LeSS

More with LeSS

Agile Gurukul all rights reserved.

Certified LeSS Practitioner Training

With Dinesh Sharma

Based on material created by Bas Vodde and Craig Larman

dinesh@agilegurukul.com

#DineshSharma_

Agile Gurukul all rights reserved.

Who am I?

Name: Dinesh Sharma

Originally from India (Delhi)

Lives in England

Organisation Change Consultant & Life Coach

Experience working with Large Scale Product Development

dinesh@agilegurukul.com

lastminute.com SOCIETE GENERALE Cognizant gamesys

Raytheon SITA BARCLAYS Argos

Henley Business School UNIVERSITY OF READING

CERTIFIED LeSS TRAINER

CERTIFIED LeSS PRACTITIONER

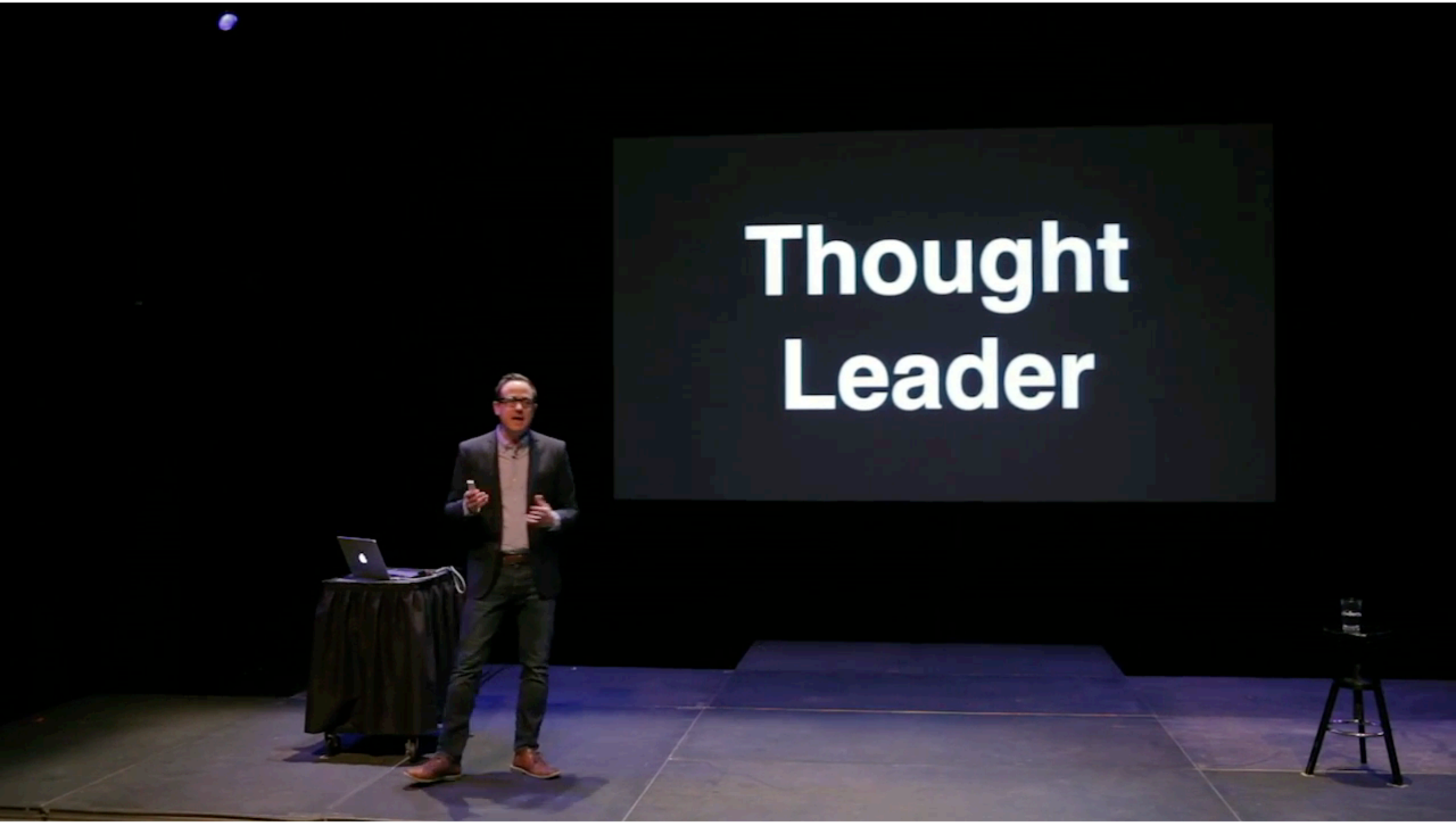
Agile Gurukul all rights reserved.

Don’t believe anything I say

fads & gurus -> **insight**

rent -> **own**

Agile Gurukul all rights reserved.



Agile Gurukul all rights reserved.

Agenda

- Introduction
- Principles
- Overview
- Organised by Customer Value

- Product
- Definition of Done
- Product Owner and Product Backlog

- Sprint Planning, Review Retrospective
- Integration and Coordination
- Management and Orientation
- Adoption
- Scrum Master

Agile Gurukul all rights reserved.

Introduction

Agile Gurukul all rights reserved.

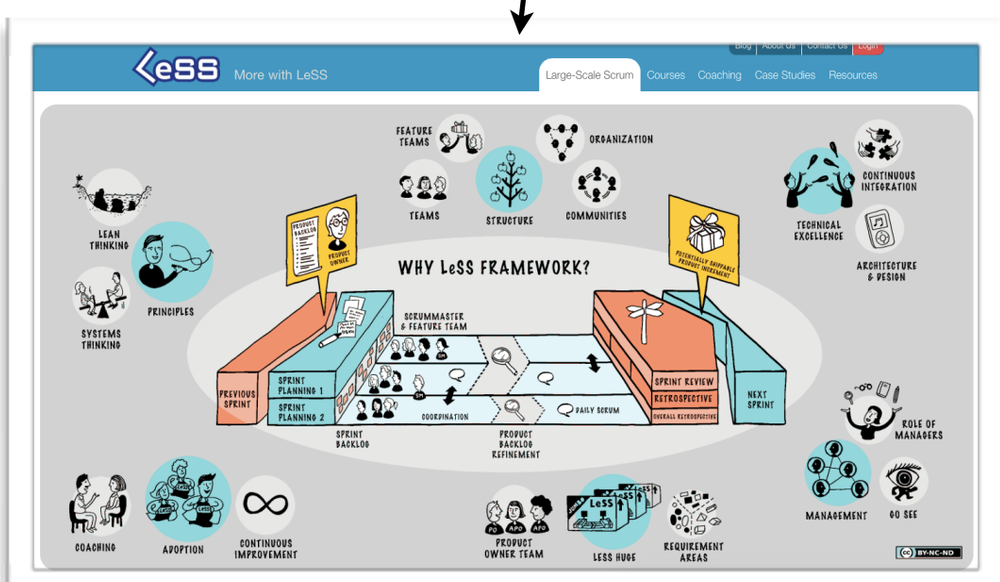
Before we start

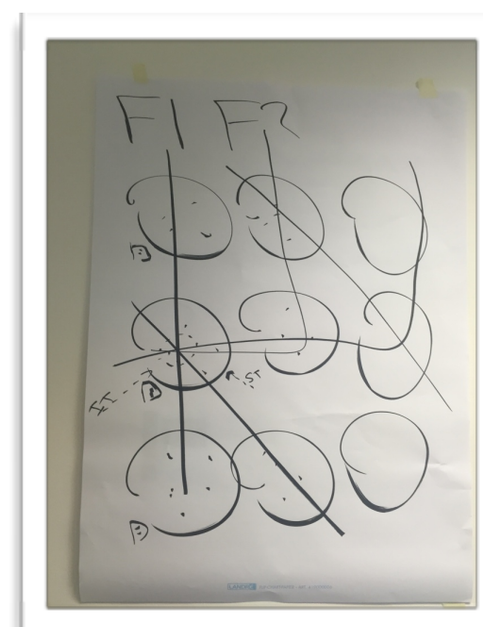
- Certified LeSS Practitioner -> The LeSS Company
- Not a Scrum course …
- Q&A and Reflection

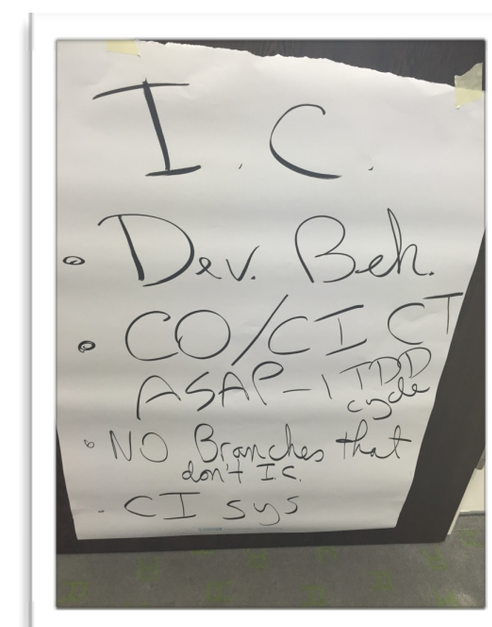
- Practicalities
  - No computers
  - Phones silent, no SMS
  - Photos and LeSS site
  - Lunch, breaks
  - Twitter
    - Hashtag: #LeSSWorks
    - Account: @Less_works

Agile Gurukul all rights reserved.

Course **PDF** & Room **Photos** at **less.works**







Agile Gurukul all rights reserved.

Who are you?

Questions

Your name?
Work history?
What you do at the company?
How does your team build products?
Experiences with LeSS?
Size of your product (in people)
Something else interesting

Agile Gurukul all rights reserved.

Exercise

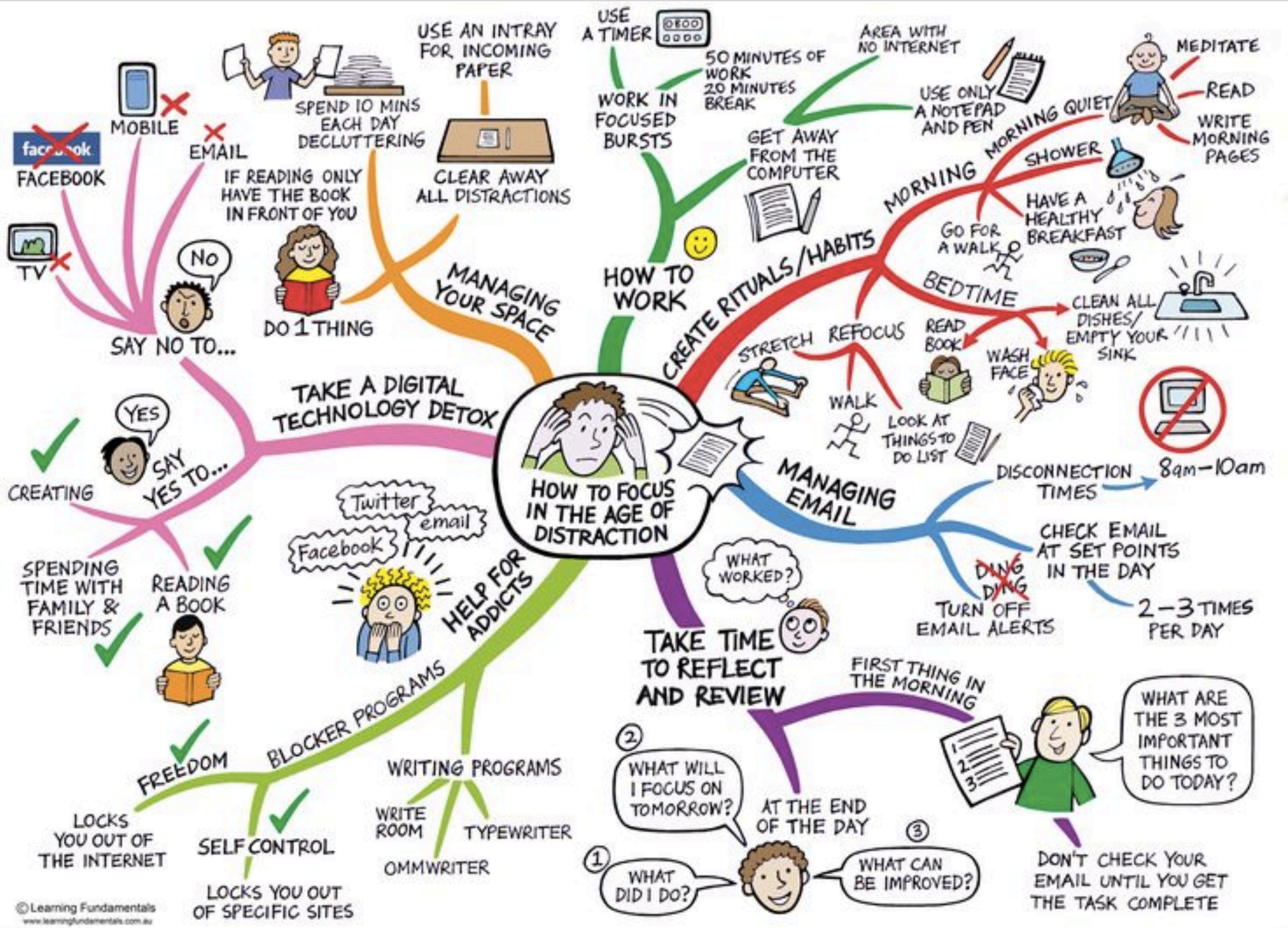
Must Cover Backlog on the wall

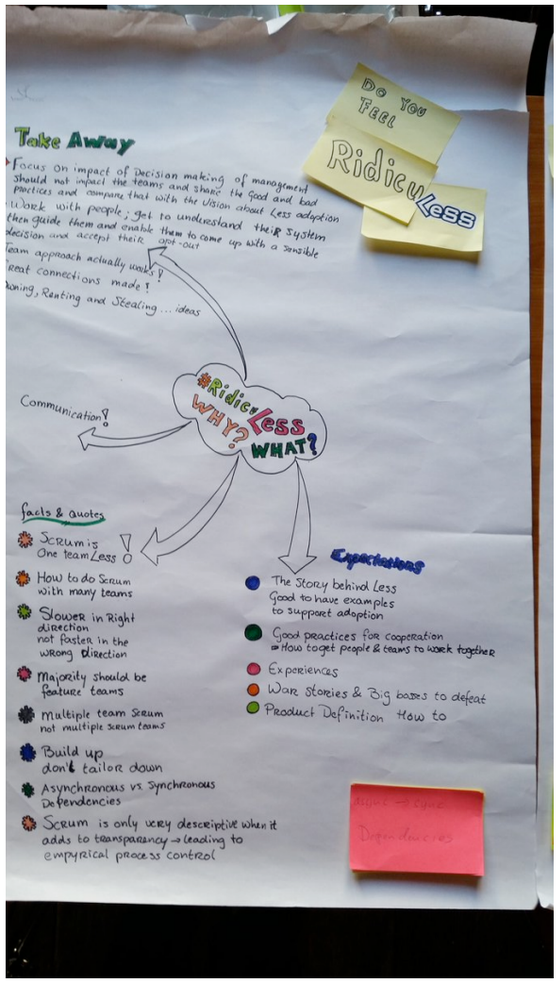
Agile Gurukul all rights reserved.

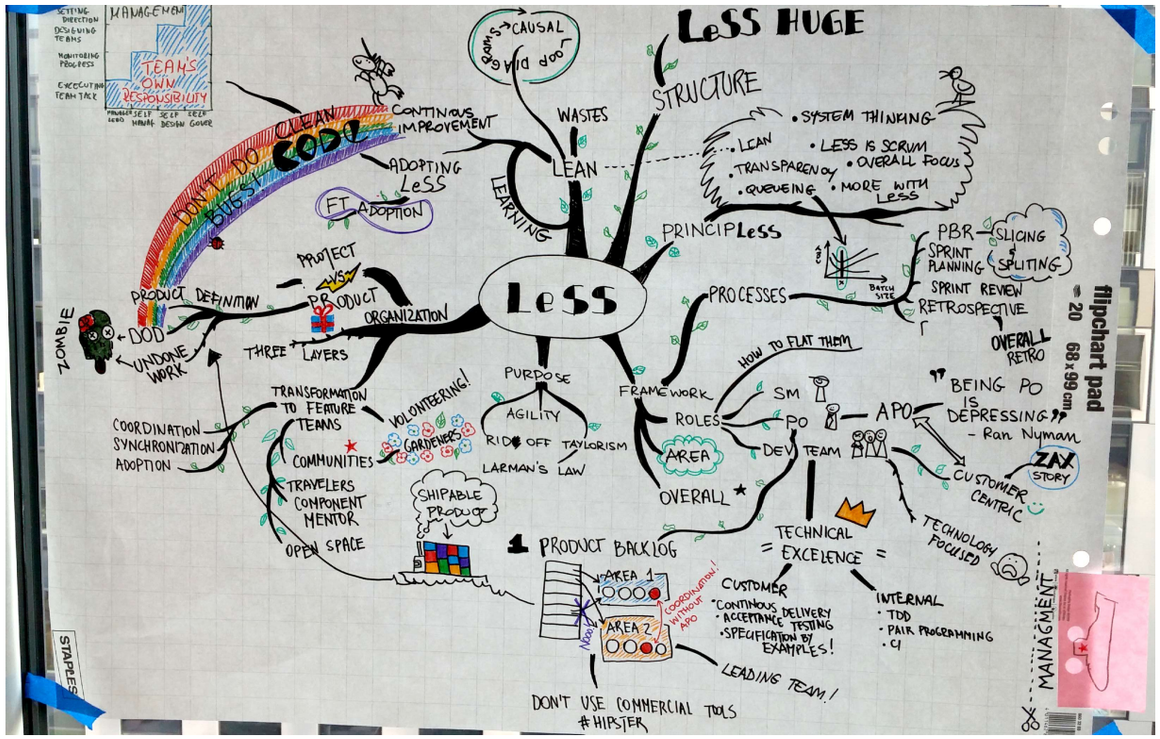
Likely Objectives: You can…

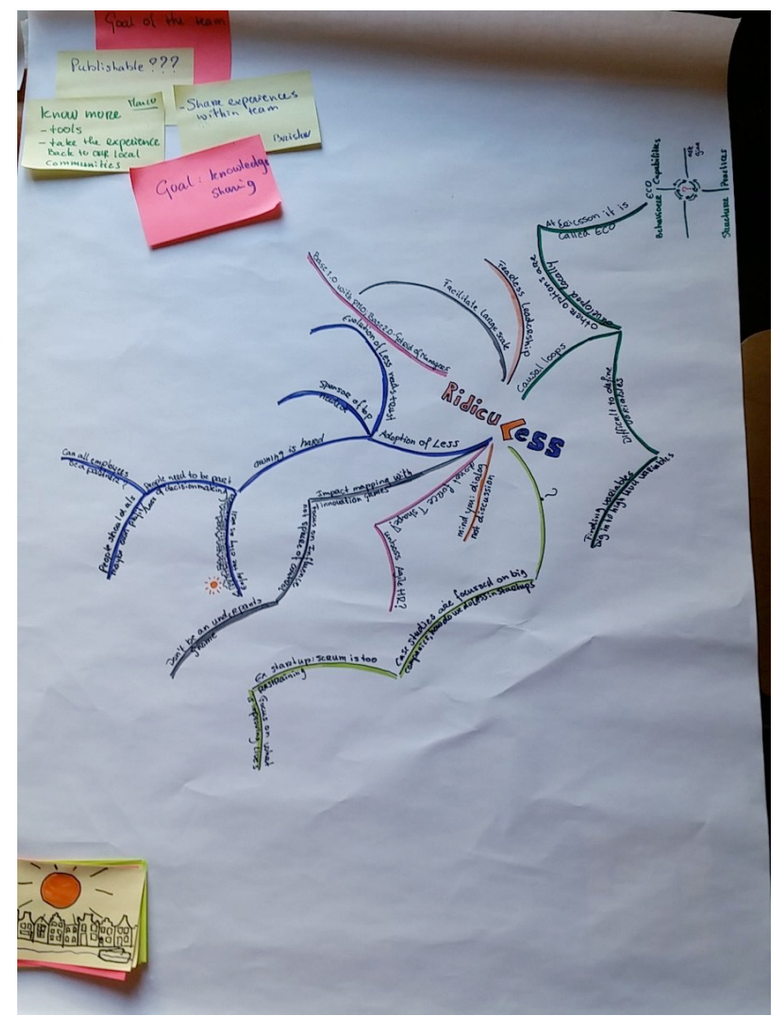
- **apply systems thinking**
- **motivate & define LeSS org design by local -> global optimisation systems analysis**
- **define a product**
- **advise on LeSS adoption**
- **explain LeSS & LeSS Huge frameworks**
- **explain several LeSS principles & make connections**
- **answer "why LeSS?"**
- **explain roles**
- **know & coach LeSS Sprint** (e.g. events, coordination, …)

Agile Gurukul all rights reserved.


HOW TO FOCUS IN THE AGE OF DISTRACTION
TAKE A DIGITAL TECHNOLOGY DETOX
SAY NO TO...
No
MOBILE
EMAIL
FACEBOOK
TV
SPEND 10 MINS EACH DAY DECLUTTERING
USE AN INTRAY FOR INCOMING PAPER
CLEAR AWAY ALL DISTRACTIONS
IF READING ONLY HAVE THE BOOK IN FRONT OF YOU
DO 1 THING
MANAGING YOUR SPACE
SAY YES TO...
Yes
CREATING
SPENDING TIME WITH FAMILY & FRIENDS
READING A BOOK
HOW TO WORK
USE A TIMER
WORK IN FOCUSED BURSTS
50 MINUTES OF WORK 20 MINUTES BREAK
GET AWAY FROM THE COMPUTER
AREA WITH NO INTERNET
USE ONLY A NOTEPAD AND PEN
CREATE RITUALS/HABITS
MORNING
MORNING QUIET
MEDITATE
READ
WRITE MORNING PAGES
SHOWER
HAVE A HEALTHY BREAKFAST
GO FOR A WALK
BEDTIME
CLEAN ALL DISHES/ EMPTY YOUR SINK
READ BOOK
WASH FACE
STRETCH
REFOCUS
WALK
LOOK AT THINGS TO DO LIST
MANAGING EMAIL
DISCONNECTION TIMES
8am–10am
CHECK EMAIL AT SET POINTS IN THE DAY
2–3 TIMES PER DAY
TURN OFF EMAIL ALERTS
DING
DING
HELP FOR ADDICTS
Twitter
email
Facebook
BLOCKER PROGRAMS
FREEDOM
LOCKS YOU OUT OF THE INTERNET
SELF CONTROL
LOCKS YOU OUT OF SPECIFIC SITES
WRITING PROGRAMS
WRITE ROOM
TYPEWRITER
OMMWRITER
TAKE TIME TO REFLECT AND REVIEW
WHAT WORKED?
FIRST THING IN THE MORNING
WHAT ARE THE 3 MOST IMPORTANT THINGS TO DO TODAY?
DON'T CHECK YOUR EMAIL UNTIL YOU GET THE TASK COMPLETE
AT THE END OF THE DAY
1 WHAT DID I DO?
2 WHAT WILL I FOCUS ON TOMORROW?
3 WHAT CAN BE IMPROVED?
© Learning Fundamentals
www.learningfundamentals.com.au







Agile Gurukul all rights reserved.

Exercise- Create Initial Mind Map

- apply systems thinking
- motivate & define LeSS org design by local -> global optimisation systems analysis
- define a product
- advise on LeSS adoption
- explain LeSS & LeSS Huge frameworks
- explain several LeSS principles & make connections
- answer "why LeSS?"
- explain roles
- know & coach LeSS Sprint (e.g. events, coordination, ...)

Agile Gurukul all rights reserved.

Overview

Agile Gurukul all rights reserved.

Agile Manifesto

Agile Gurukul all rights reserved.

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 Taylorised, 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

Agile Gurukul all rights reserved.

Konosuke Matsushita (2)

"We have passed the Taylor stage. We are aware that business has become terribly complex. Survival is very uncertain. Therefore, a company must have the constant commitment of the minds of all of its employees to survive. For us, management is the entire workforce's intellectual commitment at the service of the company.

We know that the intelligence of a few technocrats—even very bright ones—has become totally inadequate to face these challenges. Only the intellects of all employees can permit a company to live with the ups and downs and the requirements of its new environment. Yes, we will win and you will lose. For you are not able to rid your minds of the obsolete Taylorism that we never had."

Panasonic

Agile Gurukul all rights reserved.

Taylorism

or Scientific Management

A theory of management that analyses and synthesises workflows. Its main objective is improving economic efficiency, especially labour productivity.

Do we want to be a slave to Taylorism?

Frederick Winslow Taylor
The Principles of Scientific Management (1909)

There is **no question** that the cost of production is lowered

by **separating** the work of **planning** and the **brain work** as much as possible

from the **manual labor**

Agile Gurukul all rights reserved.

One Right Solution

Assuming that we have it and we just need to train it to the employees

Agile Gurukul all rights reserved.

End of Taylorism

"Wouldn't it be great if we could achieve high levels of coordination without a supervisory super-structure? Wouldn't it be terrific if we could get the freedom and flexibility of an open market with the control and coordination of a tightly knit hierarchy? If only we could manage without managers."

-HBR DECEMBER 2011 Gary Hamel

But a lot has changed Since the 1800's

We are now in the age of the knowledge worker

Agile Gurukul all rights reserved.

Exercise

Do you have division of work in your current organisation and what problem does it cause?

Agile Gurukul all rights reserved.

There is no blame!

"Peoples' behaviour (the culture) is a product of the system; when you change the system, peoples' behaviour changes."

-Culture Change is Free: This article by John Seddon was first published by Quality World in 2009

Agile Gurukul all rights reserved.

Larman's Laws of Organizational Behavior

1. **Organisations are *implicitly* optimised 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", "religious", and "needing pragmatic customisation 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, reinforcing (2) and (3).**

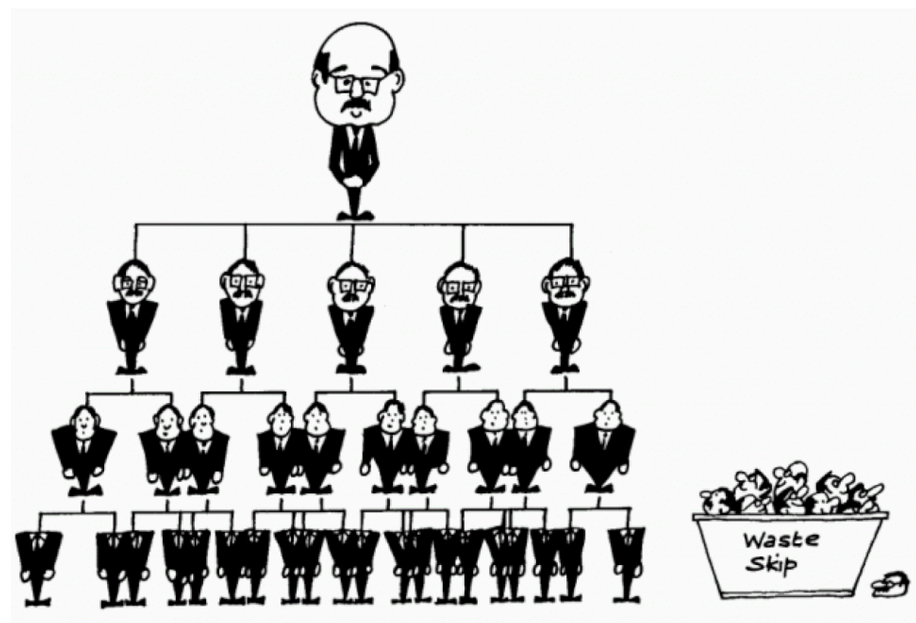
5. **Culture follows structure (or behaviour/mindset follows system)**

Agile Gurukul all rights reserved.

