
```

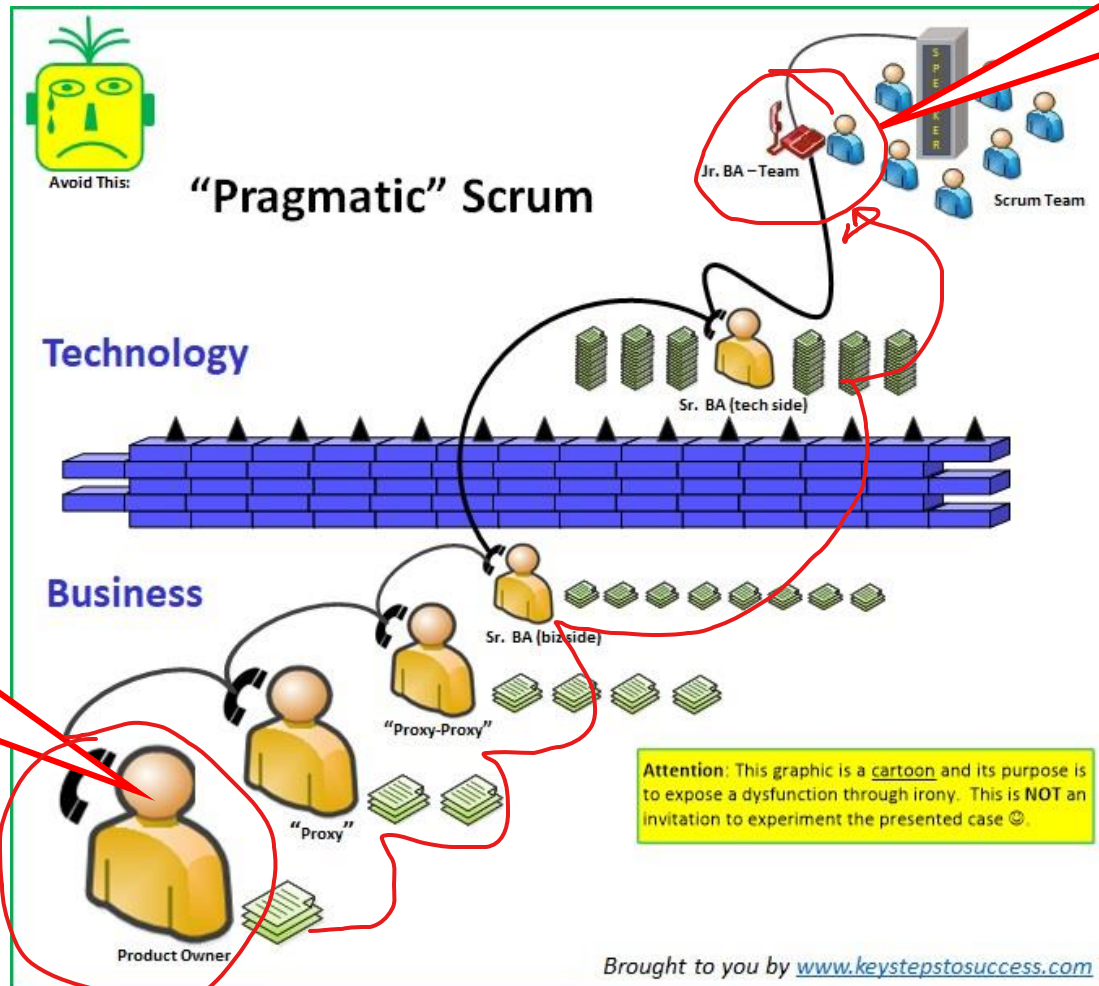
Brought to you by [www.keystepstosuccess.com](http://www.keystepstosuccess.com)

Sourced from: <https://www.keystepstosuccess.com/agile-anti-patterns-with-irony-and-satire/>



# LeSS Guides: *Product Owner*

## Why So Many Scrum Implementations Fail?



<https://www.keystepstosuccess.com/2020/11/less-review/>

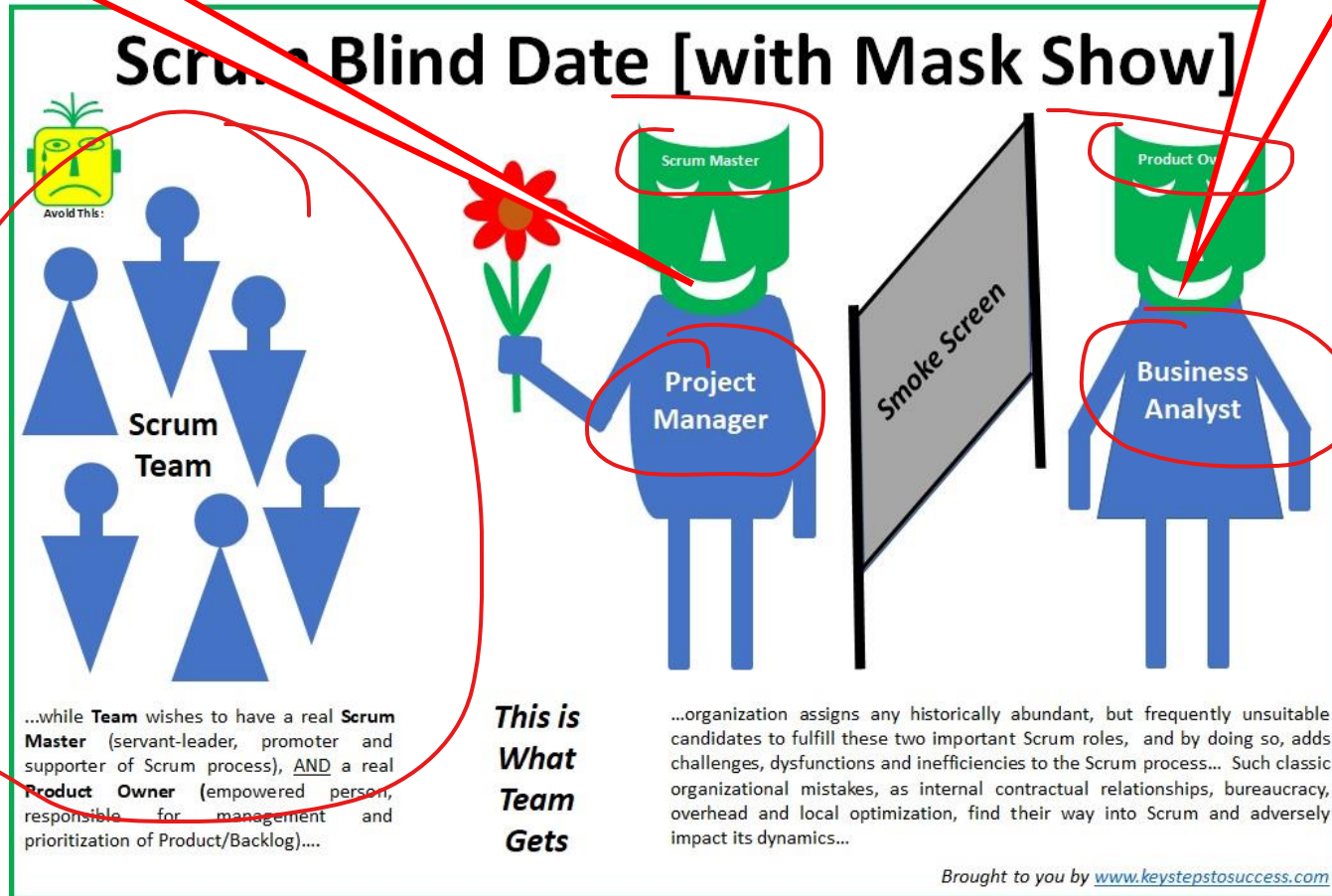


# LeSS Guides: *Product Owner*

"Oh yeah... cause my developers could not speak directly to customers and users. Now, I am their Scrum Master..."

## Why So Many Scrum Implementations Fail?

"Do you recognize me? I used to collect business requirements from users and giving to YOUR developers"

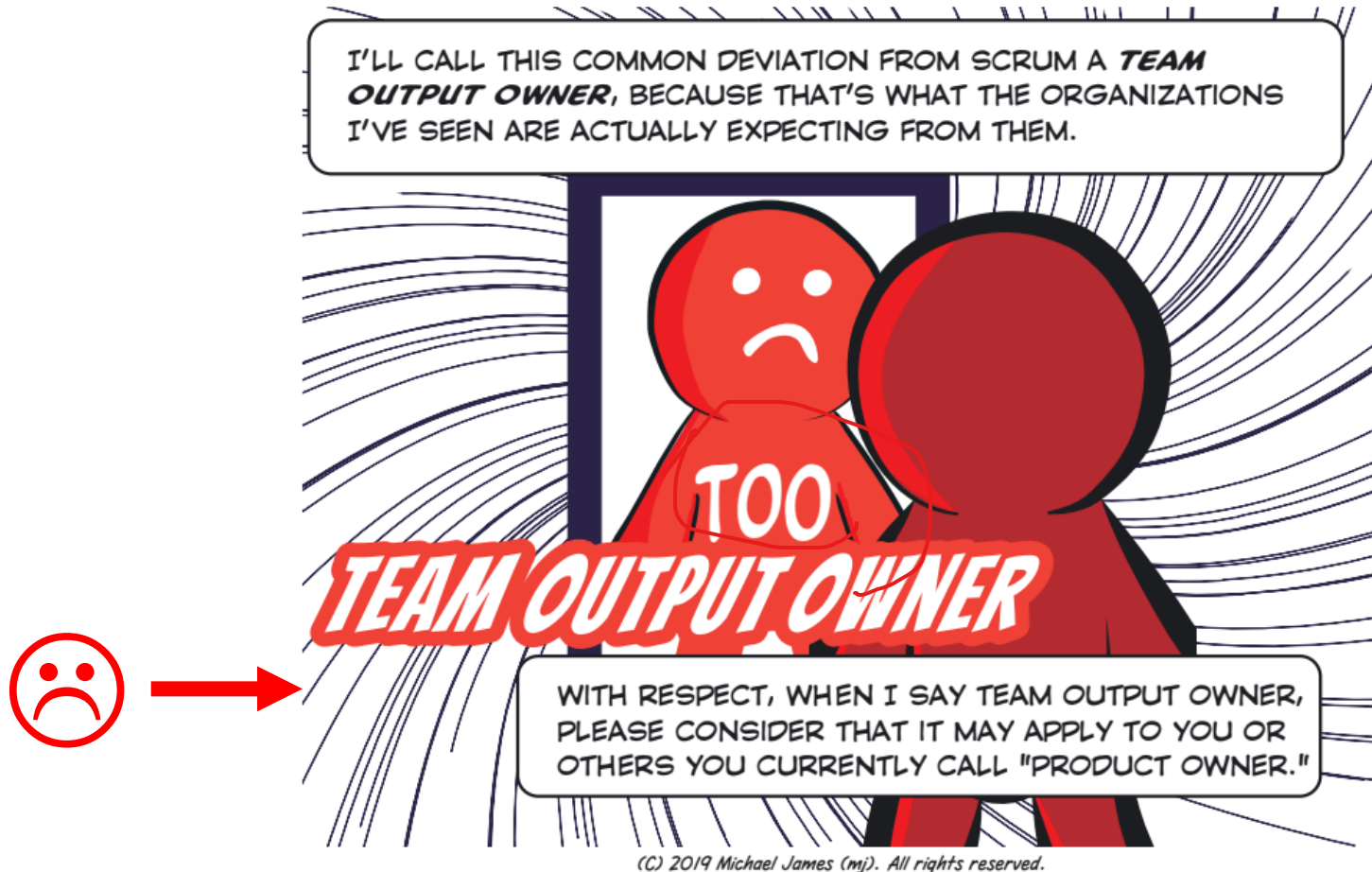


Sourced from: <https://www.keystepstosuccess.com/agile-anti-patterns-with-irony-and-satire/>



# LeSS Guides: *Product Owner*

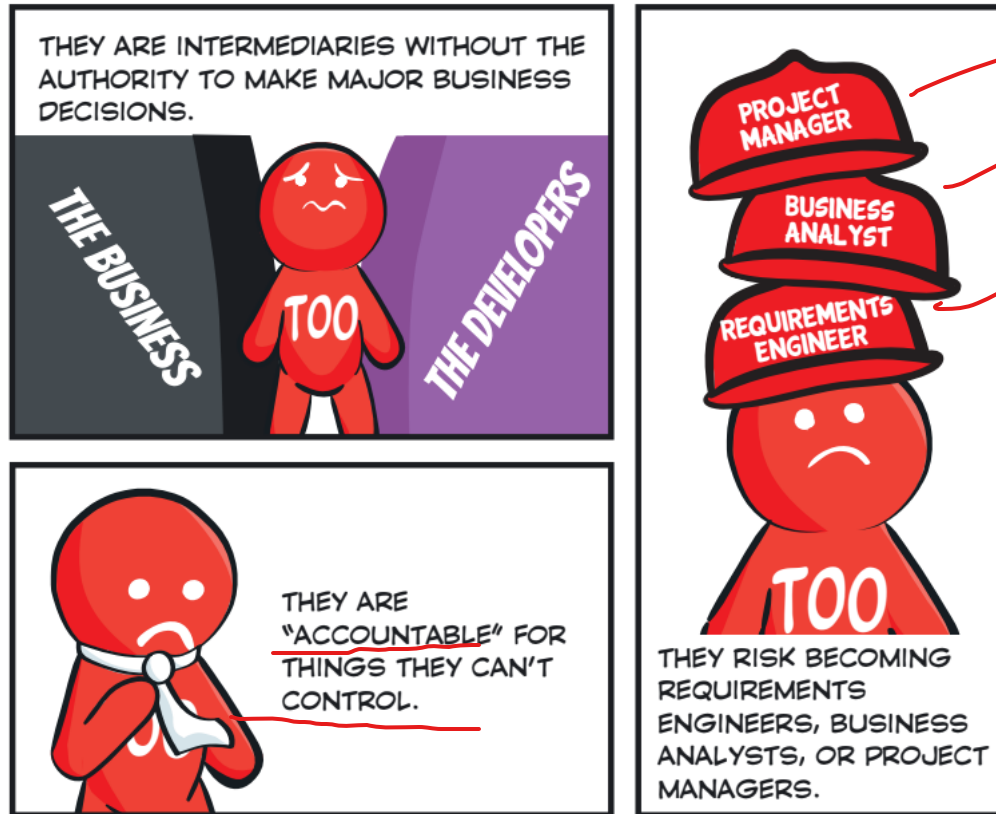
## *Why So Many Scrum Implementations Fail?*





# LeSS Guides: *Product Owner*

## Why So Many Scrum Implementations Fail?

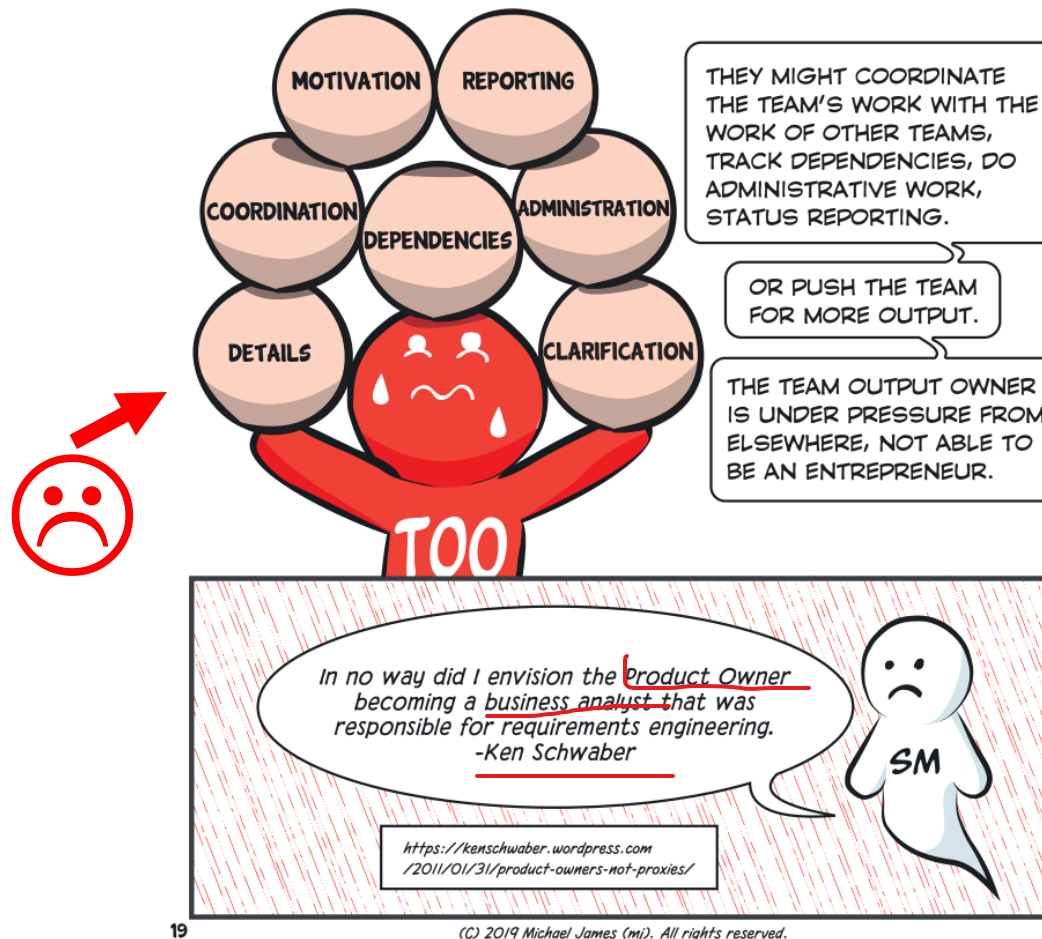


(C) 2019 Michael James (mj). All rights reserved.



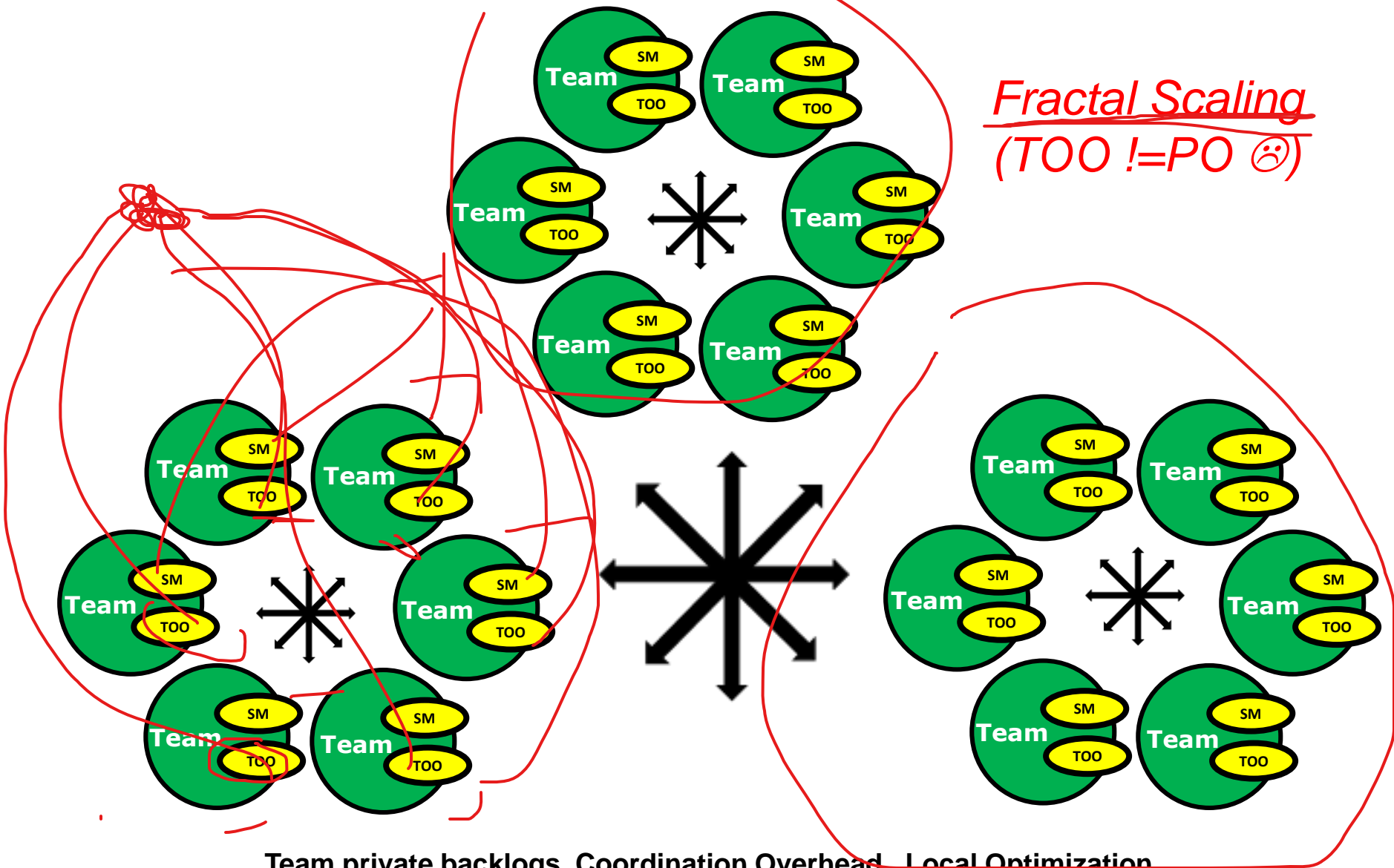
# LeSS Guides: *Product Owner*

## Why So Many Scrum Implementations Fail?





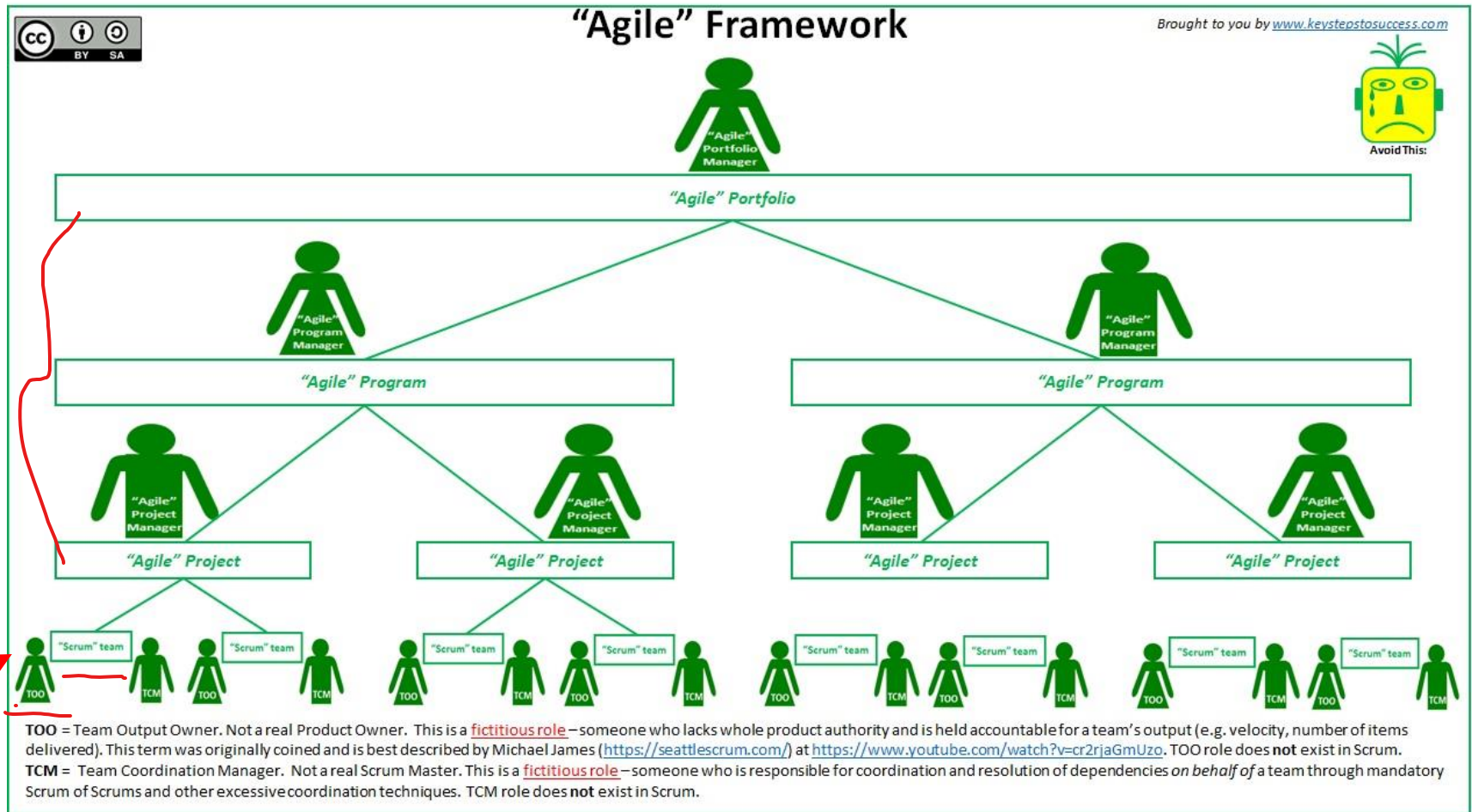
# LeSS Guides: *Product Owner*





# LeSS Guides: *Product Owner*

## Why So Many Scrum Implementations Fail?

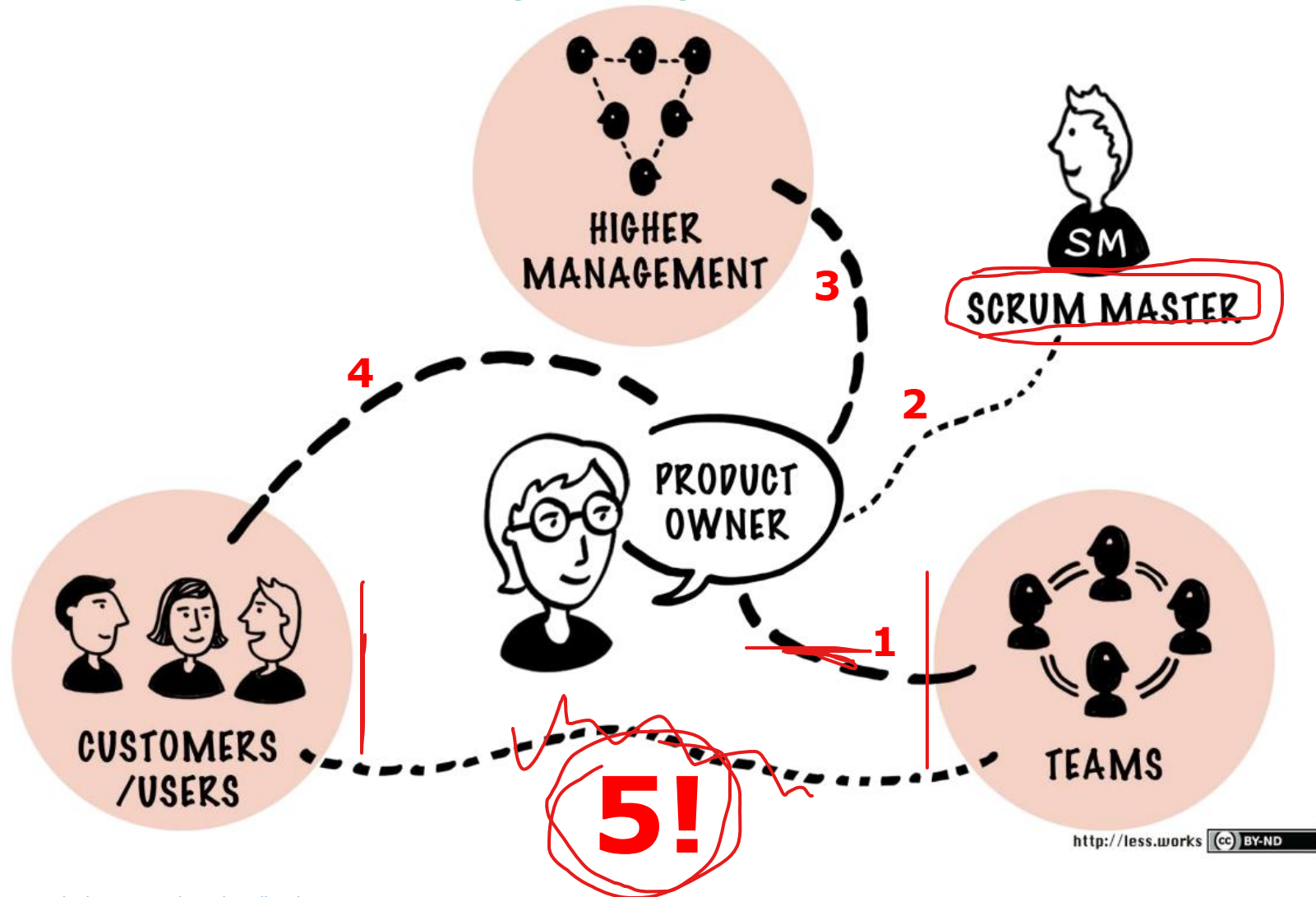


Sourced from: <https://www.keystepstosuccess.com/agile-anti-patterns-with-irony-and-satire/>



# LeSS Guides: *Product Owner*

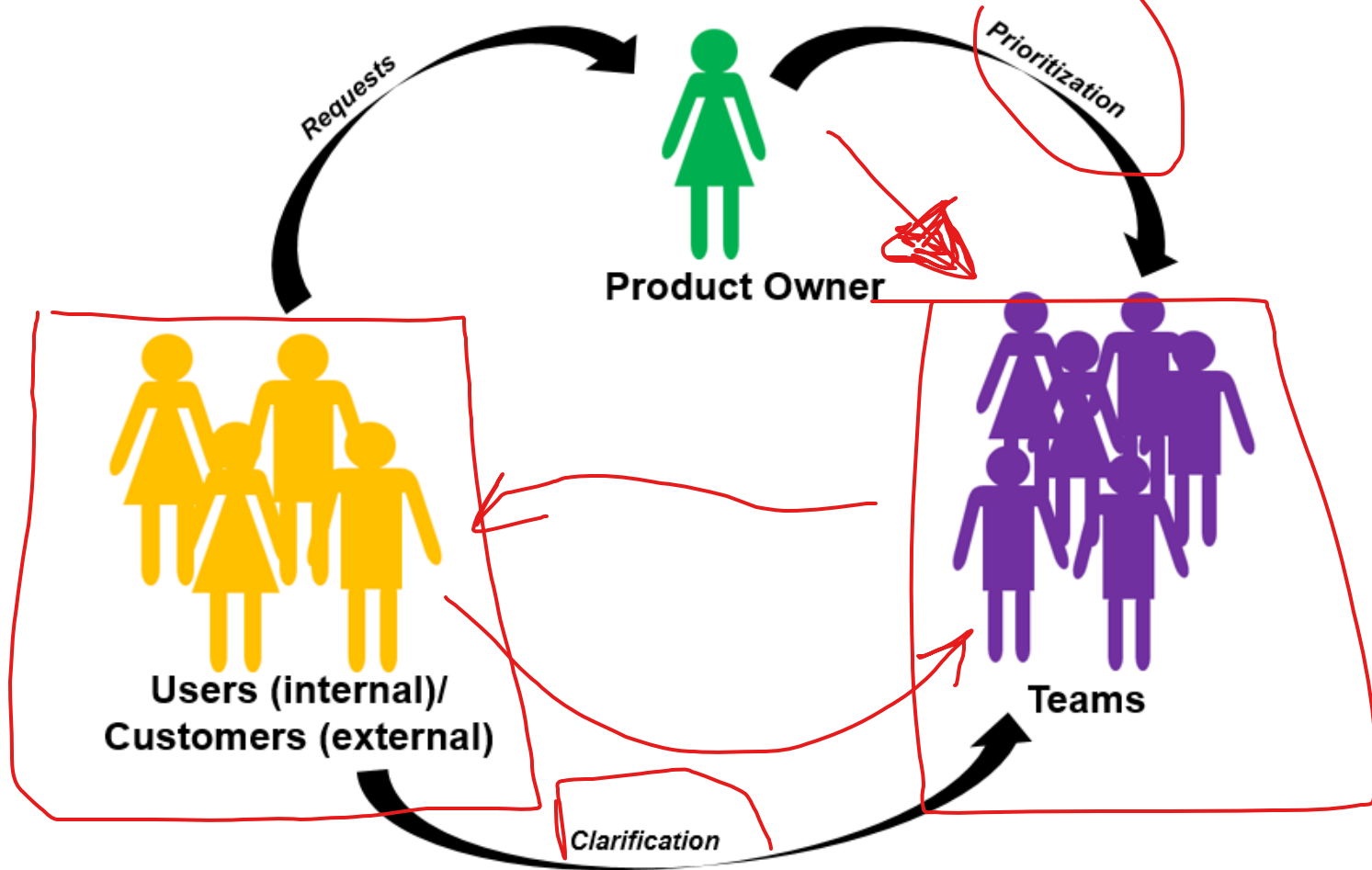
## *Proper Positioning and Alignment of Product Owner*





# LeSS Guides: *Product Owner*

## *Proper Positioning and Alignment of Product Owner*



Created by: Gene Gendel

# 5!



# LeSS Guides: *Product Owner*

## *Prioritization over Clarification*

There are two key *information flows* in Scrum related to the Product Owner: (1) Adaptively deciding the **direction** to evolve the product and reflecting that decision in Product Backlog prioritization, and (2) Discovering and clarifying the **details** of user needs and items. In the first flow (direction and prioritization), information is sought and analyzed related to profit drivers, strategic customers, business risks, etc. In the second flow (details and clarification), the objective is to discover the fine-grained behavior and qualities of items, the user experience, etc.

As Product Owner, you focus on thinking hard about direction and prioritization, but delegate to the teams as much of the detailed discovery as possible. You encourage and help teams enter in a direct conversation with users, acting as a *connector*, not an intermediary. In short, you are mostly focusing on prioritization rather than detailed clarification, which is delegated to the teams.

Sourced from: <https://less.works/resources/learning-resources/books>



# LeSS Guides: *Product Owner*

## *Proper Positioning and Alignment of Product Owner*

There is one Product Owner and one Product Backlog for the complete shippable product.

The Product Owner shouldn't work alone on Product Backlog refinement; it is mostly done by the multiple Teams working directly with customers, users, and other stakeholders.

All prioritization (ordering) goes through the Product Owner, but clarification is as much as possible directly between the Teams and customer, users, and other stakeholders.

Sourced from: <https://less.works/resources/learning-resources/books>



# LeSS Guides: *Product Owner*

## *Consumption of Product Owner's Time In LeSS*

What LeSS meetings does the Product Owner attend, and what is their *average actual* duration in a typical two-week Sprint?