Therefore

We need to focus on something that affect system.

Structure



LeSS is an organisational design framework
not a way to have more effective R&D

Agile Gurukul all rights reserved.

Taylorism **is not** a good idea.

Process prescription **is not** a good idea

LeSS **is not** a good idea

Agile Gurukul all rights reserved.

Except when…

Agile Gurukul all rights reserved.

Shu - Ha - Ri

守

Apprentice = Follow the rules and practice until you reach a good level

離

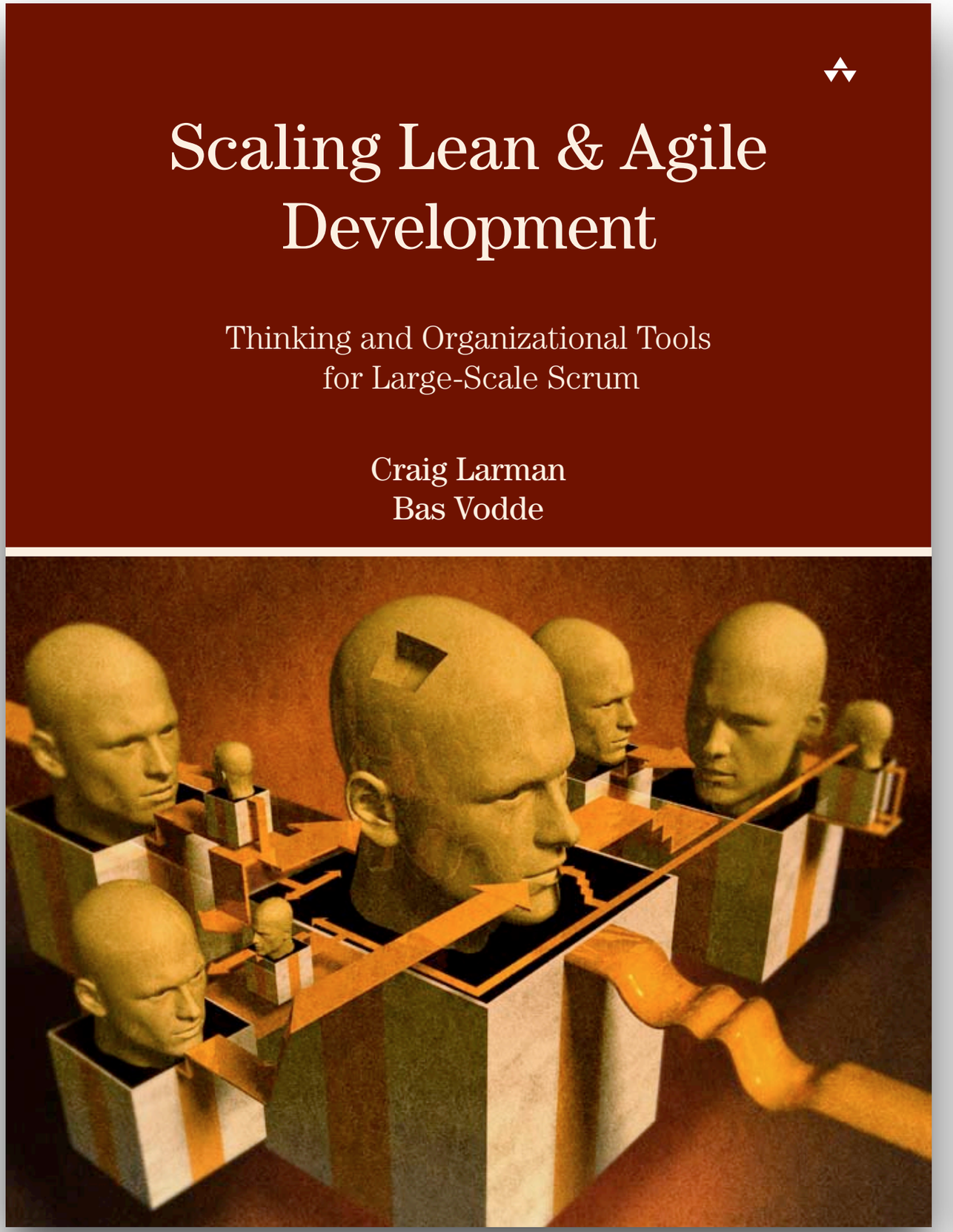
Journeyman = Start using the rules and use a sensei to explore improvement options

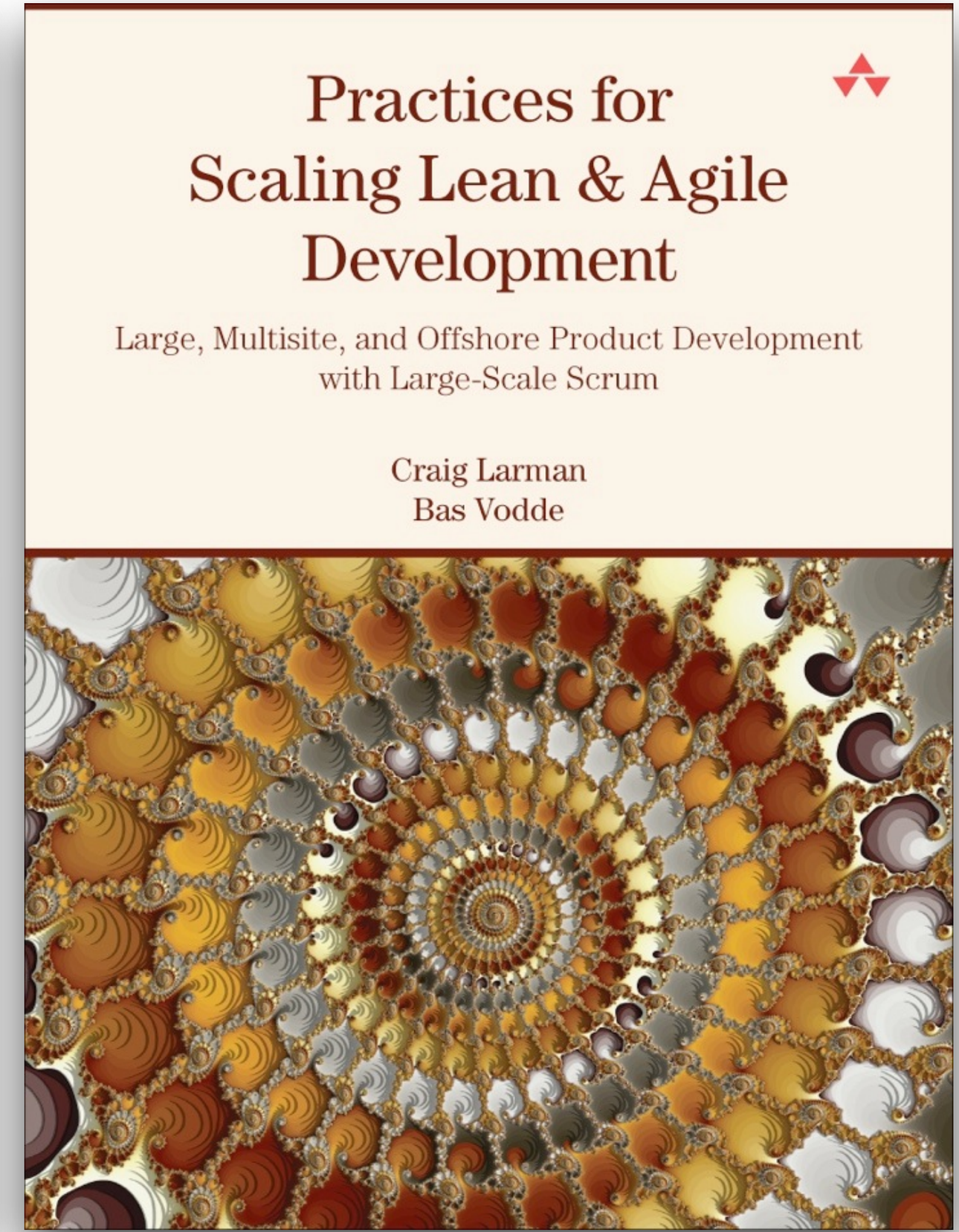
破

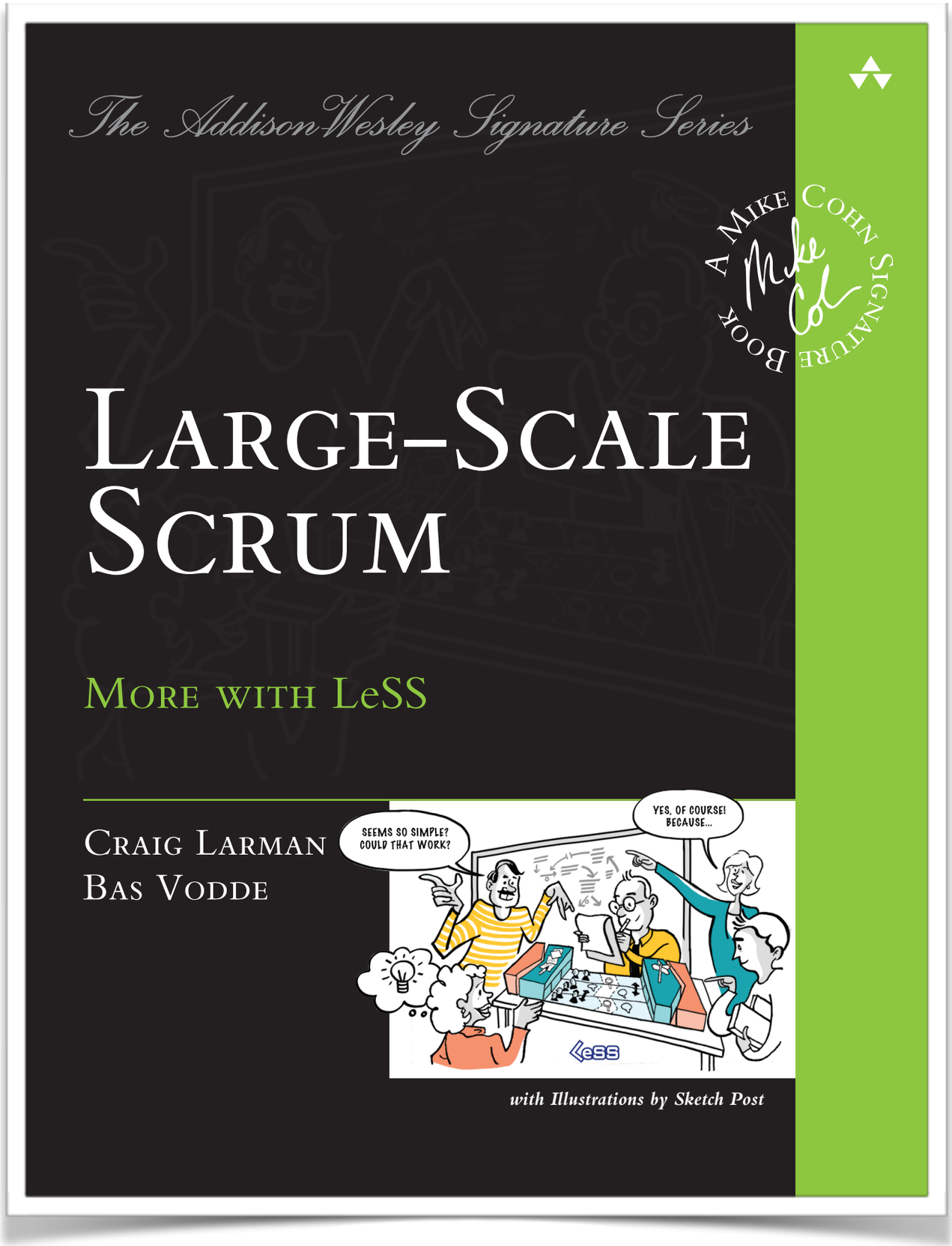
Master = Create new rules and become a sensei (teacher) for new apprentices

Agile Gurukul all rights reserved.

First Two LeSS books…







1 More with LeSS 1
2 LeSS 5

LeSS Structure

3 Adoption 41
4 Organize by Customer Value 67
5 Management 111
6 ScrumMasters 133

LeSS Product

7 Product 155
8 Product Owner 171
9 Product Backlog 197
10 Definition of Done 231

LeSS Sprint

11 Product Backlog Refinement 249
12 Sprint Planning 275
13 Coordination and Integration 285
14 Review & Retrospective 313

Agile Gurukul all rights reserved.

Scaling?

Agile Gurukul all rights reserved.

first, a caution...

Agile Gurukul all rights reserved.

One of the directors of SAGE was discussing why the programming had gotten out of hand. He was then asked, **"If you had it to do all over again, what would you do differently?"** …

Agile Gurukul all rights reserved.

His answer:

"Find the ten best people and write the entire thing themselves."

[Horowitz74]

Agile Gurukul all rights reserved.

after years working in

large

multisite

offshore

development,

our **key advice**? …

Agile Gurukul all rights reserved.

large - don't

multisite - don't

offshore - don't

Agile Gurukul all rights reserved.

but groups still ‘scale’,
for reasons…

compelling
(“create self-driving car”)

questionable
(“low-cost sites & outsourcing”)

Agile Gurukul all rights reserved.

so is LeSS for **scaling**?

Agile Gurukul all rights reserved.

Descaling & Simplifying

Agile Gurukul all rights reserved.

Agile Manifesto

Agile Gurukul all rights reserved.

“How can we apply agile at scale in our big complex organisation?”

Agile Gurukul all rights reserved.

Is this the right question?...

Agile Gurukul all rights reserved.

traditional large groups are complicated — though not because they need to be, but because their organisational designs create an illusion of 'necessary' complexity

Agile Gurukul all rights reserved.

This is an Important Question…

"How can we simplify the unnecessarily big and complex organisational design, and be agile rather than do agile?"

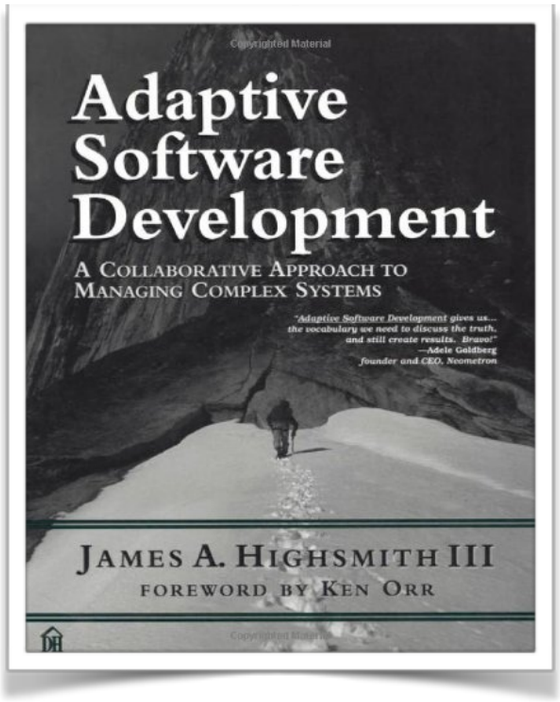
Agile Gurukul all rights reserved.

“agile”?

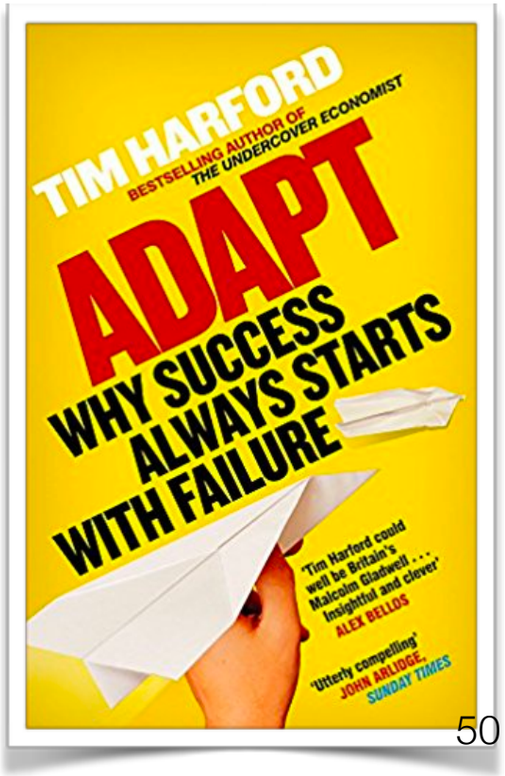
Agile Gurukul all rights reserved.

as "agile"
has become
meaningless jargon for *anything*,
will avoid it, and use…

Agile Gurukul all rights reserved.



adaptive



Agile Gurukul all rights reserved.

LeSS **descales**
organisational complexity,
(1) dissolving unnecessary complex org solutions
(2) solving in simpler ways

Agile Gurukul all rights reserved.

***be* adaptive**

not *do* adaptive

Agile Gurukul all rights reserved.

descaling

simplifying

Agile Gurukul all rights reserved.

not about "agile coaching"

organisational
design
consulting

Agile Gurukul all rights reserved.

not about "method"

organisational design

Agile Gurukul all rights reserved.

own

vs

rent

Agile Gurukul all rights reserved.

want to

vs

have to

Agile Gurukul all rights reserved.

why

Agile Gurukul all rights reserved.

don’t believe
anything I say

fads & gurus -> **insight**

rent -> **own**

Agile Gurukul all rights reserved.

Agile Gurukul all rights reserved.

"It is difficult to get a man to understand something, when his salary depends upon his **not** understanding it!"

— Upton Sinclair

Agile Gurukul all rights reserved.

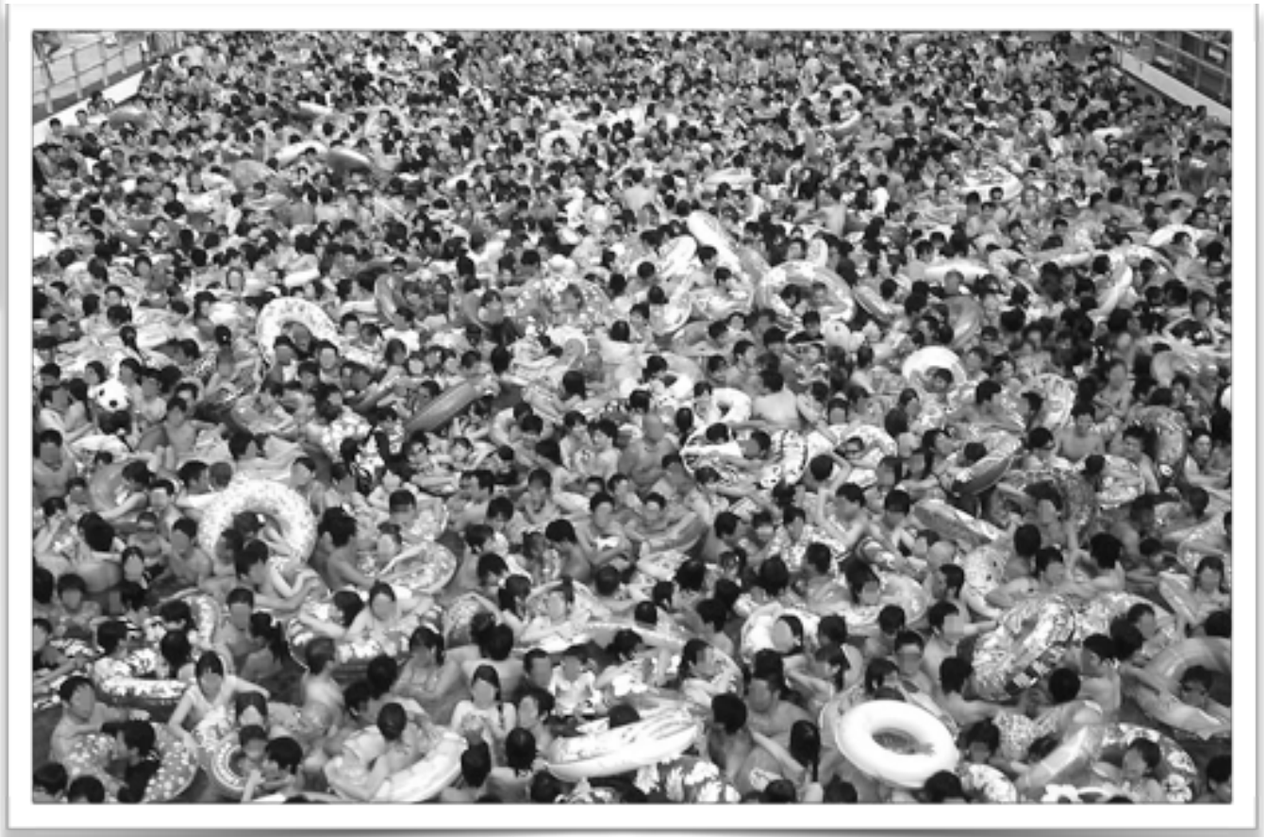
Job Safety, but not Role Safety

Agile Gurukul all rights reserved.

Adoption: **Top-Down** & **Bottom-Up**

Agile Gurukul all rights reserved.

Adoption: **Deep & Narrow** over Broad & Shallow



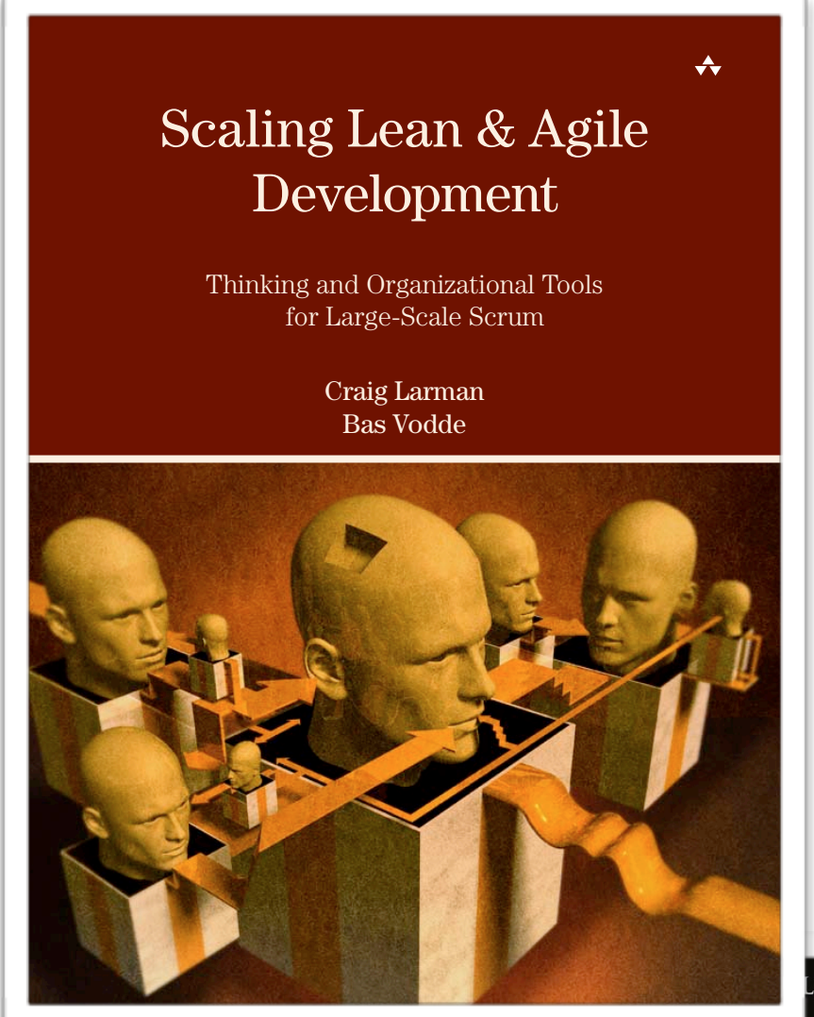
Agile Gurukul all rights reserved.

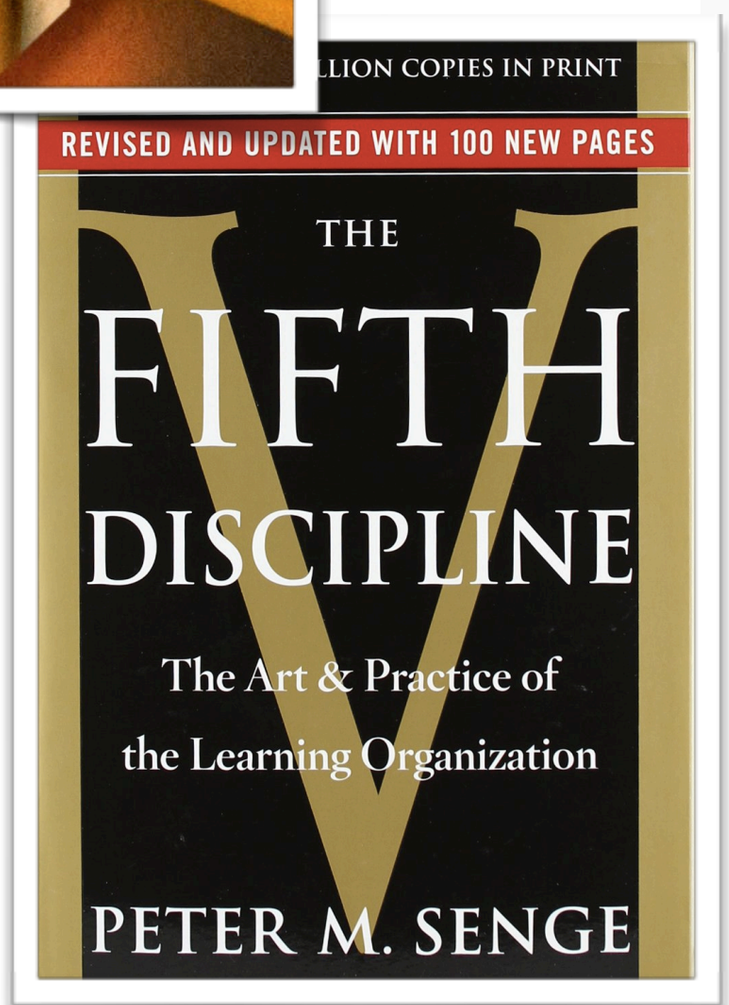
Keeping it Simple

Agile Gurukul all rights reserved.

role of management...

Agile Gurukul all rights reserved.





Thinking Tools

2. Systems Thinking
3. Lean Thinking
4. Queueing Theory
5. False Dichotomies
6. Be Agile

Organizational Tools

7. Feature Teams
8. Teams
9. Requirement Areas
10. Organization
11. Large-Scale Scrum

Agile Gurukul all rights reserved.

BIG Idea

management's remit is to
see the whole system
optimise the whole system
(not locally optimise)

Agile Gurukul all rights reserved.

BIG Idea

so… *failures* in…

"delivering quality",

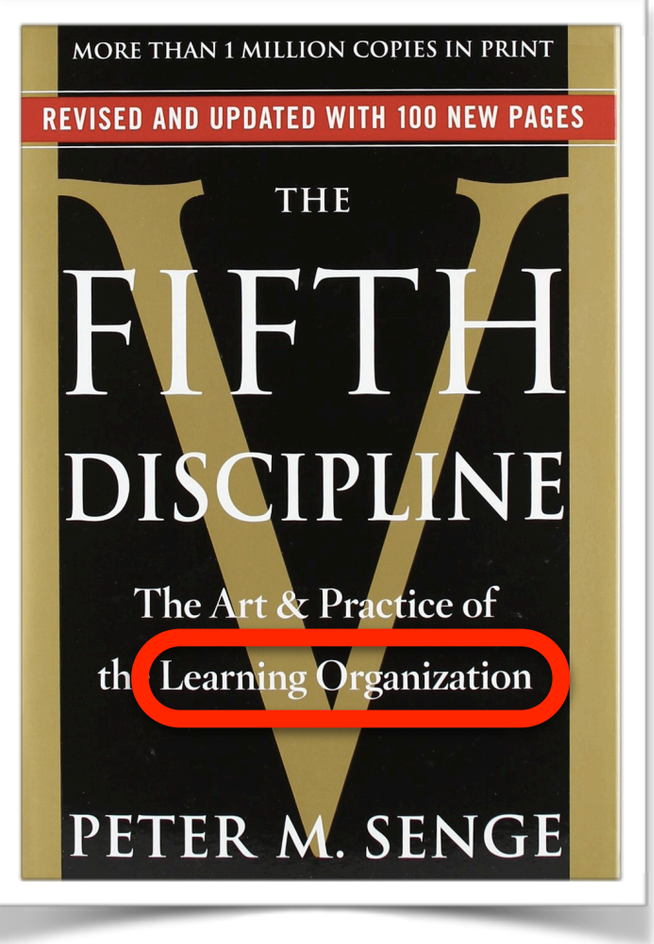
"successful change", etc

are *failures of the system*,

and thus

"failures in management"

Agile Gurukul all rights reserved.



becoming a

learning organisation

not a

copying organisation

Agile Gurukul all rights reserved.

sharing insights?...

Agile Gurukul all rights reserved.

by the end, you will understand the following:

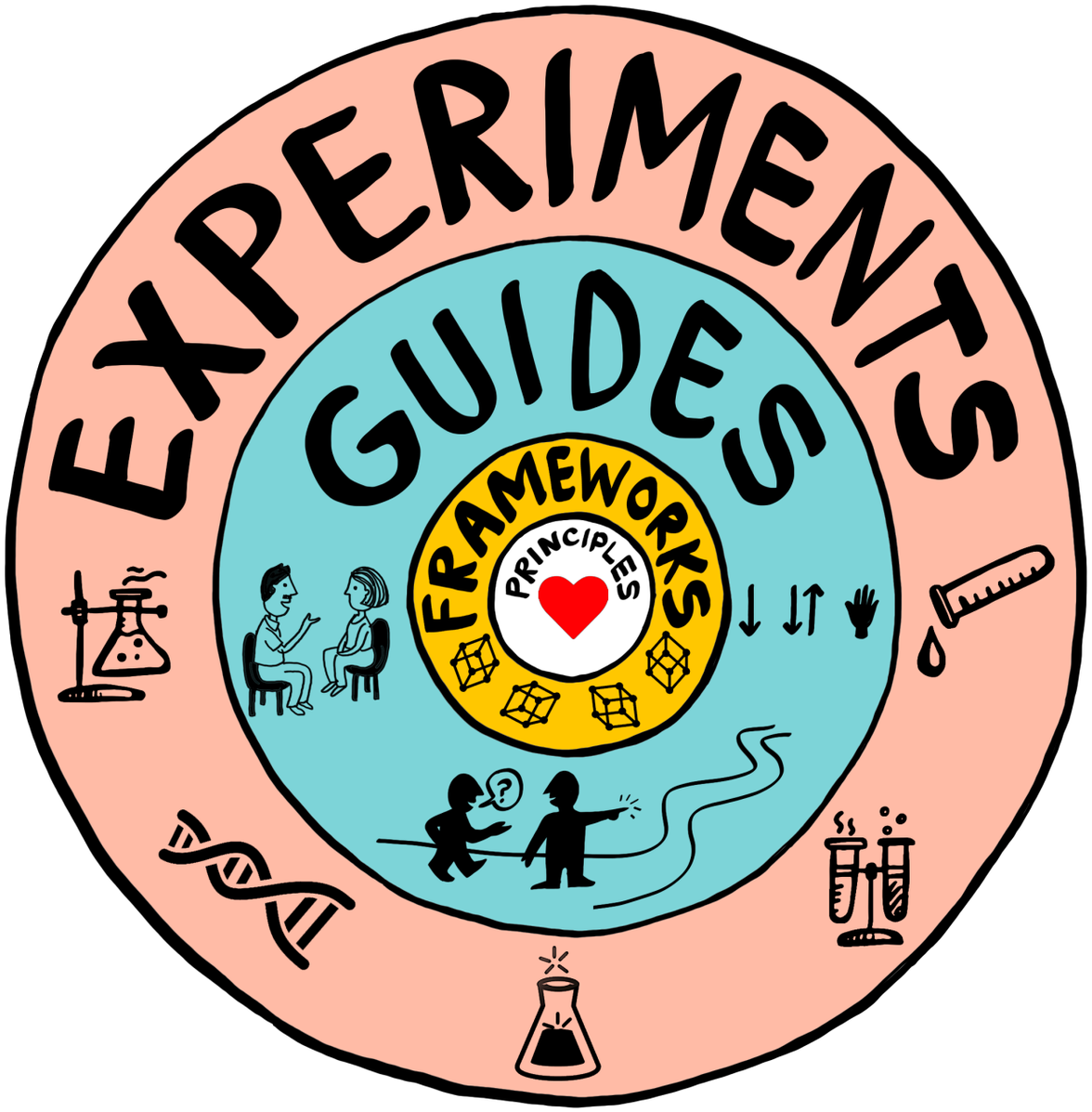
you won't *successfully* be able to return to workplace and "give a summary" of your insights; it is **futile** & **won't be understood**

Agile Gurukul all rights reserved.

LeSS
Overview

Agile Gurukul all rights reserved.

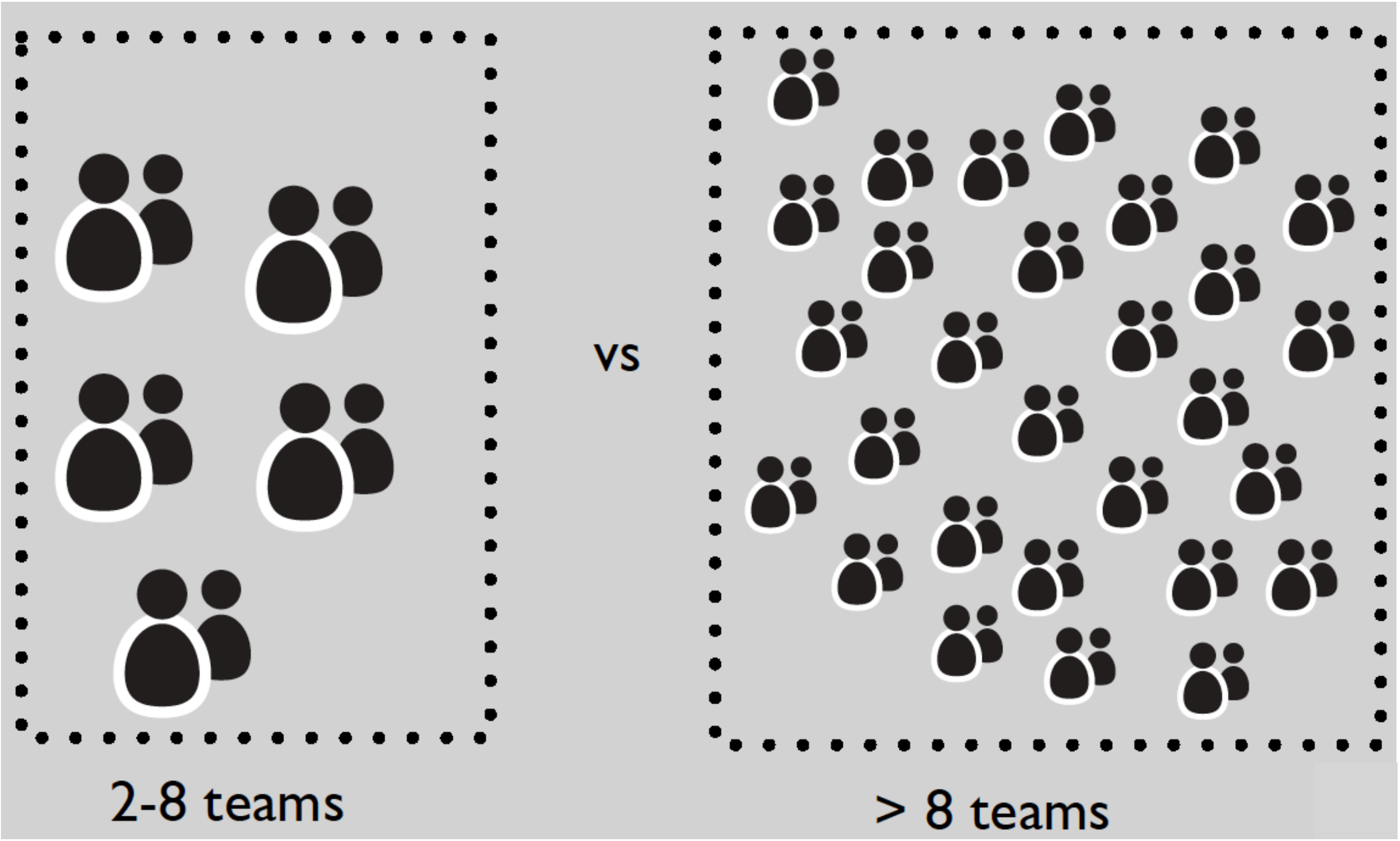
Large Scale Scrum



http://less.works BY-ND

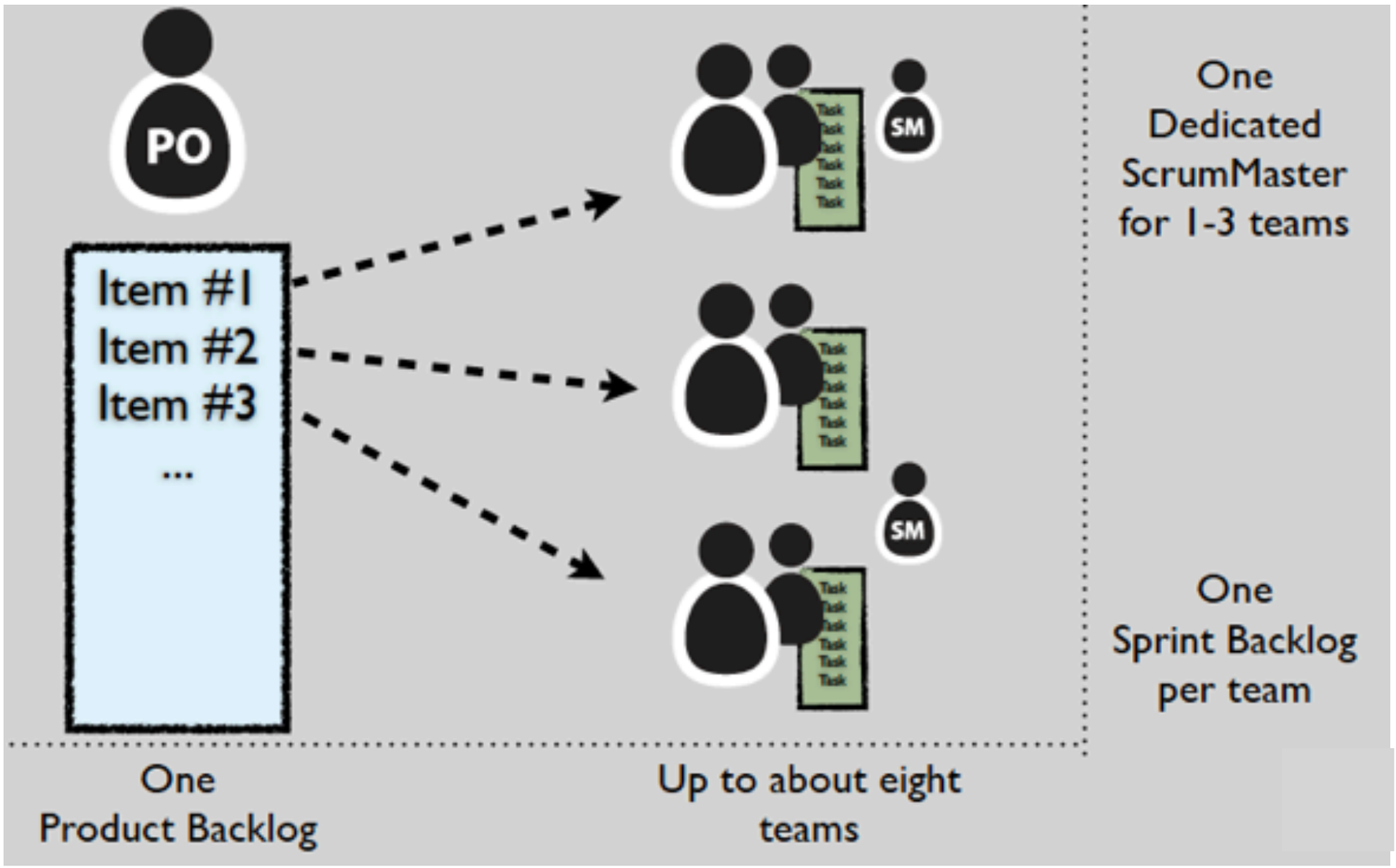
Agile Gurukul all rights reserved.

LeSS/LeSS Huge



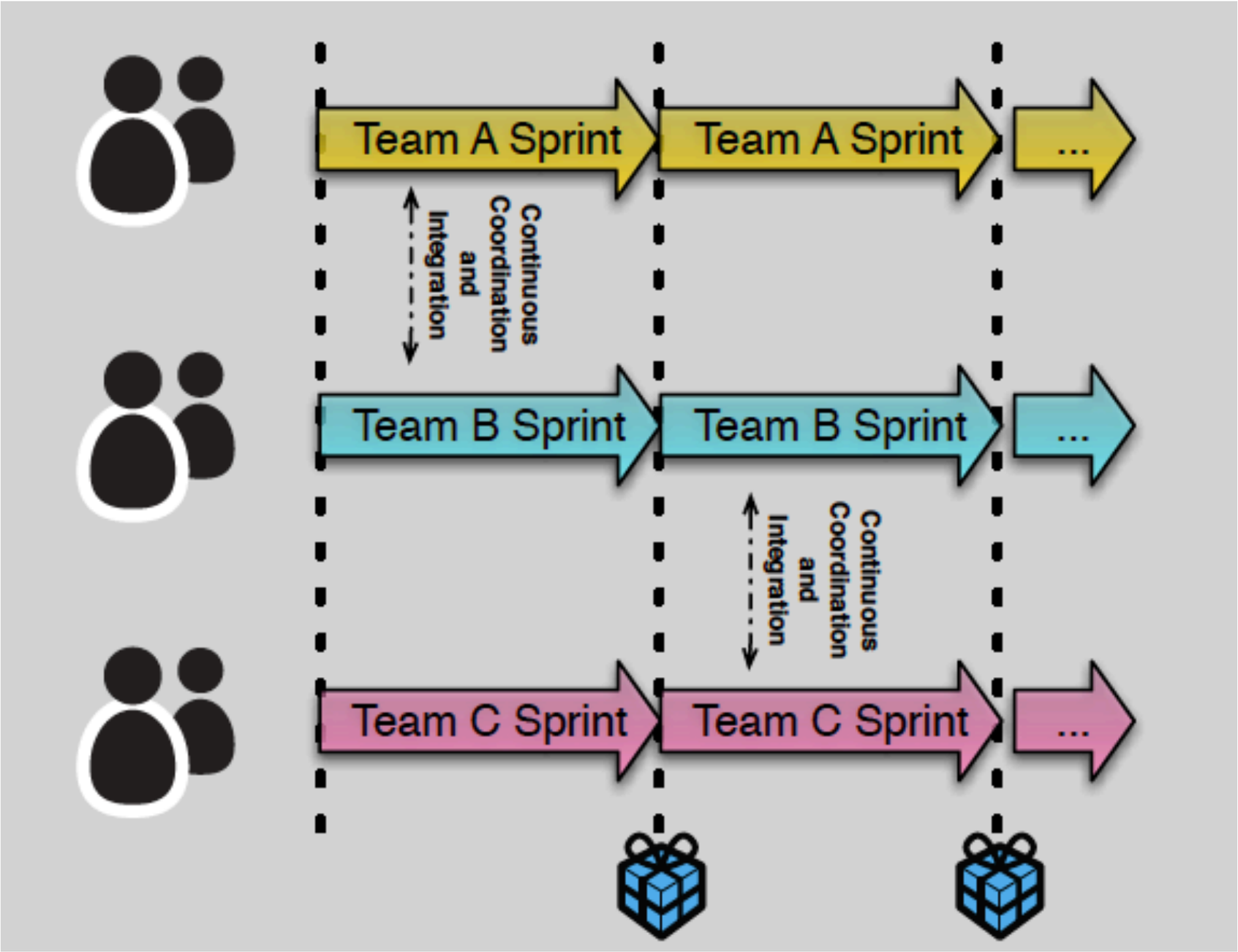
Agile Gurukul all rights reserved.

Large-Scale Scrum (LeSS)



Agile Gurukul all rights reserved.

LeSS One Sprint



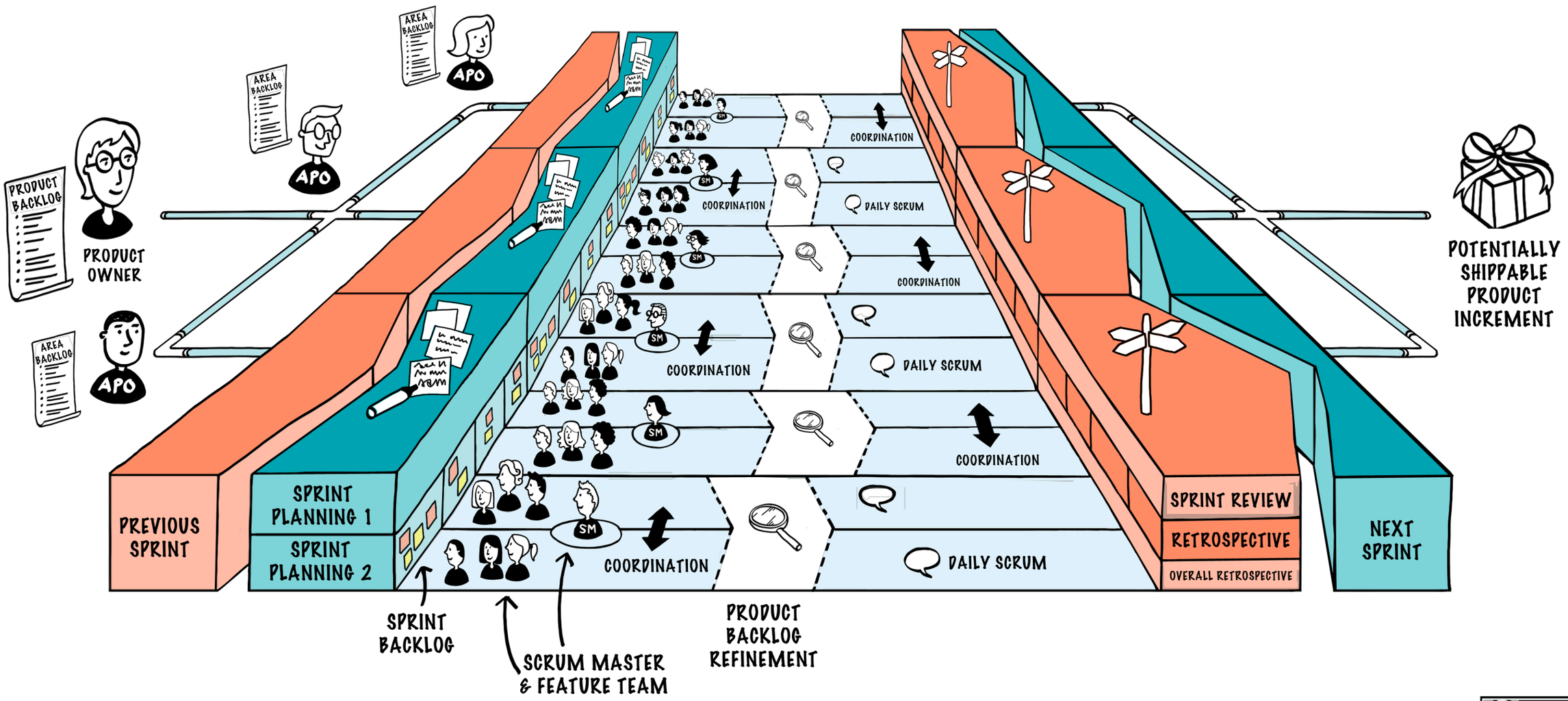
Agile Gurukul all rights reserved.

Terminology

- Overall -> Whole Product Focus
- Multi-team -> Several teams together
- Team Representative -> Someone from the team (not the ScrumMaster)
- Overall Product Owner -> Product Owner (as in not APO)
- Team -> Development Team (In Scrum)
- Scrum Team -> Not used
- Self-managed Team -> Self-organised Team (in Scrum)

Agile Gurukul all rights reserved.

LeSS Huge Framework

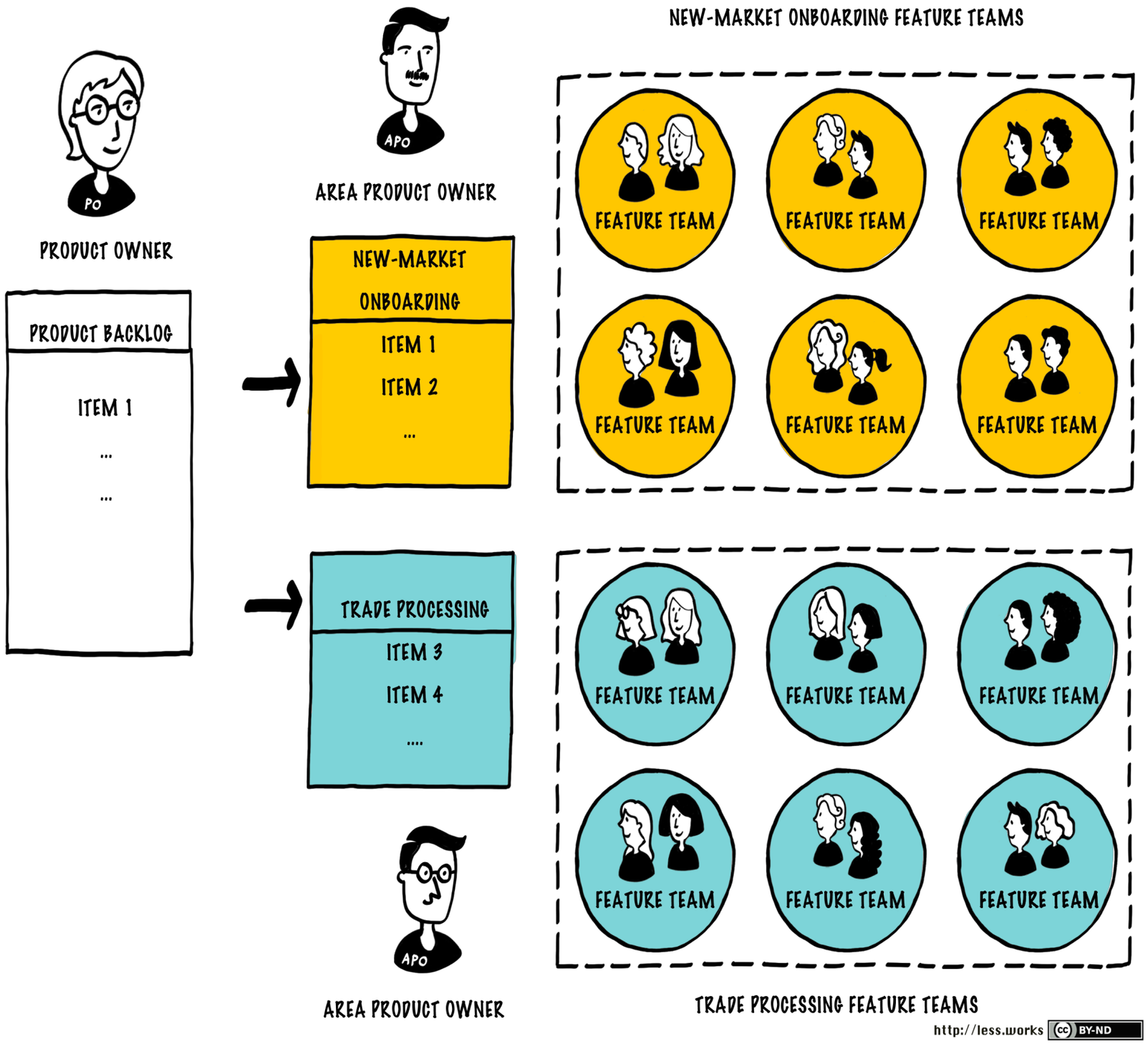


http://less.works BY-ND

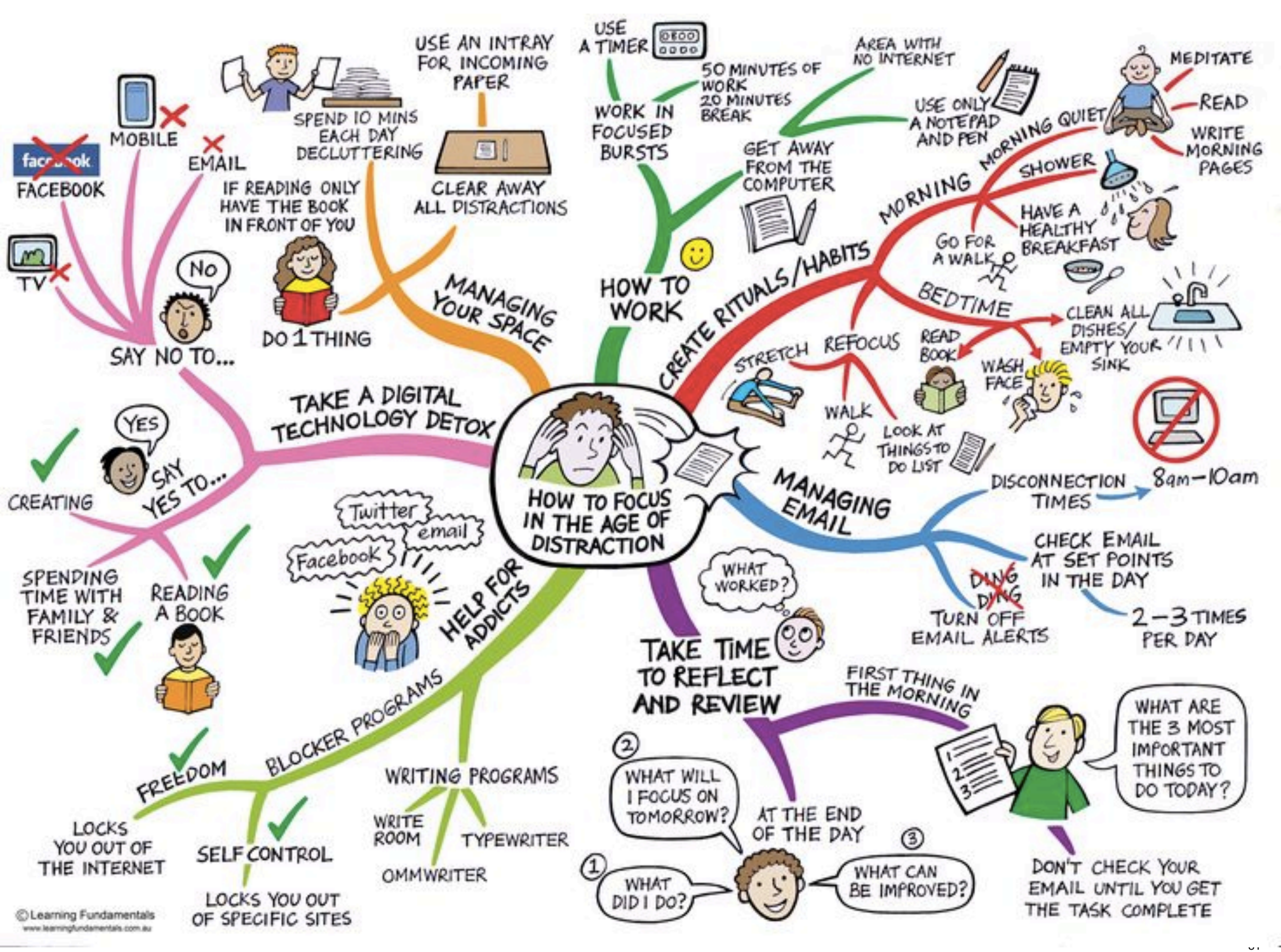
more than 8 teams

Agile Gurukul all rights reserved.