1. Sprint Planning Part One: 1 hour
2. If doing overall Product Backlog refinement: 1 hour<sup>6</sup>
3. Sprint Review: 2 hours
4. Overall Retrospective: 1.5 hours

So the total time together in meetings is less than a new Product Owner might imagine: realistically, perhaps six hours in a two-week Sprint.

Sourced from: <https://less.works/resources/learning-resources/books>

6-7 → 10h



# LeSS Guides: *Product Owner*

## Who Are Those Users/Customers?

Type	Sub-type	Who is the source of requirements? <sup>a</sup>	Who validates & gives feedback?
Product development	Innovation-centric, and/or strongly influenced by new technologies and/or standards-driven.	No real users or even proxies provide requirements. Rather, requirements come internally from <u>product managers</u> (including the Product Owner), team members, etc.	<i>Pseudo-users</i> : candidate users, internal volunteers, and users of prior related products.
Product development	Driven by customer requirements, and it's a mass market.	User proxies such as <u>product managers, marketers, team members, and other customer- or market-facing experts</u> . Use focus groups of candidate or existing users.	source
Product development	Driven by customer requirements, and there are only, say, 50 customers.	<u>Hands-on users at multiple customers.</u>	source
Internal development	Regular.	Internal hands-on users.	source
Internal development	Special change initiative, e.g. regulatory.	The source of the special change, such as a policy maker or regulator.	source
Project development		Hands-on users at the one paying customer.	source

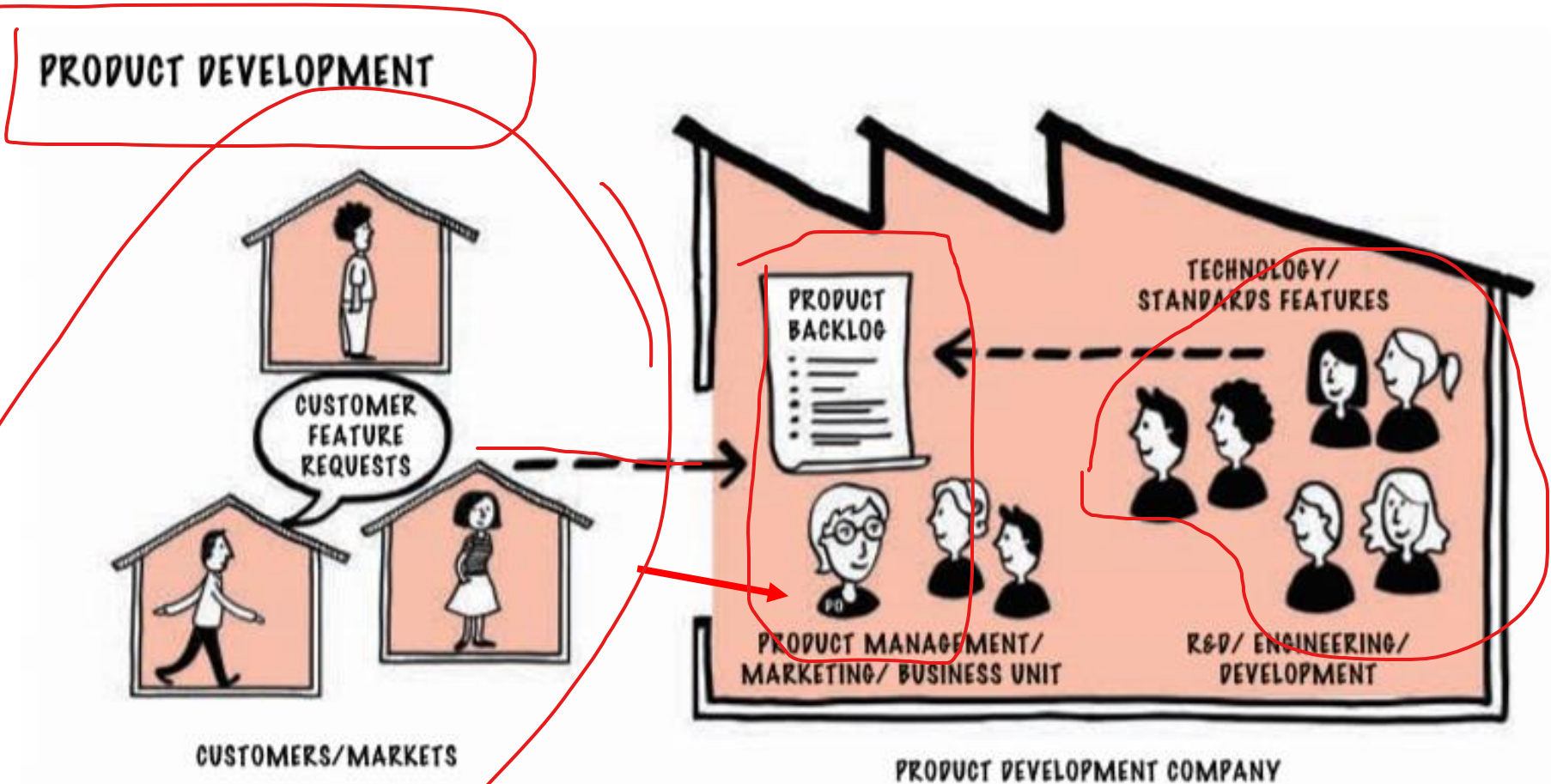
a. This is an illustrative *introduction*; not meant to be thorough or in-depth.

Sourced from: <https://less.works/resources/learning-resources/books>



# LeSS Guides: *Product Owner*

*Who Should be Product Owner?*



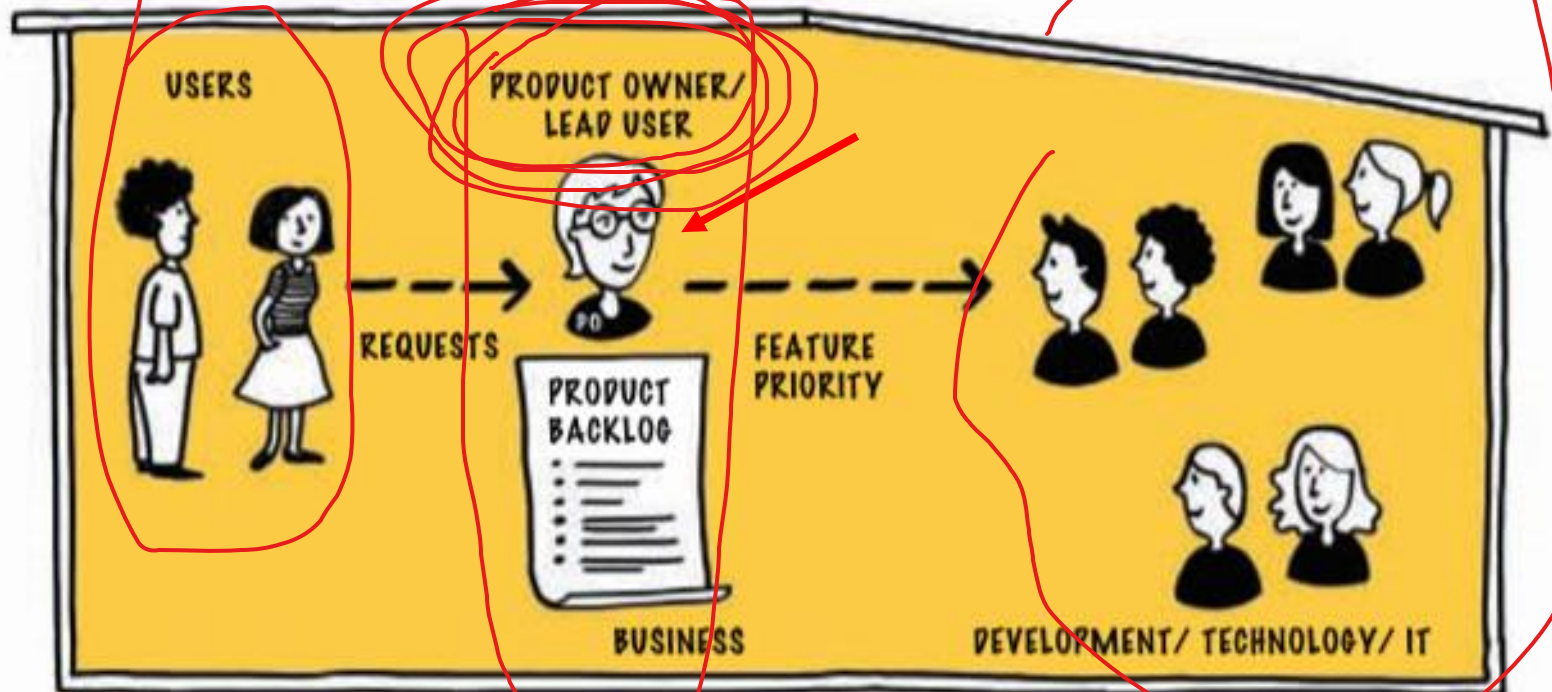
Sourced from: <https://less.works/resources/graphics/index.html>



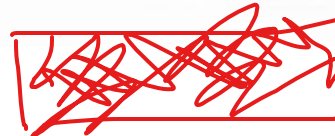
# LeSS Guides: *Product Owner*

*Who Should be Product Owner?*

## INTERNAL (PRODUCT) DEVELOPMENT



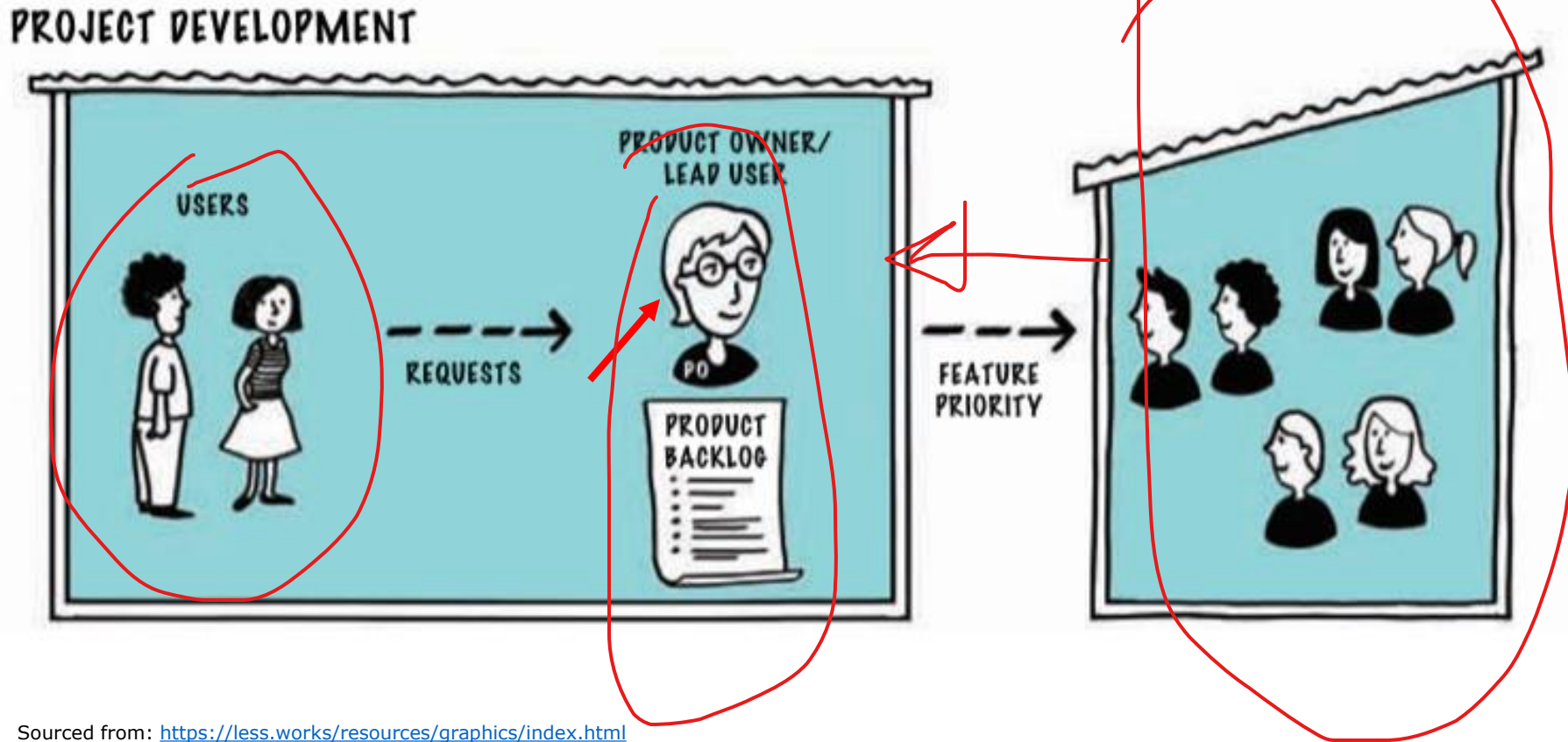
Sourced from: <https://less.works/resources/graphics/index.html>





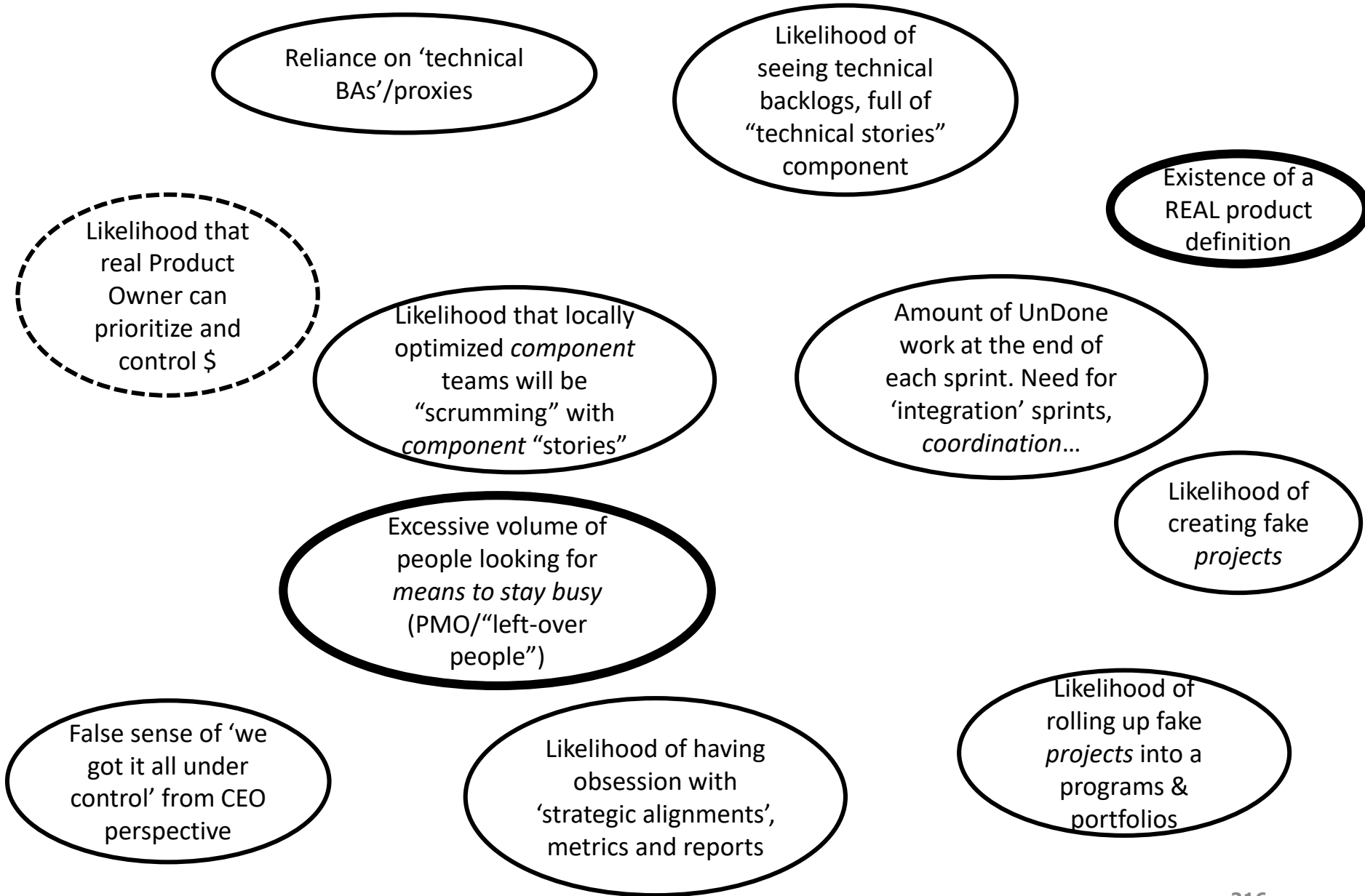
# LeSS Guides: *Product Owner*

## *Who Should be Product Owner?*



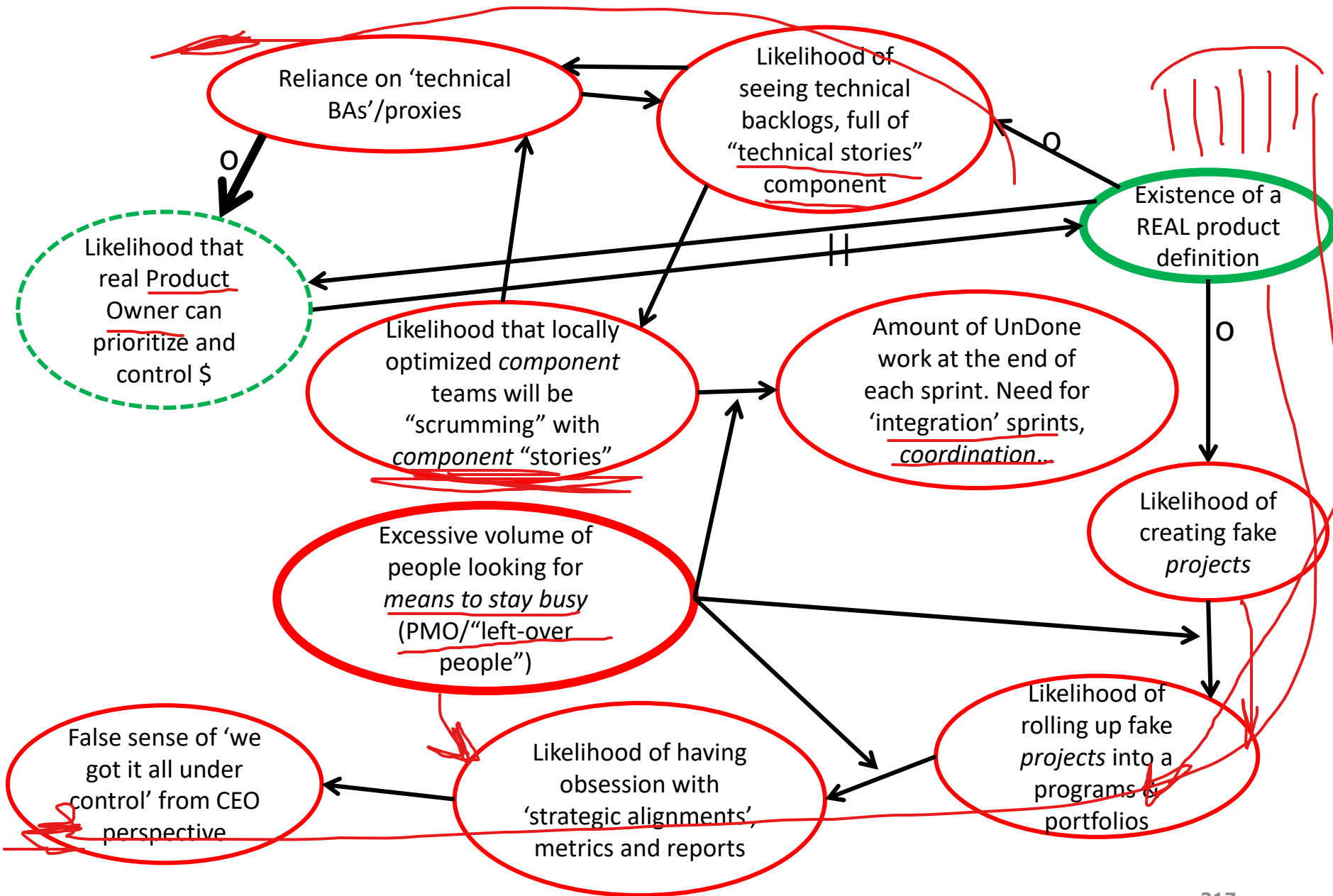


# Local Optimization in Roles & WBS - Exercise





# Local Optimization in Roles & WBS - Exercise



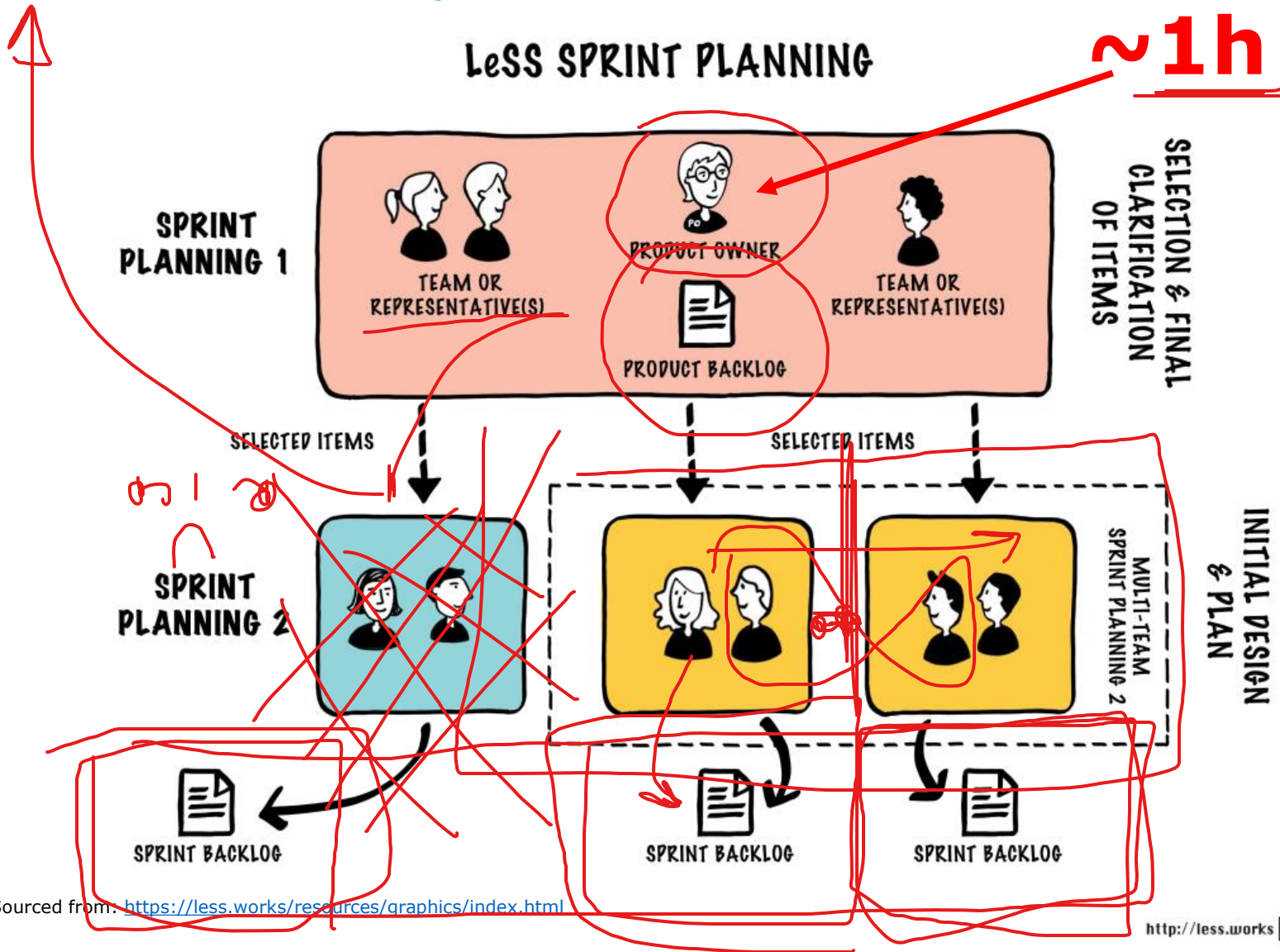






# LeSS Guides: *Product Owner*

## Consumption of Product Owner's Time In LeSS



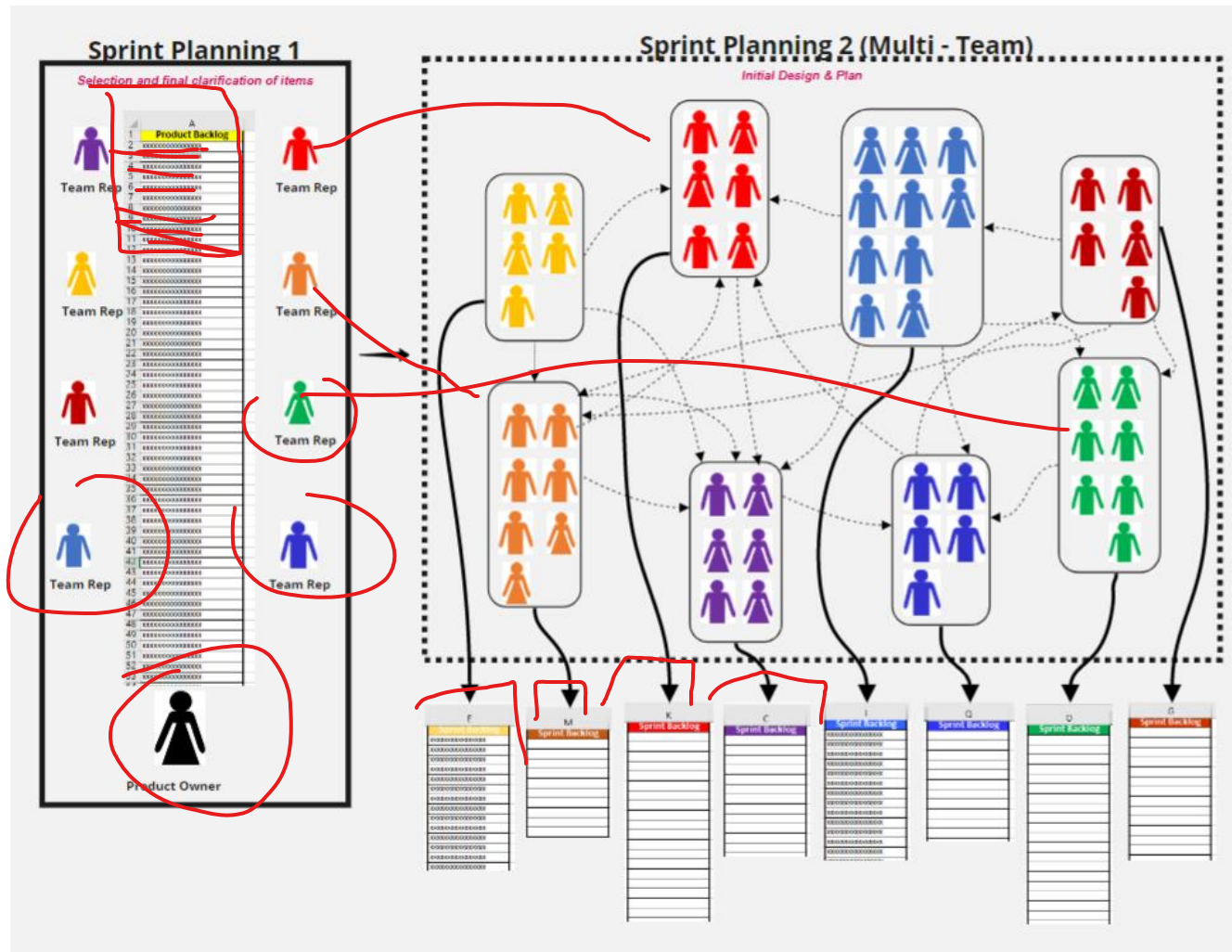
Sourced from: <https://less.works/resources/graphics/index.html>

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



# LeSS Guides: *Product Owner*

## *Consumption of Product Owner's Time In LeSS*

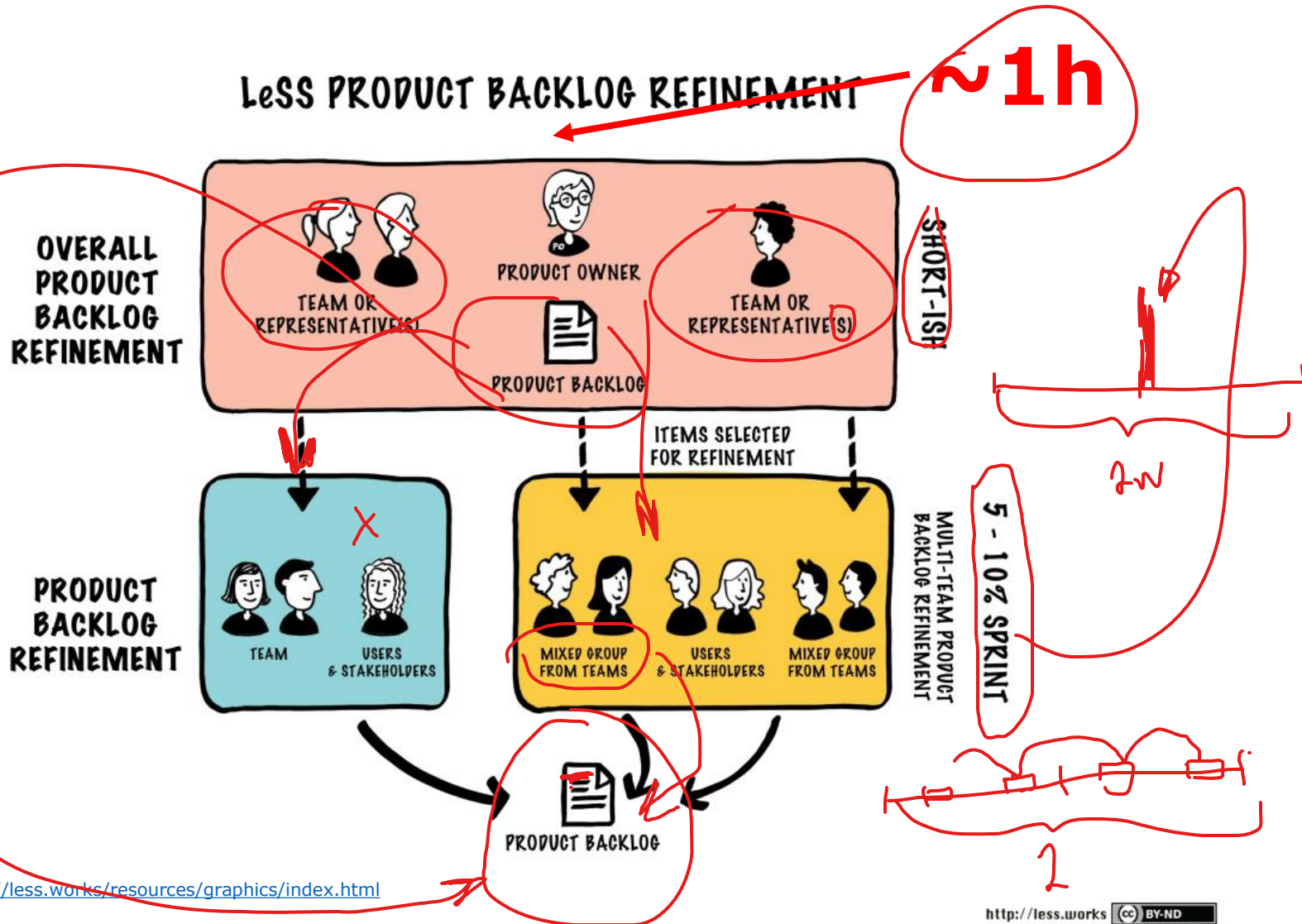


Created by: Gene Gendel



# LeSS Guides: *Product Owner*

## Consumption of Product Owner's Time In LeSS



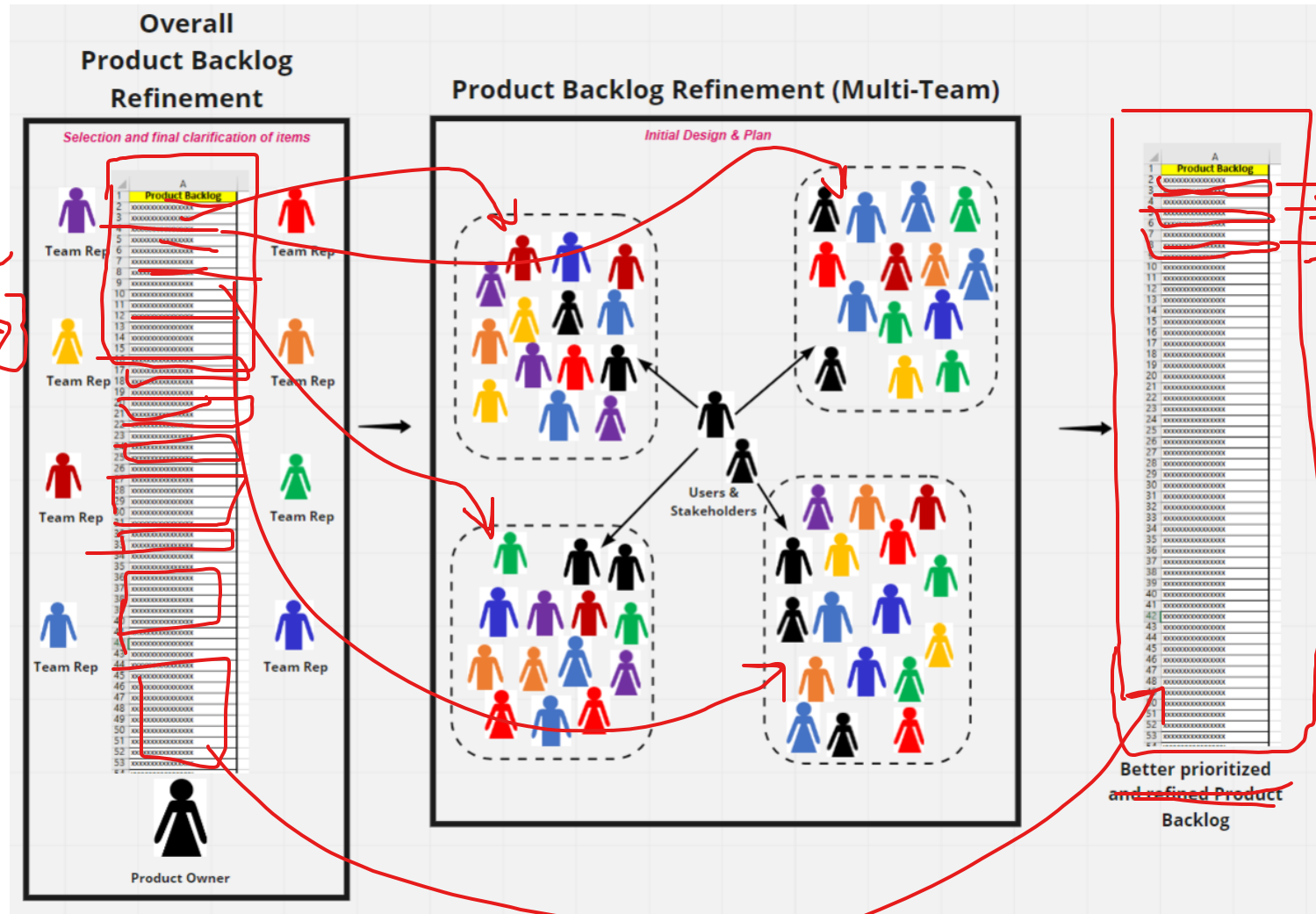
Sourced from: <https://less.works/resources/graphics/index.html>

<http://less.works> 



# LeSS Guides: *Product Owner*

## *Consumption of Product Owner's Time In LeSS*



Created by: Gene Gendel



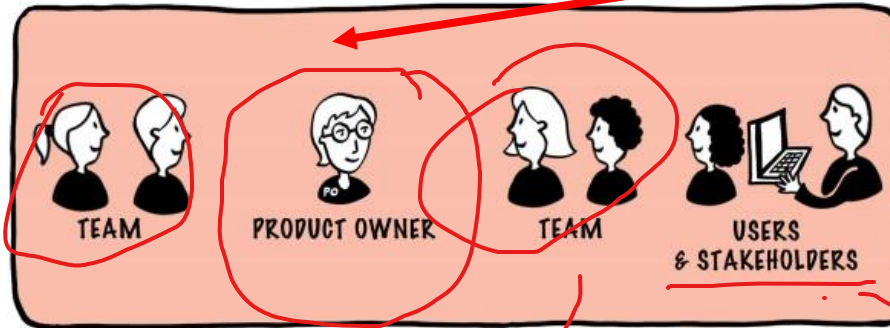
# LeSS Guides: *Product Owner*

## Consumption of Product Owner's Time In LeSS

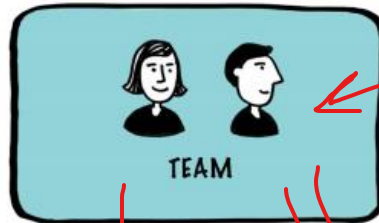
### LeSS SPRINT REVIEW & RETROSPECTIVE

~2h

SPRINT  
REVIEW



TEAM  
RETROSPECTIVE



OVERALL  
RETROSPECTIVE



~1.5h

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

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



# LeSS Guides: *Product Owner*

## *Don't Be Nice...but sometimes... Let Go*

As Product Owner you play a key role in setting expectations for teams. It will happen that the teams come to you and mention that a *set of items are only half done*. A skillful Product Owner might empathize... but doesn't "accept" it. Don't be nice. Instead make *very clear* that the items are **not done** and that you expect the teams to improve how they work so they deliver done items.