LeSS Huge - Requirement Areas

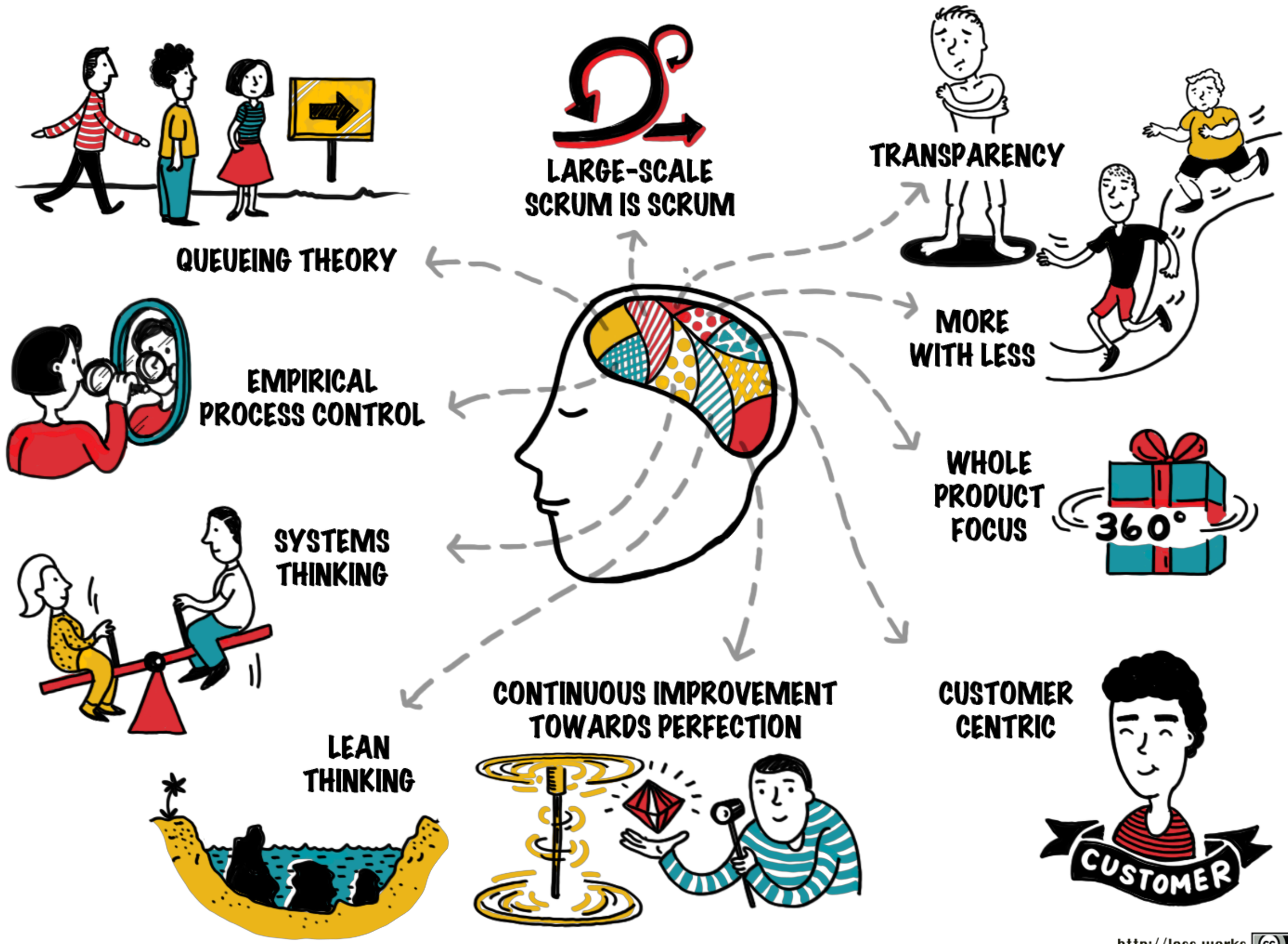


Agile Gurukul all rights reserved.


HOW TO FOCUS IN THE AGE OF DISTRACTION
TAKE A DIGITAL TECHNOLOGY DETOX
SAY NO TO...
No
MOBILE
EMAIL
FACEBOOK
TV
SAY YES TO...
Yes
CREATING
SPENDING TIME WITH FAMILY & FRIENDS
READING A BOOK
MANAGING YOUR SPACE
SPEND 10 MINS EACH DAY DECLUTTERING
USE AN INTRAY FOR INCOMING PAPER
CLEAR AWAY ALL DISTRACTIONS
IF READING ONLY HAVE THE BOOK IN FRONT OF YOU
DO 1 THING
HOW TO WORK
USE A TIMER
WORK IN FOCUSED BURSTS
50 MINUTES OF WORK 20 MINUTES BREAK
GET AWAY FROM THE COMPUTER
AREA WITH NO INTERNET
USE ONLY A NOTEPAD AND PEN
CREATE RITUALS/HABITS
MORNING
MORNING QUIET
MEDITATE
READ
WRITE MORNING PAGES
SHOWER
HAVE A HEALTHY BREAKFAST
GO FOR A WALK
BEDTIME
CLEAN ALL DISHES/ EMPTY YOUR SINK
READ BOOK
WASH FACE
STRETCH
REFOCUS
WALK
LOOK AT THINGS TO DO LIST
MANAGING EMAIL
DISCONNECTION TIMES
8am–10am
CHECK EMAIL AT SET POINTS IN THE DAY
2–3 TIMES PER DAY
TURN OFF EMAIL ALERTS
DING
DING
HELP FOR ADDICTS
Twitter
email
Facebook
BLOCKER PROGRAMS
FREEDOM
LOCKS YOU OUT OF THE INTERNET
SELF CONTROL
LOCKS YOU OUT OF SPECIFIC SITES
WRITING PROGRAMS
WRITE ROOM
TYPEWRITER
OMMWRITER
TAKE TIME TO REFLECT AND REVIEW
WHAT WORKED?
AT THE END OF THE DAY
1
WHAT DID I DO?
2
WHAT WILL I FOCUS ON TOMORROW?
3
WHAT CAN BE IMPROVED?
FIRST THING IN THE MORNING
WHAT ARE THE 3 MOST IMPORTANT THINGS TO DO TODAY?
DON'T CHECK YOUR EMAIL UNTIL YOU GET THE TASK COMPLETE
© Learning Fundamentals
www.learningfundamentals.com.au

LeSS Principle

Agile Gurukul all rights reserved.



Agile Gurukul all rights reserved.

Large-Scale Scrum is Scrum—It is not "new and improved Scrum." Rather, LeSS is about figuring out how to apply the principles, elements, and purpose of Scrum in a large-scale context.

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

More with less—(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, less roles, artifacts, and special groups.

Whole-product focus—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.

Agile Gurukul all rights reserved.

Customer-centric—Identify value and waste in the eyes of the paying customer. Reduce the cycle time from their perspective. Increase feed- back loops with real customers. Everyone understands how their work today directly relates to and benefits paying customers.

Continuous improvement towards perfection—Create and deliver a product all the time, with no cost and no defects, that utterly delights customers, improves the environment, and makes lives better. Do humble and radical improvement experiments each Sprint towards that.

Systems thinking—See, understand, and optimize the whole system (not parts), and use causal-loop modelling to explore system dynamics. Avoid the local and sub-Optimisations 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.

Lean thinking—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.

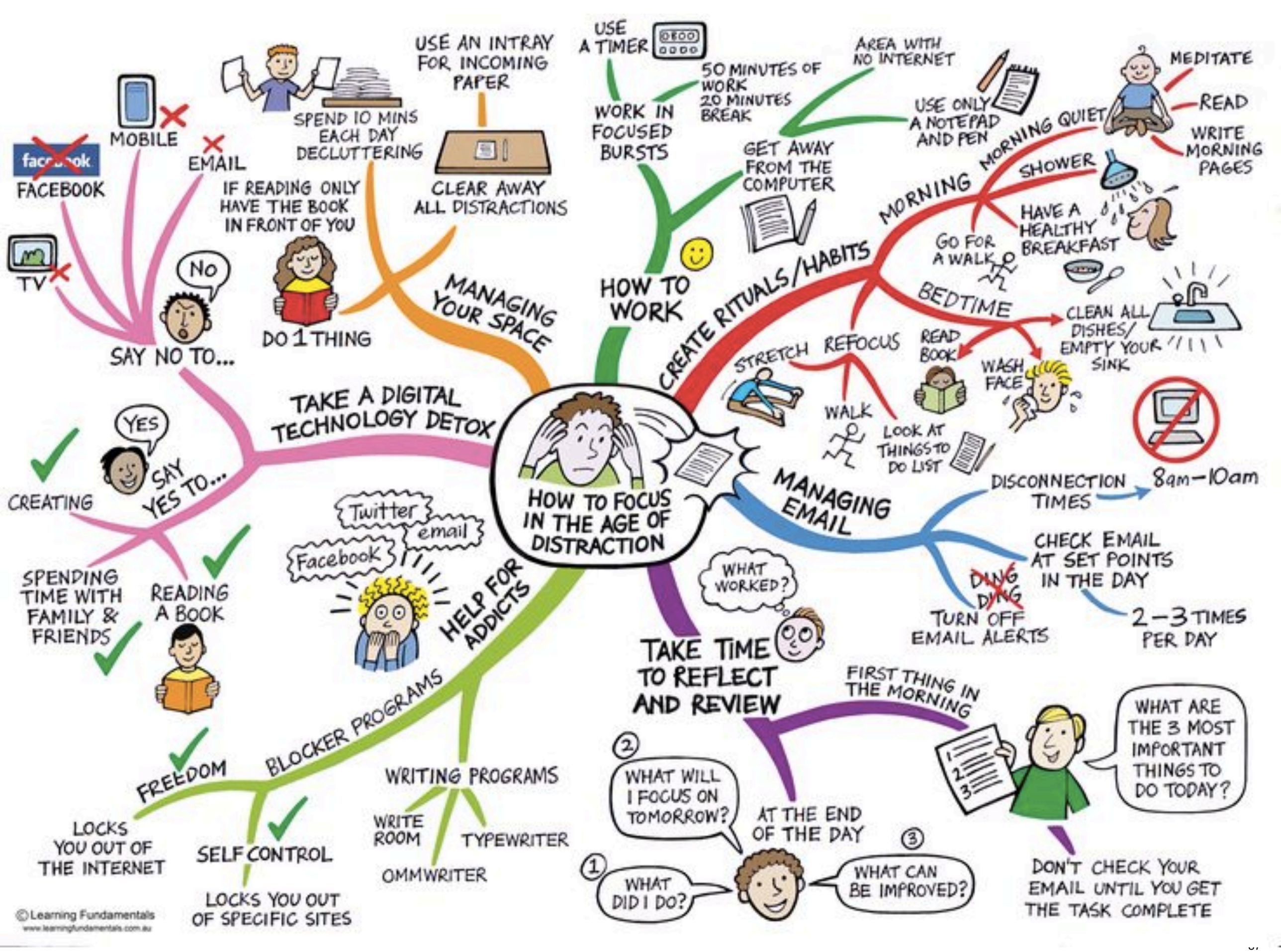
Queuing theory—Understand how systems with queues behave in the R&D domain, and apply those insights to managing queue sizes, work- in-progress limits, multitasking, work packages, and variability.

Agile Gurukul all rights reserved.

Exercise

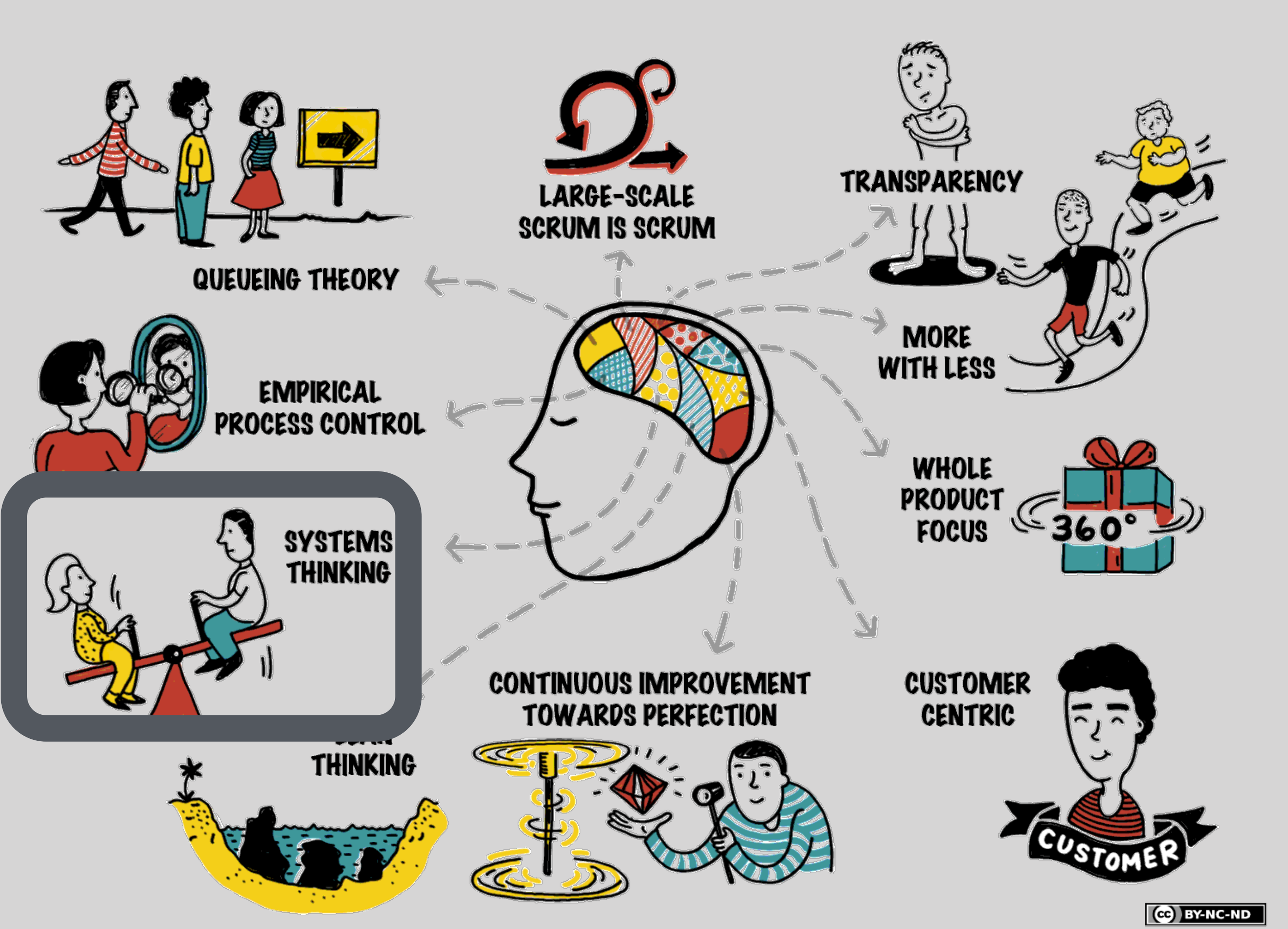
Read LeSS Principles instructed by coach.
When you have read them please stand up.

Agile Gurukul all rights reserved.


HOW TO FOCUS IN THE AGE OF DISTRACTION
TAKE A DIGITAL TECHNOLOGY DETOX
SAY NO TO...
No
MOBILE
EMAIL
FACEBOOK
TV
SAY YES TO...
Yes
CREATING
SPENDING TIME WITH FAMILY & FRIENDS
READING A BOOK
MANAGING YOUR SPACE
SPEND 10 MINS EACH DAY DECLUTTERING
USE AN INTRAY FOR INCOMING PAPER
CLEAR AWAY ALL DISTRACTIONS
IF READING ONLY HAVE THE BOOK IN FRONT OF YOU
DO 1 THING
HOW TO WORK
USE A TIMER
WORK IN FOCUSED BURSTS
50 MINUTES OF WORK 20 MINUTES BREAK
GET AWAY FROM THE COMPUTER
AREA WITH NO INTERNET
USE ONLY A NOTEPAD AND PEN
CREATE RITUALS/HABITS
MORNING
MORNING QUIET
MEDITATE
READ
WRITE MORNING PAGES
SHOWER
HAVE A HEALTHY BREAKFAST
GO FOR A WALK
BEDTIME
CLEAN ALL DISHES/ EMPTY YOUR SINK
READ BOOK
WASH FACE
STRETCH
REFOCUS
WALK
LOOK AT THINGS TO DO LIST
MANAGING EMAIL
DISCONNECTION TIMES
8am–10am
CHECK EMAIL AT SET POINTS IN THE DAY
2–3 TIMES PER DAY
TURN OFF EMAIL ALERTS
HELP FOR ADDICTS
Twitter
email
Facebook
BLOCKER PROGRAMS
FREEDOM
LOCKS YOU OUT OF THE INTERNET
SELF CONTROL
LOCKS YOU OUT OF SPECIFIC SITES
WRITING PROGRAMS
WRITE ROOM
TYPEWRITER
OMMWRITER
TAKE TIME TO REFLECT AND REVIEW
WHAT WORKED?
FIRST THING IN THE MORNING
WHAT ARE THE 3 MOST IMPORTANT THINGS TO DO TODAY?
DON'T CHECK YOUR EMAIL UNTIL YOU GET THE TASK COMPLETE
AT THE END OF THE DAY
1 WHAT DID I DO?
2 WHAT WILL I FOCUS ON TOMORROW?
3 WHAT CAN BE IMPROVED?
© Learning Fundamentals
www.learningfundamentals.com.au

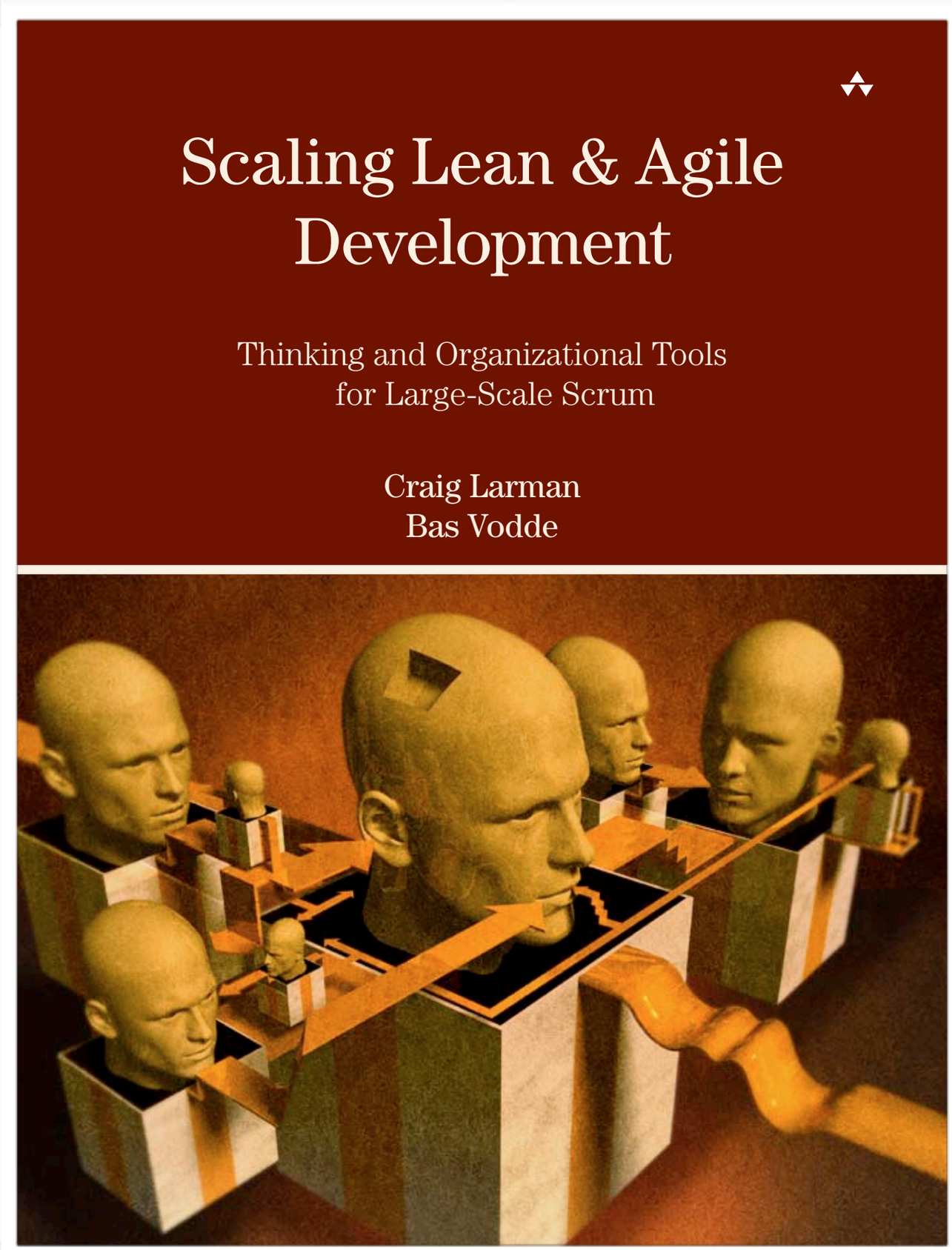
Systems Thinking

Agile Gurukul all rights reserved.



BY-NC-ND

Agile Gurukul all rights reserved.