~~"Don't be nice" doesn't mean micromanaging.~~ In an effective LeSS adoption there are self-managing, co-located feature teams that do all the work and coordinate with other teams. There is high transparency that comes from delivering (or failing to deliver) a complete product in a short cycle. So the habits of trying to control development during the Sprint can be let go.

Sourced from: <https://less.works/resources/learning-resources/books>



# LeSS Guides: *Product Owner*

## *Start Early or Messy with a Temporary Fake PO*

It's terribly important that everyone understand that the temporary fake Product Owner is... a fake. And is replaced as soon as possible. It helps to literally use the name **Fake Product Owner.**

Sourced from: <https://less.works/resources/learning-resources/books>



# Class Activity

**Class:** Review instructions of a system modelling exercise on the next slide.

**Instructor:** Provide instructions

**Duration:** next page





## Exercise

**Duration:** 10 min

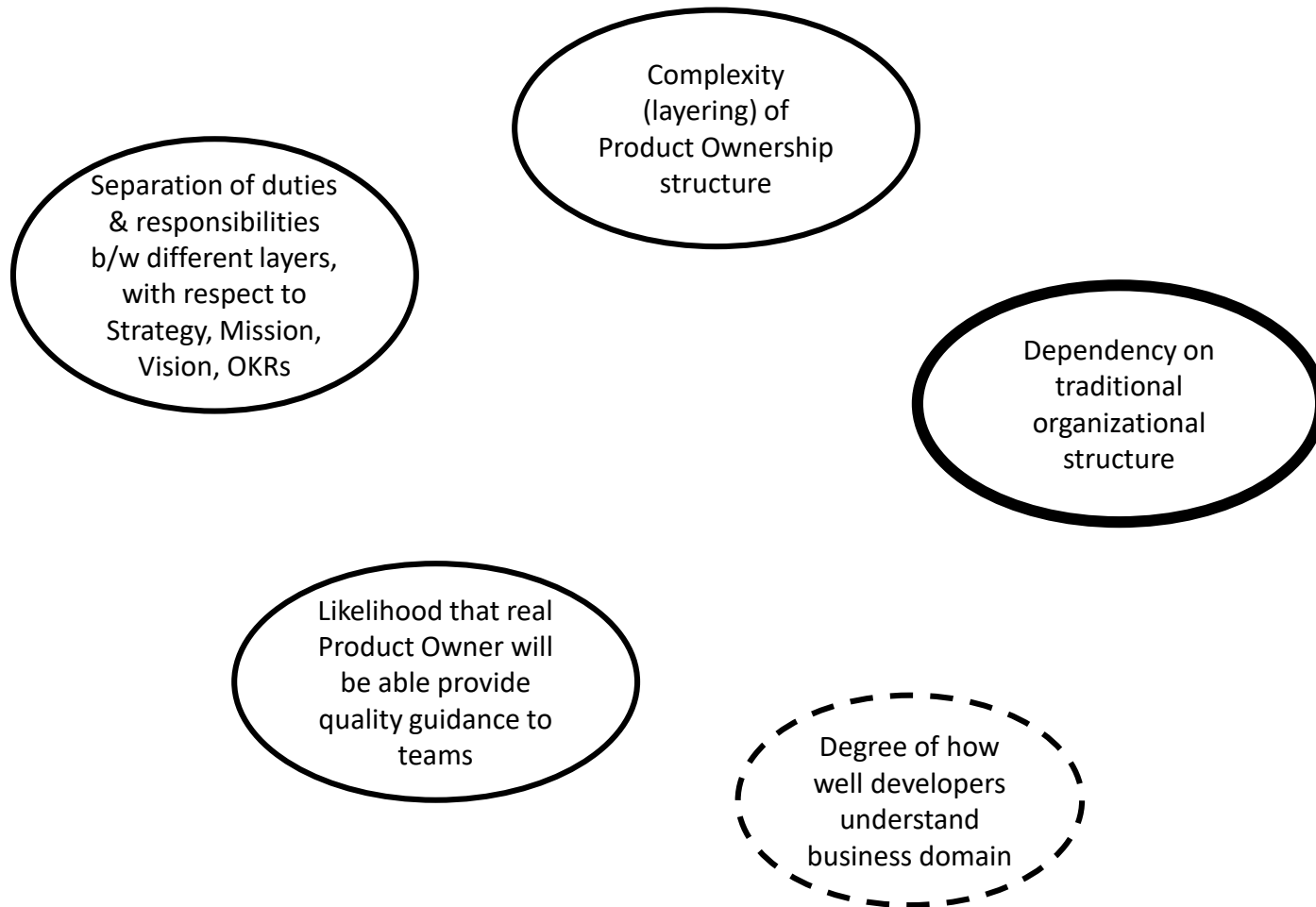
**Class:** in-groups, brainstorm some of the most common examples of ***Local Optimization in PO-ship Structure***, in your respective organizations. Work with provided system variables to create a model.

**Method/Tool:** System Modeling with CLD

**Instructor:** review with Class

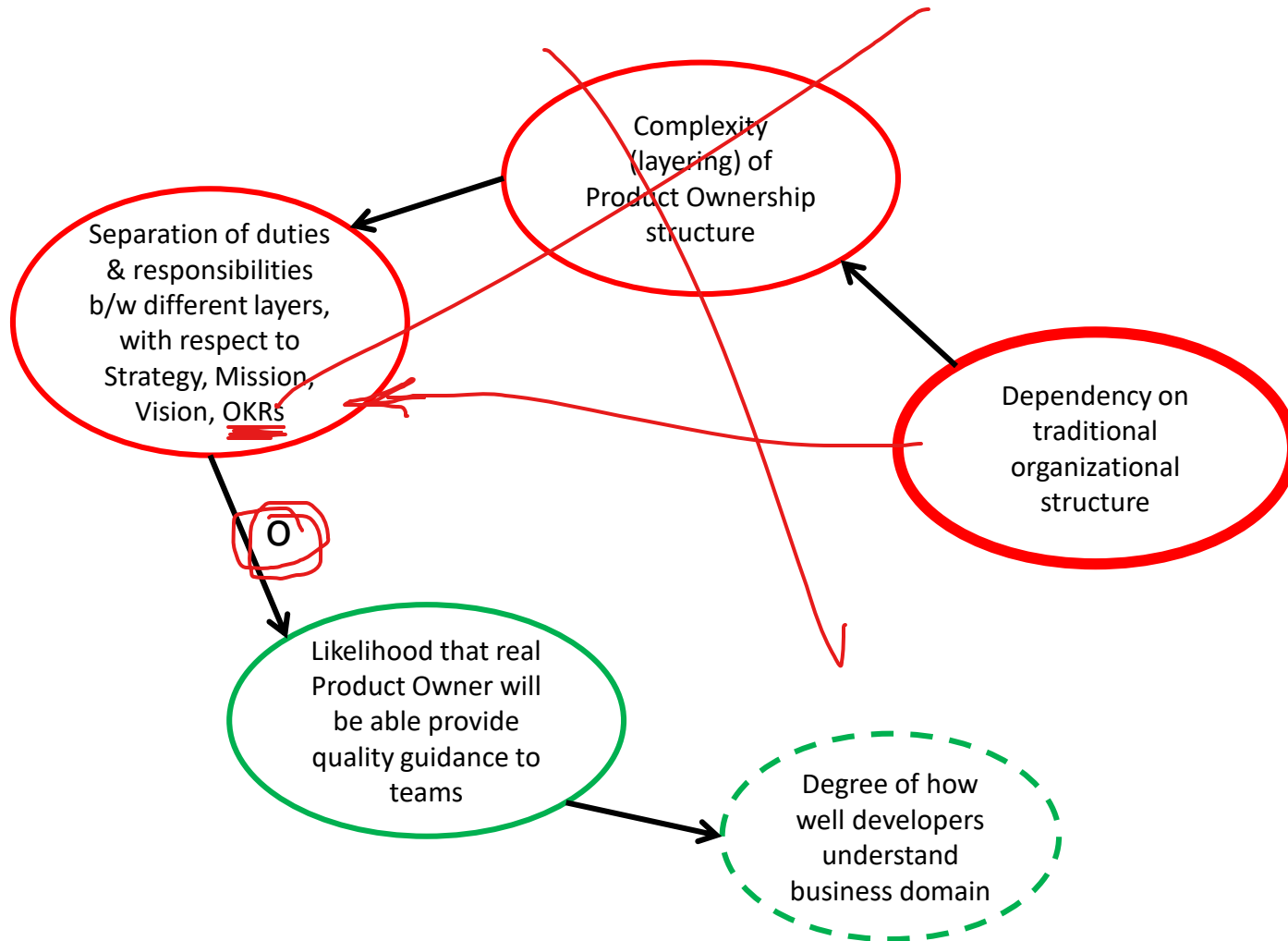


# Local Optimization in **PO-ship Structure**- Exercise





# Local Optimization in **PO-ship Structure**- Exercise









$$8 \times 3 - 5$$



# **LeSS Guides**

## **(Product Backlog)**



# LeSS Guides: *Product Backlog*

*Don't "Manage Dependencies" but Minimize Constraints*

In product development we distinguish between internal dependencies and external dependencies. Internal dependencies are between the teams within a product group, whereas external dependencies are either outside the product group or to nonfeature teams within the product group, such as in the undone department.

There are ~~no internal dependencies~~ and no dependency management with feature teams that use shared code.

Teams can benefit by working together on shared work but wouldn't depend on the output of the other team.

Sourced from: <https://less.works/resources/learning-resources/books>



# LeSS Guides: *Product Backlog*

## *Dealing with Parents*

- Removing Ancestors
- Keeping Ancestors

Removing the ancestor from the Product Backlog is like cell division (cytokinesis); the ancestor is replaced by the new items. Advantages? The first is simplicity: The backlog structure remains simple and no extra effort is required to link a parent with children. A second and subtler advantage:



The new items are naturally or obviously prioritized independently from each other and from their ancestor.

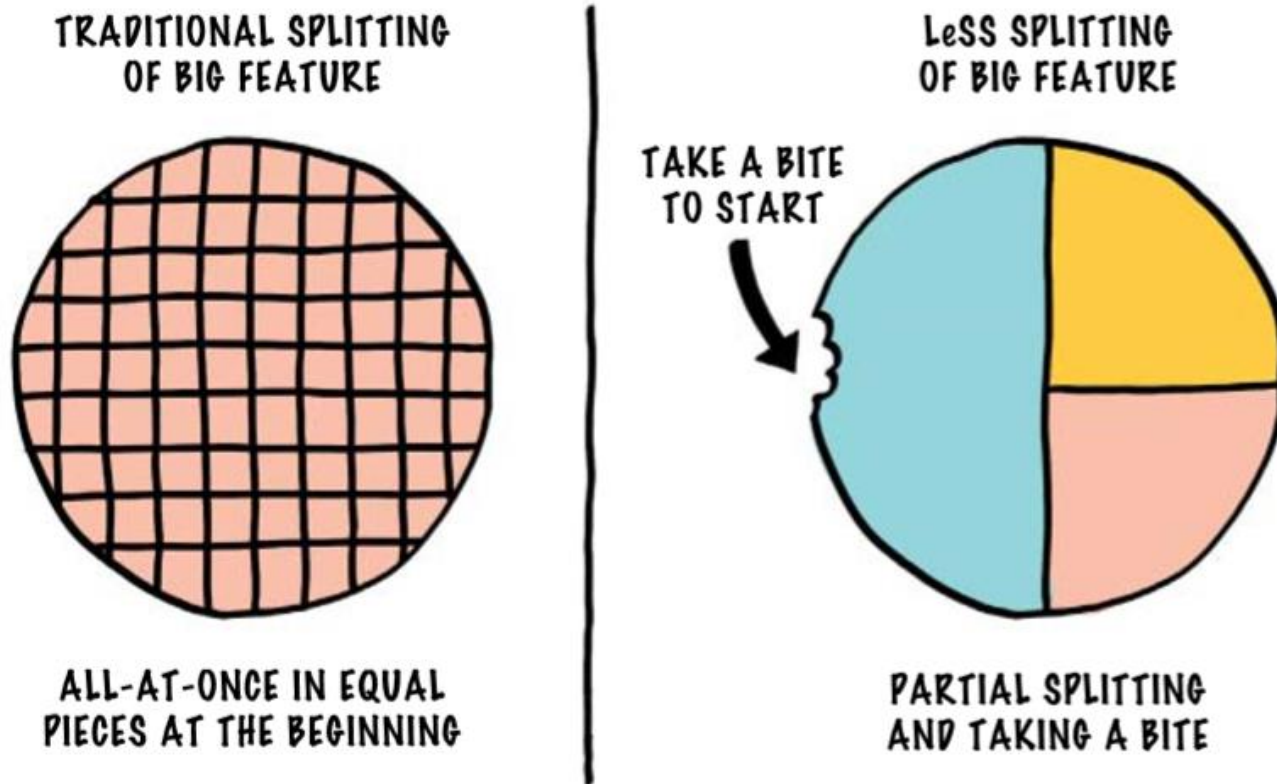
Sourced from: <https://less.works/resources/learning-resources/books>

Order	Item	noteworthy direct/indirect ancestor?
1	settle a buy	settle a trade
2	X	
3	Y	
4	settle a sell	settle a trade



# LeSS Guides: *Product Backlog*

## Take a Bite



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



# LeSS Guides: *Product Backlog*

## *Handling “Special” Items*

Express big improvement items in terms of benefit to the business and Product Owner.

Do NOT create fake “study” items for regular and repeating analysis or design activities such as business or UX analysis, UI design, or architecture analysis or design.

**Beware!**—Don’t create a “special people” group given the charter to go off and figure out that big problem. You’d be better off taking a big pile of money and lighting it on fire. At least you’d get some heat.

Sourced from: <https://less.works/resources/learning-resources/books>



# LeSS Guides: *Product Backlog*

## *Handling “Special” Items*

- **Big bug list** - keep using the defect-tracking tool until the defect count is small enough to use just the Product Backlog
- **Getting to zero** - lean thinking “stop and fix”
- **Urgent new defects** - identify a *regular feature team* as the fast-response team, rotating this responsibility each Sprint.



# LeSS Guides: *Product Backlog*

## *Tools for Large Product Backlogs*

Tools aren't agile. Agility is an organizational behavior.

What Product Backlog tool at scale?  
Use nothing more complicated than a spreadsheet and wiki.

Don't use same tool for Product Backlog and Sprint Backlogs

Sourced from: <https://less.works/resources/learning-resources/books>





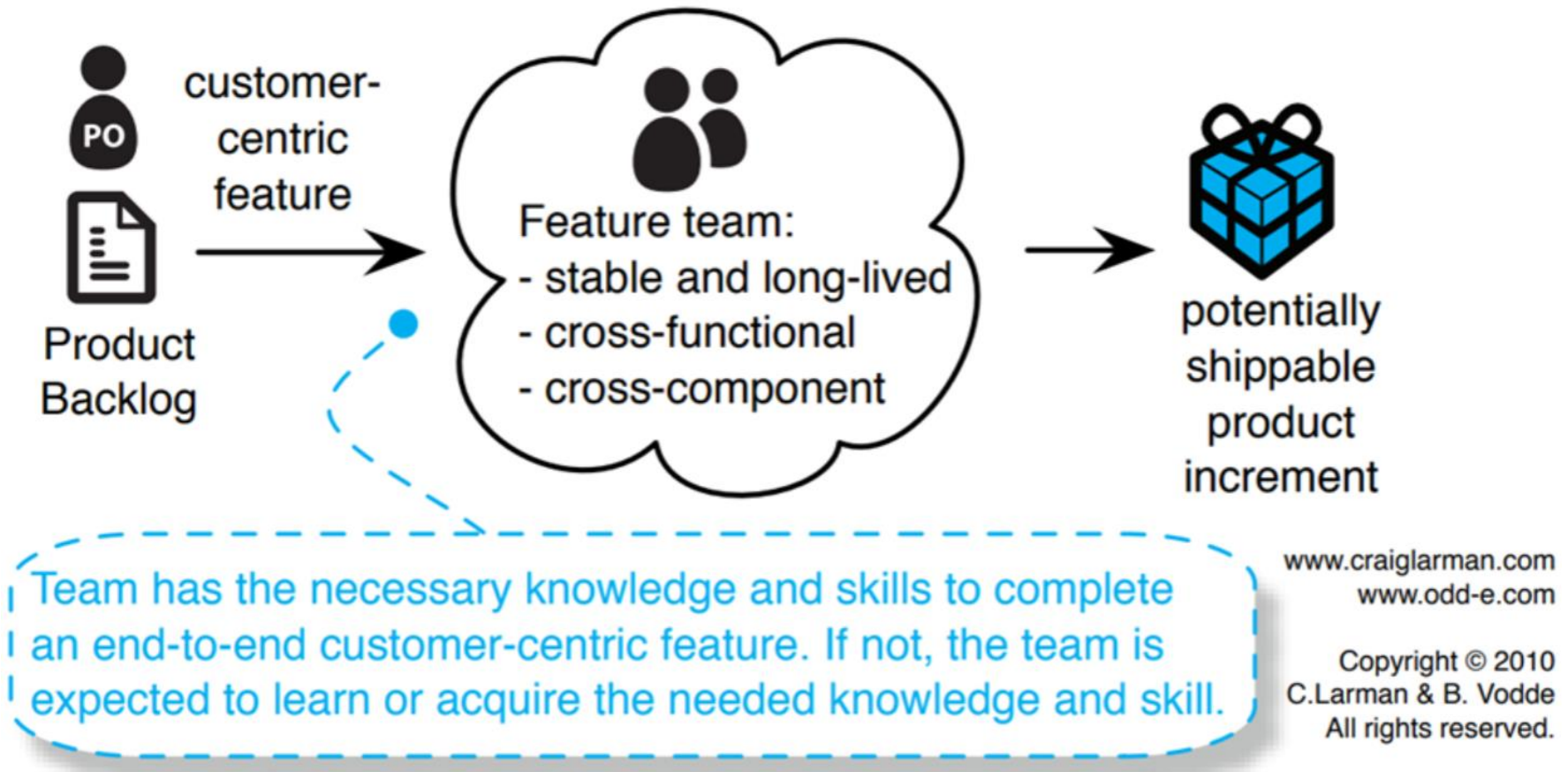


# LeSS Guides

## (Definition of Done)



# LeSS Guides: *Definition of Done*



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



# LeSS Guides: *Definition of Done*

## *Creating the Definition of Done*

One Definition of Done for the whole product  
common for all teams

Each team can have its own stronger Definition of Done by  
expanding the common one.

The perfection goal is to improve the Definition of Done so  
that it results in a shippable product each Sprint  
(or even more frequently).

Sourced from: <https://less.works/resources/learning-resources/books>



# LeSS Guides: *Definition of Done*

## Creating the Definition of Done



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

Sourced from: <https://less.works/resources/learning-resources/books>

UNDERLINED ONES ARE THE DEFINITION OF DONE

Mathematics of Done

Potentially Shippable = Definition of Done + Undone Work

Work in Sprint = Product Backlog Items × Definition of Done



# LeSS Guides: *Definition of Done*

## *Mathematics of Done*

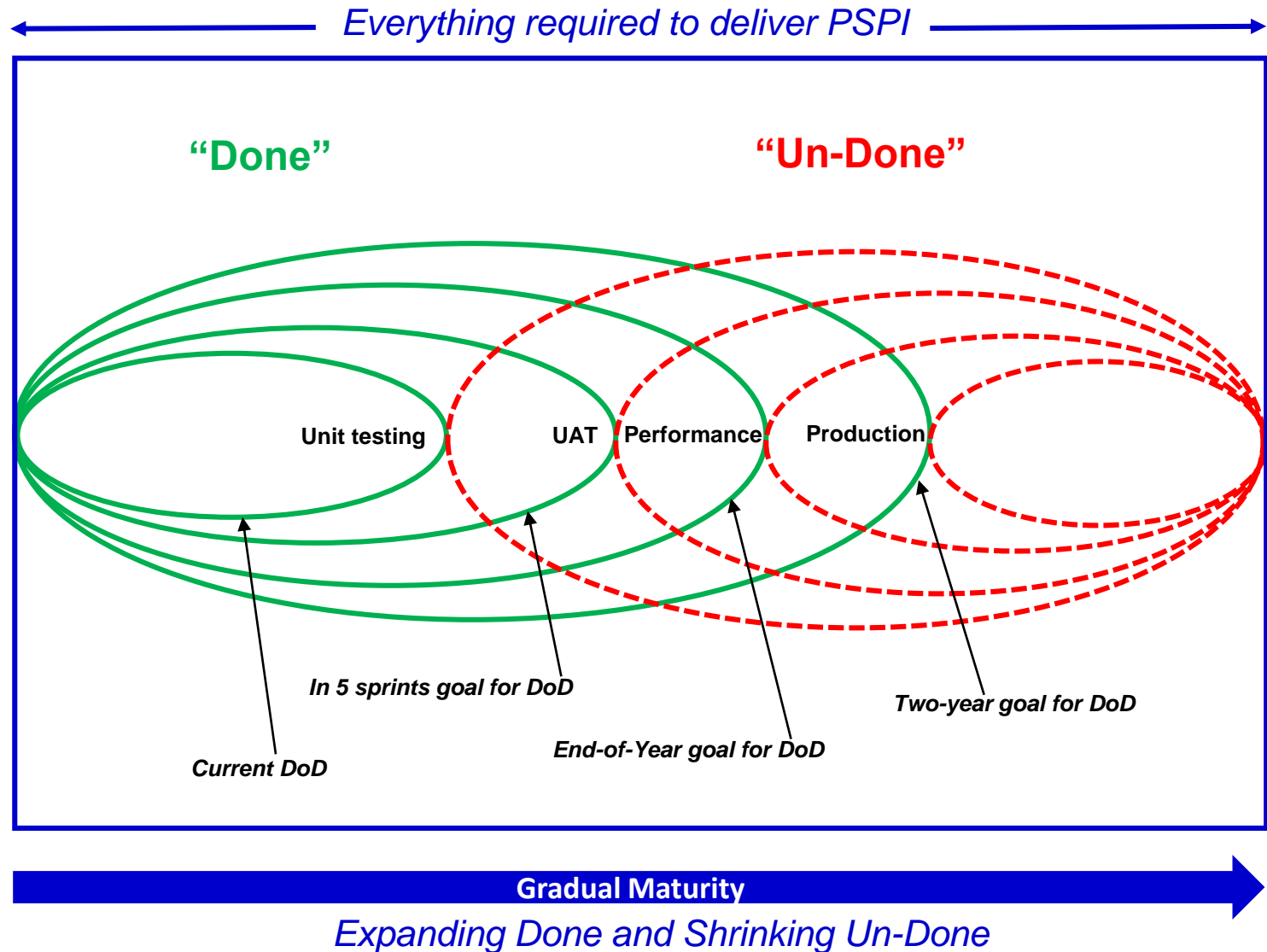
**Unfinished Work != Undone Work**

**Usually, Team's issue**

**Usually, Organizational issue**



# LeSS Guides: *Definition of Done*

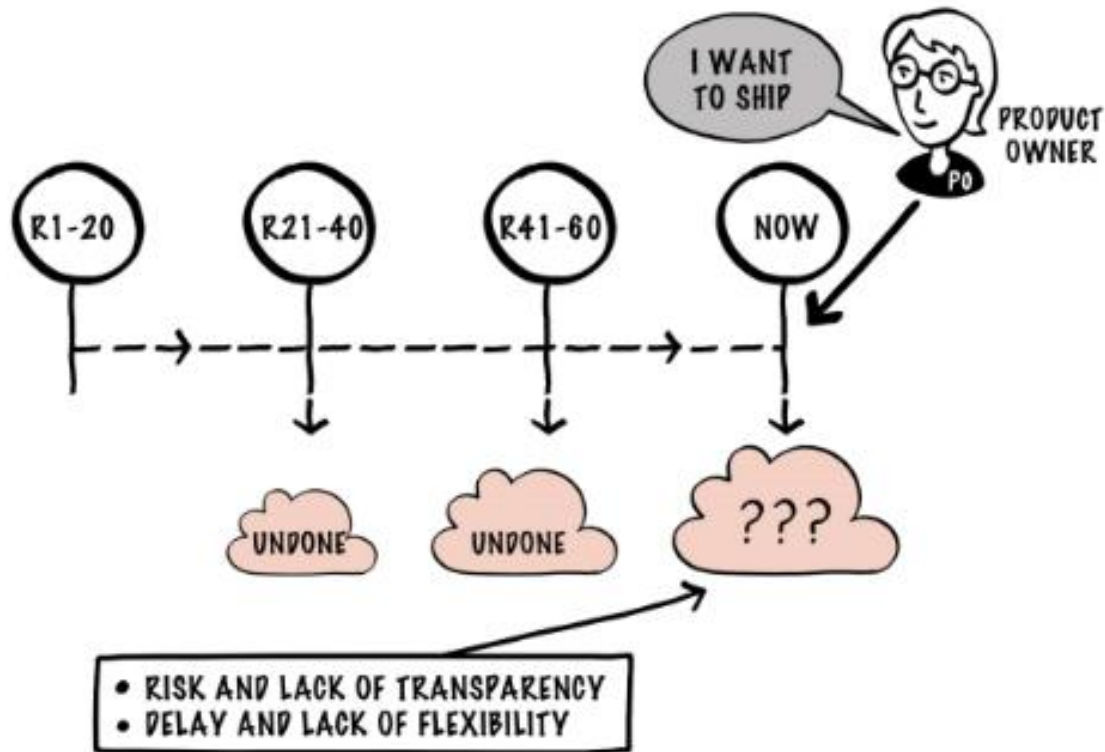


Created by: Gene Gendel



# LeSS Guides: *Definition of Done*

## *Dealing with Un-Done Work*



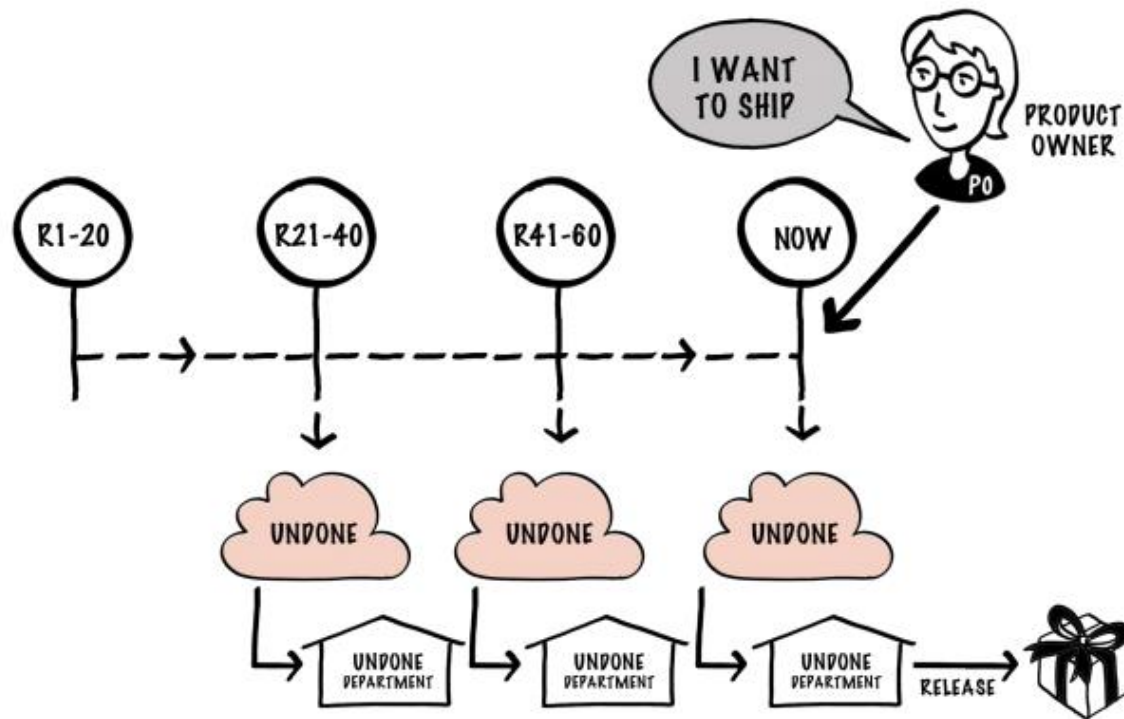
<https://less.works>

## Release Sprints



# LeSS Guides: *Definition of Done*

## *Dealing with Un-Done Work*



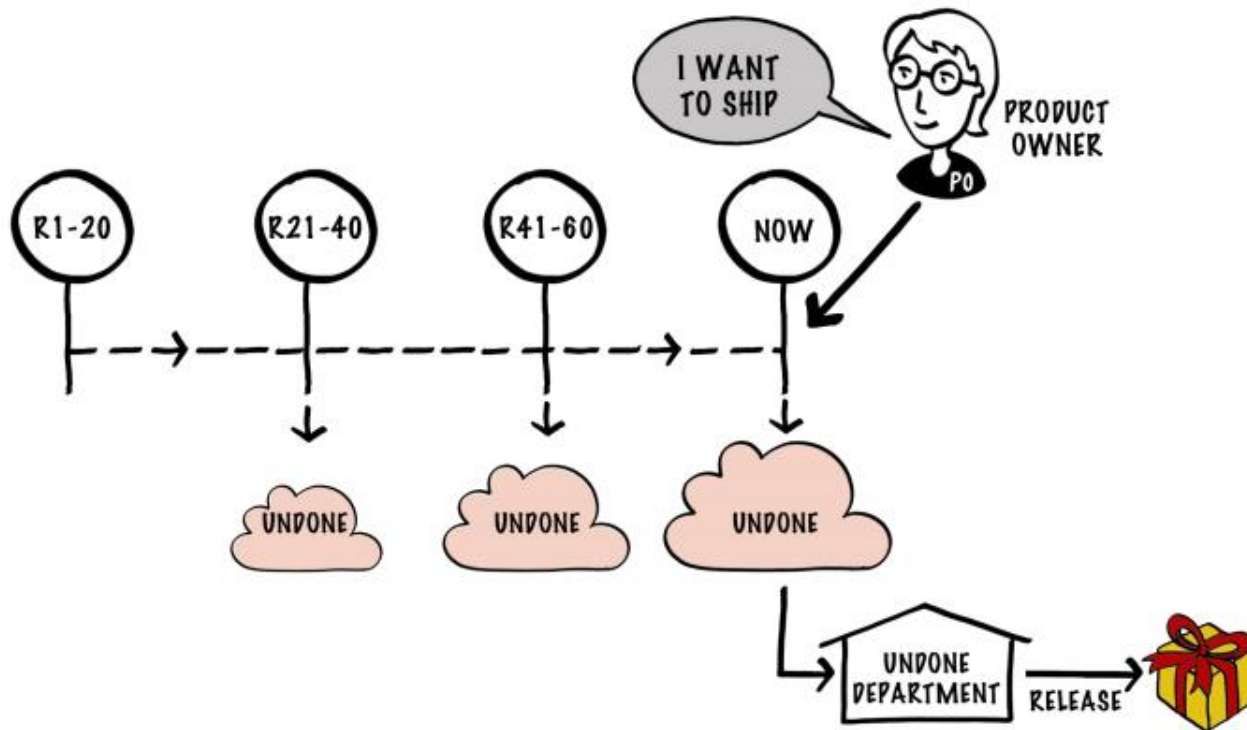
<http://less.works> CC BY-ND

## Pipelining to Undone Department



# LeSS Guides: *Definition of Done*

## *Dealing with Un-Done Work*



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

## Big-Batch Delivery to Undone Department



# LeSS Guides: *Definition of Done*



"There's No Such Thing as a "Devops Team"

Nor should there be "devops specialists" on development teams doing this work: this is core developer work, the same as writing code, and developers need to own it.

...the Devops movement addresses the dysfunction that results from organizations composed of functional silos. Thus, creating another functional silo that sits between dev and ops is clearly a poor (and ironic) way to try and solve these problems."