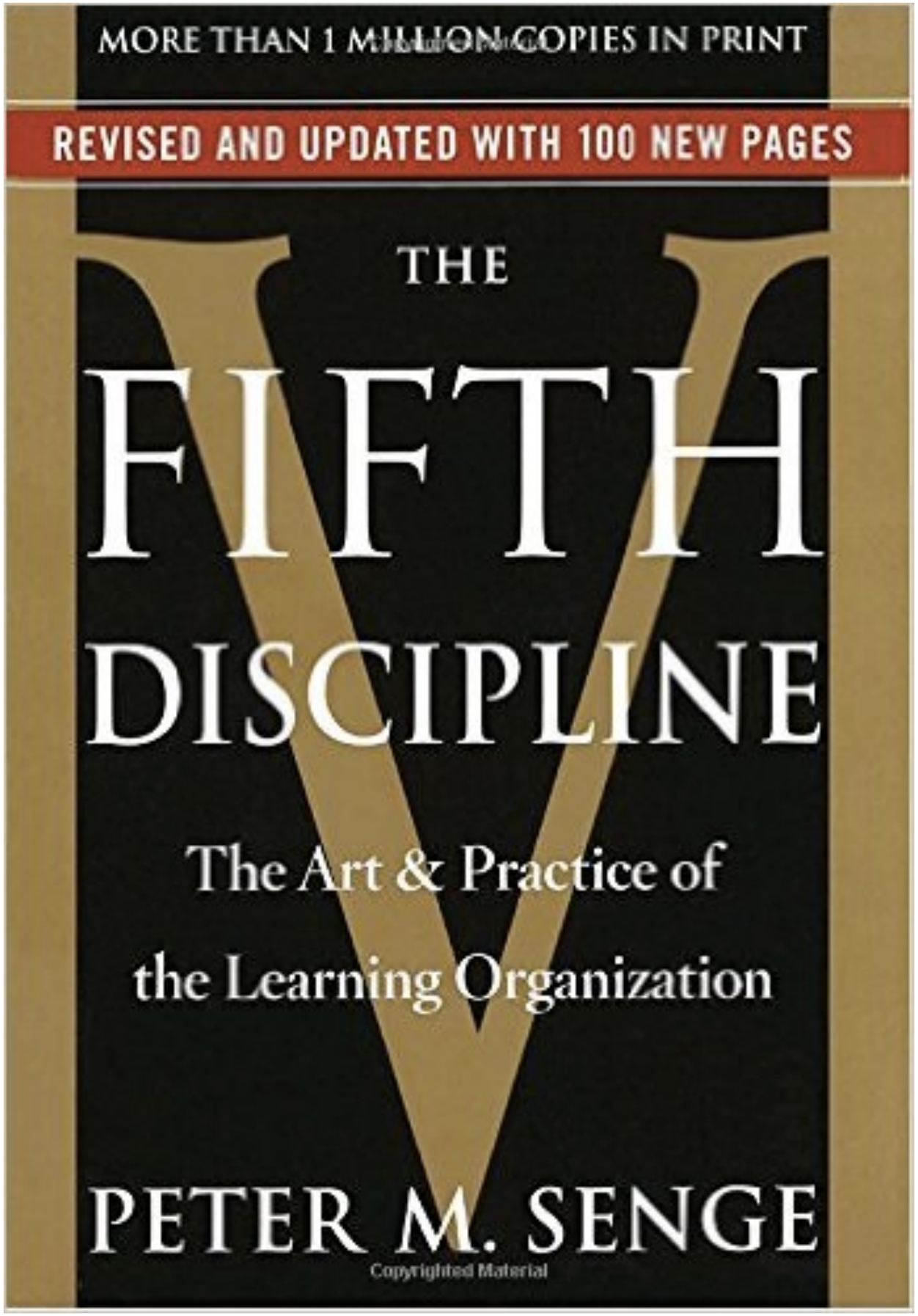
Thinking Tools

2. Systems Thinking
3. Lean Thinking
4. Queueing Theory
5. False Dichotomies
6. Be Agile

Organizational Tools

7. Feature Teams
8. Teams
9. Requirement Areas
10. Organization
11. Large-Scale Scrum

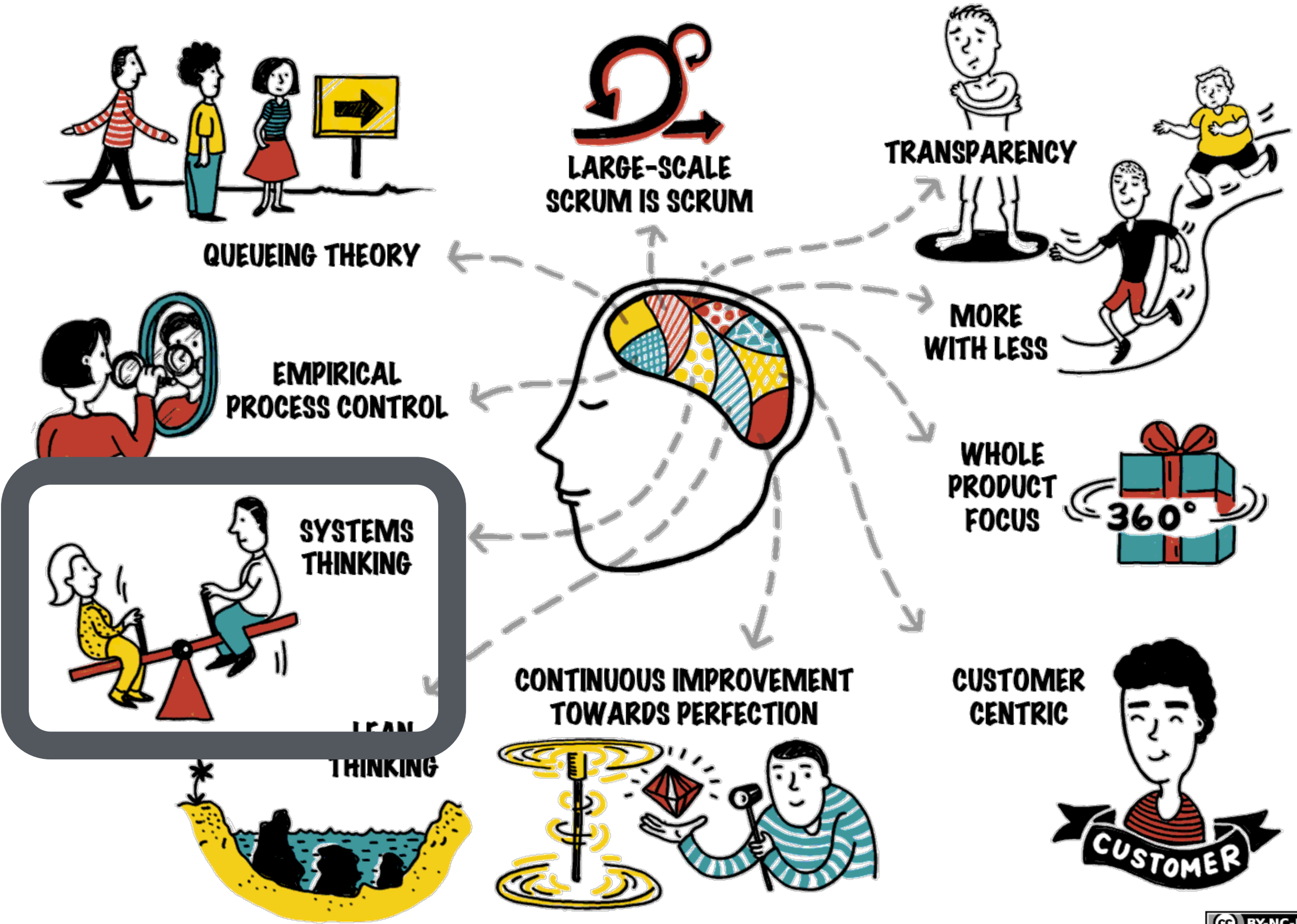
Agile Gurukul all rights reserved.



Agile Gurukul all rights reserved.

systems thinking? ...

Agile Gurukul all rights reserved.



BY-NC-ND

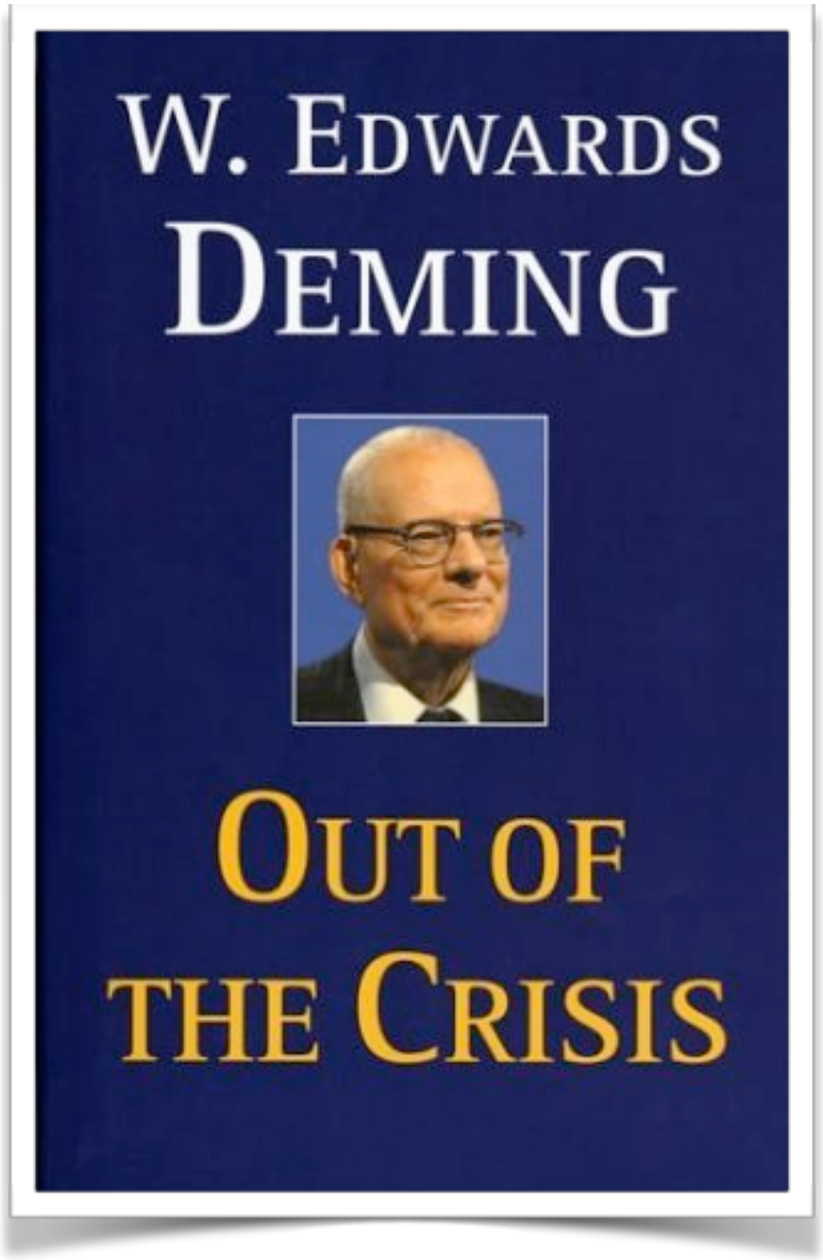
Agile Gurukul all rights reserved.

What is the **SYSTEM?**

- probably…
  - the entire company +
  - customers/markets +
  - supply chain
- *it probably isn't a group within a company*
- exceptions? separate business units

Agile Gurukul all rights reserved.

Total Quality Management



Management's job is to *optimise the system.*

… by paying attention to the *entire system.*

When we *optimise sub-components of the system we don't optimise the overall system.*

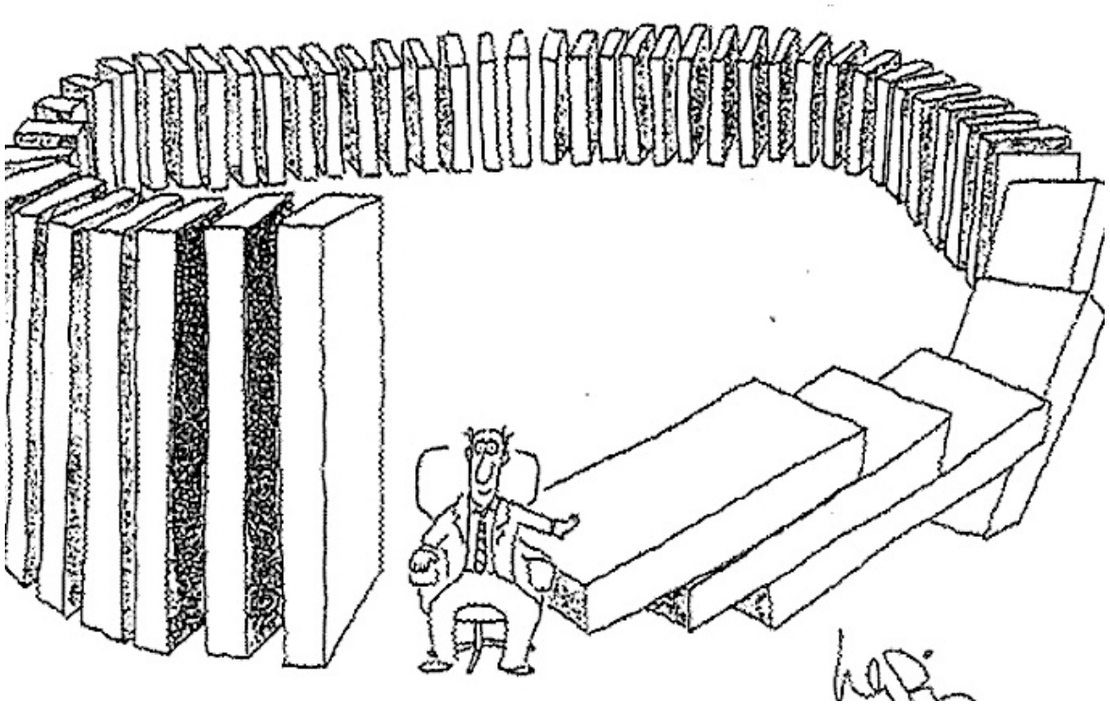
Agile Gurukul all rights reserved.

Systems Thinking

> see the whole

> optimise the whole

> focus on **interaction effects**, not on separate parts



Agile Gurukul all rights reserved.

or more detailed…

Agile Gurukul all rights reserved.

Systems Thinking

- understand there is a SYSTEM
- learn to reason about 'any' system, not 1 system
- see the whole, over space and time
- see how things influence one another and the interaction effects
- optimise the whole
- optimising the parts does not optimise the whole
- beware local-optimisation cognitive bias
- think & talk about system dynamics by drawing systems model diagrams in diverse groups

Agile Gurukul all rights reserved.

(optional) individual:

> draw a **graphic** for each systems thinking idea

> **when done, please stand**

Agile Gurukul all rights reserved.

(optional) pairs: standing

> teach each other. variant techniques the coach may assign:

> guess the icon

> explain each, …

> **please sit when done**

Agile Gurukul all rights reserved.

(optional)

local **improvements** & local optimisation ...

Agile Gurukul all rights reserved.

Local Optimisation & Local Improvement

- a "local optimisation" *tends* to be
  - (1) a *long-term & dominant objective* of the "part" (person, team, etc), and ...
  - (2) that does *not* consider its impact on the system goal (and often degrades it)
- e.g. *our team's measured objective is to increase velocity*

Agile Gurukul all rights reserved.

Local Optimisation & Local Improvement

- a "local improvement" *tends* to be
  - a *short-term & secondary objective* of the "part"
- e.g. *improve my text-editing speed*

Agile Gurukul all rights reserved.

Local Optimisation & Local Improvement

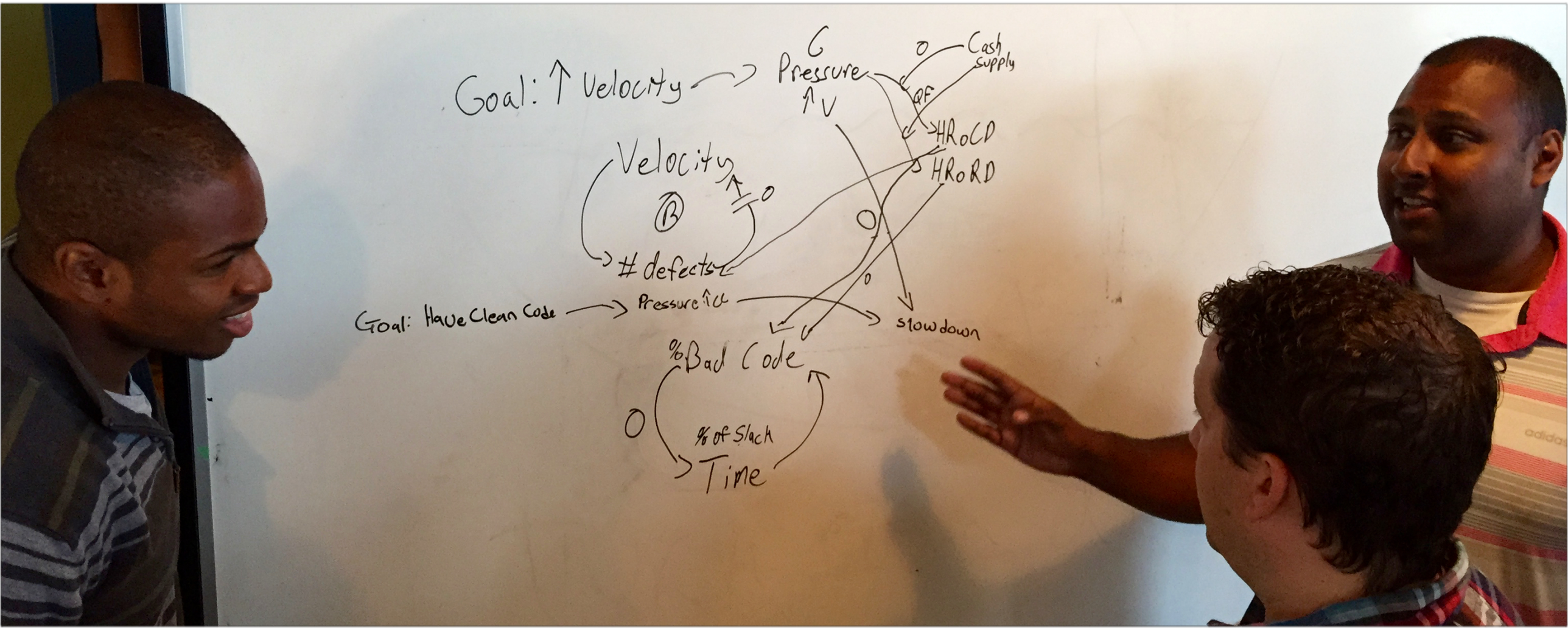
- a systems thinker would probably say:
  - *a "local improvement" is a "local optimisation" if it locally improves something but degrades the system goal*

Agile Gurukul all rights reserved.

Systems Modelling

Agile Gurukul all rights reserved.

Sketch a **System Model**



AKA **causal loop diagram**

Agile Gurukul all rights reserved.

"we model to have a conversation"

the output is shared understanding, not a model

Agile Gurukul all rights reserved.

why not just have a conversation?

Agile Gurukul all rights reserved.

focus on **why**

own vs rent

Agile Gurukul all rights reserved.

you can't

"present a model"

Agile Gurukul all rights reserved.

"all models are wrong,
but some are useful"

— George Box

Agile Gurukul all rights reserved.

"mental models"

cognitive bias

false beliefs

self-awareness
to self-doubt

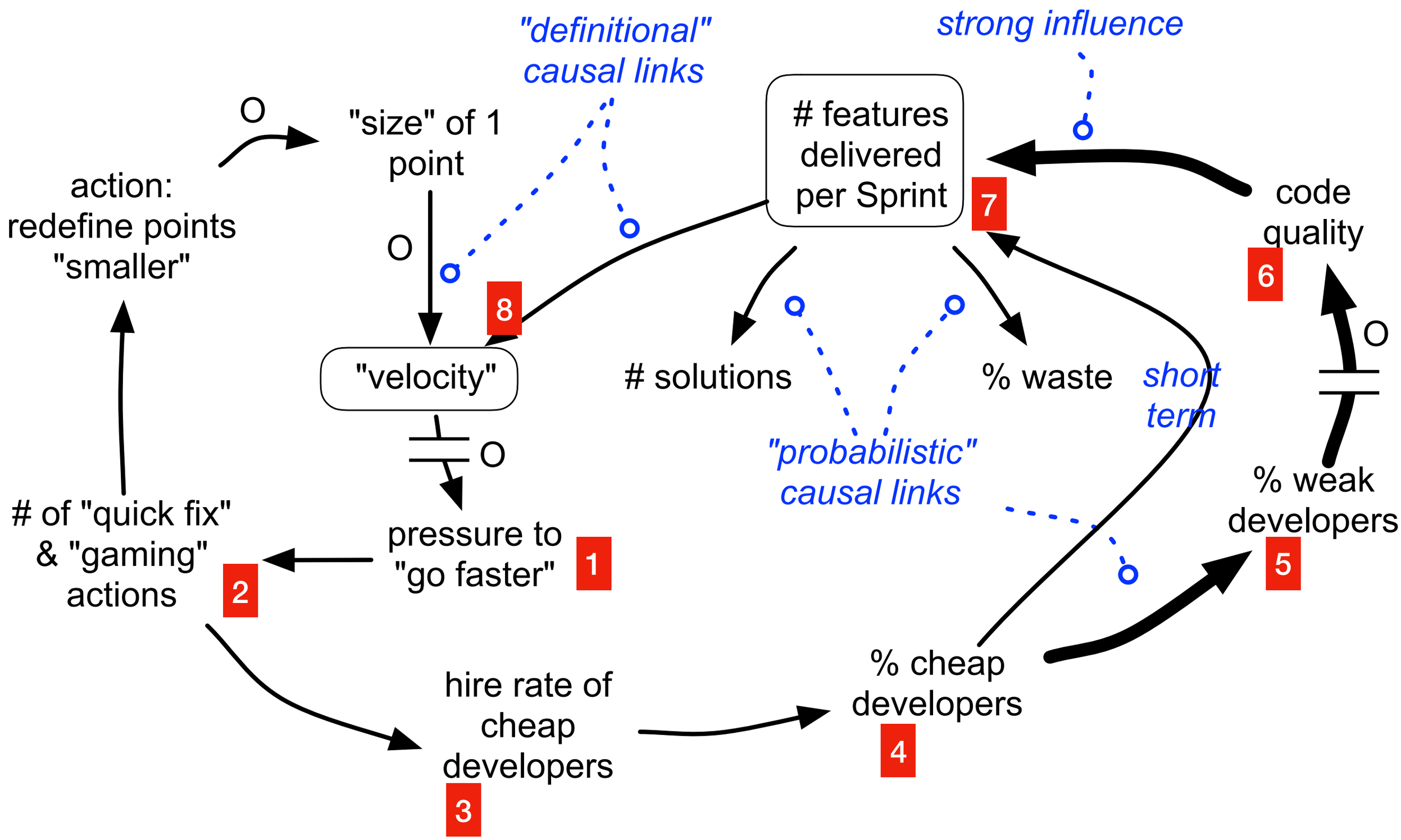
Agile Gurukul all rights reserved.

coach:

> sketch a system model

Agile Gurukul all rights reserved.

Systems Thinking with Causal Loop Diagrams



Agile Gurukul all rights reserved.

we'll start with a simple & familiar situation

to focus **first** on

notation & technique

rather than "content"...

Agile Gurukul all rights reserved.

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**
8. **effort applied to handle defects**
9. **pressure to deliver and "go faster"**
10. **dev fear**

Agile Gurukul all rights reserved.

coach: debrief

> from weaker **correlations** towards stronger **causal** influence links?

> "definitional" (often *formulaic*) links?

Agile Gurukul all rights reserved.

“we model to have a conversation”

but… our short term memories are limited

Agile Gurukul all rights reserved.

focus on **why**

own vs rent

Agile Gurukul all rights reserved.

group

> in **LeSS**, **when** do systems modelling?

> in what contexts or meetings can it help?

Agile Gurukul all rights reserved.

Local Optimisation

Agile Gurukul all rights reserved.

in traditional large-scale organisational design, the **overarching** & **repeating** theme?

local optimisation

Agile Gurukul all rights reserved.

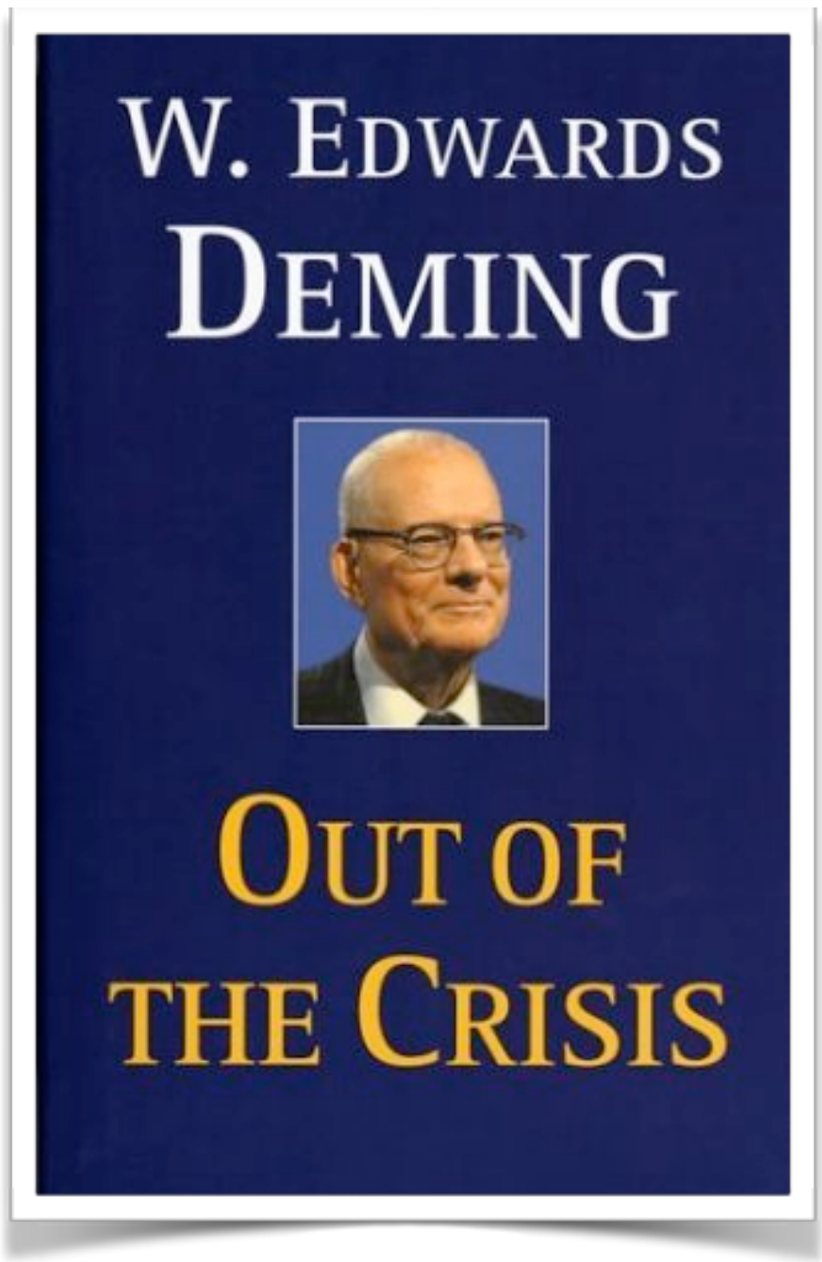
individual:

> write a **definition** of what you think is **local optimisation**

coach: review

Agile Gurukul all rights reserved.

Total Quality Management



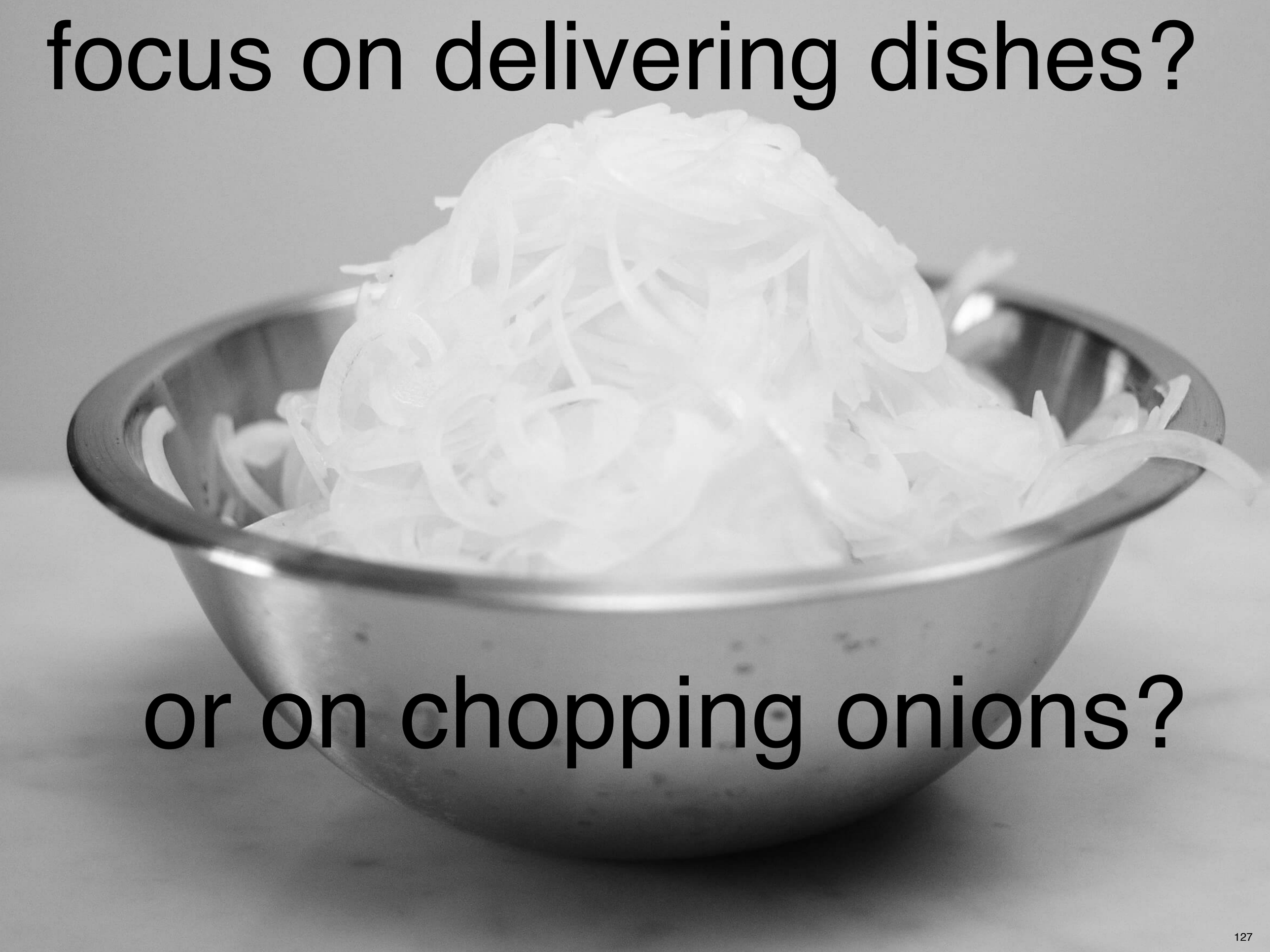
When we *optimize sub-components of the system we don't optimize the overall system.*

Agile Gurukul all rights reserved.

examples of local optimisation...