Jez Humble



**REMOVE  
UNDONE  
DEPARTMENT**



# LeSS Guides: *Definition of Done*

## *Evolve the Definition of Done (Cont.)*

**Managers**—While there is an imperfect Definition of Done, the Definition of Done is the major tool for monitoring and managing organizational change. Expanding the Definition of Done leads to organizational changes and strategic decisions and are usually the responsibility of managers.

**Scrum Masters**—Not expanding the Definition of Done is a sign of not improving. Scrum Masters are responsible for building teams that are self-managing and continuously improving, and Scrum Masters are responsible for helping the organization to improve.

Sourced from: <https://less.works/resources/learning-resources/books>



# Class Activity

**Class:** Review instructions of a system modelling exercise on the next slide.

**Instructor:** Provide instructions

**Duration:** next page





## Exercise

**Duration:** 10 min

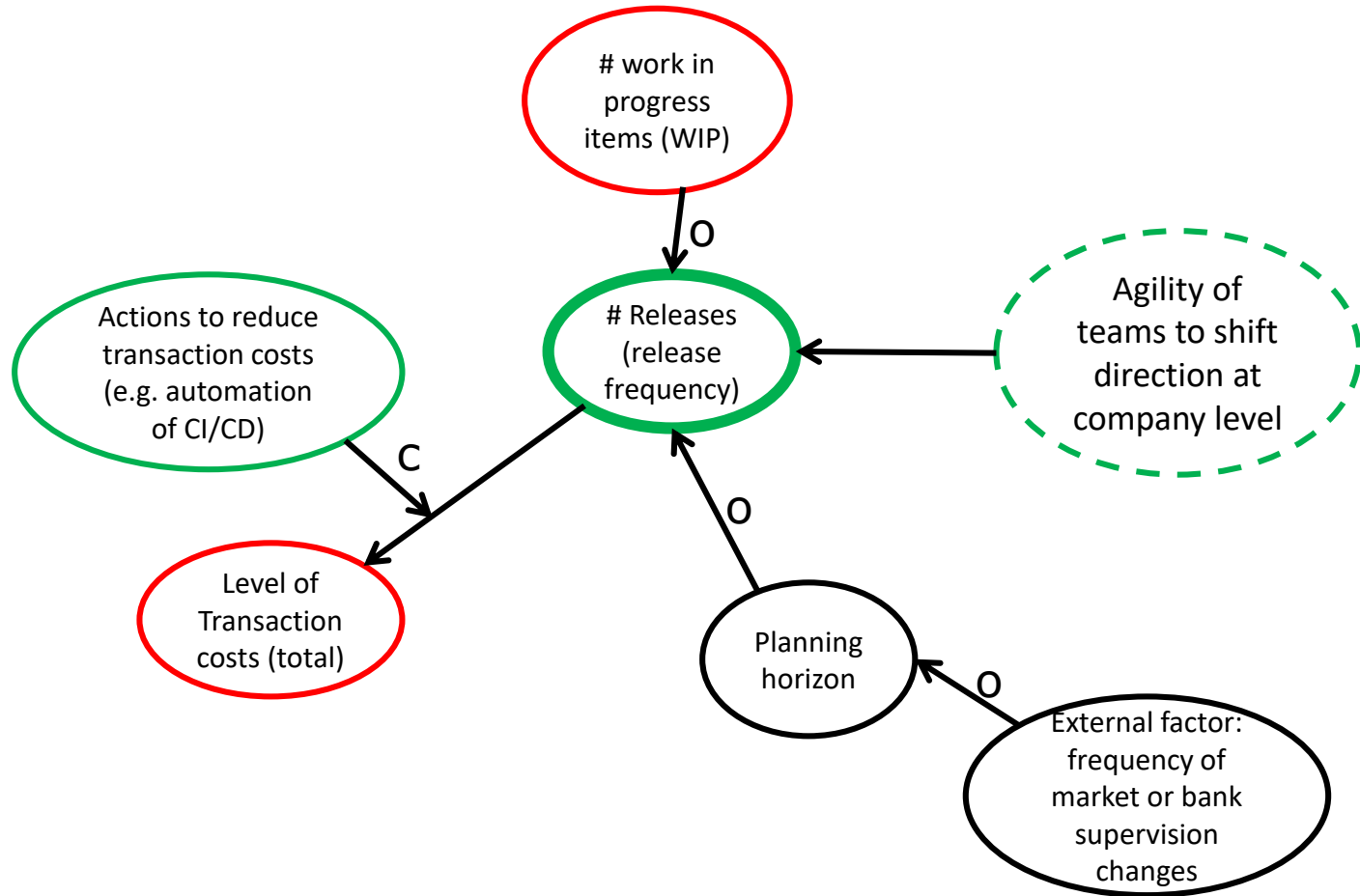
**Class:** in-groups, brainstorm some of the most common examples of ***Local Optimization in Releasing*** in your respective organizations. Work with provided system variables to create a model.

**Method/Tool:** System Modeling with CLD

**Instructor:** review with Class

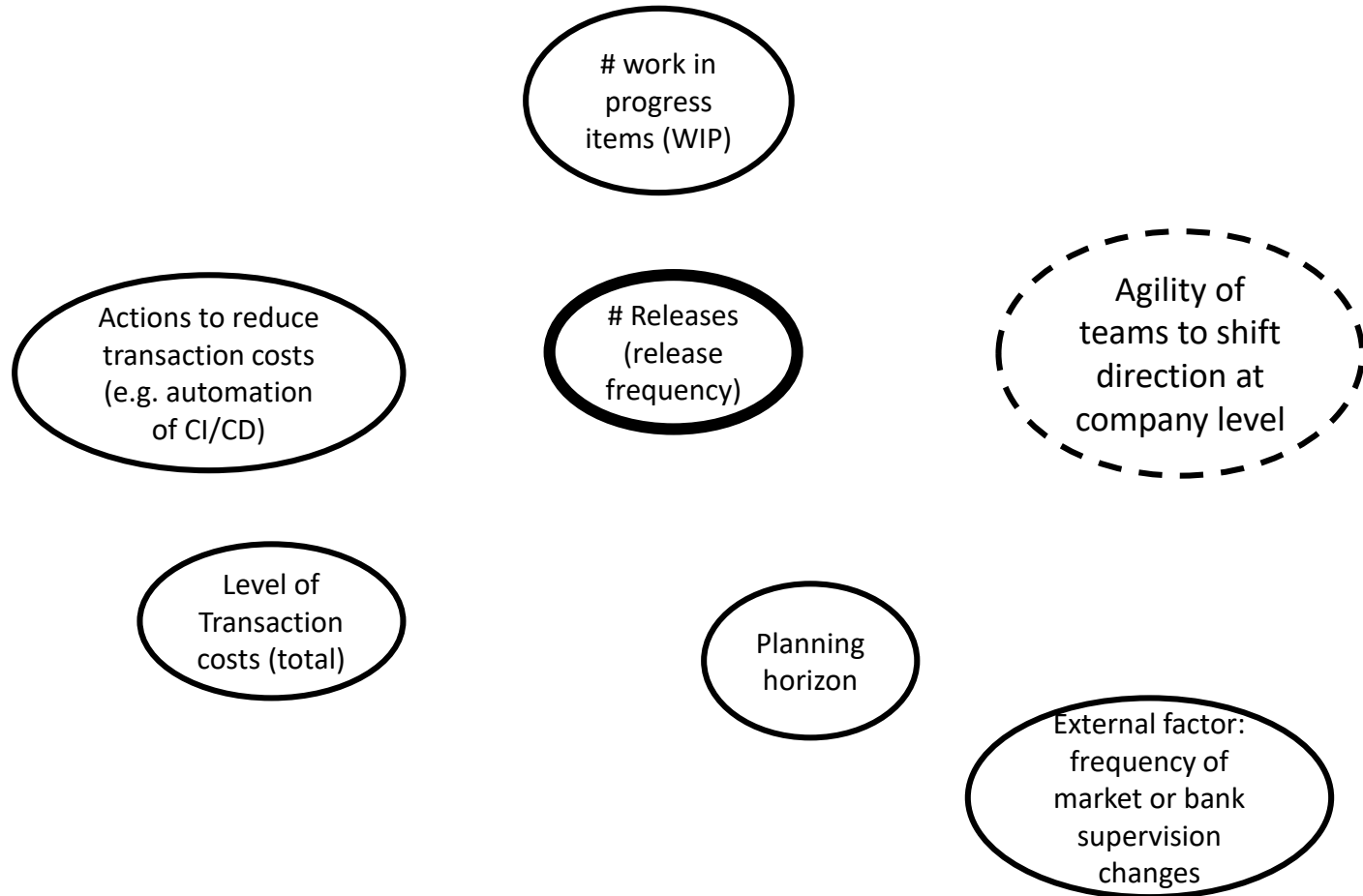


# Local Optimization in **Releasing** - Exercise





# Local Optimization in **Releasing** - Exercise









# **LeSS Guides**

## **(Product Backlog Refinement)**



# LeSS Guides: *Product Backlog Refinement*

## *Product Backlog Refinement Types*

[PBR] is an ongoing process in which the Product Owner and the Development Team collaborate on the details of Product Backlog items. ... Scrum recognizes no sub-teams in the Development Team, regardless of particular domains that need to be addressed like testing or business analysis; there are no exceptions to this rule.

Product Backlog refinement is done per team for the items they are *likely* going to implement in the future. Do multi-team PBR to increase shared understanding and exploit coordination opportunities when having closely related items or a need for broader input/learning.

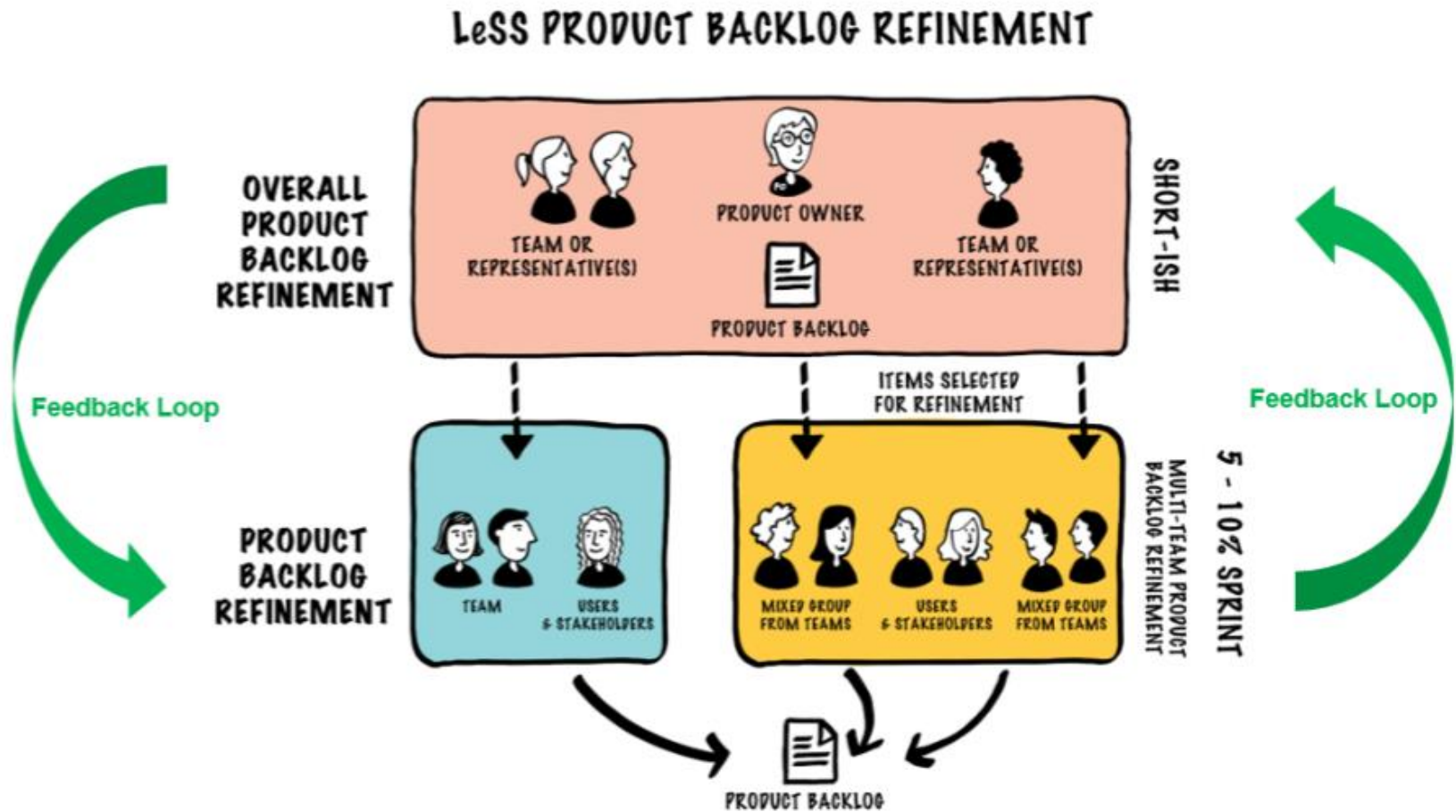
The Product Owner shouldn't work alone on Product Backlog refinement; she is supported by the multiple Teams working directly with customers/users and other stakeholders.

All prioritization goes through the Product Owner, but clarification is as much as possible directly between the Teams and customer/users and other stakeholders.

Sourced from: <https://less.works/resources/learning-resources/books>



# LeSS Guides: *Product Backlog Refinement*




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

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



# LeSS Guides: *Product Backlog Refinement*

## *Product Backlog Refinement: Duration For Each PBR 'Type'*



	Overall PBR	Multi-team PBR	Single-team PBR	Initial PBR
members from	all teams	2+ teams	1 team	all teams
includes Product Owner?	definitely	depends	rarely	definitely
includes customers/users?	rarely	probably	probably	definitely
select which teams work on which items?	yes (prefer <i>set of items with group of teams</i> )	no	done already	no
level of clarification	lightweight	in-depth	in-depth	in-depth
length	shortish	0.5–1 day	0.5–1 day	at least 2 days
typical frequency	every Sprint	most Sprints	most Sprints	once

Sourced from: <https://less.works/resources/learning-resources/books>



# LeSS Guides: *Product Backlog Refinement*

## Logical Splitting of Work

Use cases	the major work flows or use cases	Configuration	varying configuration such as OS or browser
Scenario	a specific sequence of steps (acceptance tests)	I/O channel	different interfaces, such as GUI or command line
Data part	subset of the data elements	Data format	XML...
Type	Varying types of kinds of things	Role or persona	e.g novice or power user
Risk	a part is understood, a part requires learning	Non-functional	e.g. moderate vs high throughput
CRUD	create/retrieve/update/delete	Operation	system operation such as HTTP GET
Integration	integration between (non) existing elements	Stub	working with a fake first

Sourced from creative commons repository of Certified LeSS Trainers: <https://less.works/courses/become-less-trainer>

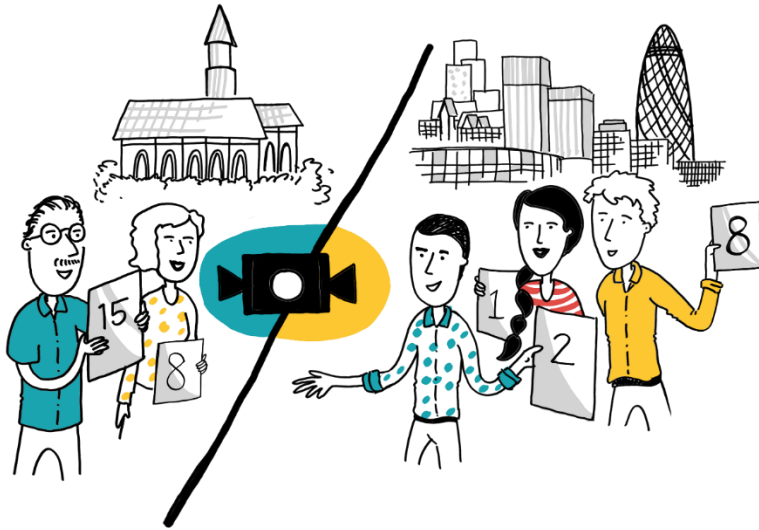
Split into thin end-to-end “vertical” requirements.

**DO NOT SPLIT ITEMS INTO INTERNAL DESIGN STEPS!**



# LeSS Guides: *Product Backlog Refinement*

## Scaling – Multi Site Estimation



<http://less.works> BRAND

Estimates don't need to be "accurate" or "precise"; they need to be *useful*, and usefulness depends on purpose.

Choose the simplest technique that matches the purpose and that fosters discussion and learning.

- Physical cards
- Instant Messenger

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



# LeSS Guides: *Product Backlog Refinement*

## *Scaling – Multi Site Estimation*

*"library of done items"*

**Calibrate against done items**—One simple approach is for teams to calibrate or compare against a set of already done items in the Product Backlog. For this to work well, there should be many such items, to increase the chance that some or many will be familiar to people.

**Synchronize in multi-team or overall PBR**—When two or more teams do PBR together and do estimation with points together, there's alignment across teams for a common meaning of relative points. Similarly, in overall PBR (with a couple of representatives from all teams), when estimation is done together, then points are synchronized.

Sourced from: <https://less.works/resources/learning-resources/books>

*normalizing a scale*



# LeSS Guides: *Product Backlog Refinement*

*Answering a BIG question....*

**What is so special  
about estimation  
in LeSS?**



# Class Activity

**Class:** Review instructions of a system modelling exercise on the next slide.

**Instructor:** Provide instructions

**Duration:** next page





## Exercise

**Duration:** 10 min

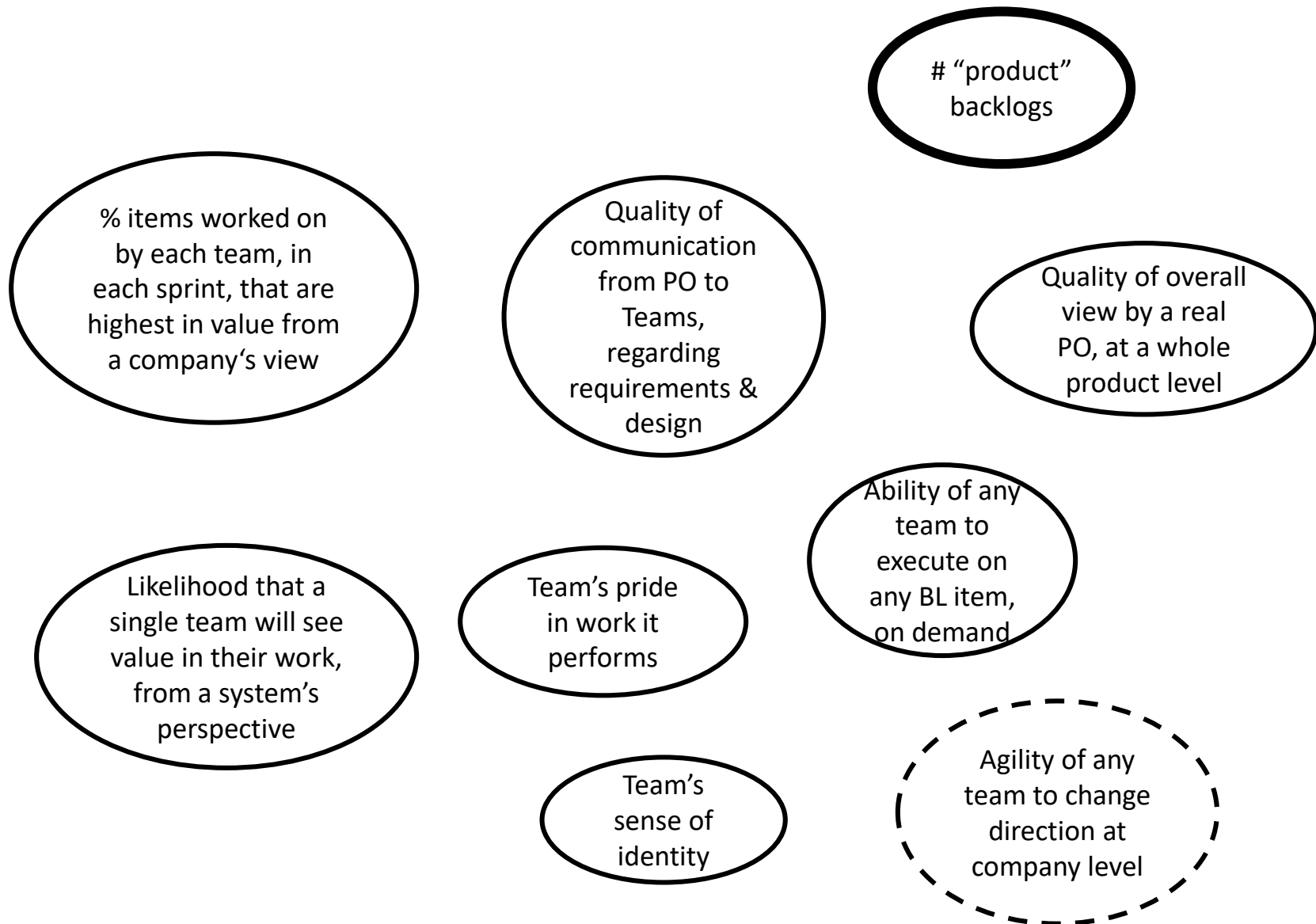
**Class:** in-groups, brainstorm some of the most common examples of ***Local Optimization in Product Backlogs*** in your respective organizations. Work with provided system variables to create a model.

**Method/Tool:** System Modeling with CLD

**Instructor:** review with Class



# Local Optimization in **Product Backlog** - Exercise





$$5 \times 5 - 4$$

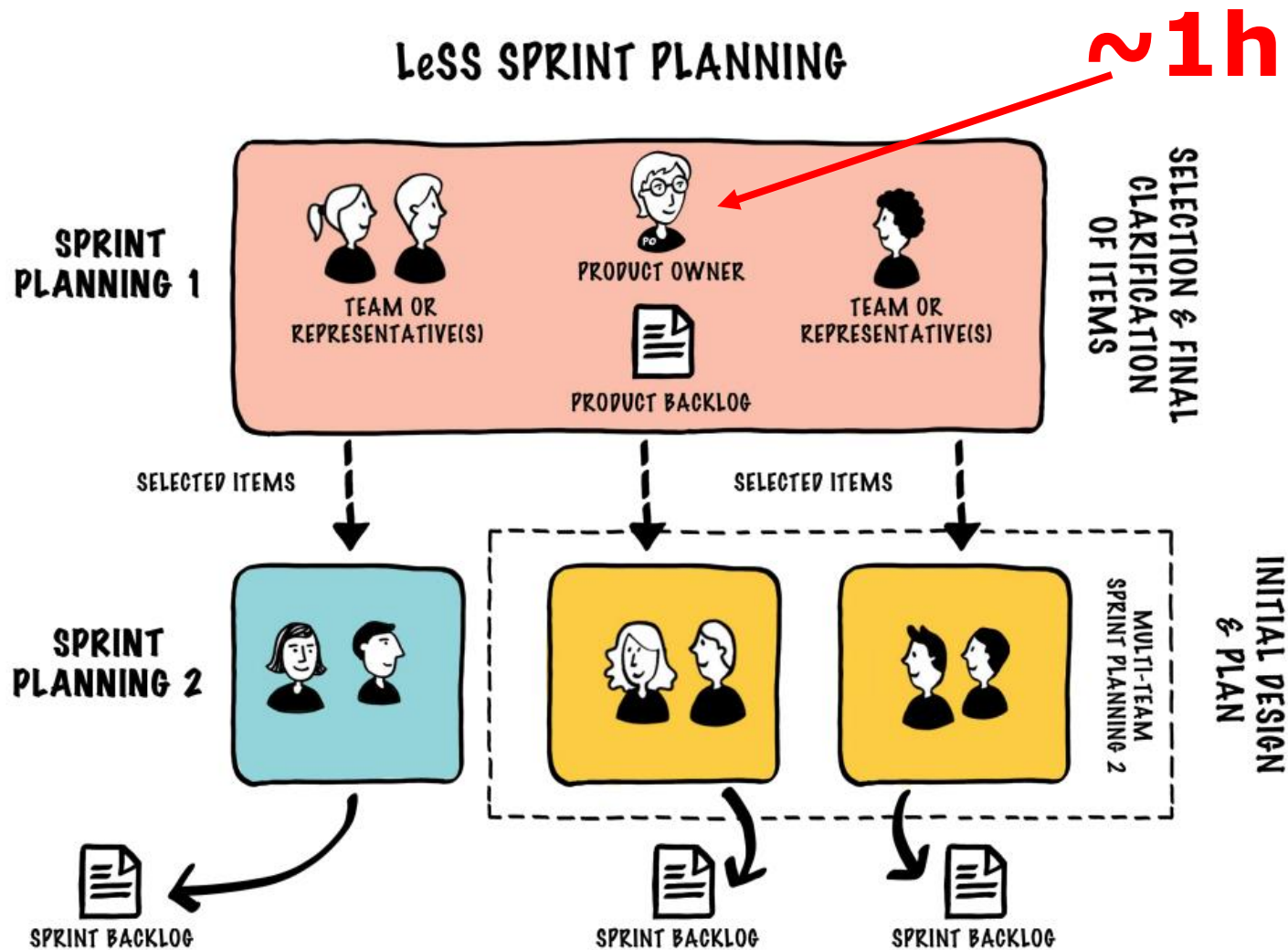


# **LeSS Guides**

## **(Sprint Planning)**



# LeSS Guides: *Sprint Planning*



Sourced from: <https://less.works/resources/learning-resources/books>

<http://less.works>



# LeSS Guides: *Sprint Planning*

*Physical and Light Digital Visual Management over Heavy Tooling*



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

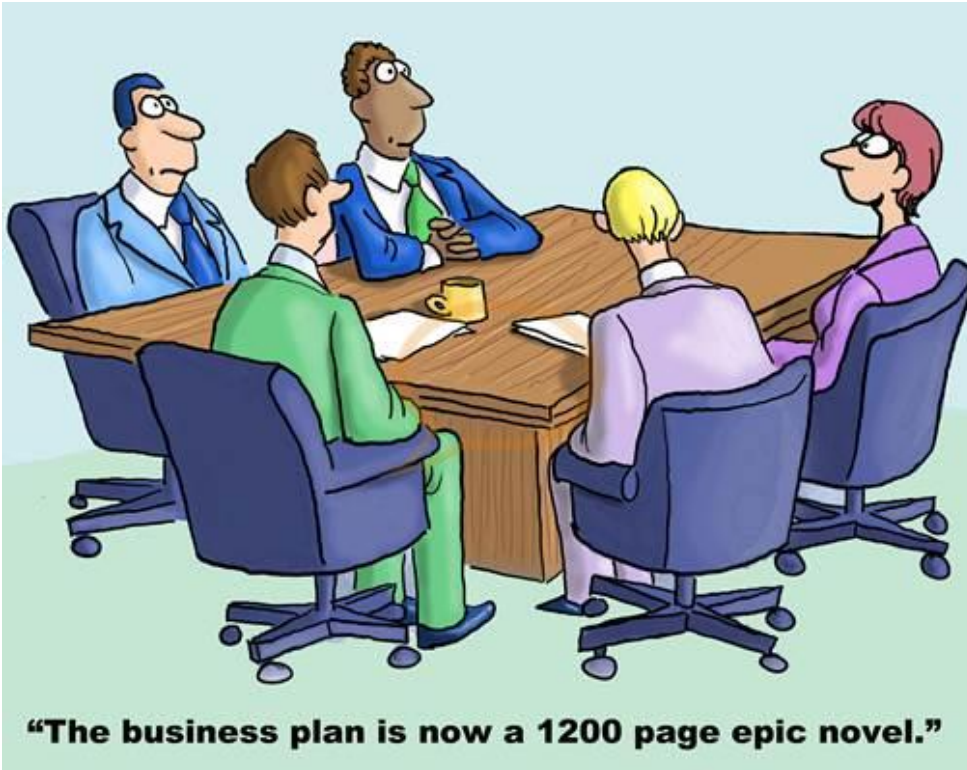
<http://less.works> 

Don't use any software tool for Sprint Backlogs;  
just use physical visual management, probably cards on a wall.



# LeSS Guides: *Sprint Planning*

*Physical and Light Digital Visual Management over Heavy Tooling*



"Let's train our PMO and BAs to use JIRA/Rally/Version1/VSTS and manage our projects in there"

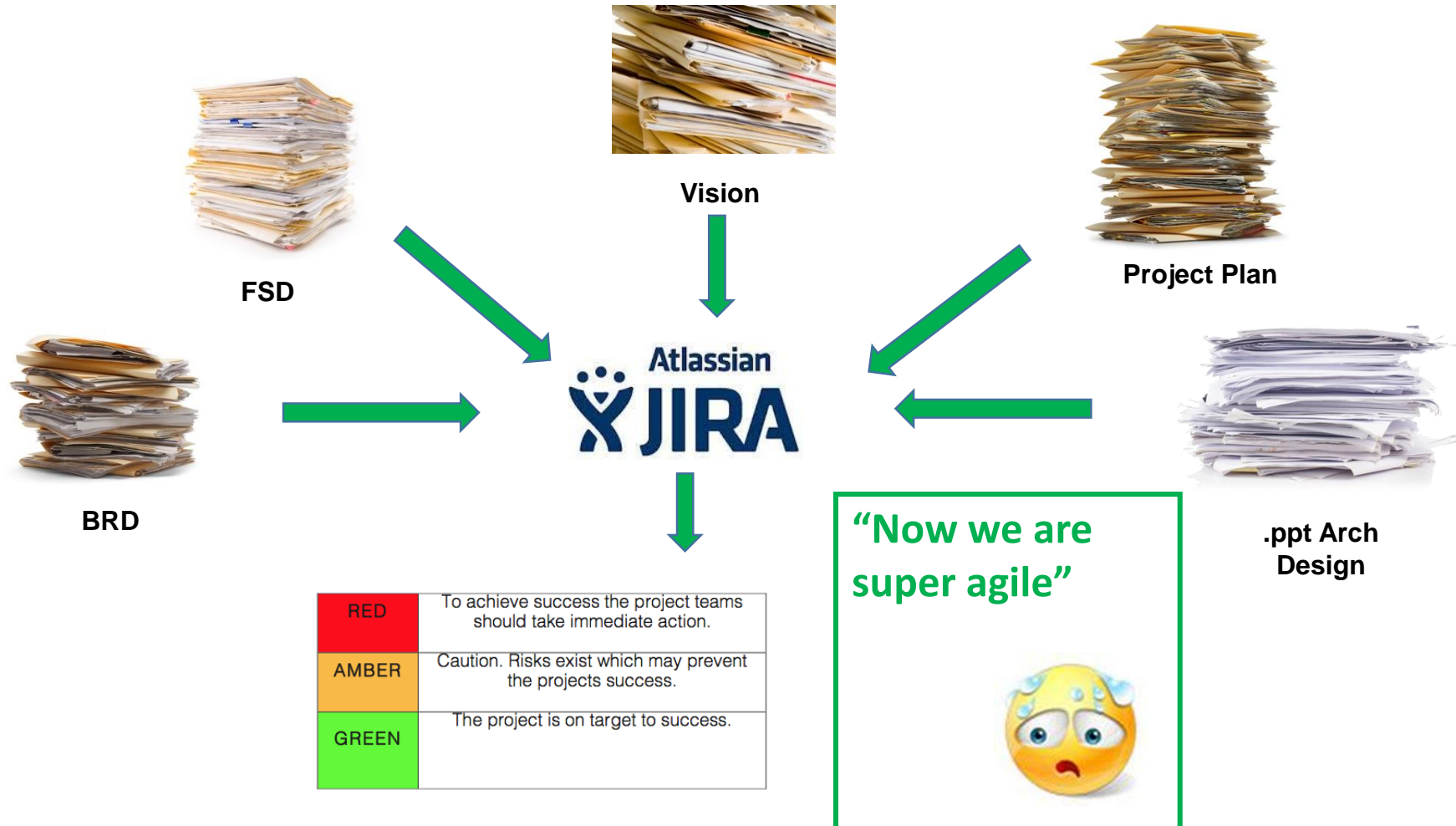


Sourced from creative commons repository of Certified LeSS Trainers: <https://less.works/courses/become-less-trainer>



# LeSS Guides: *Sprint Planning*

*Physical and Light Digital Visual Management over Heavy Tooling*



Sourced from creative commons repository of Certified LeSS Trainers: <https://less.works/courses/become-less-trainer>







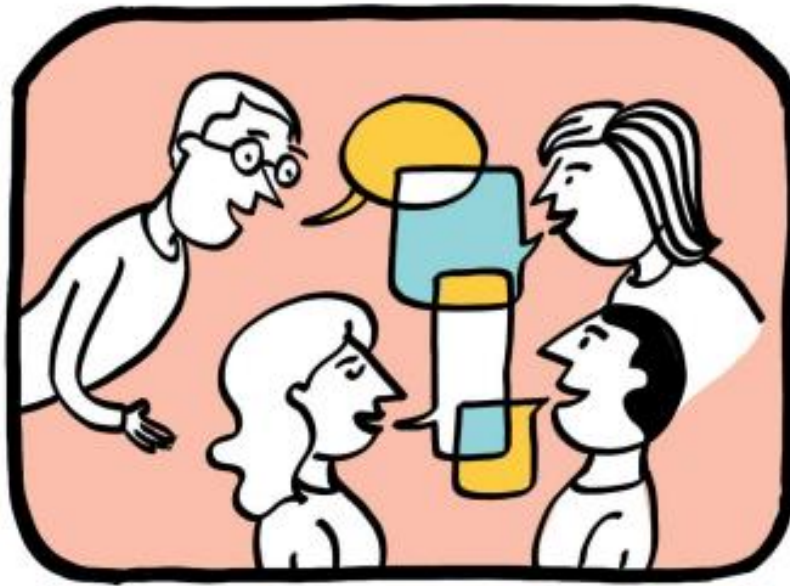
# LeSS Guides

## (Coordination & Integration)



# LeSS Guides: *Coordination & Integration*

*Just Talk in Coordination-Friendly Environment*



**JUST TALK**

<http://less.works>

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

Promote bottom-up *emergent behaviors* for coordination. Decentralized techniques support this.



# LeSS Guides: *Coordination & Integration*

## *Communicate in Code*



**COMMUNICATE IN CODE**

© 2014 by Gene Gendel

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



# LeSS Guides: *Coordination & Integration*

## *Communicate in Code*

Traditionally *coordination supported integration*,  
but we can also have *integration support coordination*.

When you branch, not only are you delaying integration, you  
are impeding coordination and cooperation between teams.

Continuous integration is...  
a developer behavior to keep a working system by small  
changes growing the system by integrating very frequently on  
a “mainline” supported by a CI system with automated tests.