Agile Gurukul all rights reserved.

focus on moving the ball?
Fly Emirates
UniCredit
or on player job title?


focus on delivering dishes?
or on chopping onions?

individual

- identify 1 *specific* example you've seen of **"working to job title" rather than "moving the ball"**
  - with the consequence of the lean waste of increased *global cycle time* (the lean waste of ***delay***)
- identify 1 *specific* example you've seen of **"chopping onions" rather than "delivering dishes"**
  - with the consequences of one or more of the other lean wastes (e.g.): ***over-processing*** *and all its children:* ***inventory/WIP, hidden defects, handoff, information scatter***

coach: review

Agile Gurukul all rights reserved.

BIG Idea

local optimisation

->

the lean wastes

Agile Gurukul all rights reserved.

Local Optimisation *Cognitive Bias*

"Everyone is **busy** and doing **their best, working efficiently** on **their task**, yet the **system** is **delivering slow** and **not delighting** the user."

Agile Gurukul all rights reserved.

Local Optimisation *Cognitive Bias*

"It's more **efficient** or **productive** when a person/ group does **one specialization**."

Agile Gurukul all rights reserved.

Local Optimisation **Consequence**

justified as:

efficient

productive

best

good

… but consequence is:

system sub-optimisation of (e.g.)

customer value

customer cycle time

customer delight

company robustness

company adaptiveness

Agile Gurukul all rights reserved.

Why Local Optimisation?

Agile Gurukul all rights reserved.

local optimisation
is a
cognitive bias

list of cognitive biases?

Agile Gurukul all rights reserved.

do they only apply to marshmallows? ...

Agile Gurukul all rights reserved.

THE *NEW YORK TIMES* BESTSELLER

THINKING, FAST AND SLOW

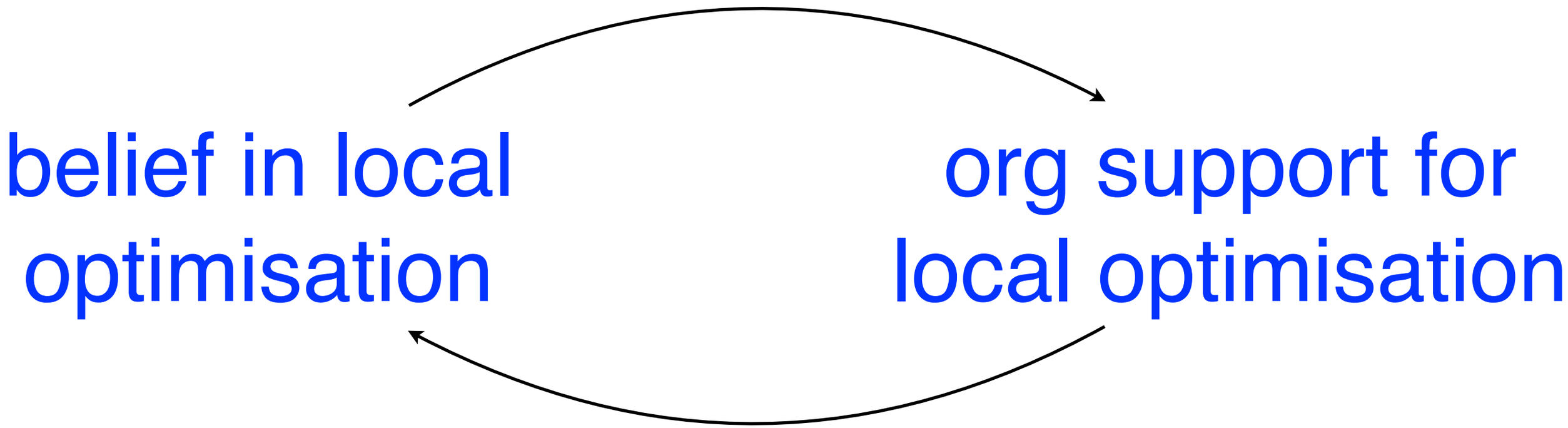
DANIEL KAHNEMAN

WINNER OF THE NOBEL PRIZE IN ECONOMICS

"[A] masterpiece . . . This is one of the greatest and most engaging collections of insights into the human mind I have read." —WILLIAM EASTERLY, *Financial Times*

Agile Gurukul all rights reserved.

Reinforcing Loop



Agile Gurukul all rights reserved.

Local Optimisation is related to…

- management by objectives
- measurement & metrics
- resource management
- performance management
- appraisals

Agile Gurukul all rights reserved.

individual & then group

> examples that led to local optimisation (& its consequences) due to:

> **metrics** or

> **objectives** & **appraisals** or

> **resource management**

Agile Gurukul all rights reserved.

System Optimisation

Agile Gurukul all rights reserved.

Systems Optimisation

"watch the **ball**,
not the players"

"deliver the **dish**,
not the onions"

Agile Gurukul all rights reserved.

what's the goal? ...

Agile Gurukul all rights reserved.

Systems Optimisation

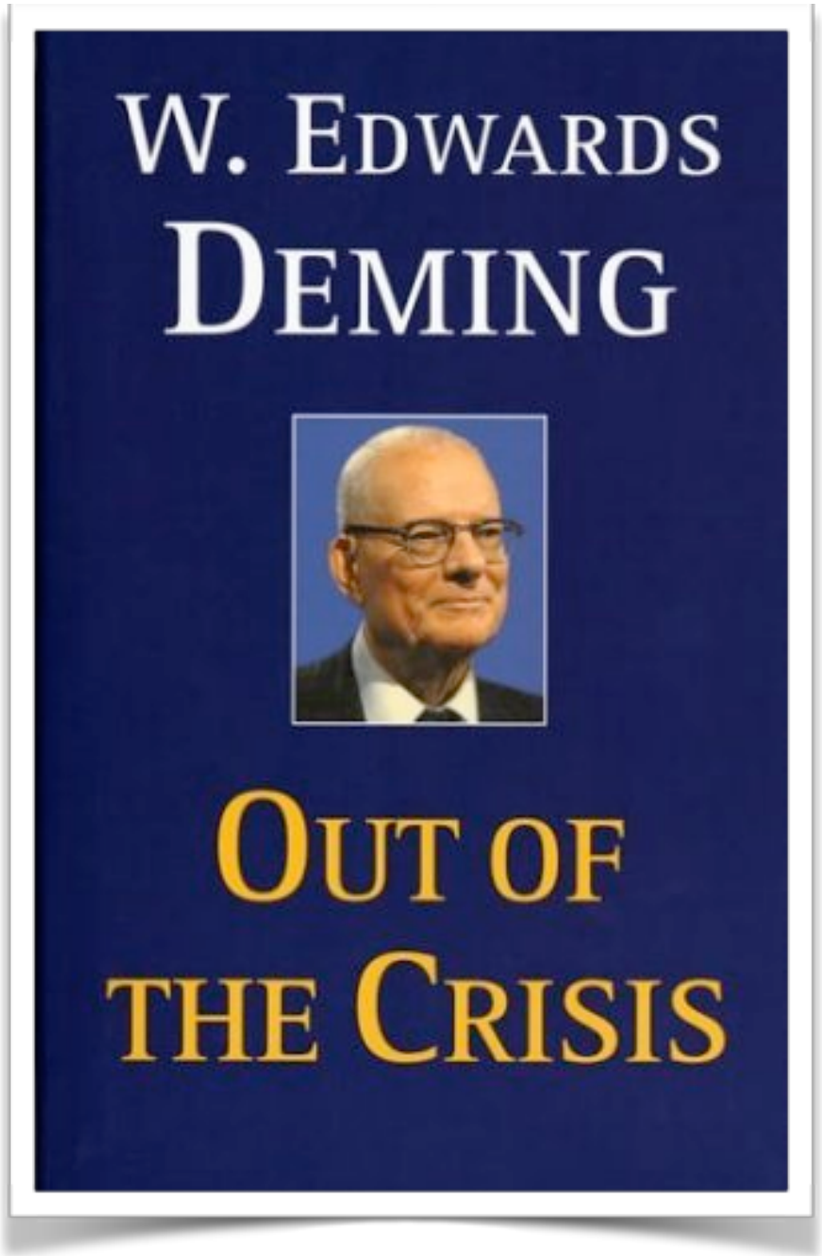
goal:

optimise

system

Agile Gurukul all rights reserved.

Total Quality Management



Management's job is to *optimise the system.*

"Eliminate management by objective.

Eliminate management by numbers, numerical goals. Substitute leadership."

Agile Gurukul all rights reserved.

the **One True** system optimising goal?

Agile Gurukul all rights reserved.

what if the CEO ...?

Agile Gurukul all rights reserved.

BIG Idea

leadership needs to agree on the system optimising goal

Agile Gurukul all rights reserved.

BIG Idea

organisational design elements should be consistent with the system optimising goal (i.e. pass the "consistency test")

Agile Gurukul all rights reserved.

System Goals vs **Indirect Wishes** & **Constraints**

- **goals** that the system can be designed to **definitely & directly influence**
  - highest value, high adaptiveness (agility), low cycle time
- **indirect wishes** (not "goals" in this usage)
  - increased market share
- **constraints** (seldom would the CEO describe these as the goal of the system or company)
  - reduced cost
  - reduced risk
- TIP! beware confusing these

Agile Gurukul all rights reserved.

Systems Optimisation Goal

An organisational system can be optimised for anything E.g.

- **Secrecy**
- **Deliver highest value**
- **E2E Cycle time**
- **Cost**
- **Manager prestige**
- **Adaptability / Agility**
- **Shareholder value**
- **etc..**

Agile Gurukul all rights reserved.

BIG Idea

LeSS is for

highest-level

(1) *adaptiveness*

in the service of

(2) *highest-level "value"*

Agile Gurukul all rights reserved.

Local Optimisation in Lists (e.g. Backlogs)

Agile Gurukul all rights reserved.

let's start to apply system modelling to "scaling agile" organisational design choices…

Agile Gurukul all rights reserved.

team:

> in reality the prior variables should remain, since it's **all one system**

> but **wall space management**…

> **tear down the old model**

Agile Gurukul all rights reserved.

> place the new model DEAD CENTRE of the free wall space

Agile Gurukul all rights reserved.

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

> **1 product, many teams, each team has a Team "Product Backlog" prioritised 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 <u>dead centre</u> 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 the company level** (i.e. cost of change & txn cost) **… at "3 o'clock"**
4. **% of items worked on each Sprint that are highest value from a company view … at "6 o'clock"**
5. **likelihood that a single team will see they may be working on low-value items, from a company view … at "8 o'clock"**
6. **local team identity … at "10 o'clock"**

Agile Gurukul all rights reserved.

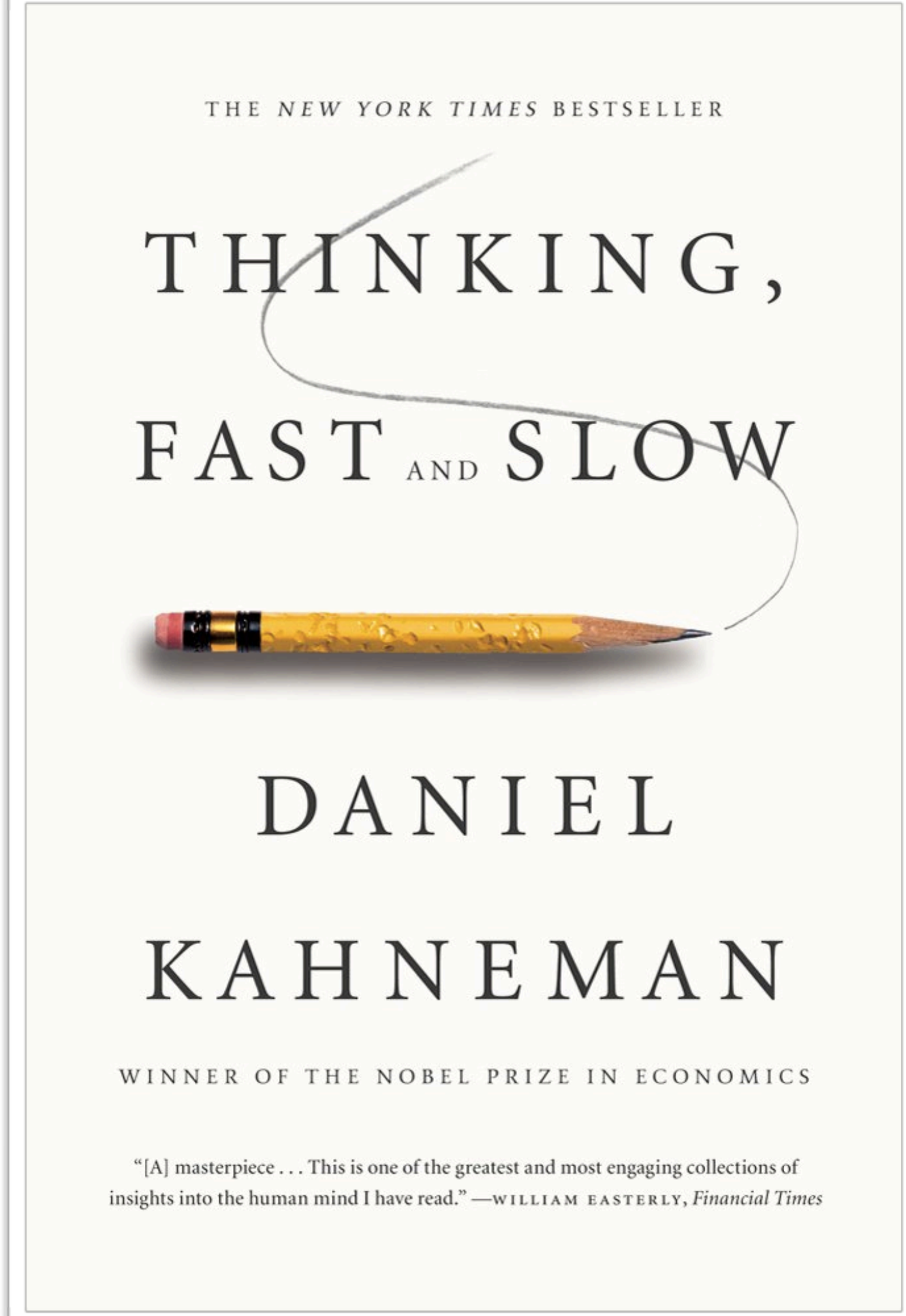
"but that's not efficient"

Agile Gurukul all rights reserved.

Local Optimisation *Cognitive Bias*

what is the misunderstanding when someone says "**efficient**" or "**productive**"?

Agile Gurukul all rights reserved.



Agile Gurukul all rights reserved.

coach: relationship of these variables?

1. **adaptiveness (agility) of teams to change direction at the company level**
2. **% of items worked on each Sprint that are highest value from a company view**

Agile Gurukul all rights reserved.

coach

> if system optimising goals are

  > **adaptiveness** for **highest value**, at company level

> … *how many backlogs?*

> … *is the answer "good" or "bad"?*

> … *did the coach tell you the answer?*

Agile Gurukul all rights reserved.

BIG Idea

the purpose of
adaptiveness/agility?

…to cheaply & easily support changing direction (**re-prioritisation**) to work on **continually newly-discovered** highest value

Agile Gurukul all rights reserved.

support **secondary concerns**
or **constraints …**
(e.g. “local team identity”)

… without sub-optimising
the system goal

Agile Gurukul all rights reserved.

"support
secondary concerns
in
secondary ways"

Agile Gurukul all rights reserved.

“local optimisation in backlogs”

Agile Gurukul all rights reserved.

therefore…

Agile Gurukul all rights reserved.

LeSS Rule(s)

1 (and only 1)
Product Backlog

(and no
Team "Product"
Backlogs)

Agile Gurukul all rights reserved.

1 Product Backlog …

and no FAKE "redefining" by calling a set of team backlogs "*part of 1 Product Backlog*" or "*views* on the 1 backlog"

Agile Gurukul all rights reserved.

Same Rule as in 1-team Scrum…

Scrum Guide:

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

Agile Gurukul all rights reserved.

Reflections

Agile Gurukul all rights reserved.

notice that
the coach has
not "taught" the
elements of LeSS

Agile Gurukul all rights reserved.

own

vs

rent

Agile Gurukul all rights reserved.

focus on **why**

Agile Gurukul all rights reserved.

want to

vs

have to

Agile Gurukul all rights reserved.

How to **Find** Useful Variables?

strongly related to…

optimising goal(s)

secondary concerns

indirect wishes

constraints

Agile Gurukul all rights reserved.

“driving variables”

Agile Gurukul all rights reserved.

highlight variables in the model strongly related to (or actually) the **system optimising goal**

Agile Gurukul all rights reserved.

definitional links

probabilistic links

Agile Gurukul all rights reserved.

preparing for the next exercise…

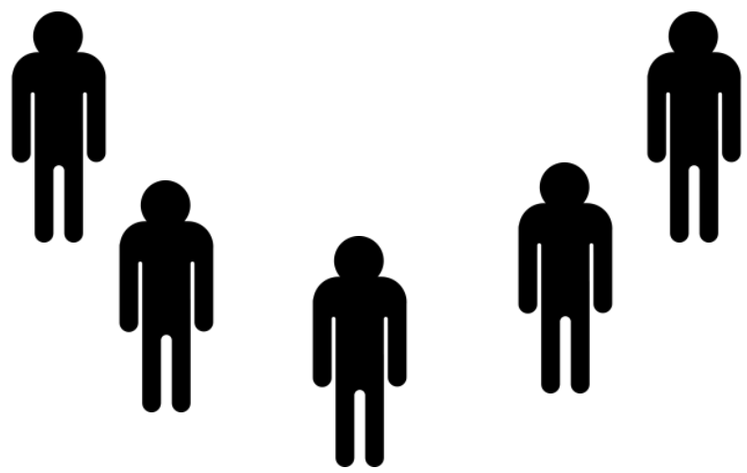
Agile Gurukul all rights reserved.

tip:

share/rotate the **PEN**

look for ways for everyone to be engaged

Agile Gurukul all rights reserved.



Agile Gurukul all rights reserved.

consider being (self) aware of not actively listening enough, or dominating conversation

Agile Gurukul all rights reserved.

"am i exploring if my colleagues understand my point clearly?"

"am i exploring if i understand my colleagues' point?"

"am i *really actively* listening?"

"am i trying to help everyone feel heard and included?"

Agile Gurukul all rights reserved.

team

> now you've worked together for a little time, have a **retrospective**

Agile Gurukul all rights reserved.

Local Optimisation in Lists/Backlogs (again)

Agile Gurukul all rights reserved.

1 Product Backlog but still

constraints due to teams knowing disjoint items…

Agile Gurukul all rights reserved.

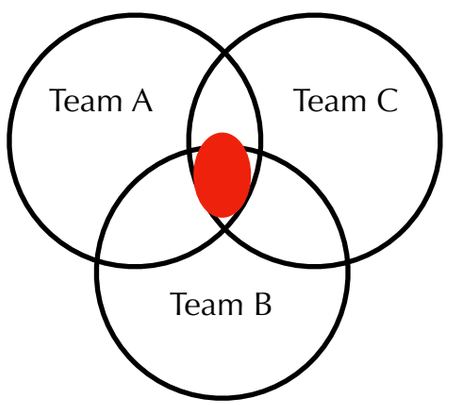
coach: visualise with Venn diagram

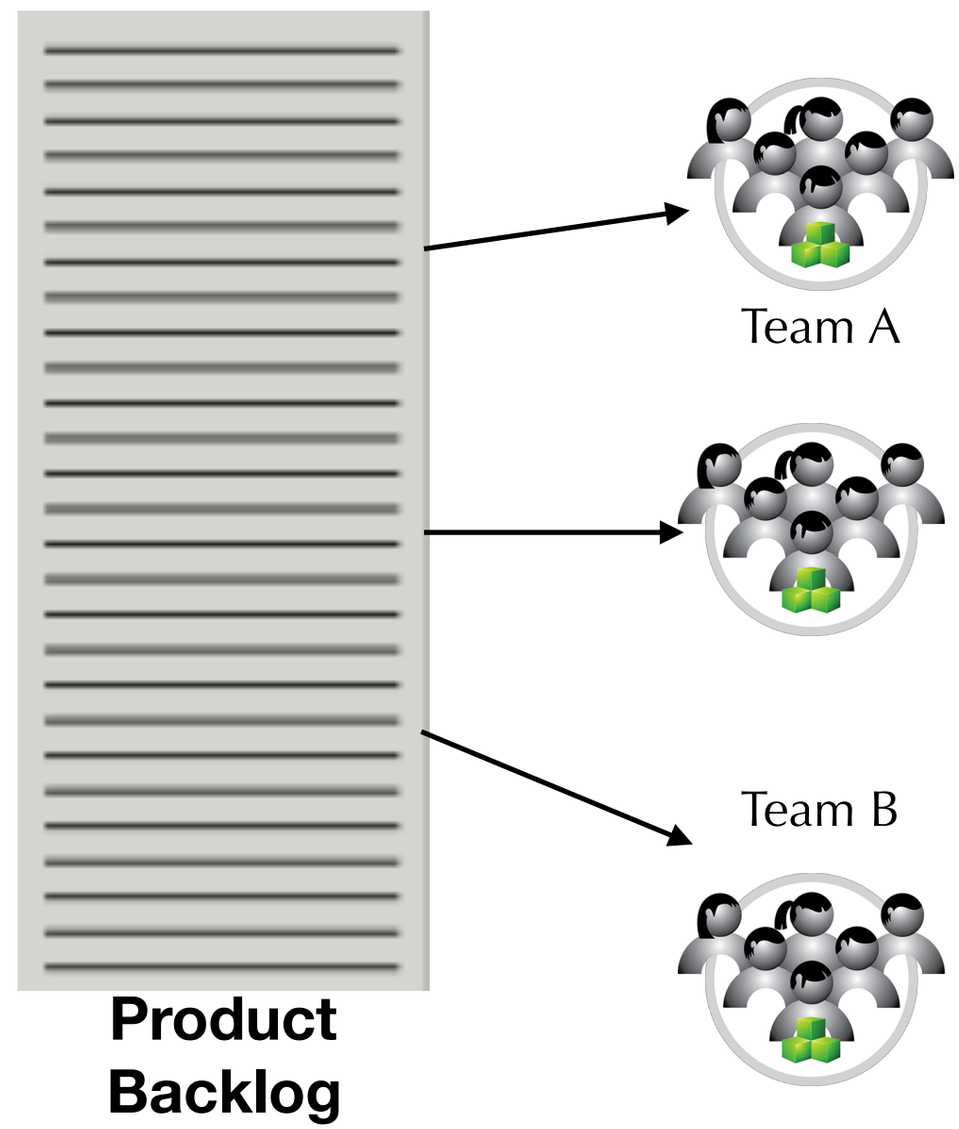
> **disjointness of item knowledge**

Agile Gurukul all rights reserved.

Implicit Backlog

What team can take in reality





Agile Gurukul all rights reserved.

team: update your system model, considering this **puzzle**:

> **"I wonder about the impact on the optimising goal, if disjoint sets are high?"**

> **start** with this variable

  > **disjointness of item knowledge** (top right side)

Agile Gurukul all rights reserved.

COACH & group

> update your system model:

> **effort applied by teams learning more items**

> **# implicit backlogs**

Agile Gurukul all rights reserved.

"**implicit** backlogs"

"**team-level views** on the 1 backlog"

Agile Gurukul all rights reserved.

if the “disjoint set”
value is high

->

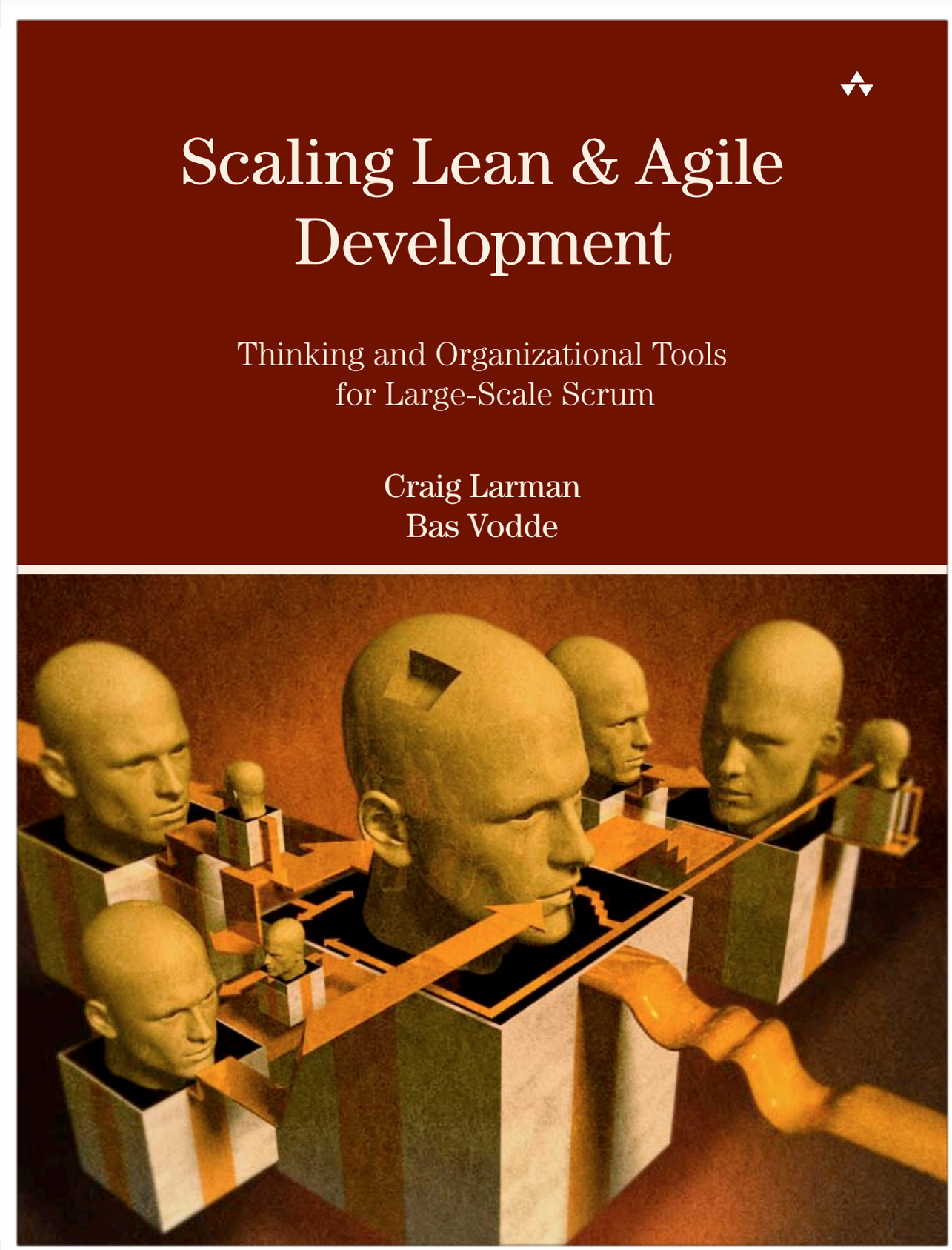
implicit backlogs
grow stronger

Agile Gurukul all rights reserved.

backlogs can create

queue dynamics

Agile Gurukul all rights reserved.