Sourced from: <https://less.works/resources/learning-resources/books>



# LeSS Guides: *Coordination & Integration*

## *Communities*



## COMMUNITIES

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

Communities cannot make decisions for the teams, but they can produce something that the teams *decide* to adopt.



# LeSS Guides: *Coordination & Integration*

## *Communities*

- Have a **community coordinator** with passion for the concern and desire to cultivate a strong community that cares; preferably someone who is an active **hands-on** practitioner
- Actively try to **recruit participation** from most teams
- are visible and easily discoverable so that everyone knows the current communities and knows how to join
- Preferably focus on **concrete problem-solving goals**— they make learning practical and concrete
- Have agreed how they work and make decisions
- Might have a **Scrum Master** who helps them work and improve, and who facilitates community meetings or workshops
- Use wiki, discussion groups, group chat meet regularly
- Are strongly encouraged within the organization; everyone knows that it is OK and indeed expected for them to join communities and spend effort in community activities

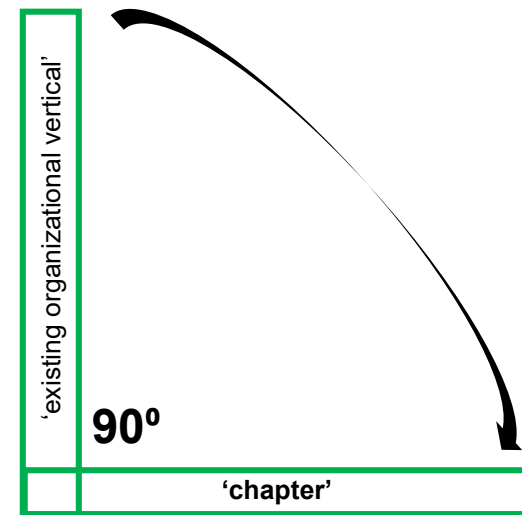


# LeSS Guides: *Coordination & Integration*

## *Avoid Fake Communities*

- Forgo a community coordinator or the coordinator doesn't care (this often happens if the person was assigned)
- Hold frequent meetings just for the sake of meeting
- Have mostly members that are not in feature teams
- Regard the community as secondary so that participation is downgraded because "we're too busy to participate."

**Fake communities**—Traditional large groups are structured as single-function teams, such as architecture, test, and so forth. Most organizations are implicitly optimized to avoid changing the status quo manager and specialist positions and power structures. Consequently, we've seen *fake communities* that are just the old single-function teams relabeled



Created by: Gene Gendel



# LeSS Guides: *Coordination & Integration*

## *Avoid Fake Communities*

### Chapters, Chapters of Chapters, Guilds ...etc... etc... etc...

Try thinking of them as COMMUNITIES that are used for functional learning and “spreading knowledge laterally”. Without hierarchy. Based on volunteering. And then, try simplifying your naming convention and just calling these structures, what they really are: Communities 😊.

Otherwise, you may end up with *“used-to-be a vertical reporting structure, flipped on its side and re-branded with fancy Spotify terminology”*, without changing much.

Read more about Communities at: <https://less.works/less/structure/communities> and study in depth how they can be Requirement Area (RA)–specific or span across multiple RAs.

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



# LeSS Guides: *Coordination & Integration*

## *Multi-Team Design Workshop*



<http://less.works> 

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



# LeSS Guides: Coordination & Integration

## Current-Architecture Workshop



<http://less.works> 

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



# LeSS Guides: *Coordination & Integration*

## *Component Mentors*



**COMPONENT MENTOR**

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



# LeSS Guides: *Coordination & Integration*

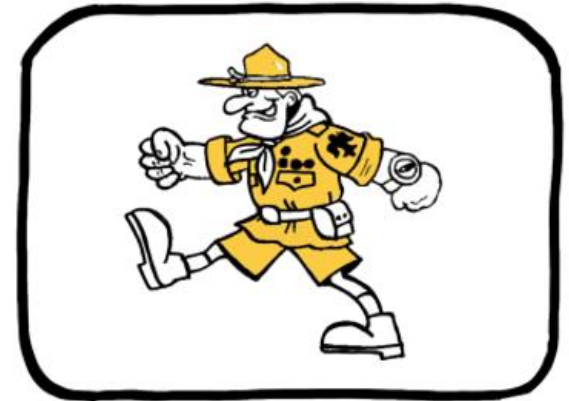
## *Open Space, Travelers, Scouts*



OPEN SPACE



TRAVELER



SCOUT

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

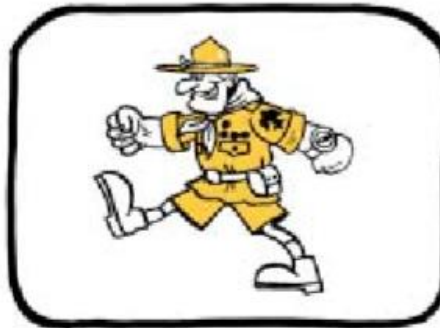


# LeSS Guides: *Coordination & Integration*

## *Mix and Match Techniques*



JUST TALK



SCOUT



OPEN SPACE



TRAVELER



COMMUNITIES



COMPONENT MENTOR

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



# LeSS Guides: Coordination & Integration

## *Maybe, Don't Do Scrum Of Scrums*

A Scrum of Scrums meeting is a Daily-Scrum-like meeting between team representatives—not Scrum Masters or managers—that's commonly held three times per week.

Scrum of Scrums is a formal centralized meeting and *thus not preferred*.

-6-

BAs – “PO Proxies” – Team Output Owners,

Borrowed resources – capacity donors.

Scrum, Scrum of Scrums and Scrum's fractal design,

Oh God!, someone get me a bottle of wine!

<https://www.keystepstosuccess.com/2020/09/agile-poetry-in-motion/>

## AGILE POETRY IN MOTION

🕒 SEPTEMBER 3, 2020 👤 GENE 💬 2 COMMENTS ✎ EDIT

Lyrics written by: [Gene Gendel](#) | Musical/voice performance by: [Erin Perry](#)









# **LeSS Guides**

## **(Review & Retrospective)**



# LeSS Guides: Review & Retrospective

## *Adapt the Product Early and Often*

There is one product Sprint Review; it's common for all teams.  
Ensure that suitable stakeholders join to contribute the information needed for effective inspection and adaptation.

Each Team has its own Sprint Retrospective.

An Overall Retrospective is held after Team Retrospectives to discuss cross-team and systemwide issues and to create improvement experiments. Attended by Product Owner, Scrum Masters, Team representatives, and managers (if any).

Sourced from: <https://less.works/resources/learning-resources/books>



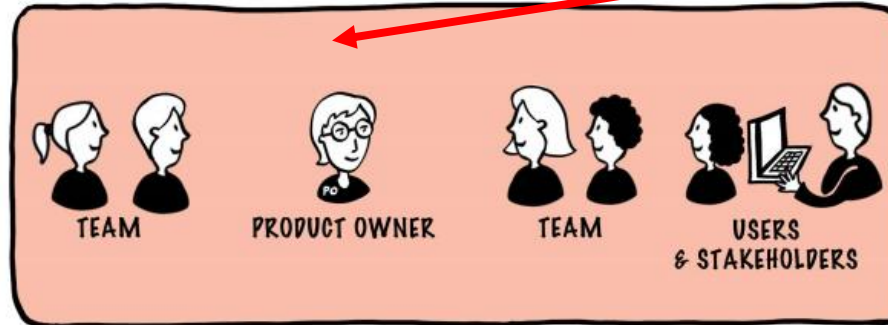
# LeSS Guides: Review & Retrospective

*Adapt the Product Early and Often*

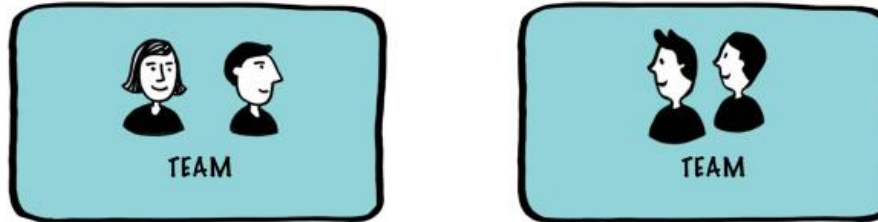
## LeSS SPRINT REVIEW & RETROSPECTIVE

~2h

### SPRINT REVIEW



### TEAM RETROSPECTIVE



### OVERALL RETROSPECTIVE



~1.5h

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

<http://less.works>



# LeSS Guides: *Review & Retrospective*

## *Review Bazaar*

A Sprint Review *bazaar* is analogous to a science fair: A large room has multiple areas, each staffed by team representatives, where the items developed are explored and discussed together with users, teams, etc.



Sourced from creative commons repository of Certified LeSS Trainers: <https://less.works/courses/become-less-trainer>



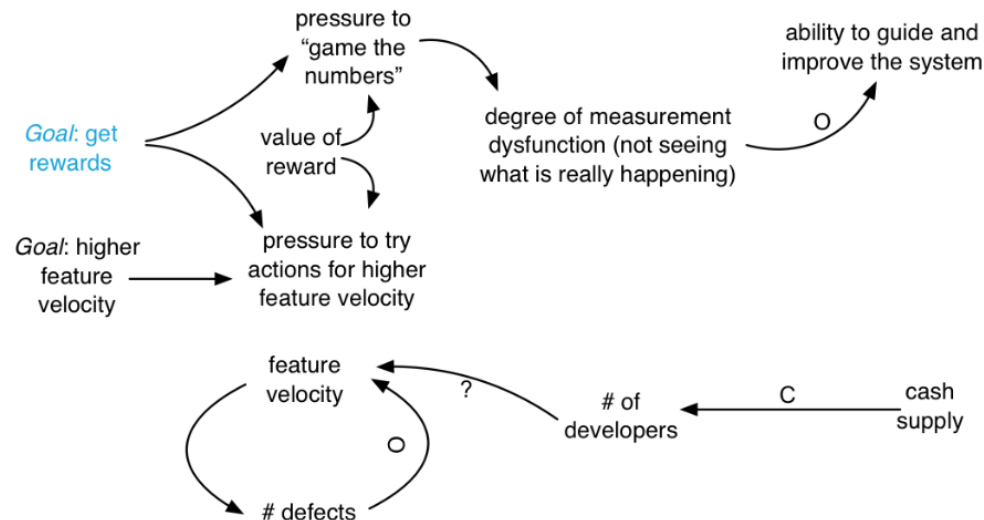
# LeSS Guides: *Review & Retrospective*

## *Improve the System*

Understand & improve the system by focusing on the system.

The first step of **systems thinking** is “simply” recognizing that there is a *whole system*, with elements that *influence* one another within a whole. These influences can have *delays*, create *reinforcing cycles*, and have *unintended* or *hidden* consequences, with a *cascade* of new influences.

Sourced from: <https://less.works/resources/learning-resources/books>



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



# LeSS Guides: *Review & Retrospective*

## *Improve the System*

### Legend for Causal Loop Diagrams

- ❖ **Goals** — 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 related variables
- ❖ **Opposite effects** — “o” annotation near an arrow; suggests that the effect of one variable on another is negative (“-” feedback loop)
- ❖ **Delayed effect** — “||” annotation that disrupts a causal link (arrow); it implies that there is a delayed effect of one variable on 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 short-term, lower-cost effect into a relationship between two related variables

### *In System Modelling, we mostly care about Causations*

- ❖ **Correlation** - relationship between two system variables (event or action), where there is no direct dependency. A relationship can be positive or negative.
- ❖ **Causation** - a.k.a. “cause and effect”, is relationship between two system variables (event or action), where change in one leads to change in another.

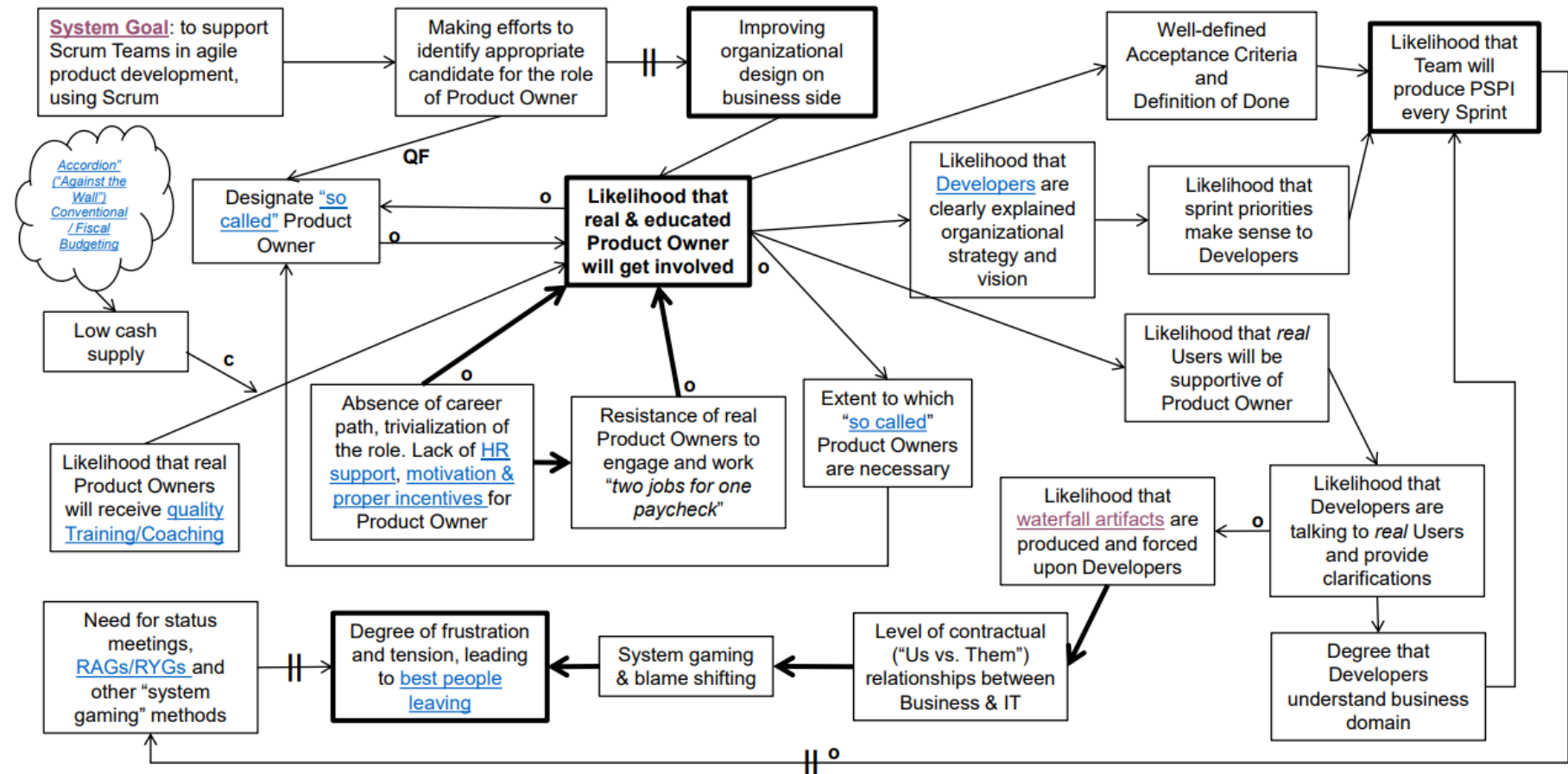
Sourced from: <https://www.keystepstosuccess.com/coach-tools/>



# LeSS Guides: *Review & Retrospective*

## Improve the System

### System Modeling for Product Owner Role



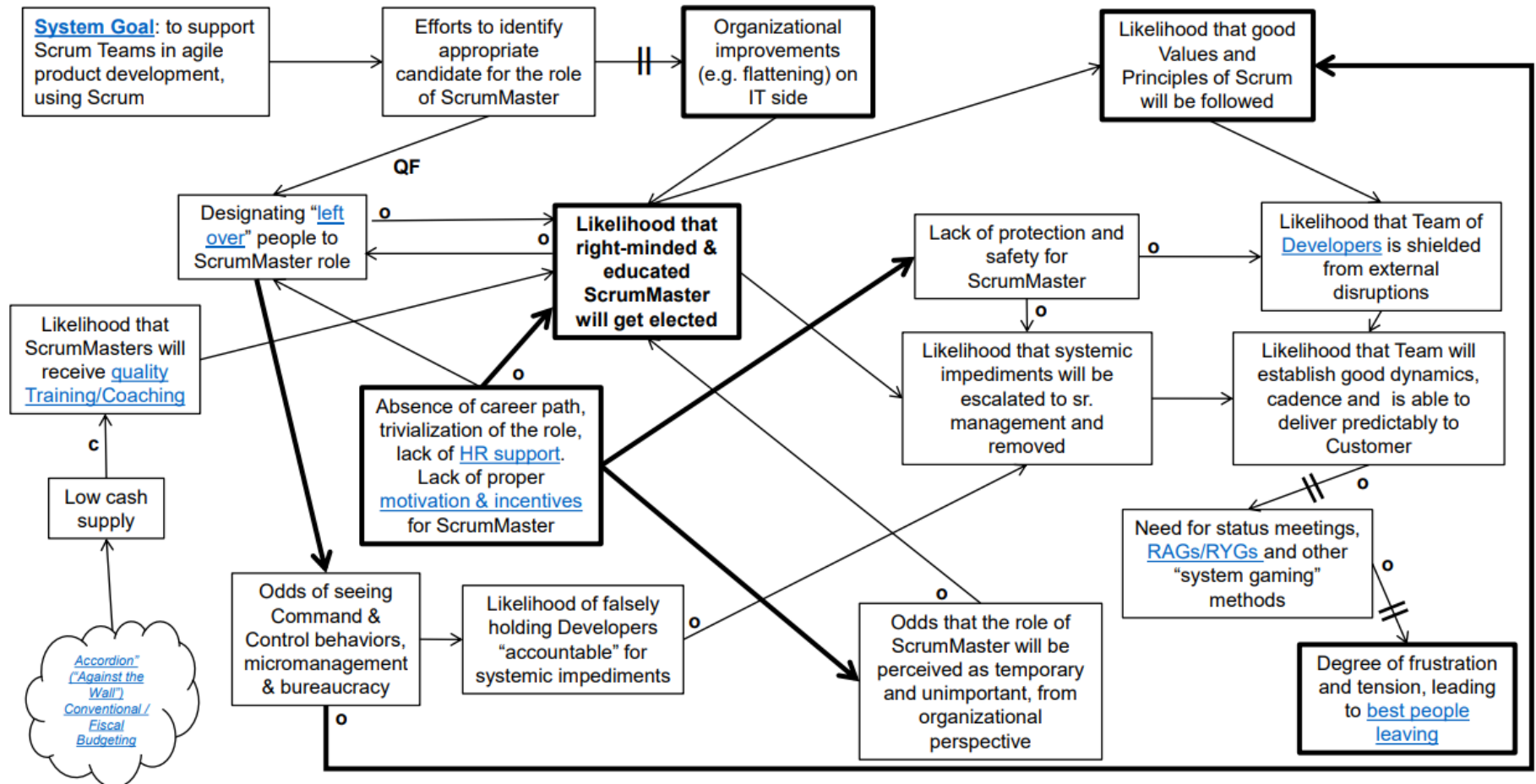
Sourced from: <https://www.keystepstosuccess.com/coach-tools/>



# LeSS Guides: Review & Retrospective

## Improve the System

### System Modeling for ScrumMaster Role



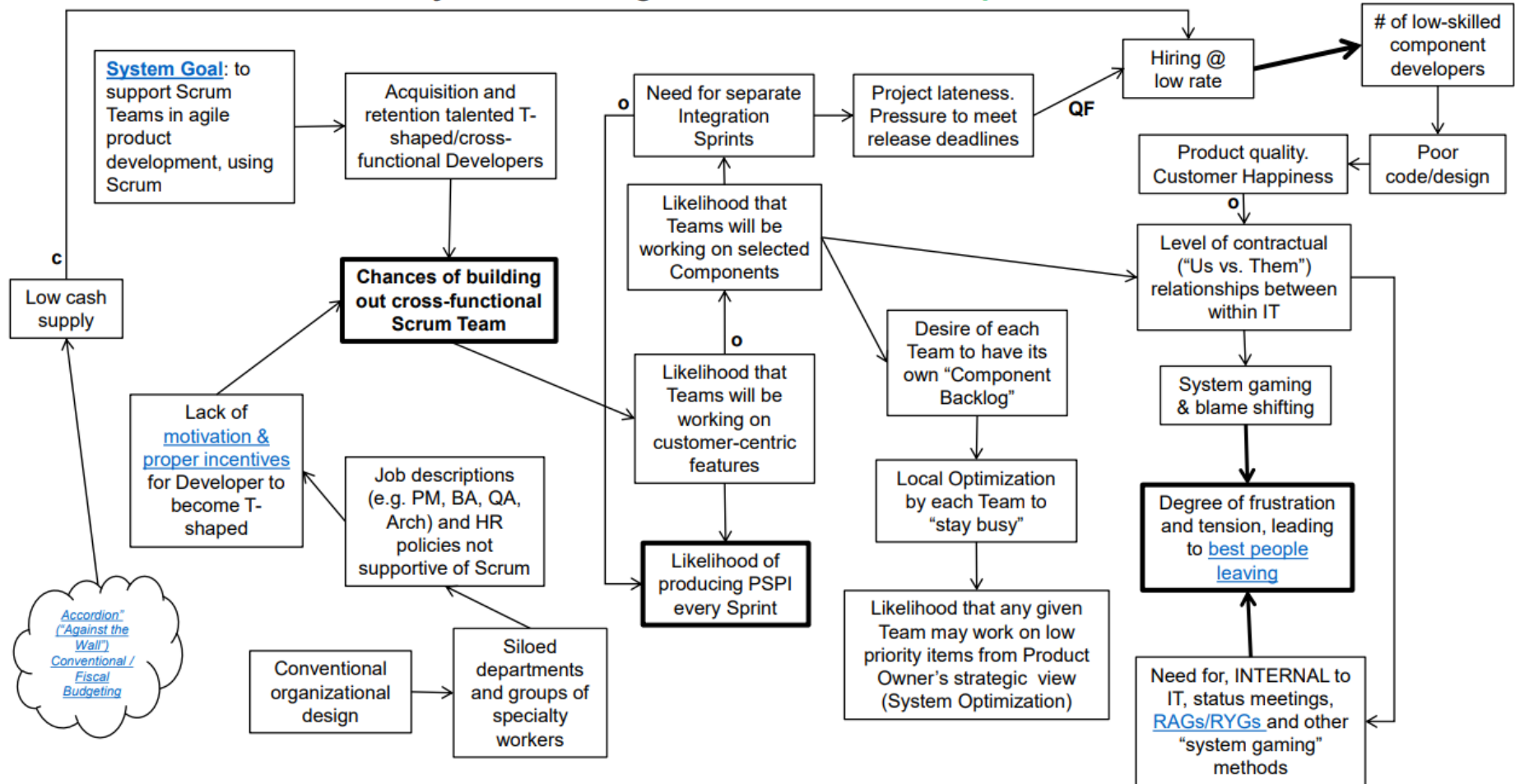
Sourced from: <https://www.keystepstosuccess.com/coach-tools/>



# LeSS Guides: *Review & Retrospective*

## *Improve the System*

### System Modeling for **Scrum Team Developer Role**



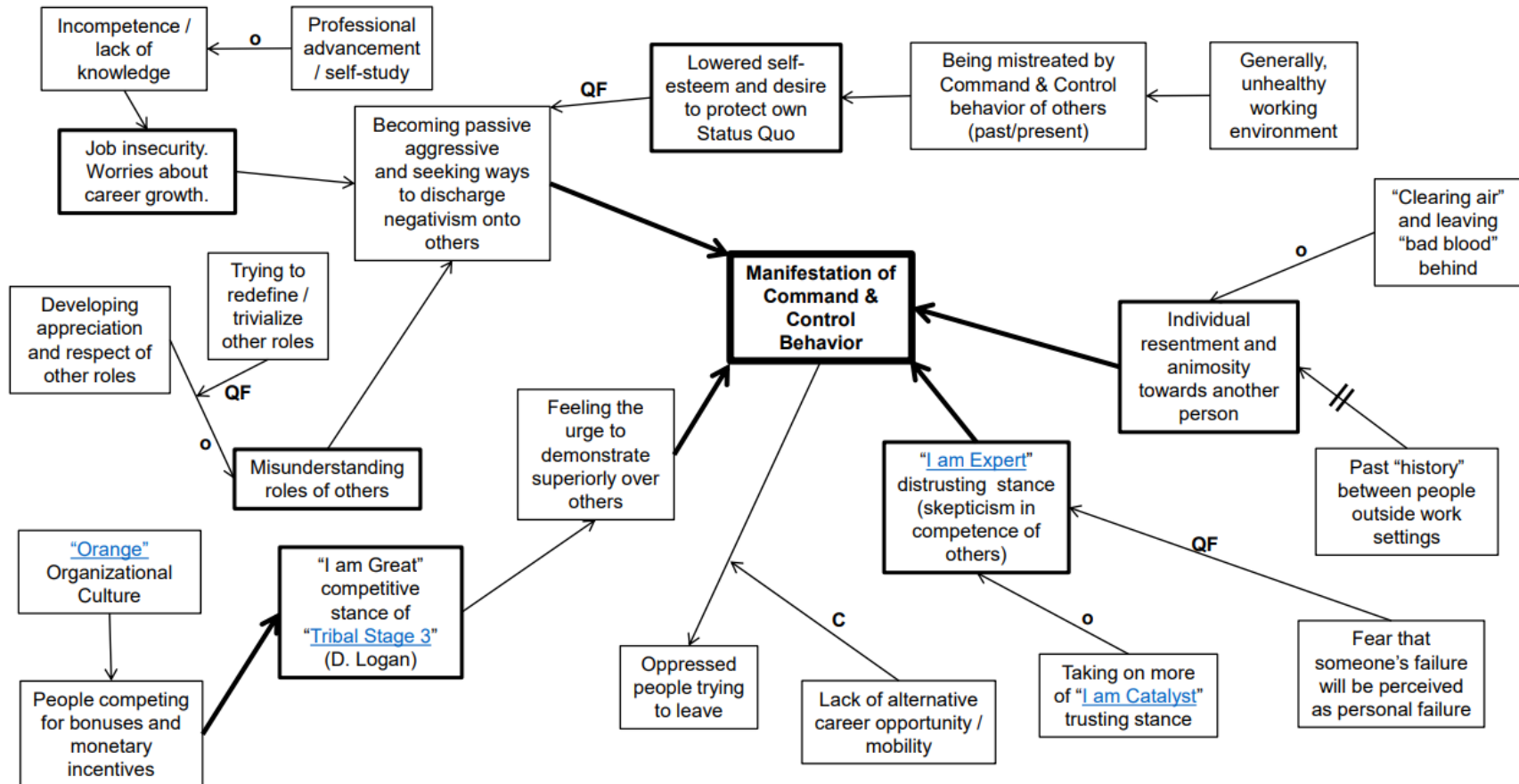
Sourced from: <https://www.keystepstosuccess.com/coach-tools/>



# LeSS Guides: *Review & Retrospective*

## Improve the System

### System Modeling for **Command & Control Behavior**



Sourced from: <https://www.keystepstosuccess.com/coach-tools/>



# LeSS Guides: *Review & Retrospective*

## *Multi-Area Reviews & Retrospective*

There are no LeSS Huge rules for reviews and retrospectives. The catch-all statement “All Sprint LeSS rules apply for each Requirement Area” implies a Sprint Review and Overall Retrospective for each separate Requirement Area. But there’s no requirement for meetings that span the entire product.

Sourced from: <https://less.works/resources/learning-resources/books>

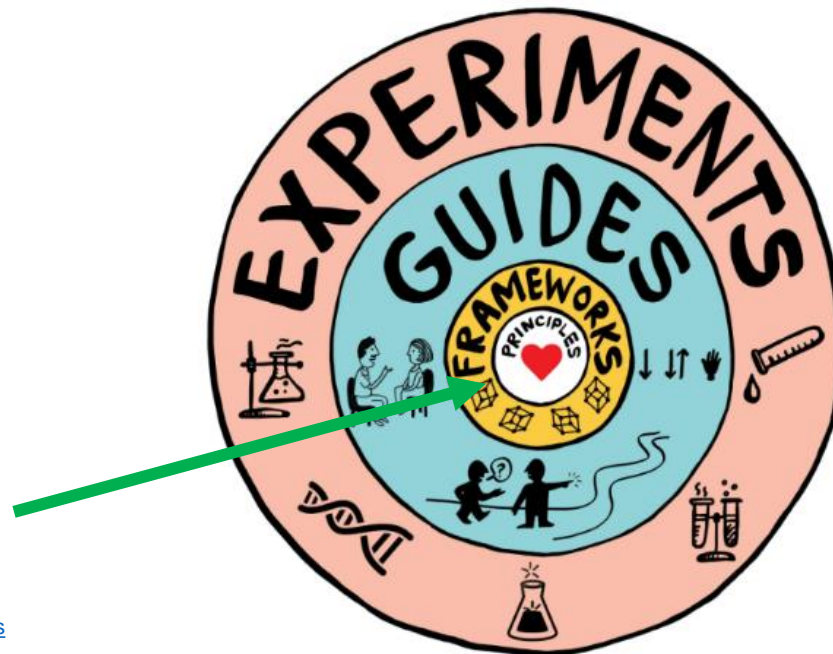
**Do them only on-demand, if you feel they can improve your system**







# LeSS Huge Rules

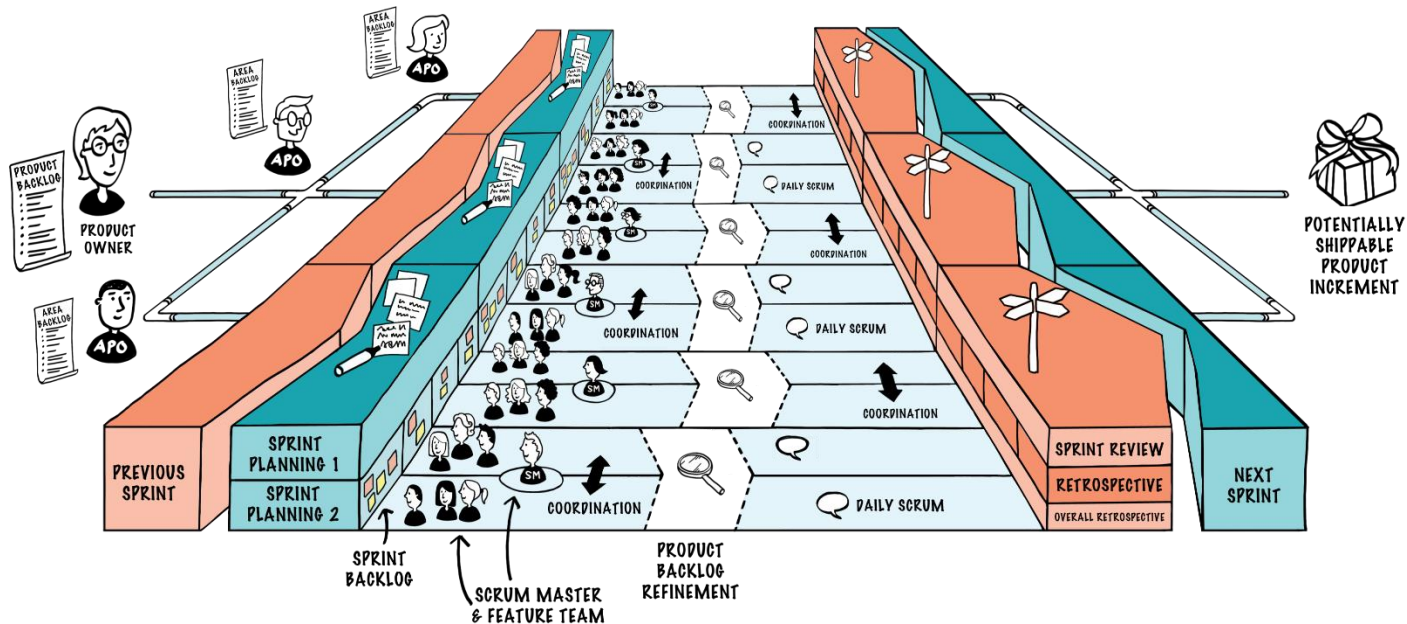


Sourced from: <https://less.works/resources/learning-resources/books>



# LeSS (Huge) Rules

## "Stacks" of LeSS



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

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



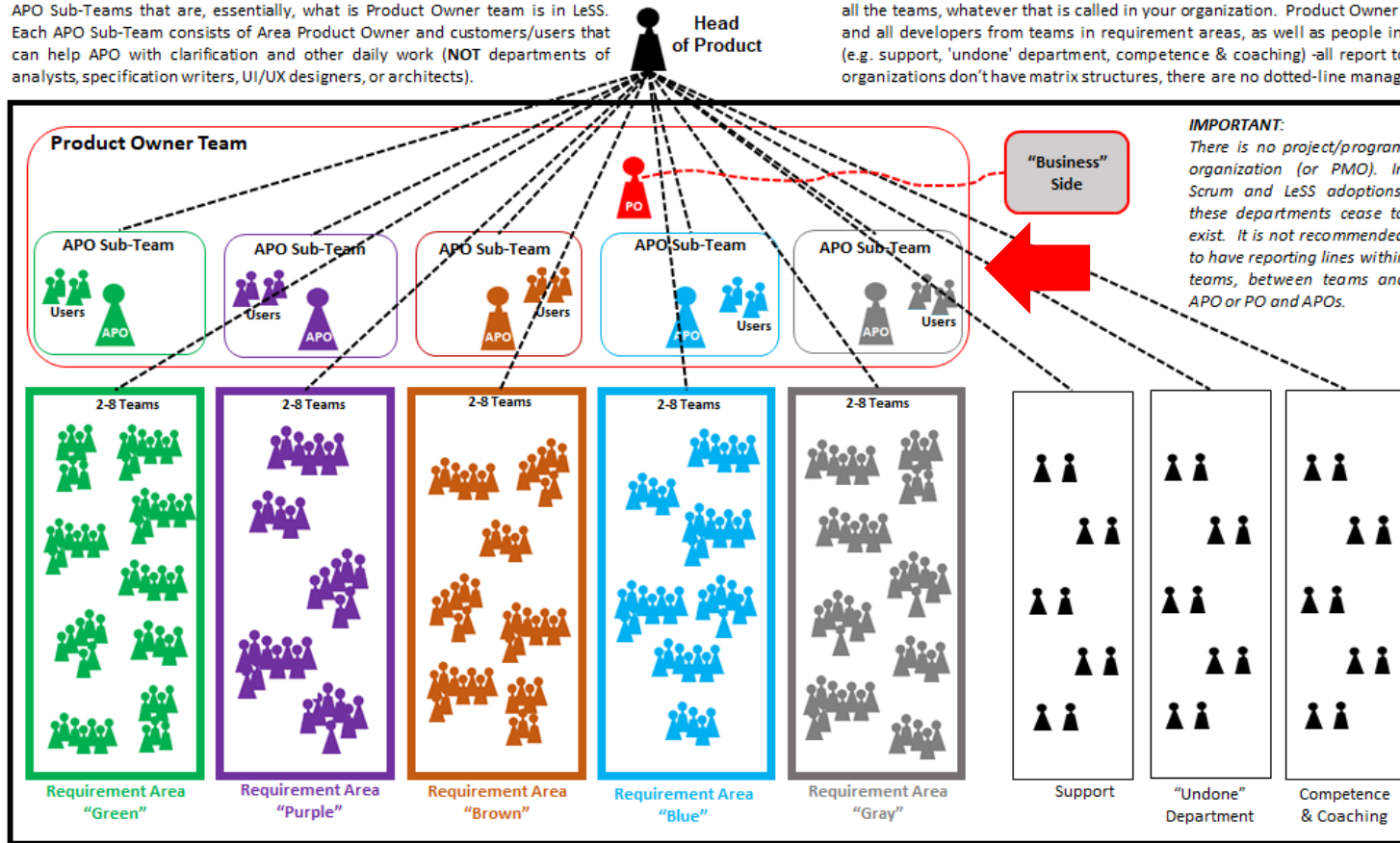
# LeSS (Huge) Rules

## Who The Product Owner Reports Into In LeSS Huge

### 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 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

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



# LeSS (Huge) Rules: *LeSS Huge Structure*

- Customer requirements that are strongly related **from a customer perspective** are grouped in Requirement Areas.