Thinking Tools

2. Systems Thinking
3. Lean Thinking
4. Queueing Theory
5. False Dichotomies
6. Be Agile

Organizational Tools

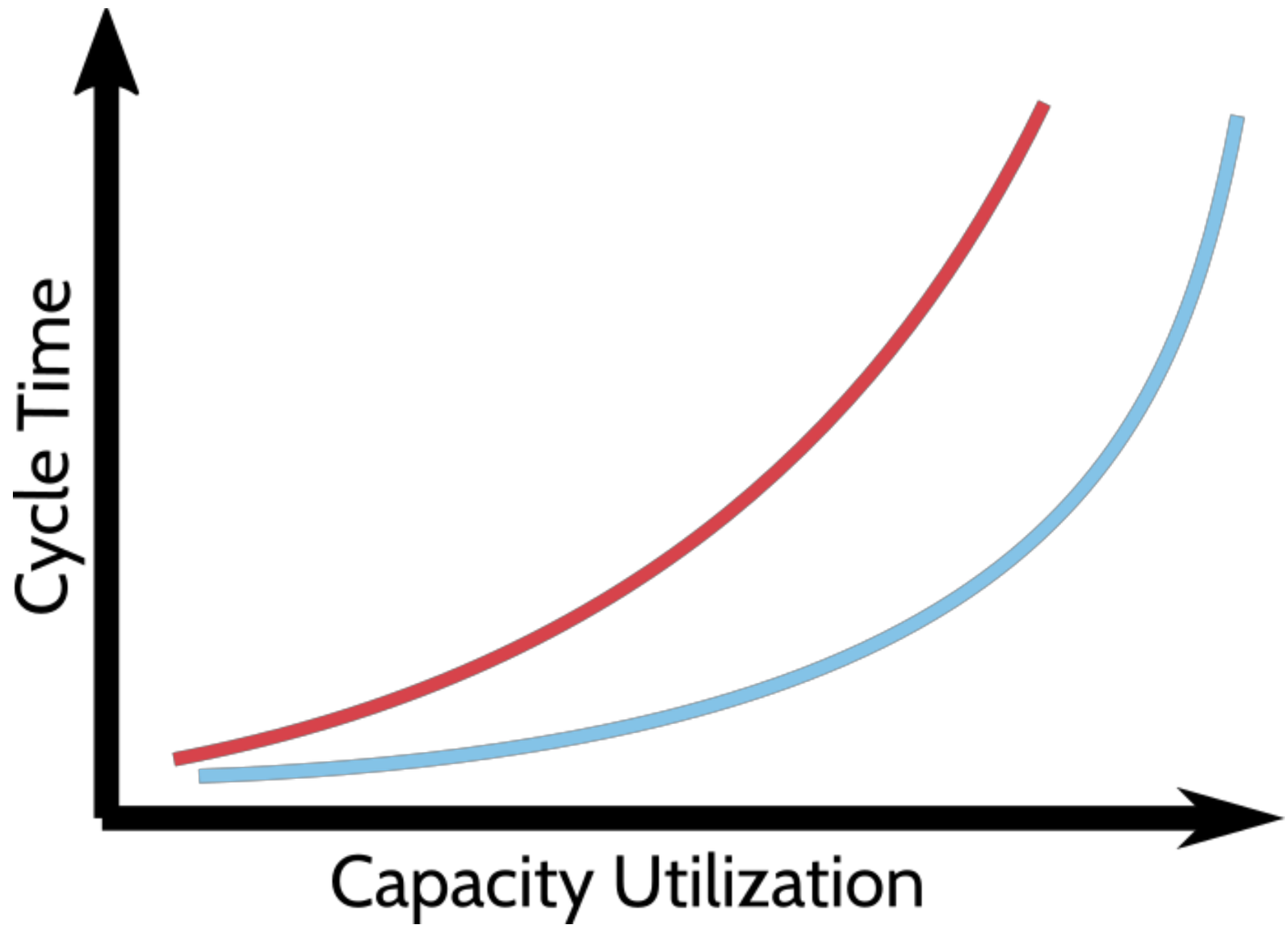
7. Feature Teams
8. Teams
9. Requirement Areas
10. Organization
11. Large-Scale Scrum

Agile Gurukul all rights reserved.

Queues & Variability

Problem with Queues

Optimize Queue management or Remove Queues?



Agile Gurukul all rights reserved.

coach

> if system optimizing goals are

> **adaptiveness** for **highest value**, at company level

> *… should teams **spend time** learning about **more** items?*

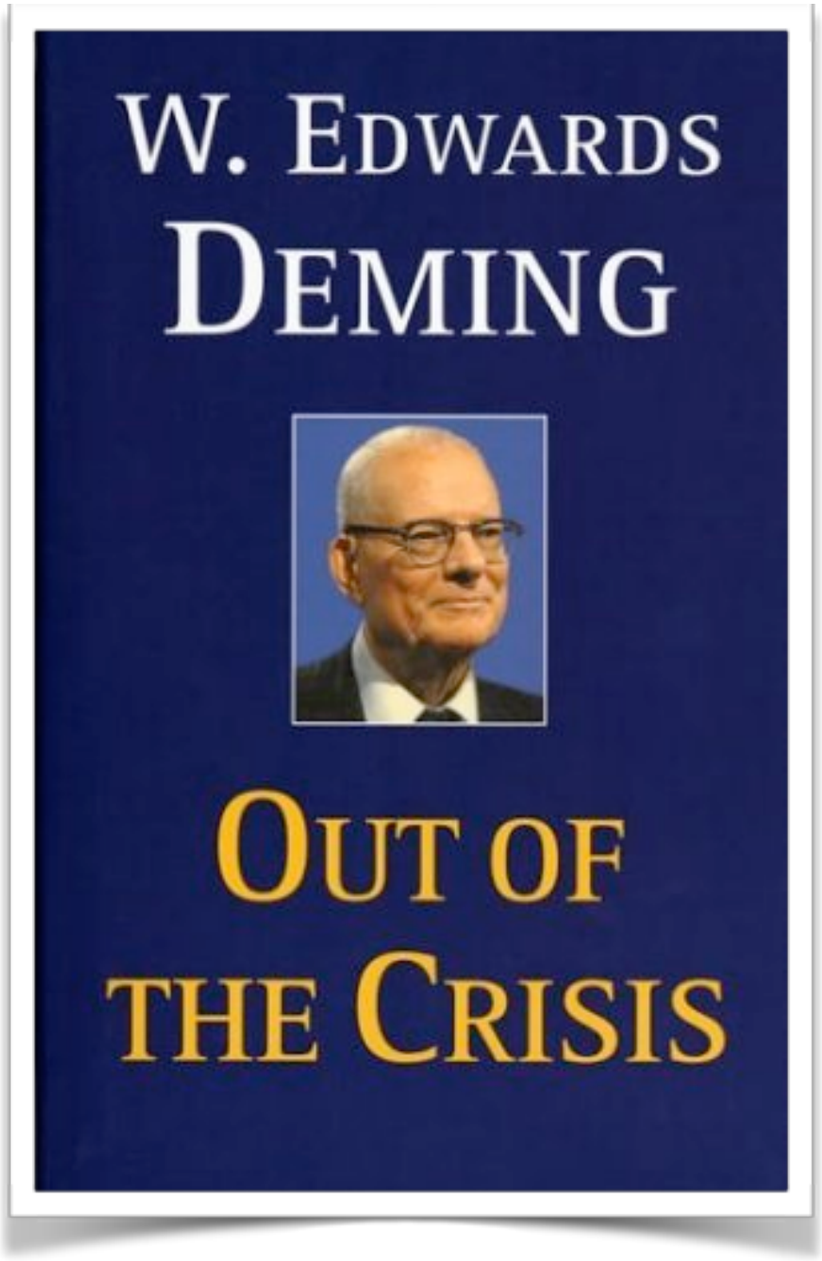
Agile Gurukul all rights reserved.

Local Optimisation Cognitive Bias

Q: "Isn't it inefficient and wasteful to have teams learning about many items?"

Agile Gurukul all rights reserved.

Total Quality Management



"constancy of purpose"

Agile Gurukul all rights reserved.

Local Optimisation Cognitive Bias

Q: "Why does each team have a team backlog, and that only does narrow learning?"

A: "Because it's **best, and most efficient**."

Agile Gurukul all rights reserved.

Descaling with LeSS

remove

local Optimisation of
backlogs…

that comes from:
team backlogs
(and all their org
design elements)

→

1 Product Backlog

(and no *hidden* “team backlogs/views”, and avoid “implicit backlogs”)

Agile Gurukul all rights reserved.

Reflections

Agile Gurukul all rights reserved.

(if not already) team

> **update** your system model to be like the **"debrief model"**

Agile Gurukul all rights reserved.

team: standing: round robin

> most **noteworthy** or **interesting** idea so far?

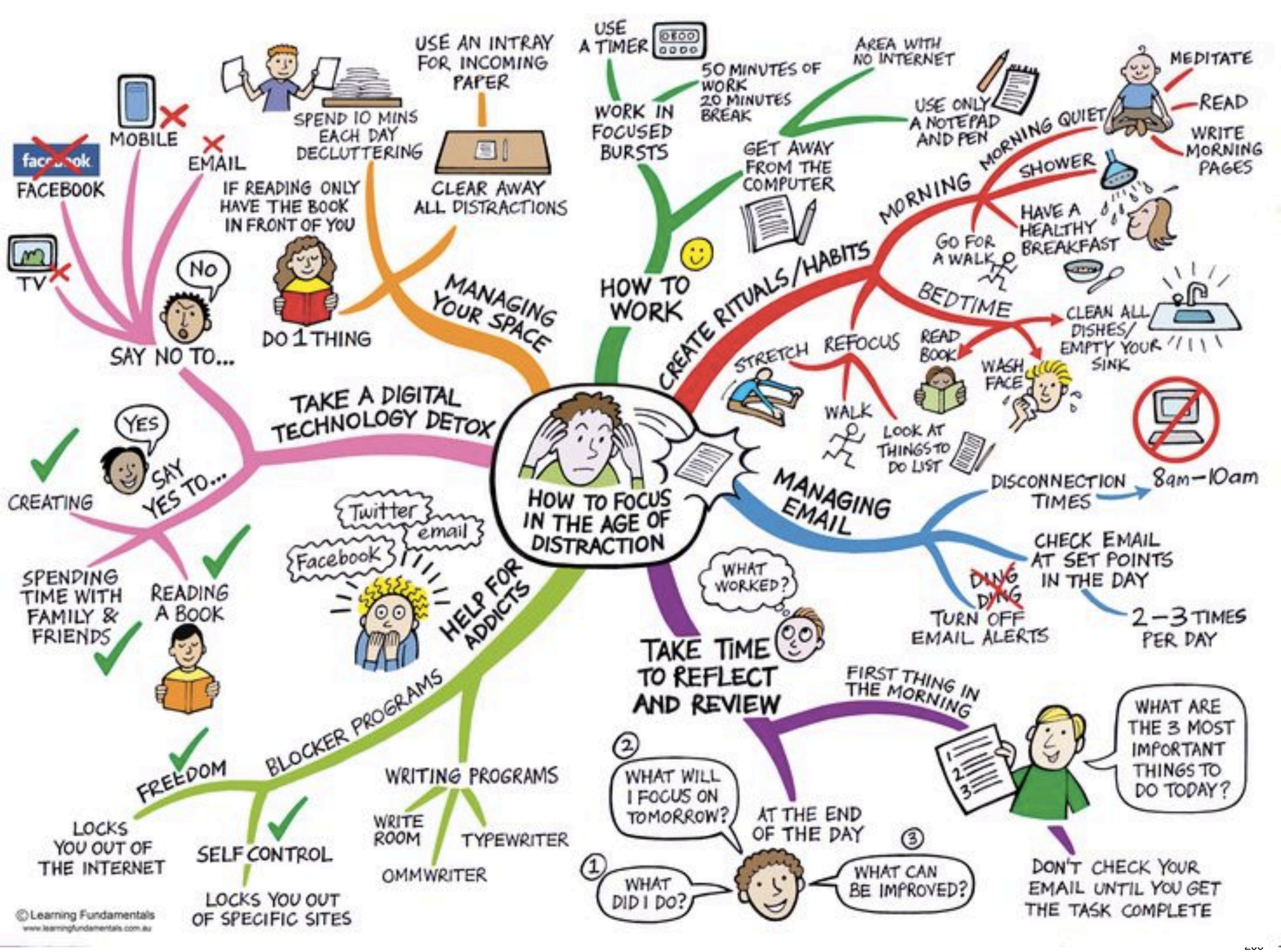
> please **sit** when team is done

Agile Gurukul all rights reserved.

individual

> each person, write a summary of “my interesting idea” on a post-it, and *post it*

Agile Gurukul all rights reserved.


HOW TO FOCUS IN THE AGE OF DISTRACTION
TAKE A DIGITAL TECHNOLOGY DETOX
SAY NO TO...
No
MOBILE
EMAIL
FACEBOOK
TV
SAY YES TO...
Yes
CREATING
SPENDING TIME WITH FAMILY & FRIENDS
READING A BOOK
MANAGING YOUR SPACE
SPEND 10 MINS EACH DAY DECLUTTERING
USE AN INTRAY FOR INCOMING PAPER
CLEAR AWAY ALL DISTRACTIONS
IF READING ONLY HAVE THE BOOK IN FRONT OF YOU
DO 1 THING
HOW TO WORK
USE A TIMER
WORK IN FOCUSED BURSTS
50 MINUTES OF WORK 20 MINUTES BREAK
GET AWAY FROM THE COMPUTER
AREA WITH NO INTERNET
USE ONLY A NOTEPAD AND PEN
CREATE RITUALS/HABITS
MORNING
MORNING QUIET
MEDITATE
READ
WRITE MORNING PAGES
SHOWER
HAVE A HEALTHY BREAKFAST
GO FOR A WALK
BEDTIME
CLEAN ALL DISHES/ EMPTY YOUR SINK
READ BOOK
WASH FACE
STRETCH
REFOCUS
WALK
LOOK AT THINGS TO DO LIST
MANAGING EMAIL
DISCONNECTION TIMES
8am–10am
CHECK EMAIL AT SET POINTS IN THE DAY
2–3 TIMES PER DAY
TURN OFF EMAIL ALERTS
DING
DING
HELP FOR ADDICTS
Twitter
email
Facebook
BLOCKER PROGRAMS
FREEDOM
LOCKS YOU OUT OF THE INTERNET
SELF CONTROL
LOCKS YOU OUT OF SPECIFIC SITES
WRITING PROGRAMS
WRITE ROOM
TYPEWRITER
OMMWRITER
TAKE TIME TO REFLECT AND REVIEW
WHAT WORKED?
FIRST THING IN THE MORNING
WHAT ARE THE 3 MOST IMPORTANT THINGS TO DO TODAY?
DON'T CHECK YOUR EMAIL UNTIL YOU GET THE TASK COMPLETE
AT THE END OF THE DAY
1
WHAT DID I DO?
2
WHAT WILL I FOCUS ON TOMORROW?
3
WHAT CAN BE IMPROVED?
© Learning Fundamentals
www.learningfundamentals.com.au

avoid "teaching" people LeSS (e.g. "1 PB")

prefer: coach them to **discover** it, with a focus on **why**

Agile Gurukul all rights reserved.

Day 2

Agile Gurukul all rights reserved.

Agenda

- ~~Introduction~~
- ~~Principles~~
- ~~Overview~~
- ~~Organised by Customer Value~~

- Product
- Definition of Done
- Product Owner and Product Backlog

- Sprint Planning, Review Retrospective
- Integration and Coordination
- Management and Orientation
- Adoption
- Scrum Master

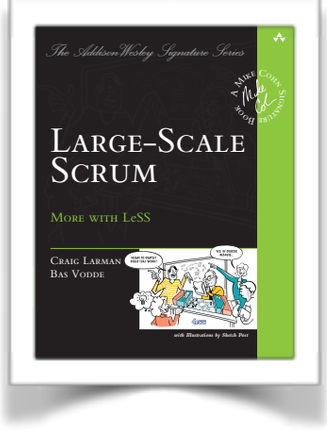
Agile Gurukul all rights reserved.

Local Optimisation in Product Definition

Agile Gurukul all rights reserved.

what are we about to learn?

Agile Gurukul all rights reserved.



Guide: Getting Started

0. Educate Everyone

1. **Define product**

2. Define ‘done’

3. Have appropriately-structured teams

4. Only the Product Owner provides work for teams

5. Keep “managers” away from teams

Agile Gurukul all rights reserved.

THE EXPERT

Products or Projects?

Products or Projects

Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

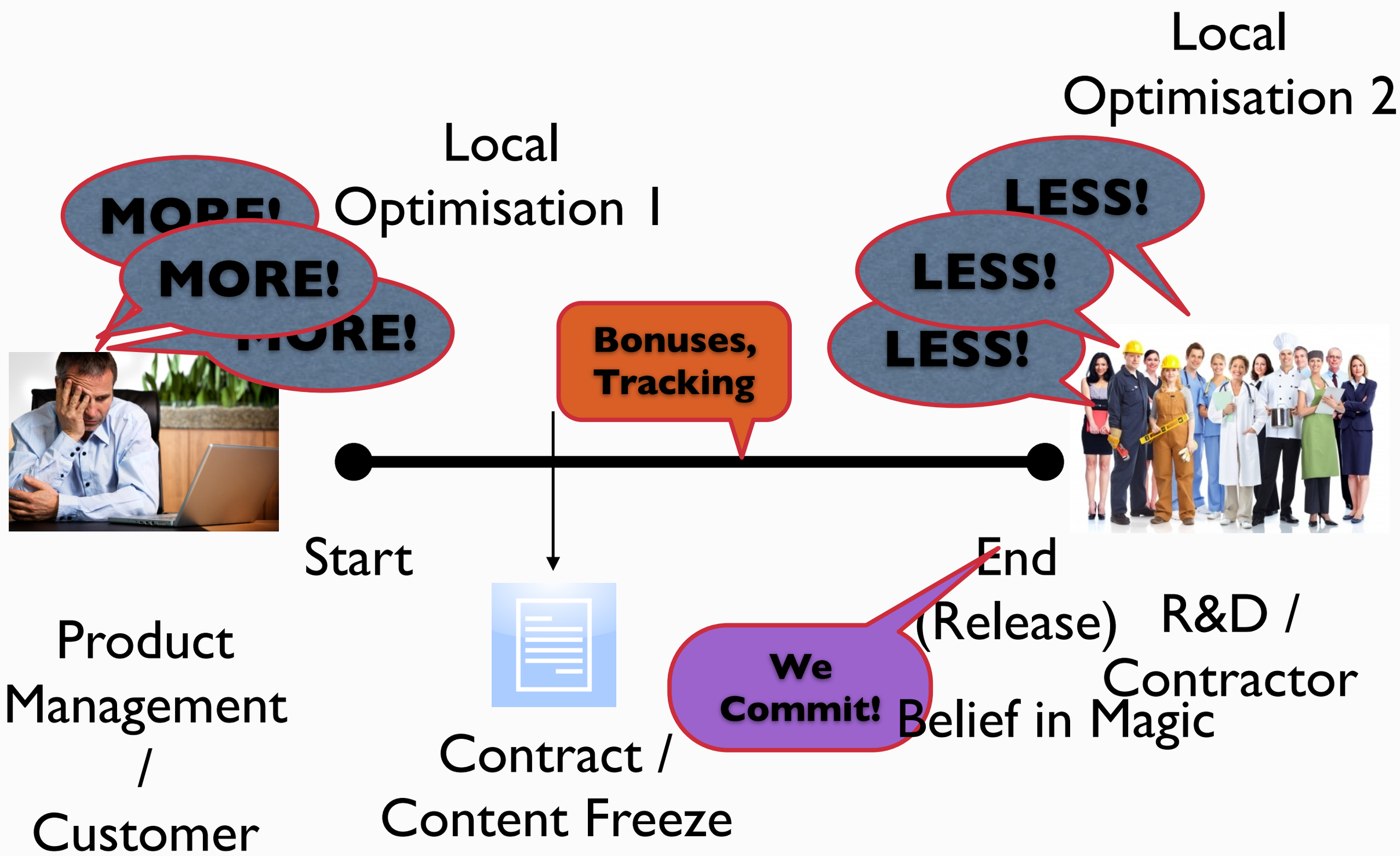
Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

Contract Game

The Contract/Commitment Game exists. e.g., there is “internal contract negotiation”, a project/program, a scope & date deadline “internal contract” with a project/program manager responsible for the project/program, “Business” “throws the project over the wall” and holds Development responsible for meeting needs, etc.

Contract Game



Scrum Ends Contract Game

The Contract/Commitment Game exists. e.g., there is "internal contract negotiation", a project/program, a scope & date deadline "internal contract" with a project/program manager responsible for the project/program, "Business" "throws the project over the wall" and holds Development responsible for meeting needs, etc.

Scrum puts this responsibility back on the Product Owner and customers through the inspect and adapt and the Sprint Review.

To End Contract Game

We need to focus on customer instead of project/programs
and move to
Product paradigm and
Adaptive planning driven by Customer needs and Business and Delivering every Sprint

Narrow vs Broad Product Definitions

Agile Gurukul all rights reserved.

group

> suppose we want to do system modelling that includes analysis of **narrow vs broad waistline**

> what is the **variable name**?

Agile Gurukul all rights reserved.

group

> suppose we want to do system modelling that includes analysis of **narrow vs broad product definition**

> what is the **variable name**?

Agile Gurukul all rights reserved.

team

> sketch a systems model, bearing in mind this **puzzle**:

> **"I wonder about the impact of narrow versus broad product definitions on the system optimising 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 from a company view
6. adaptiveness of teams to change direction at the company level

Agile Gurukul all rights reserved.

coach

> if system optimising goals are

> **adaptiveness** for **highest value**, at company level

> … *broader or narrower product definition?*

Agile Gurukul all rights reserved.

support **secondary concerns**
or **constraints**
(e.g. "product complexity")

without sub-optimising
the system goal

Agile Gurukul all rights reserved.

Coach & group (or per Team)

> **first, synch with the last debrief model**

> start with these variables verbatim: (lower right quadrant)

1. **# of inter-team task dependencies** (i.e. a team probably has to wait for another team to do "their part")
2. **strength of "private code" policies**
3. **average complete end2end customer feature cycle time**

Agile Gurukul all rights reserved.

coach

> relationship of?

> **low *complete* end2end customer feature cycle time**

> ***adaptiveness* from learning from *delivering* new items**

Agile Gurukul all rights reserved.

team

> **update** your system model to be like the **"debrief model"**

Agile Gurukul all rights reserved.

coach

> if system optimising goal is

> **low *complete* end2end customer feature cycle time**

> *… broader or narrower product definition?*

Agile Gurukul all rights reserved.

therefore…

Agile Gurukul all rights reserved.

LeSS Rule(s)

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

Agile Gurukul all rights reserved.

coach & group

> how **quickly & broadly** should we broaden the product definition?

Agile Gurukul all rights reserved.

focus on **why**

own vs rent

Agile Gurukul all rights reserved.

miscellany on product definition…

Agile Gurukul all rights reserved.

"IO Channels" & Product Definition

Google Maps?

iOS Google Maps?

Android Google Maps?

Agile Gurukul all rights reserved.

“IO Channels” & Product Definition

Google Maps?

iOS Google Maps?

Android Google Maps?

Agile Gurukul all rights reserved.

so they go to one broader product definition…

Agile Gurukul all rights reserved.

A "Broad" Product & **"Implicit Backlogs"**

"Oh yes, we have only **one broad product**, and…

Team-iOS = iOS items

Team-Android = Android items"

Agile Gurukul all rights reserved.

A "Broad" Product & **"Implicit Backlogs"**

"Oh yes, we have only **one broad product**, and…

Team-IOS = IOS items

Team-Android = Android items"

Agile Gurukul all rights reserved.

internal broad
product definition

vs

narrower external
multi-products definitions

Agile Gurukul all rights reserved.

"Product" Definitions Consistent with *The Goals*?

- common **platform/framework/service** group? (not directly sold)
- **service** or **APIs** group? (not directly sold)
- "**component**" or "**module**" or "**microservice**" or "**application**" group? (not directly sold)
- **library** group? (not directly sold)
- **front**-end group?
- **back**-end group?
- a '**project**' for some features?

Agile Gurukul all rights reserved.