- Each Team specializes in **one Requirement Area**. Teams stay in one area for a long time. When there is more value in other areas, teams might change Requirement Area
- Each Requirement Area **has one Area Product Owner**.
- Each Requirement Area has between **"4-8" teams**. Avoid violating this range.
- LeSS Huge adoptions, including the structural changes, are done with an **evolutionary**, incremental approach.
- Remember each day: LeSS Huge adoptions take months or years, infinite patience, and sense of humor.



# LeSS (Huge) Rules: *LeSS Huge Product*

- One **(overall) Product Owner** is responsible for product-wide prioritization and deciding which teams work in which Area. He works closely with **Area Product Owners**.
- Area Product Owners act as Product Owners towards their teams.
- There is **one Product Backlog**; every item in it belongs to exactly one Requirement Area.
- There is **one Area Product Backlog per Requirement Area**. This backlog is conceptually a more granular view onto the one Product Backlog.



# LeSS (Huge) Rules: *LeSS Huge Sprint*

- There is **one product-level Sprint**, not a different Sprint for each Requirement Area. It ends in one integrated whole product.
- The Product Owner and Area Product Owners synchronize frequently. Before Sprint Planning they ensure the Teams work on the most valuable items. After the Sprint Review, they further enable product-level adaptations



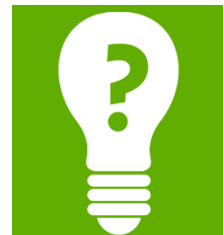
# Class Activity

## Class:

- Discuss how/why LeSS Huge rules are an *extension* of LeSS rules.
- Capture in Class Log

**Instructor:** Catalyze. Reflect.

**Duration:** 10 min

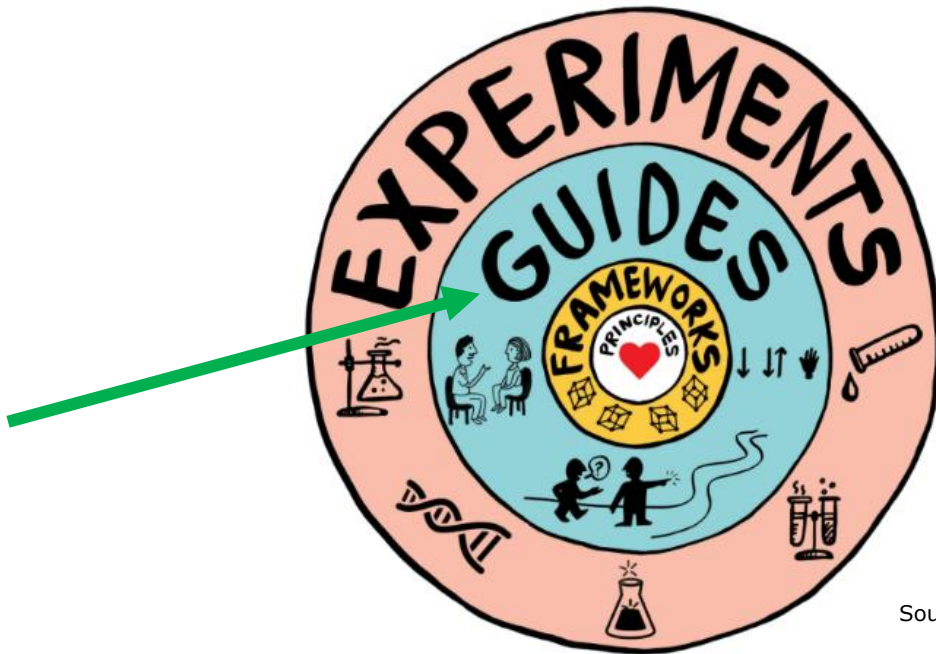




**5 x 5 - 5**



# LeSS Huge Guides



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

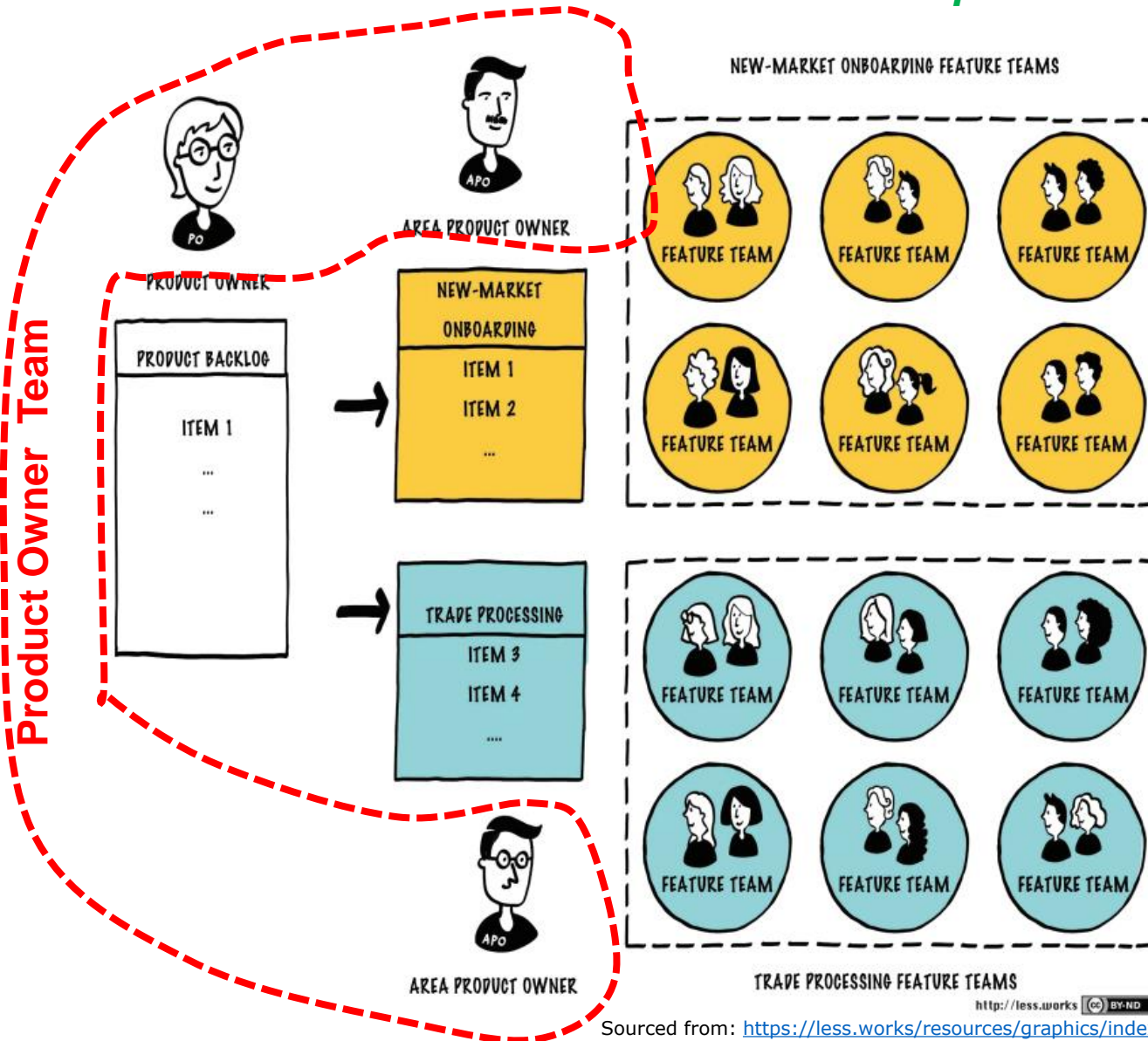






# LeSS Guides: *Product Owner*

## Area Product Owners and Requirement Areas



Customer requirements that are strongly related from a customer perspective are grouped in Requirement Areas.

Each Team specializes in one Requirement Area. Teams stay in one area for a long time. When there is more value in other areas, teams might change Requirement Area.

Each Requirement Area has one Area Product Owner.

Each Requirement Area has between "4-8" teams. Avoid violating this range.

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

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



# LeSS Guides: *Product Owner*

## LeSS Organization

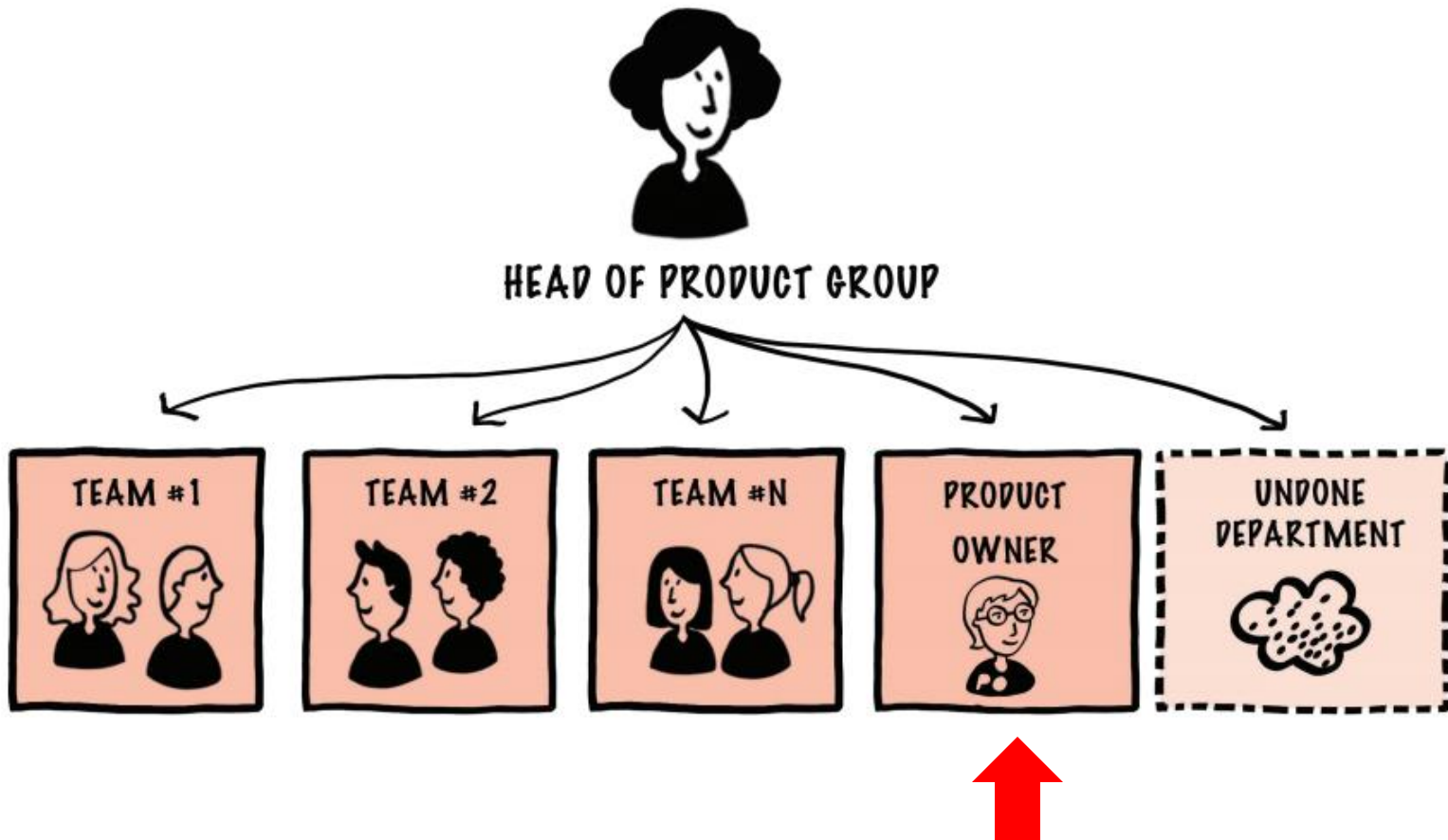
Let's examine a LeSS organization...

- **Head of the Product Group**—Most LeSS organizations still have managers including a “head of product group.” They support the teams by Go See and help them remove obstacles and improve. LeSS organizations don't have matrix structures and there are no “dotted-line” managers. “Head of Product Group” is called differently in different organization, here we mean the hierarchical manager of all the teams.
- **Feature teams**—This is where the development work is done. Each team is cross-functional, self-managing feature team with a Scrum Master. They are permanent units that stay together for the duration of a product (and sometimes longer). Avoid lots of hierarchical layers as much as possible.
- **Product Owner (Team)**—This is also commonly called “Product Management.” It can be one person but in a larger LeSS organization the Product Owner might be supported by other product managers. An important point in this organizational structure is that the Teams and the Product Owner are peers. This important to keep the power balanced between the roles. The Teams and Product Owner should have a cooperative peer relationship. A common alternative structure is when the Product Owner belongs to a different organization. This is OK though it does often require additional effort to ensure the Product Owner has a close relationship with the Teams.
- **Undone department**—This department, ideally, does not exist. But unfortunately sometimes the teams are not yet able to create a true shippable increment every Sprint. This is reflected by their “Definition of Done” not being equal to “Potentially Shippable.” Undone departments such as test, QA, architecture, or business analysis groups should never exist in the smaller LeSS framework groups as they should be integrated into the teams from the start. On the other hand, we unfortunately frequently still see an operations or production undone department in LeSS adoptions, as they often cross organizational boundaries.



# LeSS Guides: *Product Owner*

## Who The Product Owner Reports Into In LeSS



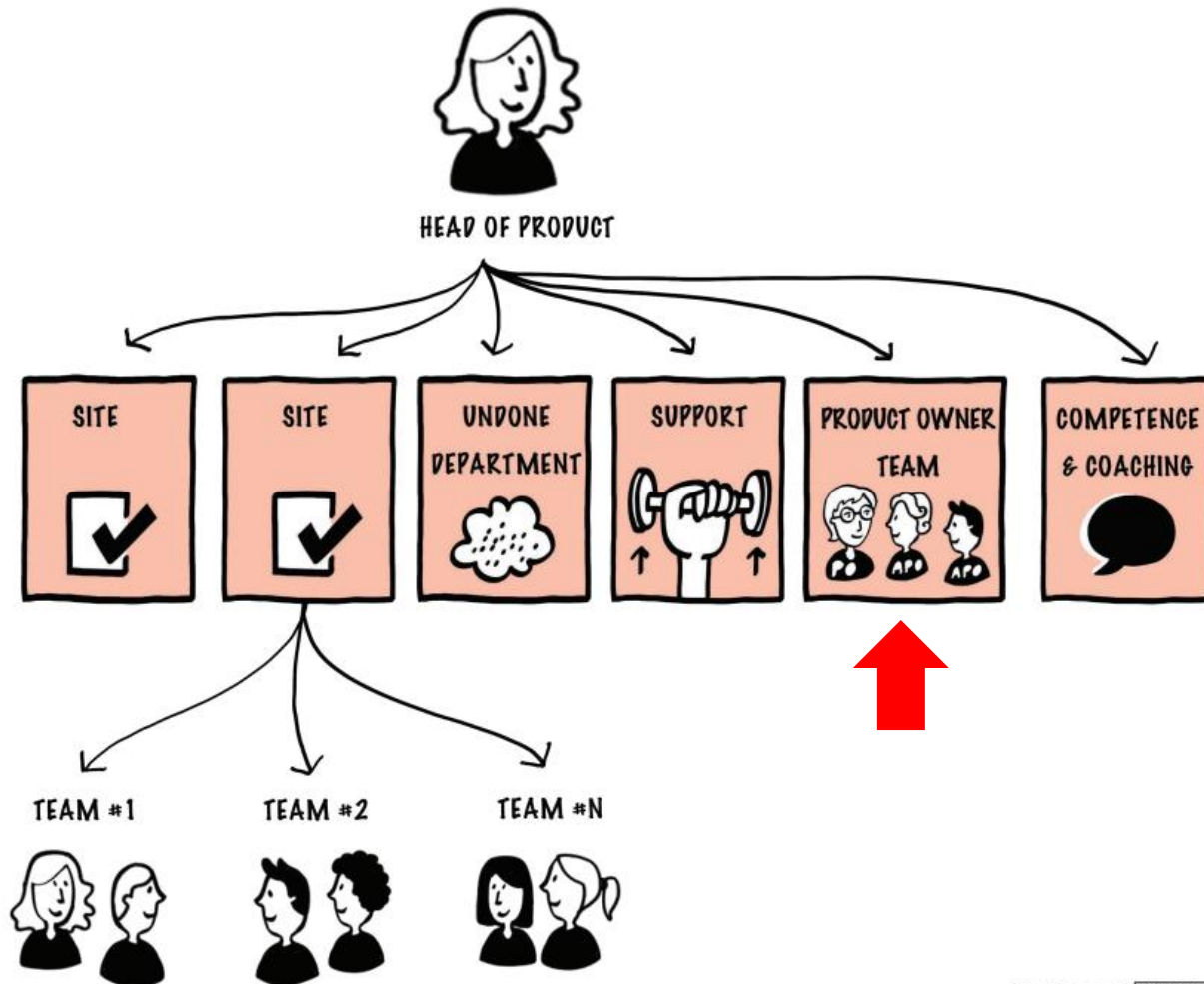
Sourced from: <https://less.works/resources/graphics/index.html>

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



# LeSS Guides: *Product Owner*

## *Who The Product Owner Reports Into In LeSS Huge*

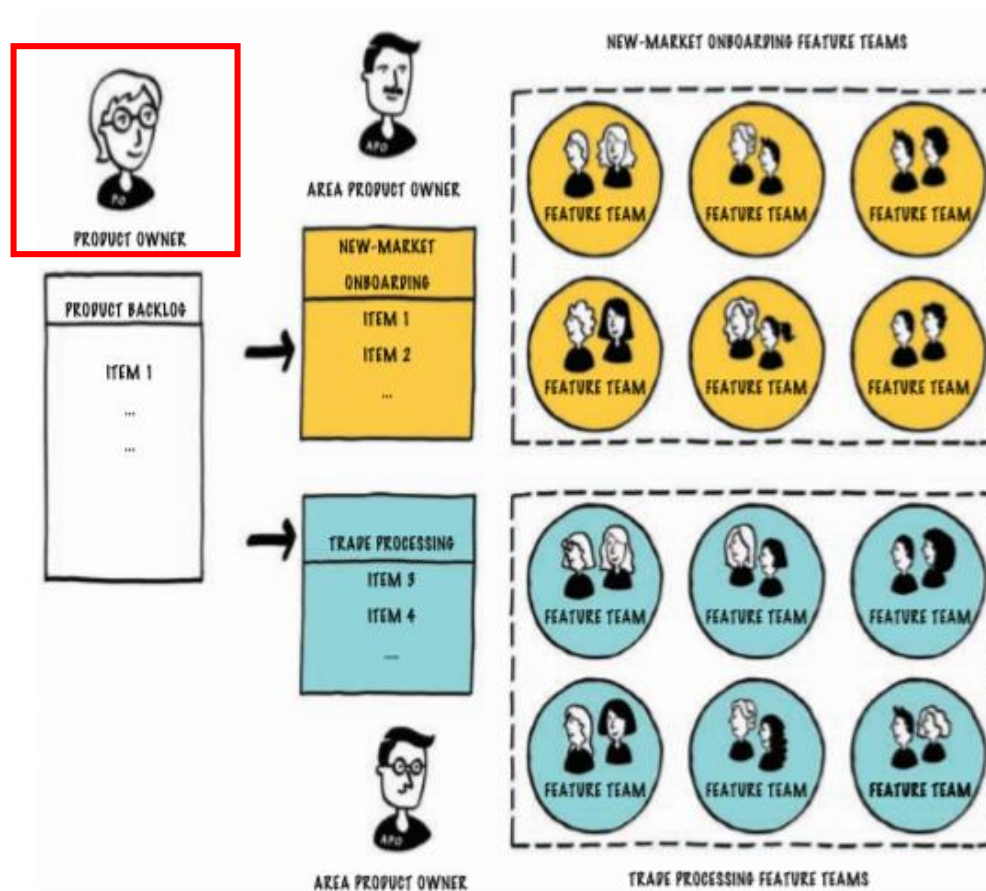


<http://less.works> BY-ND



# LeSS (Huge) Guides: *Product Owner*

## *LeSS Huge Product Owner*



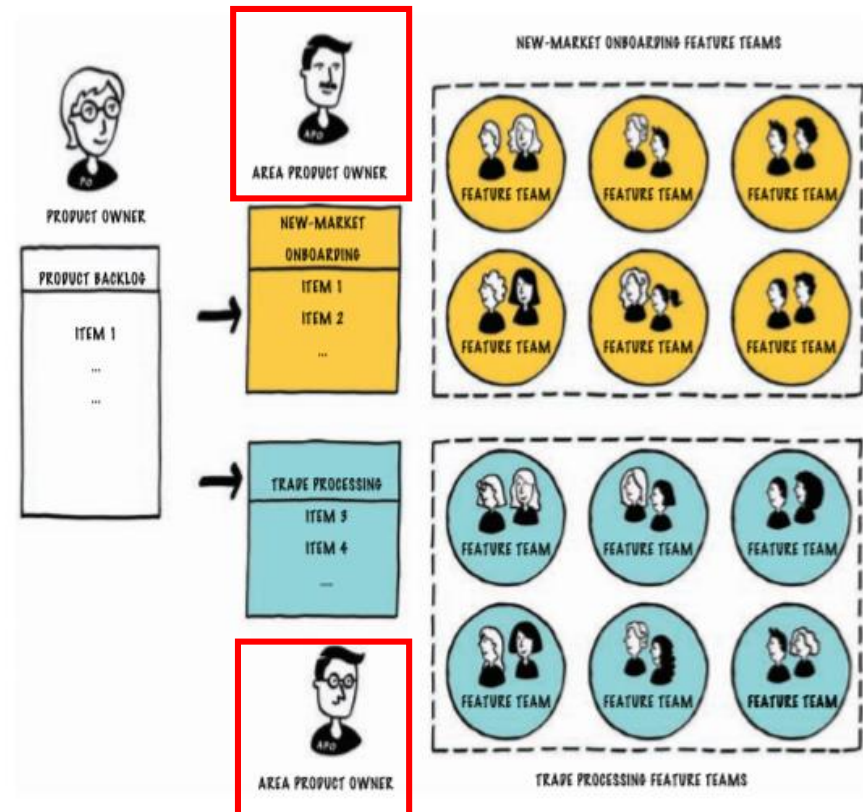
Sourced from: <https://less.works/resources/graphics/index.html>



# LeSS (Huge) Guides: *Product Owner*

## *Area Product Owners*

- Tiny areas leading to “wrong” Area Product Owners
- **Consider**: Temporary Fake Area Product Owner



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



# LeSS (Huge) Guides: *Org. by Customer Value*

## *Dynamics of Requirement Areas*

- **Birth**
  - Requirement Area grows too big
  - New, big requirement arrives
- **Midlife** – requires adjustment
- **Retirement**- if falls below 4 teams



# LeSS (Huge): *Product Backlog*

## Area Backlogs

### 1 “overall” Product Owner

Item	Requirement Area
A	On-Line Trading
B	On-Line Trading
C	Market Data
D	Customer Profile
E	Customer Profile
F	On-Line Trading
G	On-Line Trading
H	Market Data
I	Market Data
J	Market Data
K	On-Line Trading
L	Customer Profile
M	Customer Profile

### Area Product Owner

Item	Requirement Area
A	On-Line Trading
B	On-Line Trading
F	On-Line Trading
G	On-Line Trading

### Area Product Owner

Item	Requirement Area
D	Customer Profile
E	Customer Profile
F	Customer Profile
G	Customer Profile

### Area Product Owner

Item	Requirement Area
C	Market Data
H	Market Data
I	Market Data
J	Market Data

**Product Areas (“views”)**



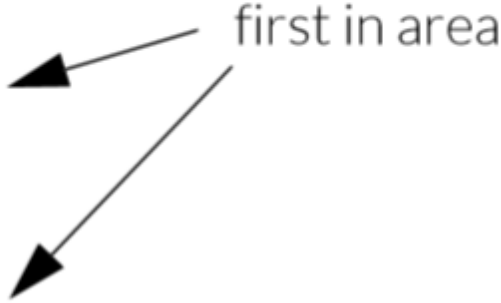
Created by: Gene Gendel



# LeSS (Huge) Guides: *Product Backlog*

## *Product Backlog with Filtered View*

Item	Requirement Area
B	market onboarding
F	market onboarding
C	trade processing
M	trade processing
...	



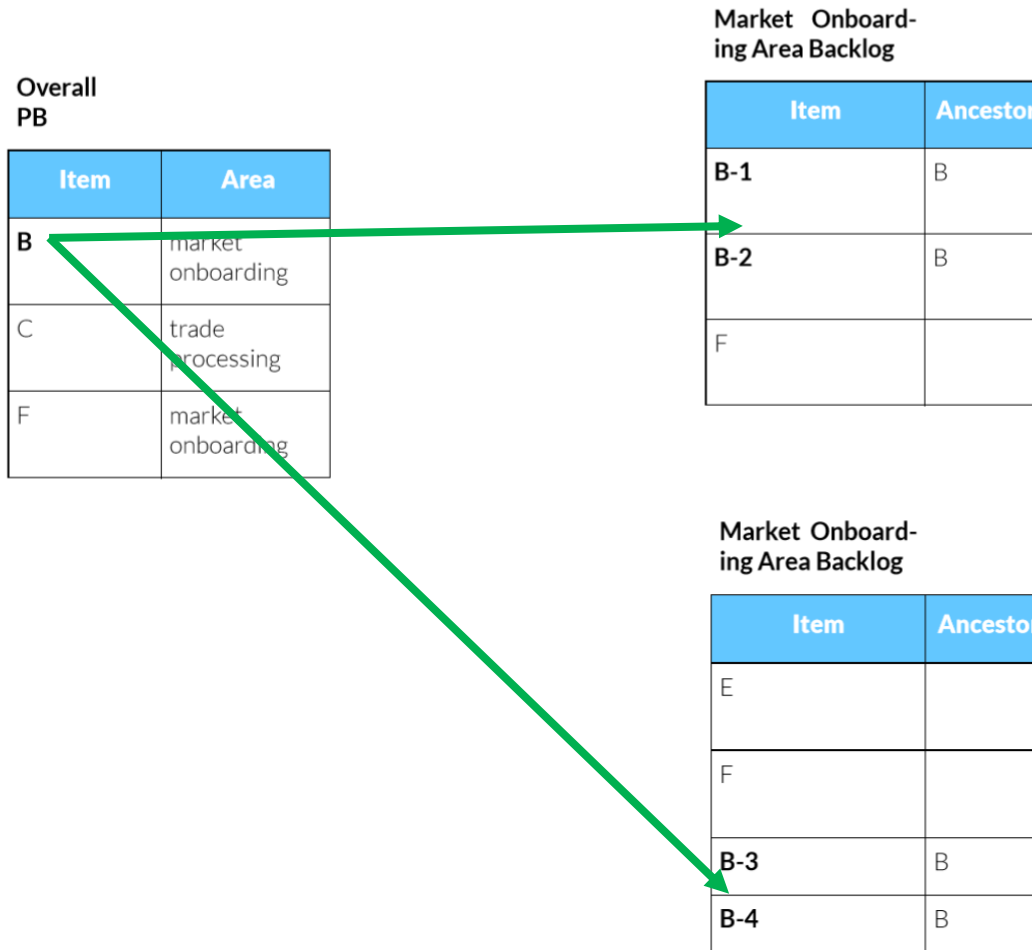
first in area

Sourced from: <https://less.works/resources/learning-resources/books>



# LeSS (Huge) Guides: *Product Backlog*

## *Splitting Within Product Backlog Area*

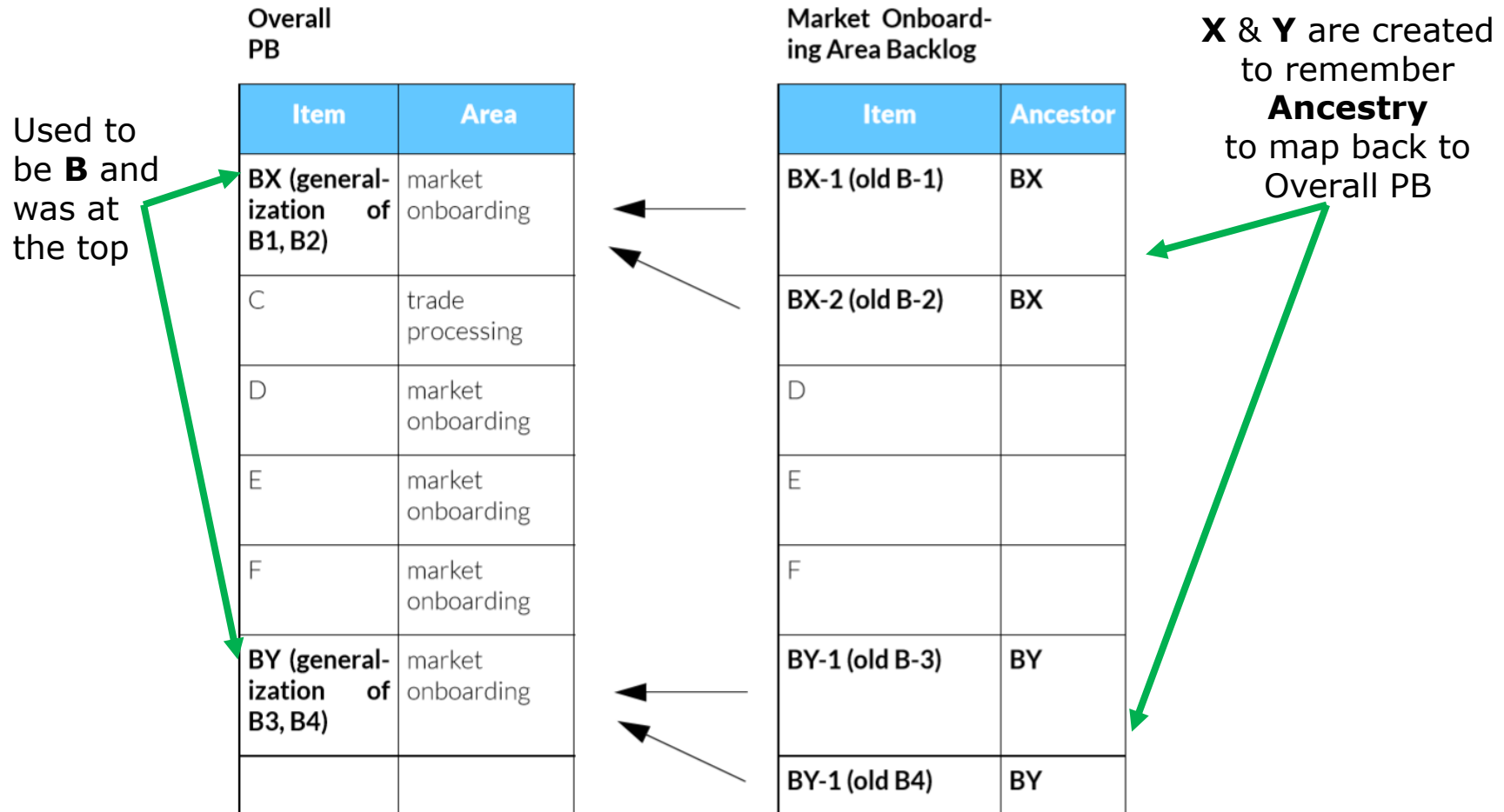


Sourced from: <https://less.works/resources/learning-resources/books>



# LeSS (Huge) Guides: *Product Backlog*

## *Splitting Within Product Backlog Area*



Sourced from: <https://less.works/resources/learning-resources/books>

**B → BX & BY → BX-1 & BX-2 + BY-1 & BY-2**



# LeSS (Huge) Guides: *Product Backlog*

## *Area Backlogs*

### Pros and Cons of Filtered Views versus Separate Artifacts

**Filtered Views—Advantages:** (1) simple, (2) no synchronization issues, (3) easy to keep overview. **Drawbacks:** (1) filters make prioritization harder, (2) Product Owner sees all details of all areas, which at first might seem an advantage but drowns her in details and can lead to the temptation of “micromanaging” priorities in an area, creating conflict in responsibility between the PO and APOs.

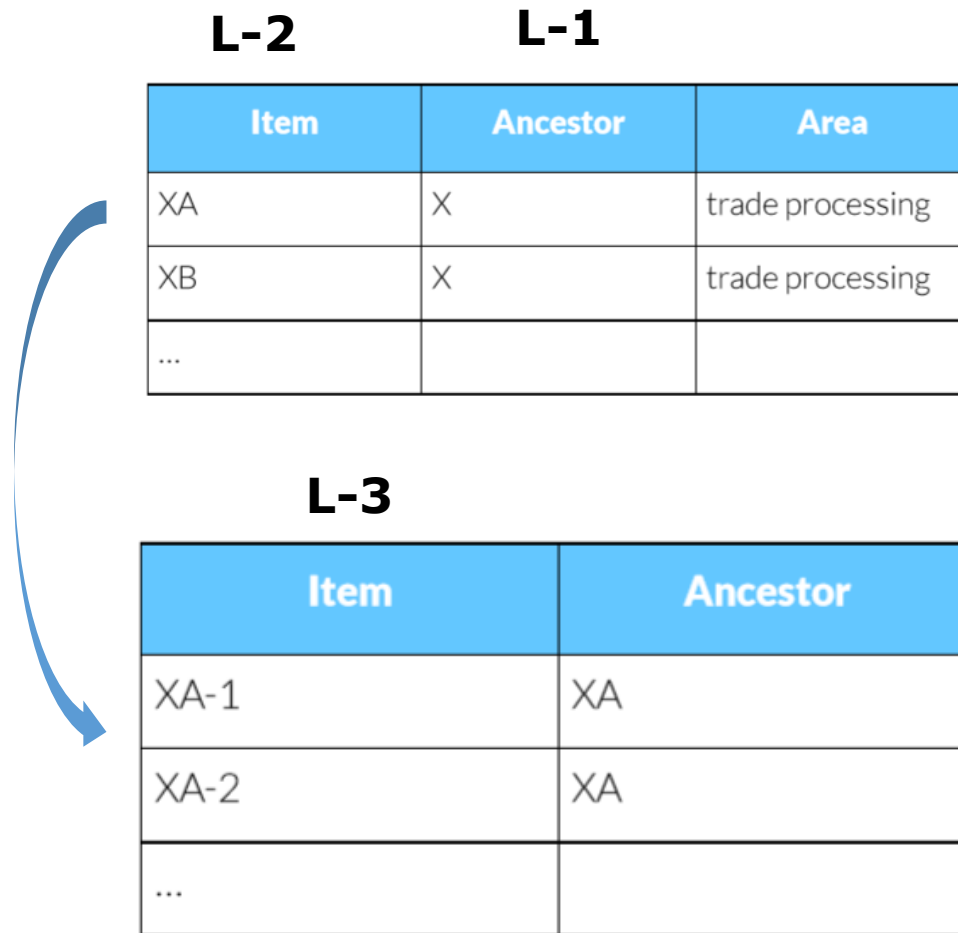
**Separate Artifacts—Advantages:** (1) Overall Backlog stays at higher level and then the PO doesn't drown in details, (2) APO can easily prioritize his backlog, (3) Supports clear separation of responsibilities between PO and APOs. **Drawbacks:** (1) Synchronization between different backlogs, (2) Priority differences not visible in overall Product Backlog, (3) Increased chance of silo mentality in each area rather than APOs caring about whole-product focus.

Sourced from: <https://less.works/resources/learning-resources/books>



# LeSS Guides: *Product Backlog*

## *Levels of Splitting*

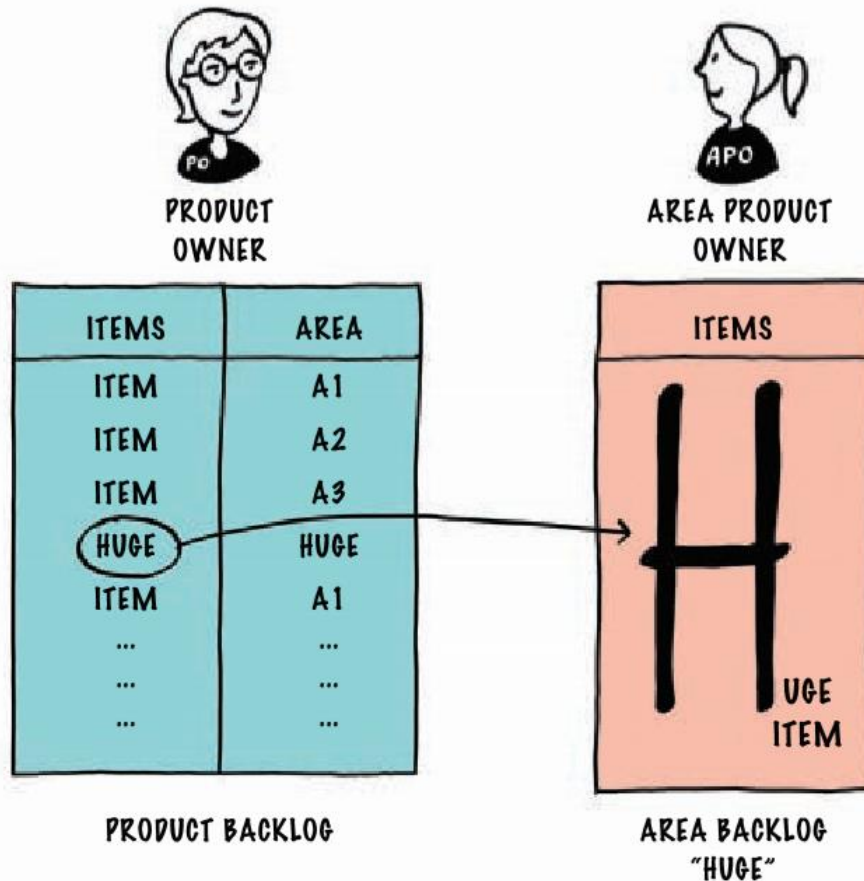


Sourced from: <https://less.works/resources/learning-resources/books>



# LeSS (Huge) Guides: *Product Backlog*

## *New Area for Giant Requirement*

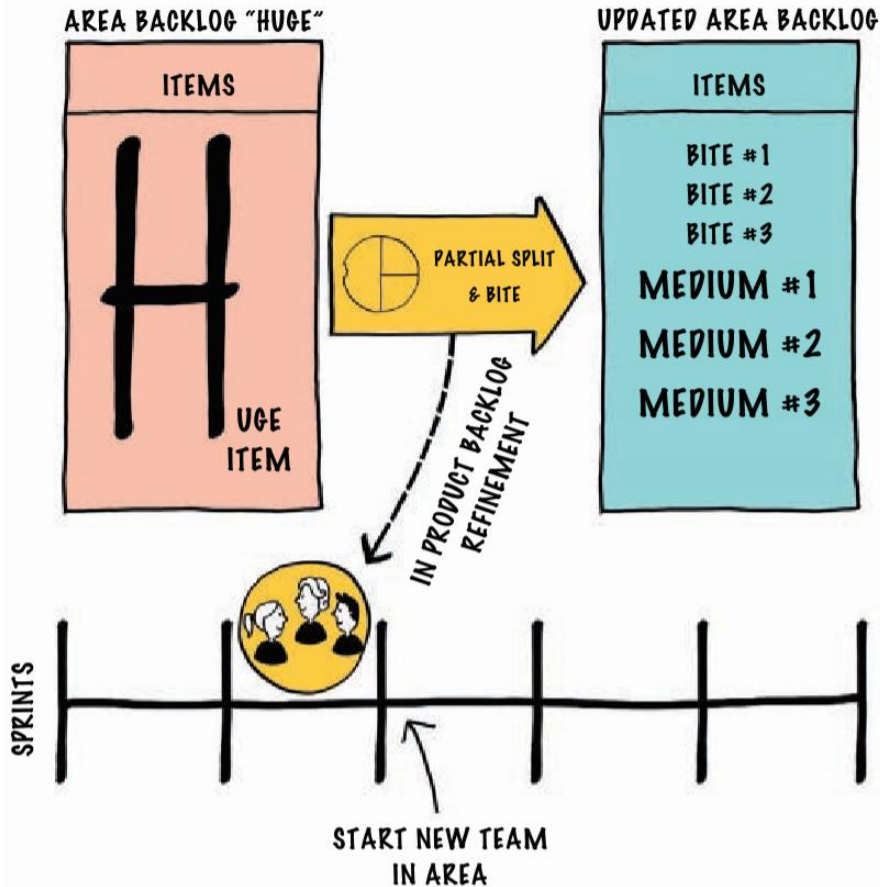


Sourced from: <https://less.works/resources/graphics/index.html>



# LeSS (Huge) Guides: *Product Backlog*

## *Handling Gigantic Requirements*

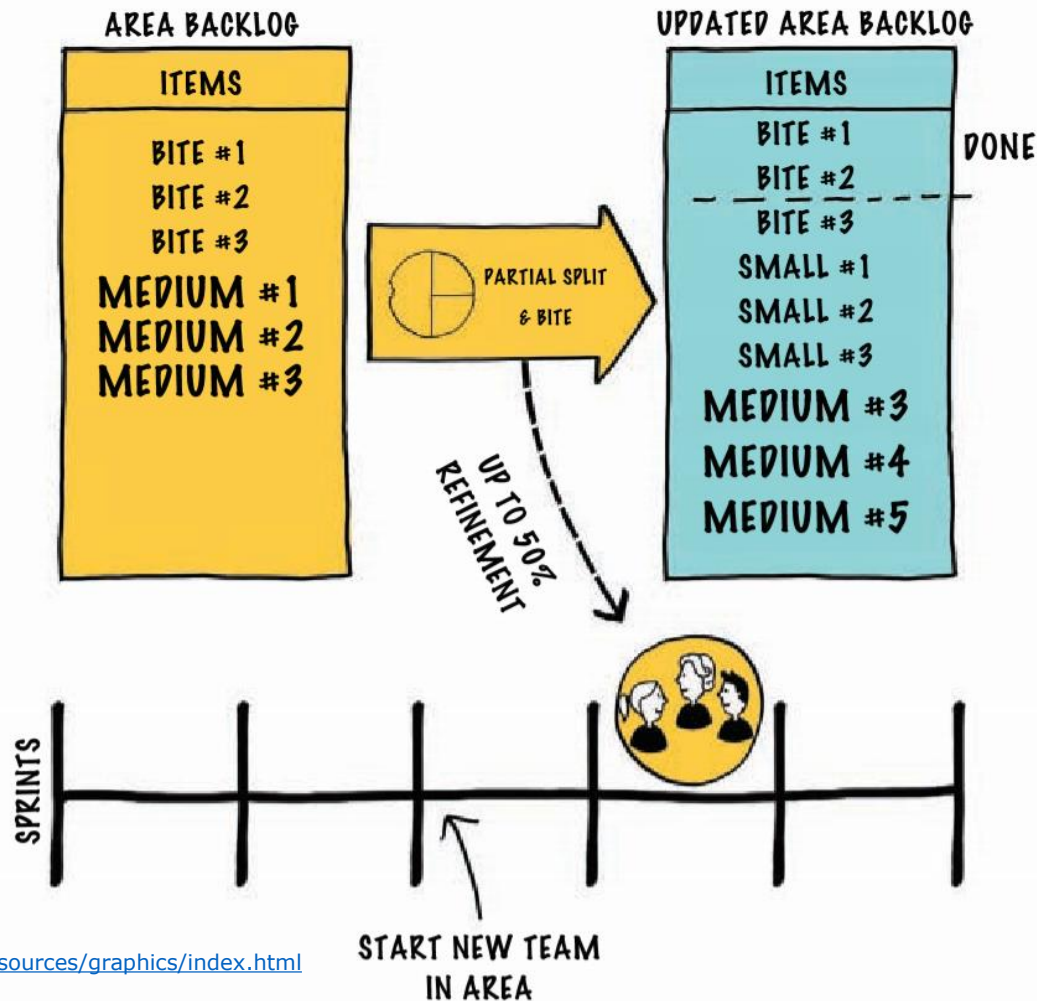


Sourced from: <https://less.works/resources/graphics/index.html>



# LeSS (Huge) Guides: *Product Backlog*

## *Handling Gigantic Requirements*

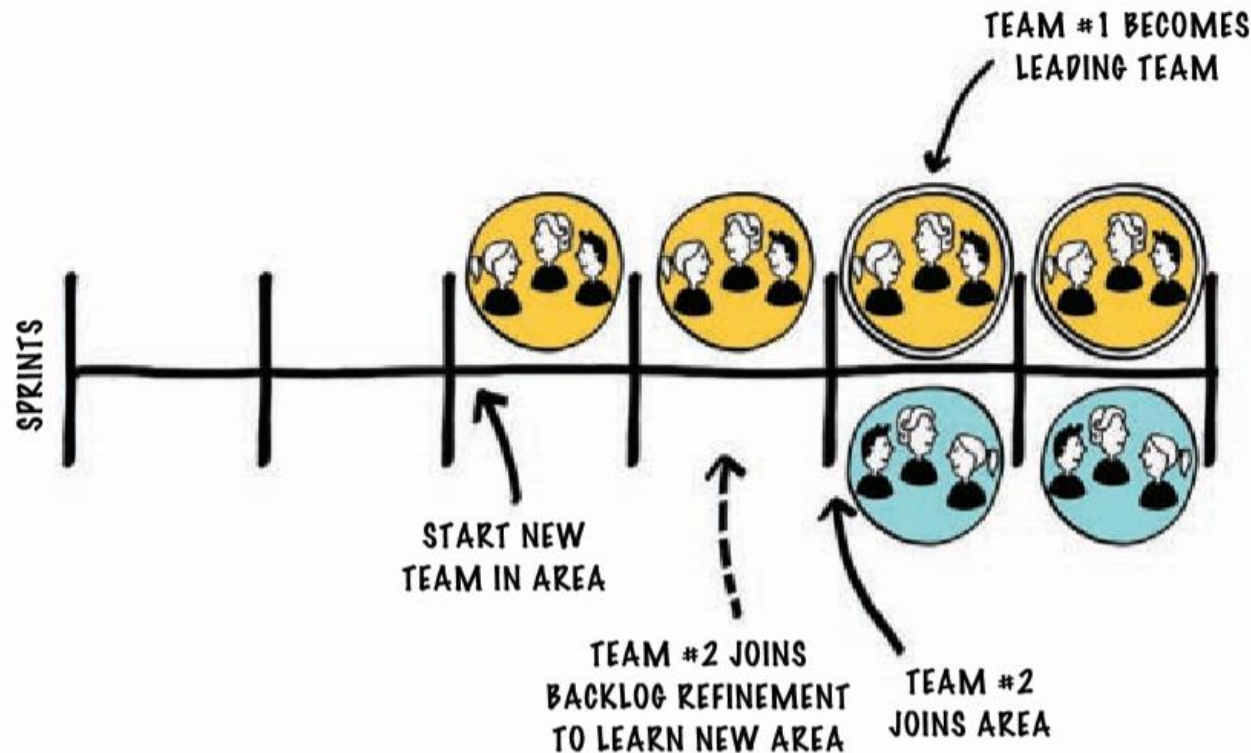


Sourced from: <https://less.works/resources/graphics/index.html>



# LeSS (Huge) Guides: *Product Backlog*

## *Handling Gigantic Requirements*



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



# LeSS (Huge) Guides: *Product Backlog*

## *Handling Gigantic Requirements*

- Create new **Requirement Area** for huge requirement
- Not all teams are equal: start with **more experienced**
- Partial splitting; **Take a Bite**
- Spend up to **50%** of Sprint on **refinement**, while building the **bite**
- Gradually **grow** a new Requirement Area
- Use multi-team PBR for learning
- Initial team becomes **leading team** with additional mentoring and overview responsibilities.







# **Product Definition & Exploration Workshop in LeSS**



## Product Definition & Exploration Workshop

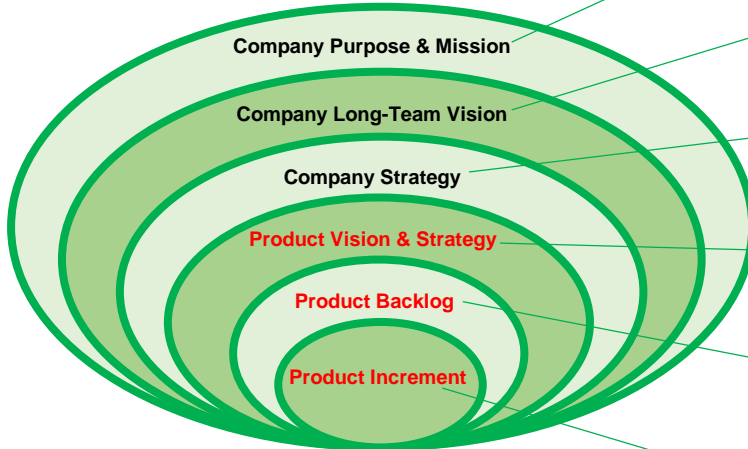
- **Understand a 'big picture' (Strategy, Mission, Vision, OKRs).**
- **Select & discuss a product.**
- **Define & explore a product with Product Canvas, applying expanding & constraining forces.**
- **Run exercises (e.g. Item mapping, value stream mapping, customer journey, etc, personas.)**
- **Create an *initial* Product Backlog.**
- **Create Definition of Done (DoD). Agree on what is "UNDONE".**
- **Identify functional expertize & technical skillset required (each Team; LeSS Product Group)**
- **Prototype a team 'blueprint' (*Do NOT create teams, on behalf of teams*)**
- **Prototype a product group 'blueprint' (beyond Teams; entire LeSS Product Group)**
- **Identify (all known) organizational design impediments and assign them to executives**
- **Run team self-design exercise**
- **Start sprinting**



# Product Definition & Exploration Workshop

Notes:

- Discuss what each 'onion' layer below means in the context of your organization
- Modify the 'onion', based on organizational specifics (add/remove layers)
- In respective areas on the right side, provide some examples and details for each layer
- Discuss and agree where to keep information (e.g. tool, directory, other repository)
- Discuss and agree how 'onion' layers will be maintained and updated over time

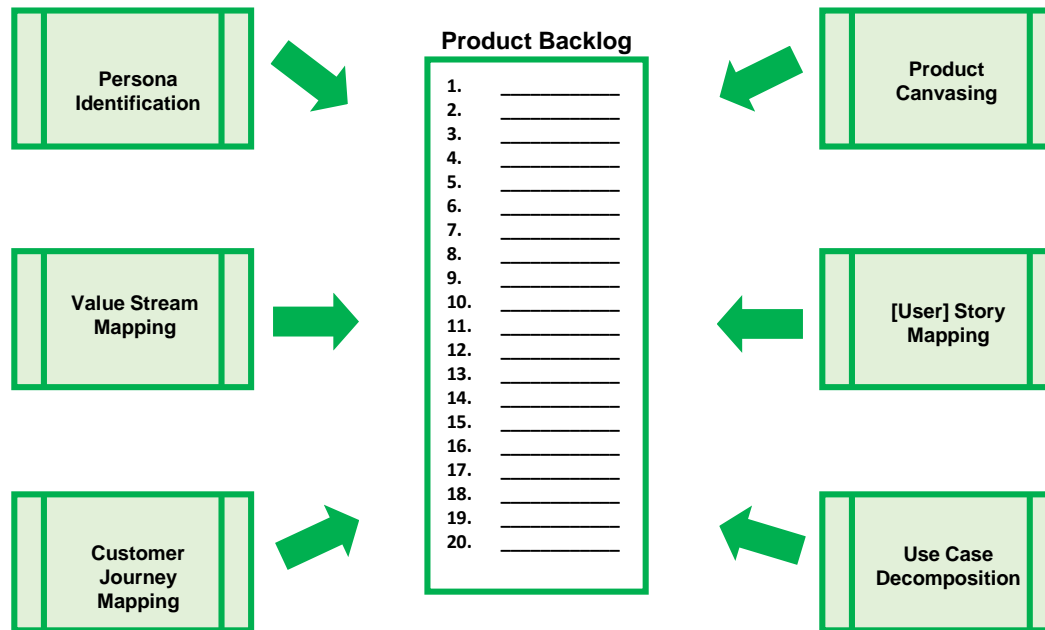


*Note: Since customers buy and users use Products, in agile development, we focus on ProDUCTs, not ProJECTs, not ProGRAMs. We also challenge the term Portfolio, unless it is a portfolio or real Products. Please, be mindful of this, as you add layers to the 'onion'.*



# Product Definition & Exploration Workshop

To help yourself with PRODUCT BACKLOG creation, run various exercises and discovery activities:



**Note:** Remember that the goal of all supplementing exercises and activities is to produce a single Product Backlog, to be prioritized by Product Owner and shared by all teams that will be working on a product.



# Product Definition & Exploration Workshop

## *Persona Identification Template*

Attribute	Description
Picture	
Name	
Age/Gender	
Personality	
Interests	
Title/Organizational Position	
Career Path/Journey	
Technical Skills	
Business Expertize	
Personal Goals	
What is today's solution?	
Reasons to buy/use a product	
Comments	



# Product Definition & Exploration Workshop

## *Use Case Template*

Attribute	Description
Name	
ID	
Description	
Actors	
Assumptions	
Benefits	
Pre-Conditions	
Post-Conditions	
Main Path (Steps)	1. -----step----- 2. -----step----- 3. -----step----- 4. -----step----- 5. -----step----- 6. -----step----- 7. -----step-----
Alternative Path (Steps)	1. -----step----- 2. -----step----- 3. -----step----- 4. -----step-----
Exceptions	
Comments	
Related Use Cases	e.g. <<extend>>, <<include>>



# Product Definition & Exploration Workshop

## Customer Journey Map Template

Phase/Stage of Journey	Phase 1	Phase 2	Phase 3	Phase 4			
Specific Activities/Steps	1. Activity 2. activity 3. activity 4. activity 5. activity 6. activity	1. Activity 2. activity 3. activity 4. activity 5. activity 6. activity	1. Activity 2. activity 3. activity 4. activity 5. activity 6. activity	1. Activity 2. activity 3. activity 4. activity 5. activity 6. activity	1. Activity 2. activity 3. activity 4. activity 5. activity 6. activity	1. Activity 2. activity 3. activity 4. activity 5. activity 6. activity	1. Activity 2. activity 3. activity 4. activity 5. activity 6. activity
Sentiments, Perceptions, Feelings, Emotions							
Comments, Ideas, Improvements							



# Product Definition & Exploration Workshop

## [User] Story Mapping Template




**Note:** Try not to exceed three levels of business-centric decomposition




# Product Definition & Exploration Workshop

Define & Explore Product (using Product Canvas): use expanding & constraining forces

PRODUCT CANVAS (1/2)				Product	Organization	Date	Version
<b>Vision</b> What do we strive to be? What do we do that changes the world? What differentiates us? Tagline? Visual image?		<b>Revenue Streams</b> How do we fund our product? Where does our revenue come from?		<b>Cost Factors</b> What cost factors are there to discover, deliver, market, sell, and service our product? What activities and assets are most expensive? Most risky?		<b>Channels</b> How do we keep our customers coming back? How do we obtain, retain, and grow customer awareness of our product?	
<b>Product Partner: Business</b> Who are the people in the business (ours, others) who legitimize, authorize, pay for, or help us promote the product? Who has domain experience we need to draw upon?	<b>Value</b> What outcomes do business partners want (e.g., increased revenue, cost reduction, risk reduction, brand projection, customer satisfaction, etc.)?	<b>Product Partner: Customer</b> Who needs or uses our product? Who are our target customers? What market or market segment do we serve?	<b>Value</b> What outcomes do customer partners want (e.g., save time, reduce effort, simplify access, appeal to aesthetics, etc.)?	<b>Product Partner: Technology</b> Who designs, builds, tests, deploys, services, trains, supports the product?	<b>Value</b> What outcomes do technology partners want (e.g., increase architectural readiness, improve reliability, reduce future cost to service, develop technical expertise, etc.)?		
<b>Competitive/Leader Landscape</b> Who competes with us? What are their strengths? Weaknesses? Who is the leader? What is their superior capability? Competition: _____ Capability/Threat: _____ Leader: _____ Superior Capability: _____				<b>Innovation Waves (technology   platforms   standards)</b> What changes are happening now or possibly in future to hardware, software, platforms, or standards? What trends might impact the design, development, acquisition, deployment or servicing of the product?			



This work is licensed under the Creative Commons Attribution-ShareAlike 4.0 International License  
 Download the Product Canvas at: <https://www.ebgconsulting.com/ProductCanvas.pdf>  
 DESIGNED BY: Ellen Gottesdiener, EBG Consulting | @ellengott | version 20.02











**Source:**  
<https://www.ebgconsulting.com/ProductCanvas.pdf>  
 ebconsulting.com




# Product Definition & Exploration Workshop

Define & Explore Product (using Product Canvas): use expanding & constraining forces

PRODUCT CANVAS (2/2)				Product	Organization	Date	Version
 <b>User</b> People, systems, and devices that interact with the product	 <b>Interface</b> Connections to and from users, systems, and devices.	 <b>Action</b> Capabilities provided for users	 <b>Data</b> Data and useful information the product stores and uses	 <b>Control</b> Constraints, policies, regulations, and business rules enforced by the product	 <b>Environment</b> Technology platforms (software and hardware components, standards, and physical properties) the product conforms with	 <b>Quality Attribute</b> Measurable properties that qualify the product's operation and development	

 This work is licensed under the Creative Commons Attribution-ShareAlike 4.0 International License  
Download the Product Canvas at: <https://www.ebgconsulting.com/ProductCanvas.pdf>  
DESIGNED BY: Ellen Gottesdiener, EBG Consulting | @ellengott | version 20.02

Source: <https://www.ebgconsulting.com/ProductCanvas.pdf>  ebgconsulting.com



## **Product Definition & Exploration Workshop**

Define & Explore Product (using Product Canvas): use expanding & constraining forces

### ***Additional dimensions for product definition***

- **Value-adding steps, activities** - What do you need to do, to achieve desirable results?
- **Assets/Resources** - What will you need to adequately support/sponsor your product?
- **Business Relationships** – What relationships with business and strategic partners/sponsors to consider?
- **Customer Base** – Who are your key customers? Target audience? Key Personas?
- **Costs** – What do you charge your customers and how do you justify your cost structure?



## Product Definition & Exploration Workshop

### Done

1. \_\_\_\_\_
2. \_\_\_\_\_
3. \_\_\_\_\_
4. \_\_\_\_\_
5. \_\_\_\_\_
6. \_\_\_\_\_
7. \_\_\_\_\_
8. \_\_\_\_\_
9. \_\_\_\_\_
10. \_\_\_\_\_
11. \_\_\_\_\_
12. \_\_\_\_\_
13. \_\_\_\_\_
14. \_\_\_\_\_
15. \_\_\_\_\_

*Note: A check list of items that must be done, by the end of each sprint, in order deliver a product increment (PI), as per agreement, by all teams, and with Product Owner. In LeSS, all teams share the same Definition of Done. The goal should be to expand Done, over time.*

### "Undone"

1. \_\_\_\_\_
2. \_\_\_\_\_
3. \_\_\_\_\_
4. \_\_\_\_\_
5. \_\_\_\_\_
6. \_\_\_\_\_
7. \_\_\_\_\_
8. \_\_\_\_\_
9. \_\_\_\_\_
10. \_\_\_\_\_
11. \_\_\_\_\_
12. \_\_\_\_\_
13. \_\_\_\_\_
14. \_\_\_\_\_
15. \_\_\_\_\_

*Note: A check list of items that are explicitly known, as 'undoable', due to technical and/or organizational limitations. The goal should be to reduce 'Undone' over time.*



# Product Definition & Exploration Workshop

## Technical Skill Set (per Team)

1. \_\_\_\_\_
2. \_\_\_\_\_
3. \_\_\_\_\_
4. \_\_\_\_\_
5. \_\_\_\_\_
6. \_\_\_\_\_
7. \_\_\_\_\_
8. \_\_\_\_\_
9. \_\_\_\_\_
10. \_\_\_\_\_
11. \_\_\_\_\_
12. \_\_\_\_\_
13. \_\_\_\_\_
14. \_\_\_\_\_
15. \_\_\_\_\_
16. \_\_\_\_\_
17. \_\_\_\_\_
18. \_\_\_\_\_
19. \_\_\_\_\_
20. \_\_\_\_\_

*Note: What technical skill set (e.g. technology, language) a team needs to have, in order to perform work in a backlog, to meet Definition of Done?*

## Functional Domain Expertize (per Team)

1. \_\_\_\_\_
2. \_\_\_\_\_
3. \_\_\_\_\_
4. \_\_\_\_\_
5. \_\_\_\_\_
6. \_\_\_\_\_
7. \_\_\_\_\_
8. \_\_\_\_\_
9. \_\_\_\_\_
10. \_\_\_\_\_
11. \_\_\_\_\_
12. \_\_\_\_\_
13. \_\_\_\_\_
14. \_\_\_\_\_
15. \_\_\_\_\_
16. \_\_\_\_\_
17. \_\_\_\_\_
18. \_\_\_\_\_
19. \_\_\_\_\_
20. \_\_\_\_\_

*Note: What domain expertise (e.g. line of business) each team needs to have, in order to perform work in a backlog, to meet Definition of Done?*



## Product Definition & Exploration Workshop

### Prototype Team 'Blueprint'

1. \_\_\_\_\_
2. \_\_\_\_\_
3. \_\_\_\_\_
4. \_\_\_\_\_
5. \_\_\_\_\_
6. \_\_\_\_\_
7. \_\_\_\_\_
8. \_\_\_\_\_
9. \_\_\_\_\_

*Note: Team 'blueprint' is a template of what a team may look like, given technical skillset and domain expertise required, in order to perform work in a backlog? Do not assign individuals to teams: teams will be created during a self-design workshop.*

### Prototype Product Group 'Blueprint' (beyond Teams)

1. \_\_\_\_\_
2. \_\_\_\_\_
3. \_\_\_\_\_
4. \_\_\_\_\_
5. \_\_\_\_\_
6. \_\_\_\_\_
7. \_\_\_\_\_
8. \_\_\_\_\_
9. \_\_\_\_\_
10. \_\_\_\_\_
11. \_\_\_\_\_
12. \_\_\_\_\_
13. \_\_\_\_\_
14. \_\_\_\_\_
15. \_\_\_\_\_

*Note: Product Group 'blueprint' is a template of what the whole group may look like, including teams, Product Owner and users/customers, in order to function effectively and independently?*



# Product Definition & Exploration Workshop

## [Initial] Product Backlog

High Priority

1.	
2.	
3.	
4.	
5.	
6.	
7.	
8.	
9.	
10.	
11.	
12.	
13.	
14.	
15.	
16.	
17.	
18.	
19.	
20.	

Low Priority

*Note: Initial product backlog should provide enough work ("feed" work) for teams during the first few sprints of LeSS adoption. This takes place during Initial Product Backlog Refinement event. Participation of an entire product group is required.*



# Product Definition & Exploration Workshop

## Organizational Impediment Backlog

*High Priority*

1. \_\_\_\_\_
2. \_\_\_\_\_
3. \_\_\_\_\_
4. \_\_\_\_\_
5. \_\_\_\_\_
6. \_\_\_\_\_
7. \_\_\_\_\_
8. \_\_\_\_\_
9. \_\_\_\_\_
10. \_\_\_\_\_
11. \_\_\_\_\_
12. \_\_\_\_\_
13. \_\_\_\_\_
14. \_\_\_\_\_
15. \_\_\_\_\_
16. \_\_\_\_\_
17. \_\_\_\_\_
18. \_\_\_\_\_
19. \_\_\_\_\_
20. \_\_\_\_\_

*Low Priority*

*Note: What organizational impediments (e.g. HR policies, reporting structure, career path, location strategies, budgeting, etc.) may potentially hinder healthy dynamics of a product group in LeSS adoption? Who will self-assign this responsibility (e.g. sr. manager, executive)? How do you plan on measuring progress of impediment resolution?*



# Product Definition & Exploration Workshop

## Run Team Self-Design Exercise

<p><i>Team</i></p> <ol style="list-style-type: none"><li>1. _____</li><li>2. _____</li><li>3. _____</li><li>4. _____</li><li>5. _____</li><li>6. _____</li><li>7. _____</li><li>8. _____</li><li>9. _____</li></ol>	<p><i>Team</i></p> <ol style="list-style-type: none"><li>1. _____</li><li>2. _____</li><li>3. _____</li><li>4. _____</li><li>5. _____</li><li>6. _____</li><li>7. _____</li><li>8. _____</li><li>9. _____</li></ol>	<p><i>Team</i></p> <ol style="list-style-type: none"><li>1. _____</li><li>2. _____</li><li>3. _____</li><li>4. _____</li><li>5. _____</li><li>6. _____</li><li>7. _____</li><li>8. _____</li><li>9. _____</li></ol>	<p><i>Team</i></p> <ol style="list-style-type: none"><li>1. _____</li><li>2. _____</li><li>3. _____</li><li>4. _____</li><li>5. _____</li><li>6. _____</li><li>7. _____</li><li>8. _____</li><li>9. _____</li></ol>
<p><i>Team</i></p> <ol style="list-style-type: none"><li>1. _____</li><li>2. _____</li><li>3. _____</li><li>4. _____</li><li>5. _____</li><li>6. _____</li><li>7. _____</li><li>8. _____</li><li>9. _____</li></ol>	<p><i>Team</i></p> <ol style="list-style-type: none"><li>1. _____</li><li>2. _____</li><li>3. _____</li><li>4. _____</li><li>5. _____</li><li>6. _____</li><li>7. _____</li><li>8. _____</li><li>9. _____</li></ol>	<p><i>Team</i></p> <ol style="list-style-type: none"><li>1. _____</li><li>2. _____</li><li>3. _____</li><li>4. _____</li><li>5. _____</li><li>6. _____</li><li>7. _____</li><li>8. _____</li><li>9. _____</li></ol>	<p><i>Team</i></p> <ol style="list-style-type: none"><li>1. _____</li><li>2. _____</li><li>3. _____</li><li>4. _____</li><li>5. _____</li><li>6. _____</li><li>7. _____</li><li>8. _____</li><li>9. _____</li></ol>

*Note: Use this sheet to capture names of team members (each team) during Team Self-Formation Workshop. Managers do NOT create teams on behalf of teams.*



# Product Definition & Exploration Workshop (example)

## PRODUCT CANVAS (1/2)

Product: Squeeky Klean

Organization: Example Inc.

Date: 2019-01

Version: v1.06

**Vision**  
  

Your clear view to your world

**Revenue Streams**  
  

Customer Referrals

Real Estate Agent Referrals

Home-owners

Builder Group Contracts

**Cost Factors**  
  

Liability Insurance

Equipment Costs

Franchise Legal Fees

Local Tax Changes

**Channels**  
  

Real Estate Agency Flyers

Referral Fees

Local Web searches

Groupon Coupons

Promotional Emails

**Product Partner: Business**  
  

VP Product

SK Ops

Temp Workers

**Value**  

Branding

Customer Ratings

Reputation

Reliable Cleaning Staff

Capacity Fulfillment

Reduce Unpaid Invoice Risk

Worker Safety

**Product Partner: Customer**  
  

Home-owners

Land-lords

Real estate agents

Contractors

Schedulers

**Value**  

Not losing time waiting time

Convenience

Competitive Cost

Save Time

Personal Safety

**Product Partner: Technology**  

Handheld Device Partner

SK Tech Support

AWS

**Value**  

Reduce Costs of Onsite Problem Tickets

Profile Security

Tix Through-put

Scalability

**Competitive/Leader Landscape**  

Competition:

Windows R Us

Window Genie

Capability/Threat:

Cleaners are employee

Patented Cleaning Soap

Lowest Cost

Leader:

Sparkle Home

Superior Capability:

High Search Hits

Pre-payment Policy

Mobile App

Home Repair Add-on Services

**Innovation Waves (technology | platforms | standards)**  
  

Cloud Based Data Storage

Handheld payment devices

AI Estimating & Scheduling

PCS Compliance Standard

Ladder Safety Standard

This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License  
Download the Product Canvas at: <https://www.ebgconsulting.com/ProductCanvas.pdf>

DESIGNED BY: Ellen Gottesdiener, EBG Consulting | @ellengott | version 19.01

**ebg**  
ebgconsulting.com

<https://www.ebgconsulting.com/wp/wp-content/uploads/2019/01/SK-Ex-ProductCanvas-1EBG.png>



# Product Definition & Exploration Workshop (example)

## PRODUCT CANVAS (2/2)

Product	Organization	Date	Version
Squeeky Klean	Example Inc.	2019-01	v1.06

### User

**Persona**

Description: To provide convenience with customers and contractors, Squeeky Klean's customers must be friendly, courteous, easy. Besides all activities surrounding customer price, scheduling, scheduling, estimating, scheduling, etc. all in line with Squeeky Klean's goal to be the best in the industry.

Needs: Being ready to assist, up to date customer and property data. Being ready to communicate with customers. Being ready to assist with customer data. Being ready to assist with customer data.

Goals: Provide efficient and personal service. Provide the best service to all customers.

**User Role Map**

**Estimator Prototype**

### Interface

**Context Diagram**

**User Role Map**

**Estimator Prototype**

### Action

**Features**

- Scheduling
- Estimation
- Analytics
- Invoicing
- Inventory Management
- Account Management

**Business Process Diagram**

**Story Map**

### Data

**Data Model**

**State Diagram**

### Control

**Scheduling**

Customer must have no outstanding payments

Service must be an offering

**Cancellation**

Cancellation less than 8 hours prior to schedule....

Preferred customers must not be charged penalty

**Estimation**

Calculate estimate as: ....

Determine discount using (table D)

**Decision Table**

Property type	Total paid services	Discount %
Residential	0-\$100	0%
Residential	\$100-\$150	1%
Residential	\$150-\$200	5%
Residential	>\$200	10%
Commercial	0-\$500	0%
Commercial	\$500-\$1000	5%
Commercial	\$1000-\$1500	10%
Commercial	>\$1500	15%

### Environment

**Onsite**

Handheld: Apple iOS

**Office**

Laptop: Windows Chrome 64.0.324

**S/W Platform**

Anxious Anti-Virus

Cassidy 3500 Router

**Fast Switch 73**

**Open Connect Protocol**

**Architecture Diagram**

### Quality Attribute

**Response Time**

Calculated price shows within 3 seconds of selection

**Availability**

95% M-F EST hours 8:30 am - 6 pm

**Recoverability**

2 hours MTR



This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License  
Download the Product Canvas at: <https://www.ebgconsulting.com/ProductCanvas.pdf>

DESIGNED BY: Ellen Gottesdiener, EBG Consulting | @ellengott | version 19.01

<https://www.ebgconsulting.com/wp/wp-content/uploads/2019/01/SK-Ex-ProductCanvas-2EBG.png>

ebg

ebgconsulting.com



# Product Definition & Exploration Workshop

Notes:



# Simulation: Product, Definition, DoD, Team

**<https://www.keystepstosuccess.com/2020/11/less-trainers-class-experience-report-product-definition-dod-team-blueprint-exercise/>**



# Case Studies



# What is Next?



or



*It depends*



Site:

<https://less.works/>

Discussion Group:

<https://groups.google.com/forum/#!forum/largescalescrum>

Twitter:

@less\_works

Facebook <https://www.facebook.com/less.works>



# Appendix



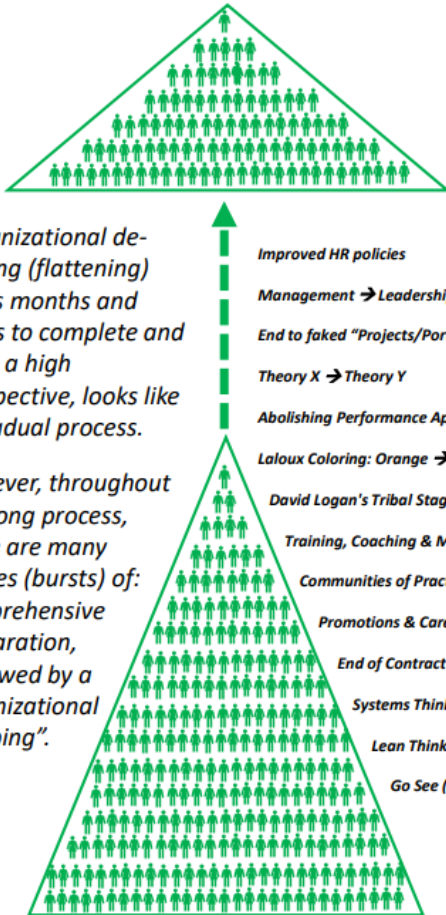
# Appendix

## 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**.

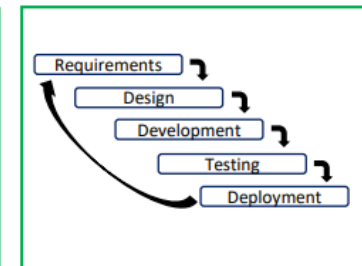
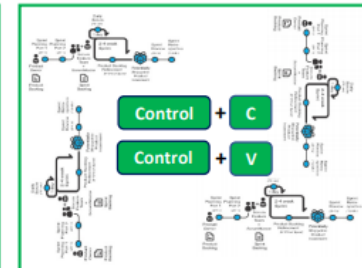
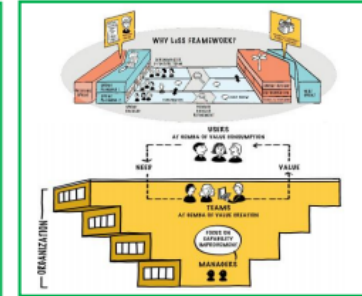
*Transition from independent basic Scrum, performed by multiple teams to LeSS, is a desirable approach, when a product is widely defined and real Product Owner is identified*

### 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



# Appendix

## How to ensure effective Coaching?



Senior Leadership needs to understand what organizational adaptiveness (agility) really means. What systemic implications of “going agile” really are? Leadership needs to be ready and comfortable to take a close look at an existing organizational design, norms, values, principles. There should be willingness not only to “bless others, in spirit” and delegate an entire effort to organizational layers beneath but get really involved, do real Gemba (“Go See”) – go, where real action is.

Once senior leadership is properly informed and has all necessary information to make a conscious and responsible decision – they give an “informed consent” (of course, it is a figure of speech) to one another, their respective organizational verticals and organizational design consultants that were hired to assist an organization in its journey. There must be an explicit agreement that everyone agrees ☺.

Properly crafted and consistent messages are communicated directly from the top of an organization, all throughout organizational layers and domains of control. Messages must include reasons, purpose/goals and desirable outcomes. Not just ‘what’ but ‘why’. People throughout an entire organization need to clear understanding what upcoming changes mean to them, career-wise and personally. Inappropriate messages may cause confusion and misalignment among people, as well as introduce an element of unsafety.

All organizational functions (ideally, whole departments) should be involved, without exception. For example, agile development is best to be done, based on flexible budgeting and rolling wave forecasts. Therefore, Finance needs to get involved. Organizational re-design, team-based performance, abolishing some old/outdated evaluation methods of individuals may require involvement of HR.

Everything that an organization (or organizational unit) works on has to be assessed: products, services, applications, platforms. This has to be done from a standpoint of a ‘buying customer’ and along the lines of business value flow. This means that existing, projects, programs and portfolios should be looked at from a stand-point of “what would a customer want to carve out for herself, if she was given a scalpel”. Frequently, it means that existing components, sub-components or sub-product centric containers of work are challenged.

Corollary to the above, Teams should be structured (or re-structured) in ways that best supports customer-centric work. It means that existing groups, ‘teams’ and organizational spheres of control (e.g. by line manager or team lead) are reassessed and possibly changed

Once teams are properly structured, they are put through comprehensive, structured training that best supports their future needs. Ideally, Teams should be trained alongside with their business counterparts. If multiple Teams are to be working closely together (e.g. scaled scrum, for the same product owner, on the same widely defined product), they should be also trained together. Training is typically superseded by mandatory self-study (reading) and other preparatory work. “Not compromised training – means that training content is not ‘weakened’ to fit organizational “uniqueness”

This is a prolonged phase that consists of on-going support by a coach who uses various coaching tools, techniques and methods to enable Team(s) become efficient and independent. This includes Team coaching, individual coaching and role playing (especially, at the beginning), etc. In more complex environments, if access to senior leadership is granted, it also includes organizational coaching.

Gradually, a coach will strive to ‘coach himself out of a job’ and give autonomy and control back to Team(s). Coaching intensity usually changes over time (is lowered, as time goes by and team(s) become more mature).



# Appendix

## Agile Self-Assessment Worksheet (ASAW) - 1

### Organizational Credo, Values, Strategy and System-Optimizing Goals

10

**Guideline:** List some most commonly heard messages, slogans and motivational statements, syndicated throughout your organization by senior management.

...and then decide if state/status of each of the below domains further supports the above....

#### Organizational Structure/Design

2

**Tip:** Is your organization flat/simple or multi-tiered/complex? How much emphasis does your organization put on centralized, top-down control, subordination and compliance?

<http://www.keystepstosuccess.com/articles-organizational-design/>

#### Organizational Culture

3

**Tip:** What types of relationships and behaviors prevail among employees in your organization?

<http://www.keystepstosuccess.com/articles-corporate-psychology/>

#### Budgeting & Finance

4

**Tip:** Does your organization do budget planning, based on adaptive business cycles or fixed cycles (fiscal year-end)?

<http://www.keystepstosuccess.com/2016/08/implementing-beyond-budgeting/>

#### HR Policies

5

**Tip:** What does your organization value more: individual performance or team performance? Are individual performance appraisals still in effect?

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

#### Sales & Marketing

6

**Tip:** Does your organization create sales plans and conduct marketing campaigns? Is internal competition encouraged?

[http://www.keystepstosuccess.com/wp-content/uploads/2019/03/s-way\\_2019\\_framework.pdf](http://www.keystepstosuccess.com/wp-content/uploads/2019/03/s-way_2019_framework.pdf)

#### Globalization/Site Strategies

7

**Tip:** How much emphasis does your organization put on cheap human resources? What factors define your site strategy?

<http://www.keystepstosuccess.com/2019/01/survival-list-to-vendor-selection-on-agile-projects/>

#### KPIs & Metrics

8

**Tip:** How much emphasis does your organization put on metrics and KPIs? How many organizational layers do metrics have to pass, on their way from a source (e.g. teams) to destination (e.g. senior leadership)?

<http://www.keystepstosuccess.com/2017/10/addressing-problems-caused-by-amms/>

#### Relationships with Vendors/Partners

9

**Tip:** Does your organization do proper vetting and assessment of vendors/third parties, before engaging with them on agile initiatives?

[http://www.keystepstosuccess.com/wp-content/uploads/2019/04/Survival\\_Sheet\\_Vendor\\_Selection\\_Agile\\_Projects.pdf](http://www.keystepstosuccess.com/wp-content/uploads/2019/04/Survival_Sheet_Vendor_Selection_Agile_Projects.pdf)

### Agile Guidance & Leadership

1

**Tip:** Do you have coaches, trainers and guide-level experts available, to assist you in your journey? <https://www.infoq.com/articles/agile-coaching-lessons> | <http://www.keystepstosuccess.com/2017/07/you-get-what-you-ask-for-agile-coaches-centaurs/>

Once you thoroughly study and fully understand the below fundamentals, please assess various domains of your Organizational Ecosystem in the above blocks...

#### 4 Agile Manifesto Postulates & 12 Supporting Principles

<https://agilemanifesto.org>

#### System Thinking

[https://en.wikipedia.org/wiki/Systems\\_theory](https://en.wikipedia.org/wiki/Systems_theory)

#### Lean Thinking

[https://en.wikipedia.org/wiki/Lean\\_thinking](https://en.wikipedia.org/wiki/Lean_thinking)

#### Design Thinking

[https://en.wikipedia.org/wiki/Design\\_thinking](https://en.wikipedia.org/wiki/Design_thinking)

Direction of Self-Assessment



# Appendix

## Agile Self-Assessment Worksheet (ASAW) - 2

*Additional Notes:*

1	
2	
3	
4	
5	
6	
7	
8	
9	
10	

Sourced from: <https://www.keystepstosuccess.com/coach-tools/>

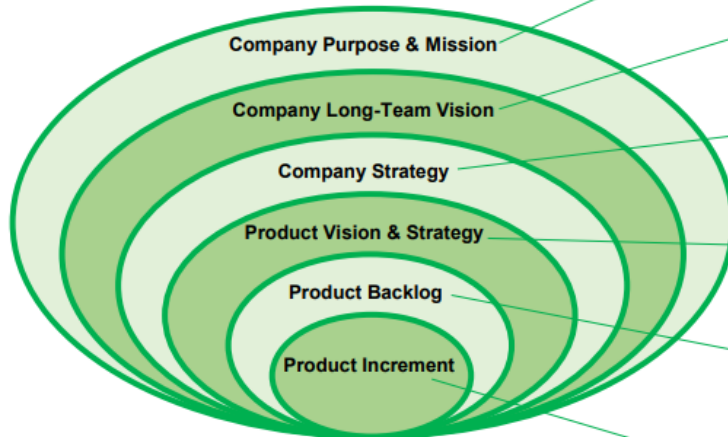


# Appendix

## Product Definition Expansion Worksheet

### What to do:

- Discuss what each 'onion' layer below means in the context of your organization
- Modify the 'onion', based on organizational specifics (add/remove layers)
- In respective areas on the right side, provide some examples and details for each layer
- Discuss and agree where to keep information (e.g. tool, directory, other repository)
- Discuss and agree how 'onion' layers will be maintained and updated over time



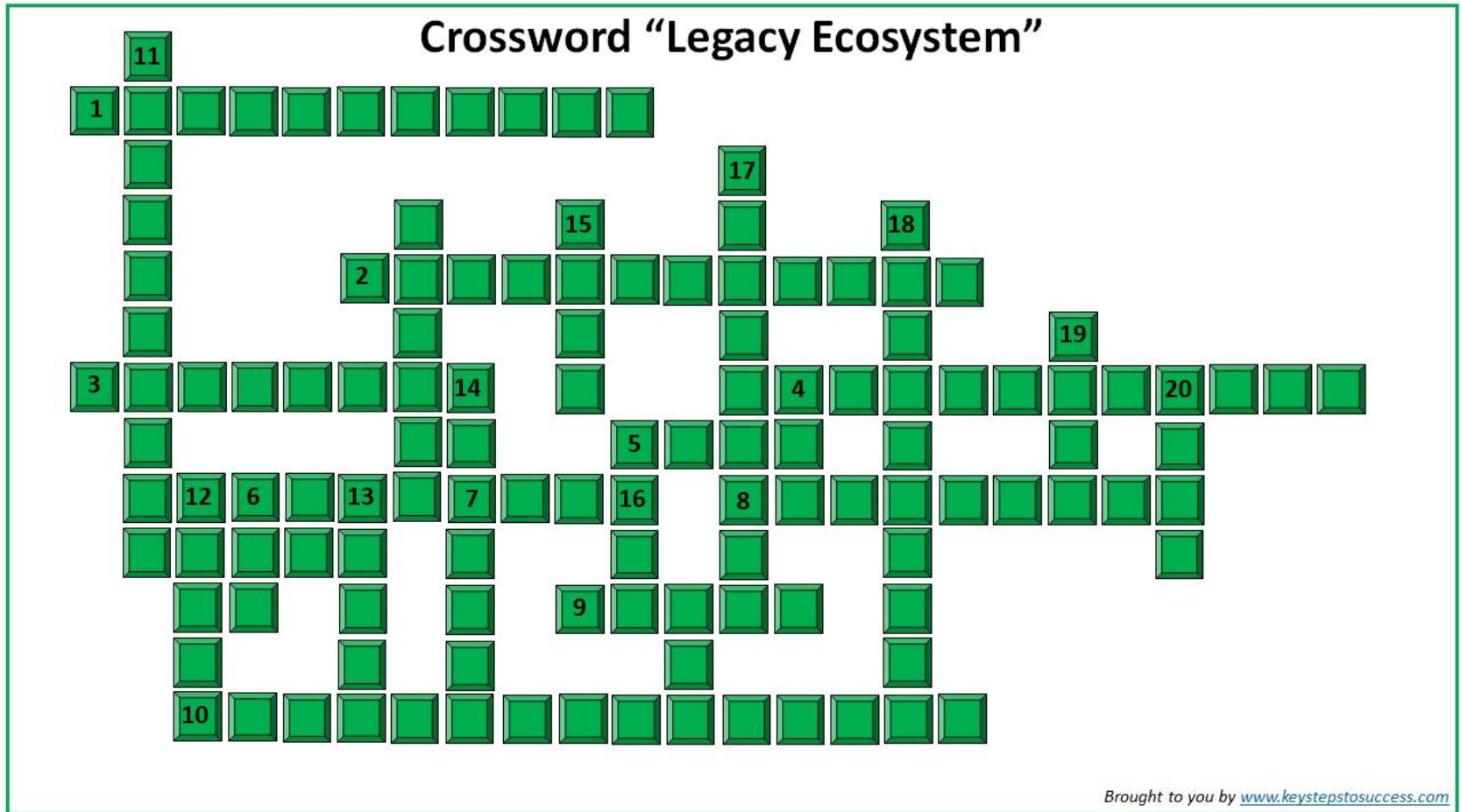
Sourced from: <https://www.keystepstosuccess.com/coach-tools/>







# Appendix



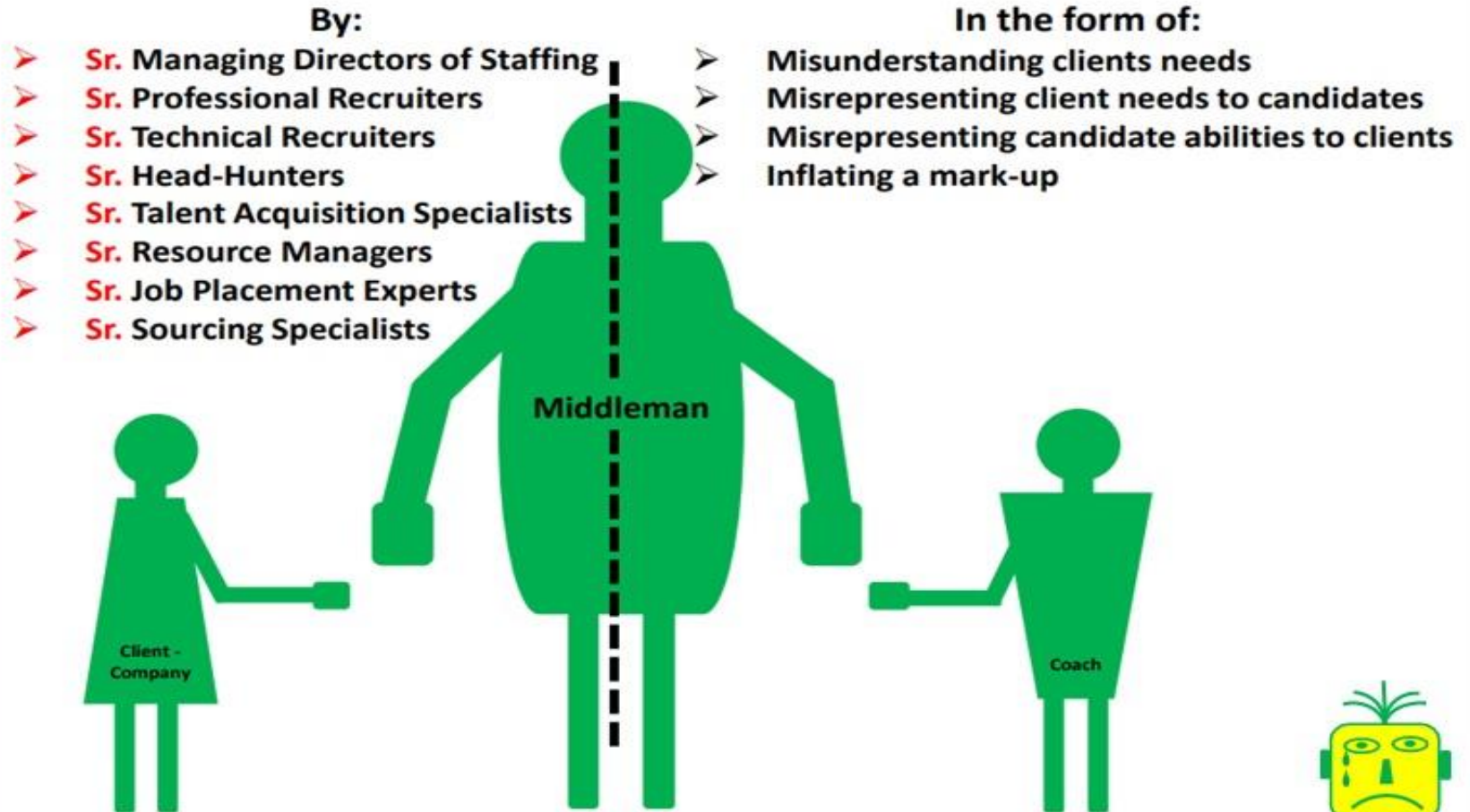
[http://www.keystepstosuccess.com/wp-content/uploads/2020/02/crossword\\_legacy\\_ecosystem.pdf](http://www.keystepstosuccess.com/wp-content/uploads/2020/02/crossword_legacy_ecosystem.pdf)

Sourced from: <https://www.keystepstosuccess.com/coach-tools/>



# Appendix

## Middleman Problem: Low (very) Value Proposition



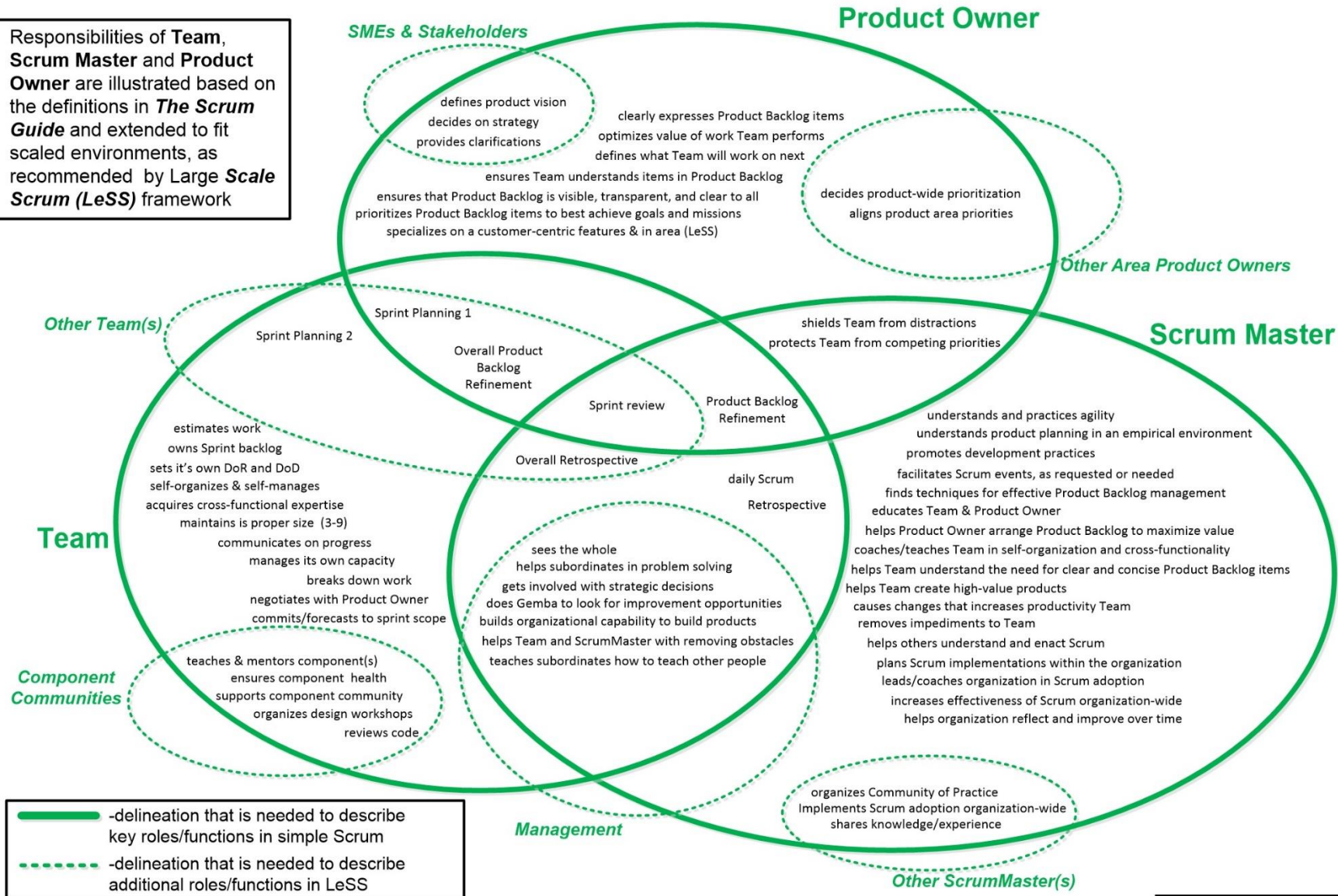
Avoid This:

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



# Appendix

Responsibilities of **Team**, **Scrum Master** and **Product Owner** are illustrated based on the definitions in *The Scrum Guide* and extended to fit scaled environments, as recommended by Large *Scale Scrum (LeSS)* framework



Gene Gendel, Certified Enterprise Coach [www.keystepstosuccess.com](http://www.keystepstosuccess.com)

Sourced from: <https://www.keystepstosuccess.com/coach-tools/>





team : sketch a system model, considering this **puzzle**:

- > **“We don’t have time to create clean code, because we are too busy going slow because of dirty code.”**
- > start with these variables; write the **bold words** verbatim, and put them in a big **circle** on the wall, in their **“clock position”**
  1. **% clean code** (place this at **1 o’clock** in the circle)
  2. **effort required to craft clean code** (2 o’clock)
  3. **effort applied to craft clean code** (*applied vs required: not same*)
  4. **effort required to create a new feature**
  5. **velocity** (...of delivering new features)
  6. **# technical defects** (vs. requirement defects)
  7. **effort required to handle defects** (technical & requirement)
  8. **effort applied to handle defects** (technical & requirement)
  9. **pressure to deliver and “go faster”**
  10. **dev fear**





team: sketch a system model, bearing in mind this **puzzle**:

- > **1 product, many teams, each team has a Team “Product Backlog” prioritized by a Team “Product Owner”**
- > **start** with these variables **verbatim**
  1. **# backlogs/lists** (e.g. 1 backlog per team (e.g. 5 if 5 teams), 1 backlog for 2 teams, 1 backlog for all teams)
    - *put this variable physically dead center of a star pattern*
  2. **% of total (product) items a team knows well (requirements & design)**
    - *put this variable at “12 o’clock”*
  3. **adaptiveness (i.e. agility) of teams to change direction at global level (i.e. txn cost & switching cost (AKA cost of change)) ... at “3 o’clock”**
  4. **% of items worked on each Sprint that are highest value at global level ... at “6 o’clock”**
  5. **likelihood that a single team will see they may be working on low-value items at global level ... at “8 o’clock”**
  6. **local team identity ... at “10 o’clock”**





team

- > sketch a systems model, bearing in mind this **puzzle**:
  - > **“I wonder about the impact of narrow versus broad product definitions on the system optimizing goal?”**
- > start with these variables verbatim:
  1. **size (breadth) of product definition** (right, middle level)
  2. **# products** (right, below)
  3. **product complexity (re. tech & requirements)** (right, above)
  4. # backlogs/lists
  5. % of items worked on each Sprint that are highest value at global level
  6. adaptiveness of teams to change direction at global level





## Coach & group (or per Team)

- > **\*\*\* first, synch with the last debrief model \*\*\***
- > sketch a systems model, bearing in mind this **puzzle**:
  - > **“I wonder about impact of size of product definition on end2end customer-feature cycle time?”**
- > start with these variables verbatim: **(lower right quadrant)**
  1. **# of inter-group/team task dependencies** (i.e. a group probably has to wait for another group to do “their part”)
  2. **strength of “private code” policies**
  3. **average complete end2end customer feature cycle time**





team

- > **\*\*\* first, synch with the last debrief model \*\*\***
- > start with these variables, and given this **puzzle**:
- > **“What happens to people’s heads as product gets broader?”**
  1. **cognitive “fullness” of one Product Owner to re-prioritize and have whole-product overview** (e.g. my head hurts! it’s full!) (far right, top)
  2. **cognitive “fullness” of people in teams to know N items** (i.e. PB items) (far right, lower)
  3. **heterogeneity in an item** (tech & domain) (far right, lower)
  4. **average size of item a team implements** (far right, lower)
  5. size (breadth) of product definition





coach

> other variables of relevance to last exercise?

1. **# items to re-prioritize each Sprint**  
(at least enough for Sprint Planning)

2. **# teams**





## COACH & group (or per team)

- > \*\*\* **first, synch with the last debrief model** \*\*\*
- > start with these variables, and given this **puzzle**:
- > **“How big should RAs be? What are the sub-optimizations from adding RAs?”**
- > (lower left quadrant of wall)
  1. **size (breadth) of a RA definition**
  2. **# RAs**
  3. **# teams in a RA**
  4. **# backlogs/lists**
  5. **size (breadth) of product definition**

265





## COACH & group

- > which is more consistent with the *system goal* of adaptiveness?
- > Requirement Area **Bond Trading: 4** teams, **AND** ... Requirement Area **Stock Trading: 4** teams
- > ... or ...
- > Requirement Area **Bond & Stock Trading: 8** teams

268

268





## COACH & group, or per TEAM

- > sketch a systems model, considering this **puzzle**:
  - > **“We are ‘scaling agile’. So, somebody told us we need ‘agile’ portfolio management.”**
- > start with these variables (**left area of wall**):
  1. **need for and activities of “portfolio management”**
  2. **# people involved in “portfolio management”**
  3. **ease of *first making* & executing large-direction decisions**
  4. **ease of *changing* & executing large-direction decisions**
  5. **# programs/projects**
  6. size (breadth) of product definition
  7. # of products





coach

- > if LeSS and if there is a **broad** product definition, **who** makes large direction decisions in the product?





team

- > sketch a systems model, considering this scenario **puzzle**:
  - > **1 product, 1 Product Backlog, many teams**
  - > **1 *real* Product Owner** re-prioritizes the **1 Product Backlog** (no team-level “product backlogs”)
  - > each Team has a so-called (team) **“Product Owner”**, who is *not* doing hands-on development, AND/OR there are many separate **“Product Owner helpers/proxies”** **“Technical Product Owners”** or **“analysts”** or **“Product Managers”** who are not doing development

396

396





team

› start with these variables

1. **# so-called "Product Owners"** (and PO "helpers/proxies" or "Product Managers" or "analysts")
  - **lay out the cards in a circle pattern, with this at the center, in some empty region of the wall**
2. **likelihood so-called "Product Owners" are doing lots of analysis & talking to users or other "POs"**  
**"2 o'clock"**
3. **likelihood so-called "Product Owners" create intermediate artifacts** **"3 o'clock"**
4. **wastes** (e.g. overproduction, inventory/WIP, handoff, information scatter, waiting, ...) **"4 o'clock"**
5. **likelihood developers are doing lots of analysis & talking to customers/users** **"5 o'clock"**
6. **skill of developers to communicate effectively with C/Us** **"6 o'clock"**
7. **degree developers have empathy and awareness of C/Us** **"7 o'clock"**
8. **degree developers understand the business domain** **"8 o'clock"**
9. **degree developers can independently make informed fine-grained decisions** **"9 o'clock"**
10. **degree features "hit the mark" of what C/Us really need** **"10 o'clock"**
11. **degree so-called "Product Owners" are a bottleneck** **"11 o'clock"**
12. **degree developers are prevented from talking to C/Us** **"12 o'clock"**
13. % of total (product) items a team knows well (requirements & design)
14. average complete end2end customer feature cycle time
15. % of items worked on each Sprint that are highest value from a global view
16. (if you have this variable) effort required to learn new items





## TEAM

- > sketch a systems model, considering this **puzzle**:
  - > **large, detailed (roles, processes, artifacts, ...) framework with many claimed “best practices” is pushed onto group by managers or consultants**
  - > start with these variables (**somewhere on wall**):
    1. **amount of detail, roles, processes, artifacts, prescription & “best practices”**
    2. **degree that new FW conveys feeling of “control” and “familiarity” similar to traditional model**
    3. **degree of feeling of ownership and engagement by hands-on people in their processes & structures, and in improving them**
    4. **degree of *learning* organization**
    5. **degree of *copying* organization**
    6. **degree of acceptance of specious arguments** (e.g. “best practices”, “gurus”, “sacred texts”, “they do it”, ...)
    7. **degree of fear if dissent**
    8. **degree of unnecessary or inappropriate processes & practices**
    9. **adaptiveness at global level** (new variable summarizing the existing ones)
    10. % of items worked on each Sprint that are highest value at global level
    11. average complete end2end customer feature cycle time





## TEAM

- > sketch a systems model, considering this **puzzle**:
  - > **large, detailed (roles, processes, artifacts, ...) scaling framework with many claimed “best practices” is pushed onto group by managers or consultants, then group is asked to tailor it down**
- > start with these variables:
  1. **degree of explicit/implicit goal to remain similar to status quo**
  2. **expectation to “get our money’s worth” by installing many FW elements**
  3. **expertise by group to customize the FW**
  4. **degree of desire to shift blame “to the FW” when problems**
  5. **size of FW adoption and ability to shift blame to it**
  6. **amount of actual simplification (tailoring down) of FW**

613



Sourced from creative commons repository of Certified LeSS Trainers: <https://less.works/courses/become-less-trainer>

*Gene Gendel, Certified Enterprise & Team Coach (CEC-CTC), Certified LeSS Trainer (CLT)*