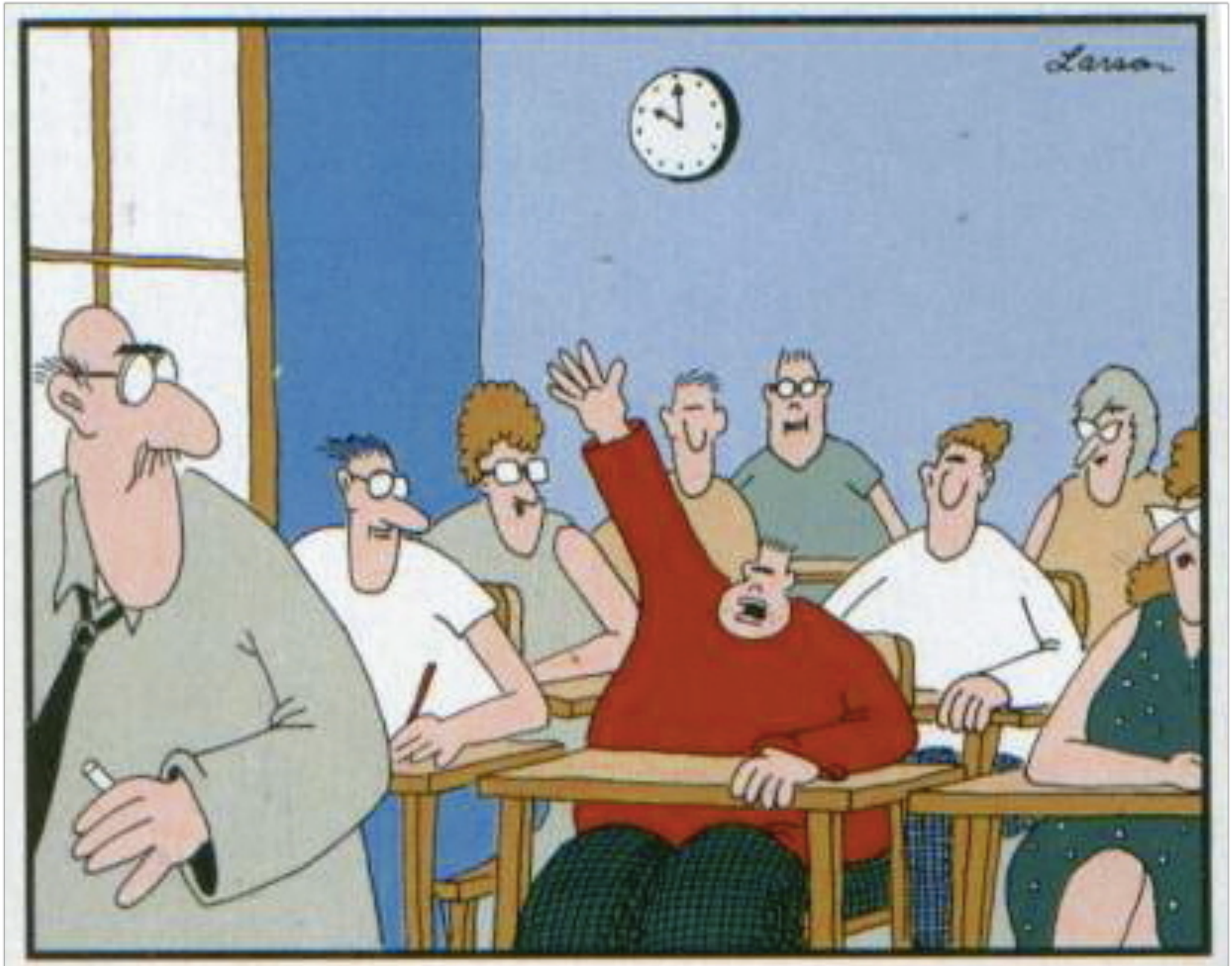
Product Definitions Consistent with *The Goals*?

- common **platform/ framework/service** group? (not directly sold)
- **service** or **APIs** group? (not directly sold)
- "**component**" or "**module**" or "**microservice**" or "**application**" group? (not directly sold)
- **library** group? (not directly sold)
- **front**-end group?
- **back**-end group?
- a '**project**' for some features?

Agile Gurukul all rights reserved.

limits of **expanding**
# of teams with
1 (and only 1)
Product Backlog &
Product Owner? ...

Agile Gurukul all rights reserved.



"Mr. Osborne, may I be excused?
My brain is full."

Agile Gurukul all rights res

team

> 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-prioritise 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 each item** (tech & domain) (far right, lower)
4. **average size of item a team implements** (far right, lower)
5. size (breadth) of product definition

Agile Gurukul all rights reserved.

coach

> other variables of relevance to last exercise?

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

Agile Gurukul all rights reserved.

support **secondary concerns**
or **constraints**
(e.g. “PO head should not explode”)

without sub-optimising the system goal

Agile Gurukul all rights reserved.

What happens to the Product Owner and Developers as the product gets broader and broader?

Agile Gurukul all rights reserved.

but no matter what we do to help the people's brains, at some point, we will reach a limit...

Agile Gurukul all rights reserved.

therefore…

Agile Gurukul all rights reserved.

LeSS Huge

Agile Gurukul all rights reserved.

divide

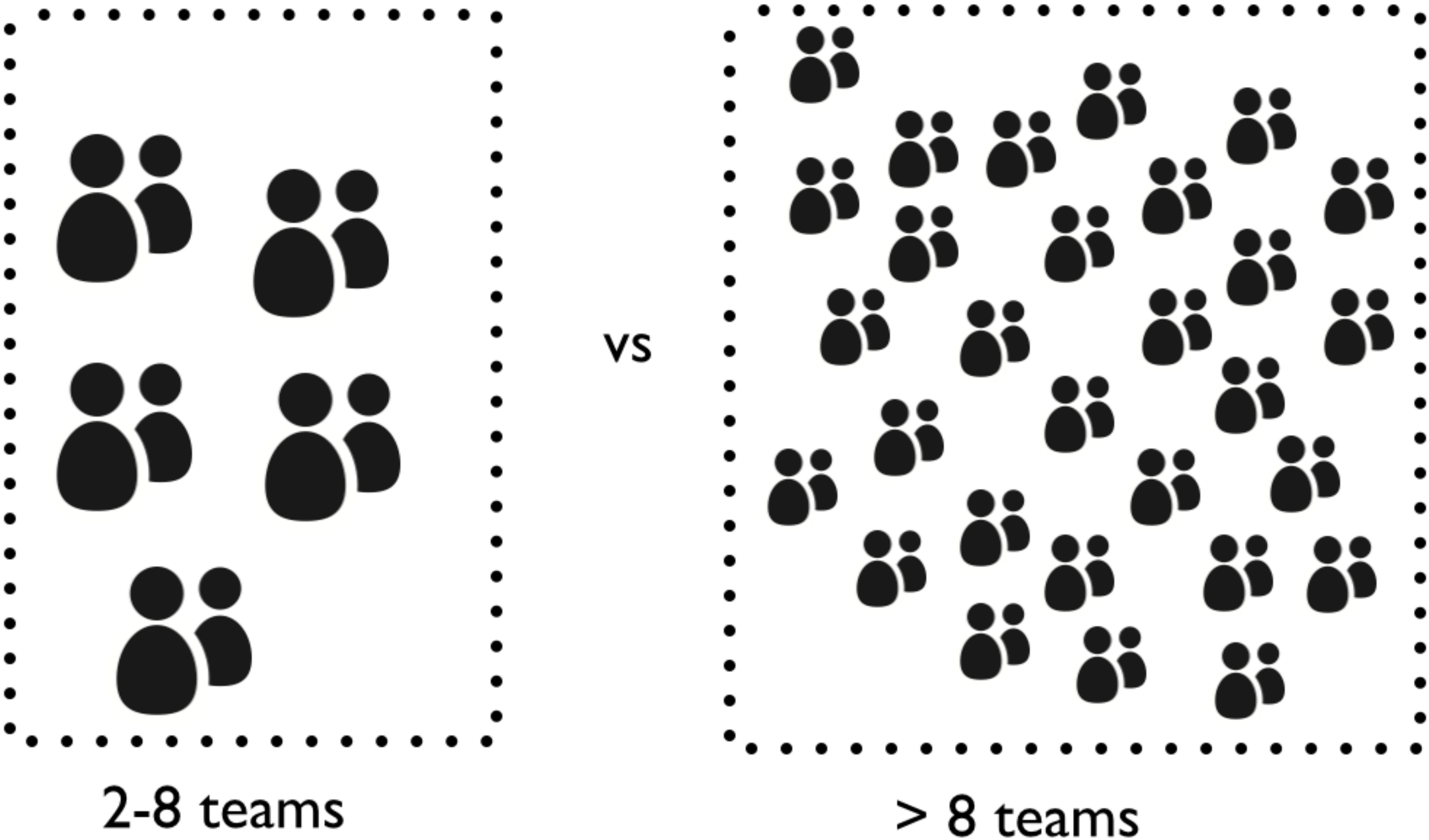
warning: dividing leads to **local Optimisation**

Agile Gurukul all rights reserved.

are there any
proxy variables
correlated with "the
heads are hurting", and
need to transition to
LeSS Huge?

Agile Gurukul all rights reserved.

LeSS Huge: After “8” Teams



Agile Gurukul all rights reserved.

but…

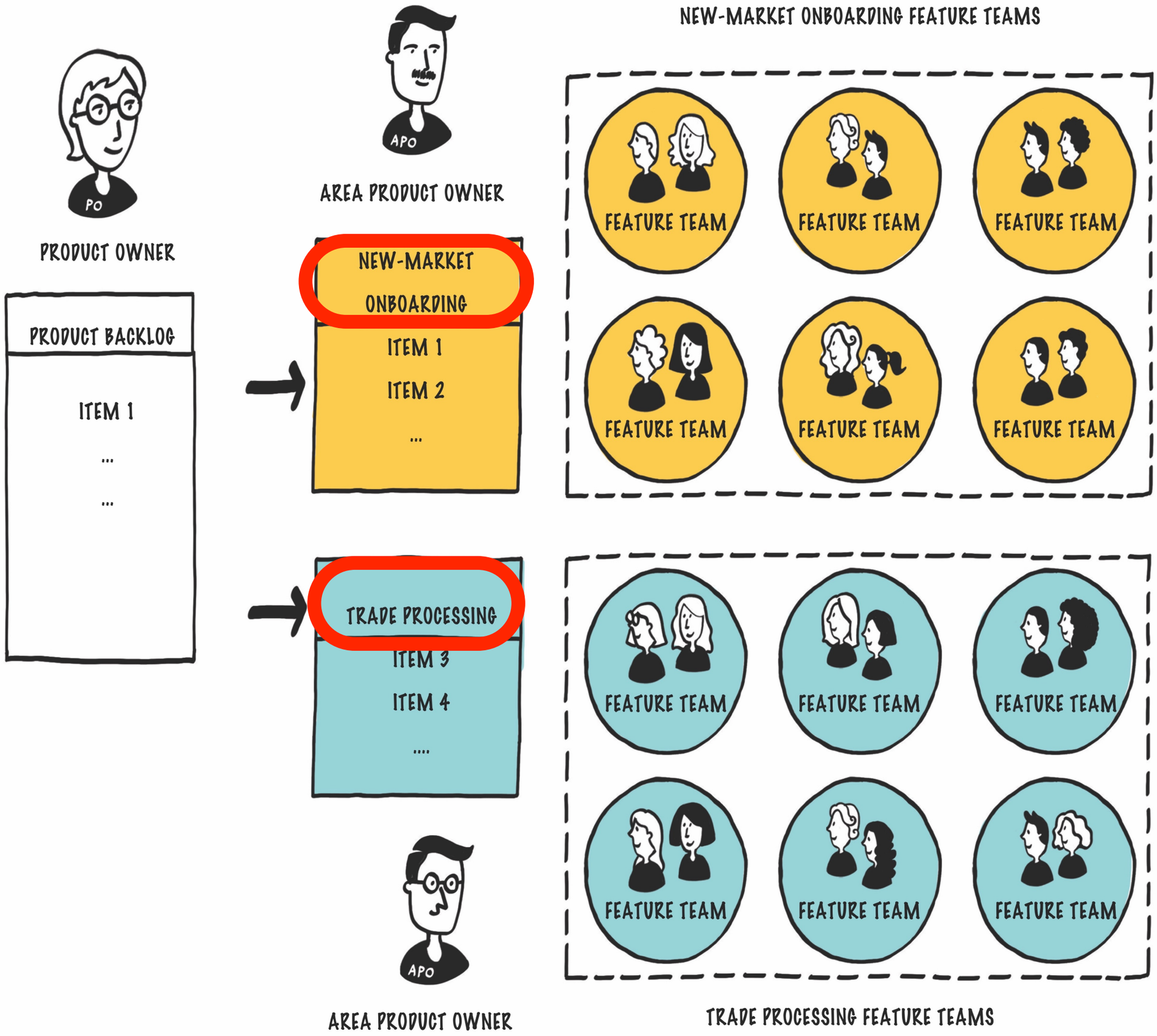
divide by what

dimension?

(architecture, …?)

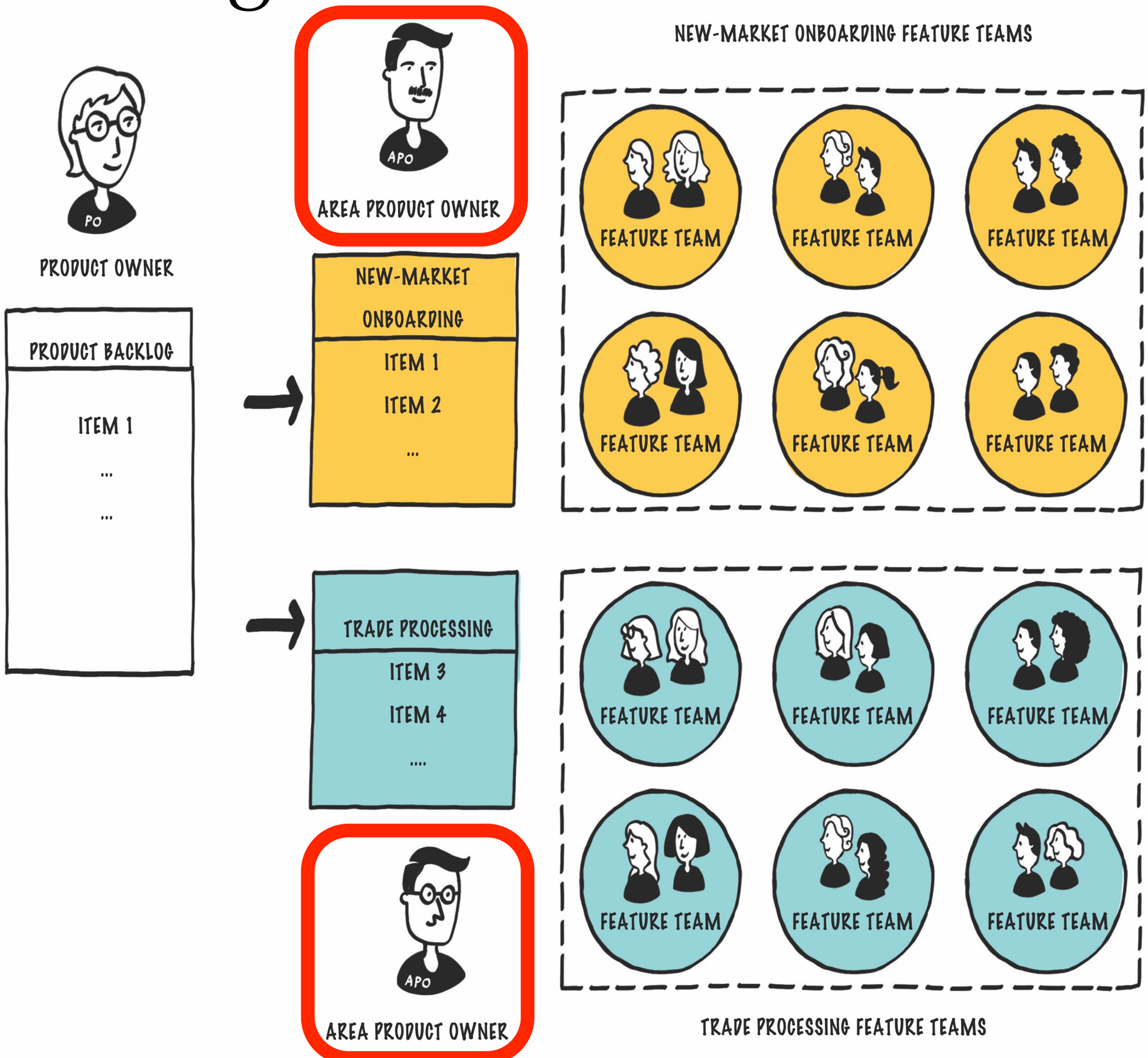
Agile Gurukul all rights reserved.

LeSS Huge: **Requirement Areas**



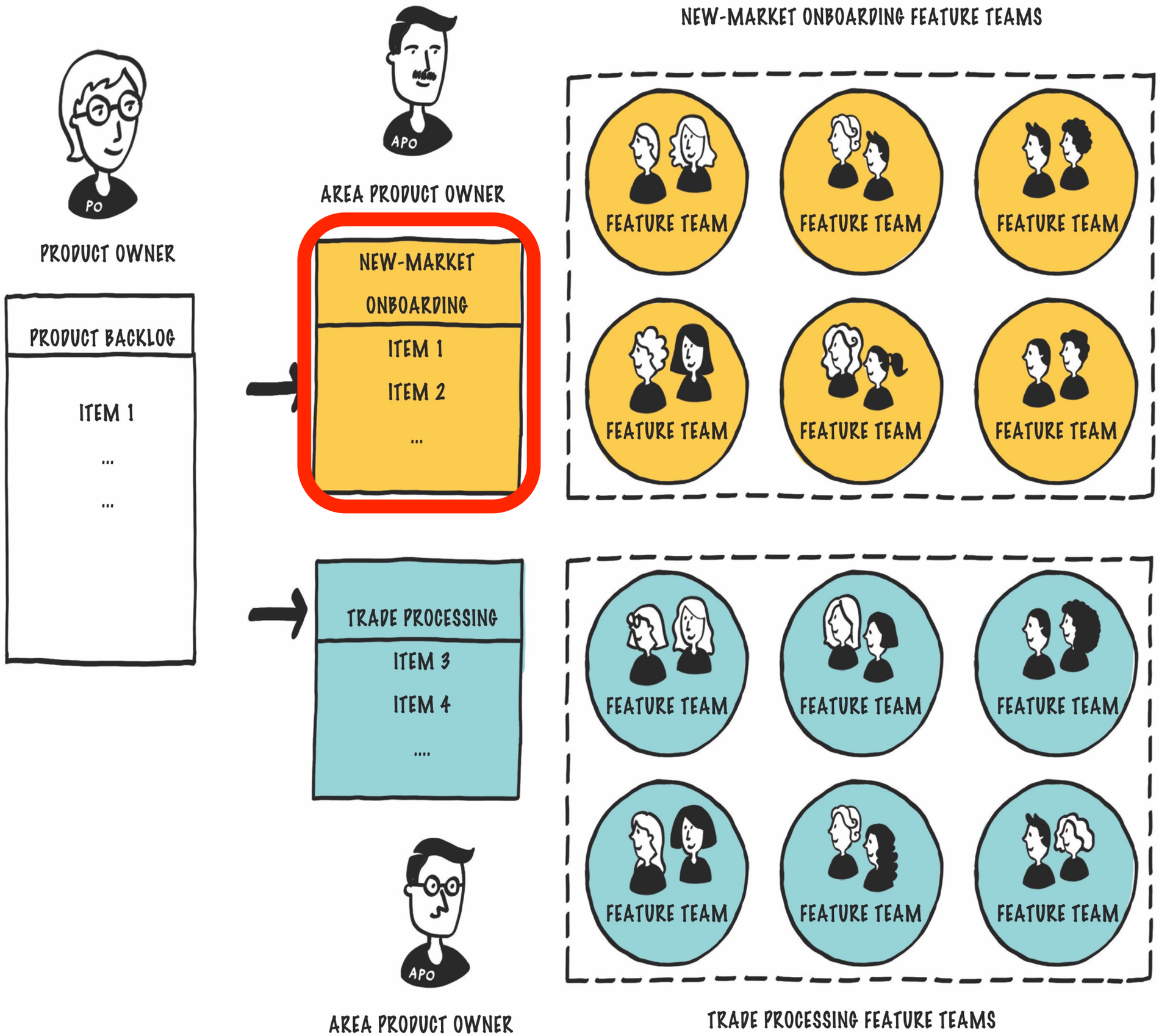
Agile Gurukul all rights reserv

LeSS Huge: **Area Product Owners**



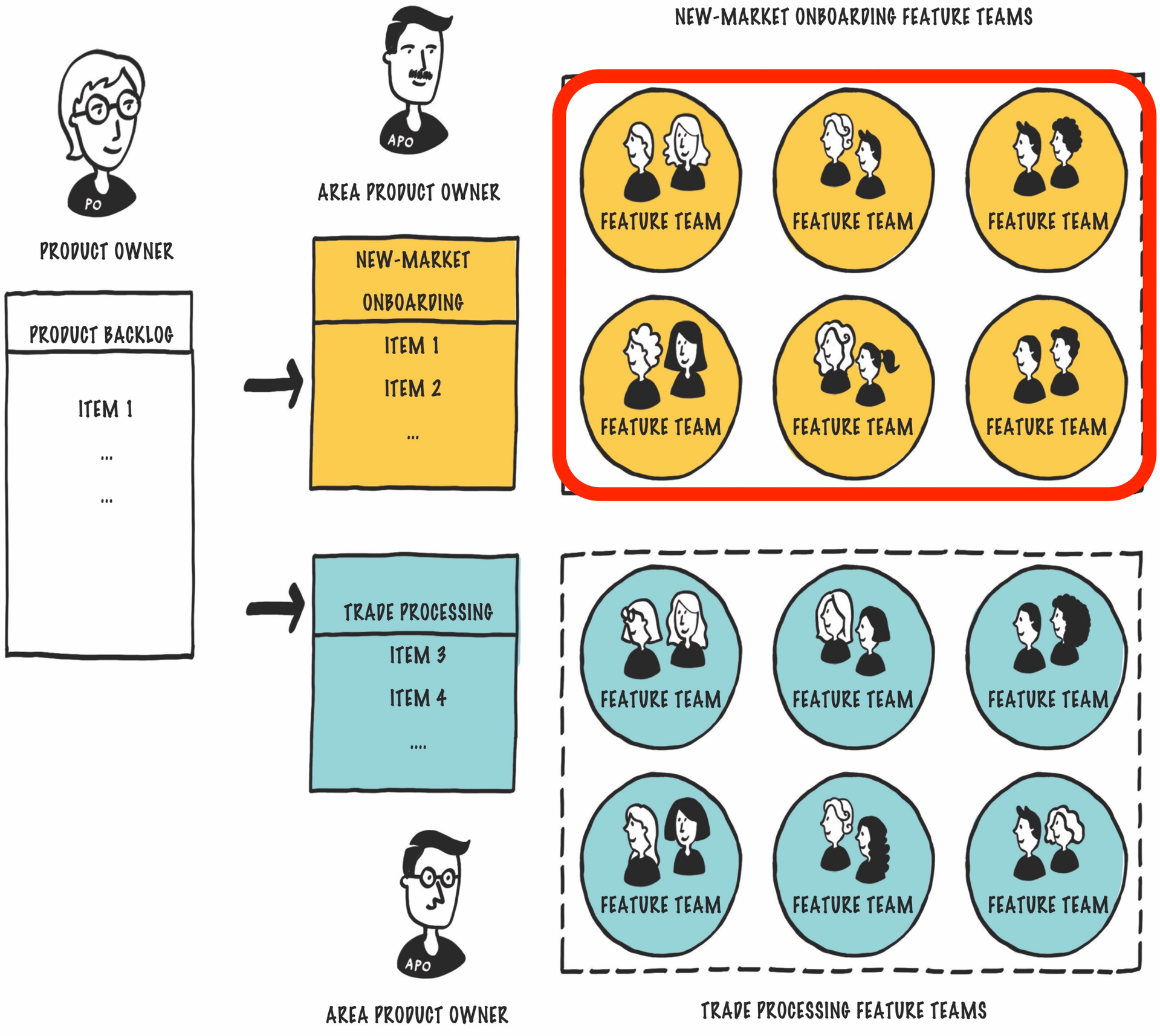
Agile Gurukul all rights reserv

LeSS Huge: **Area Backlog**



Agile Gurukul all rights reserv

LeSS Huge: **Area Teams**



Agile Gurukul all rights reserv

LeSS Huge framework

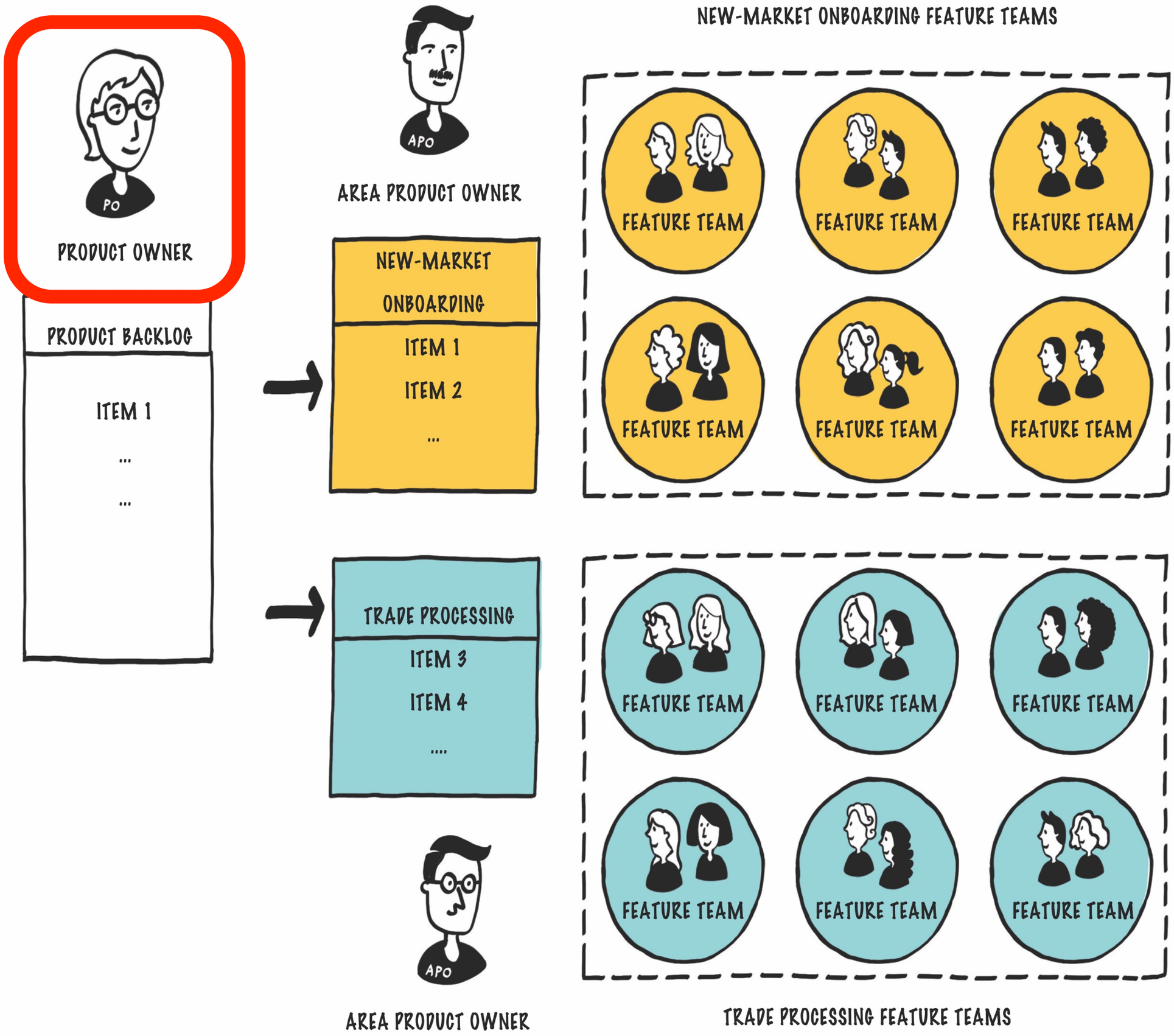
- LeSS Huge framework is NOT per se desirable; why? …
- **dividing -> local Optimisation**
- an “uncomfortable art of the possible” so Product Owner & Developer heads don’t explode ;)

Agile Gurukul all rights reserved.

why not divide into separate **products**?

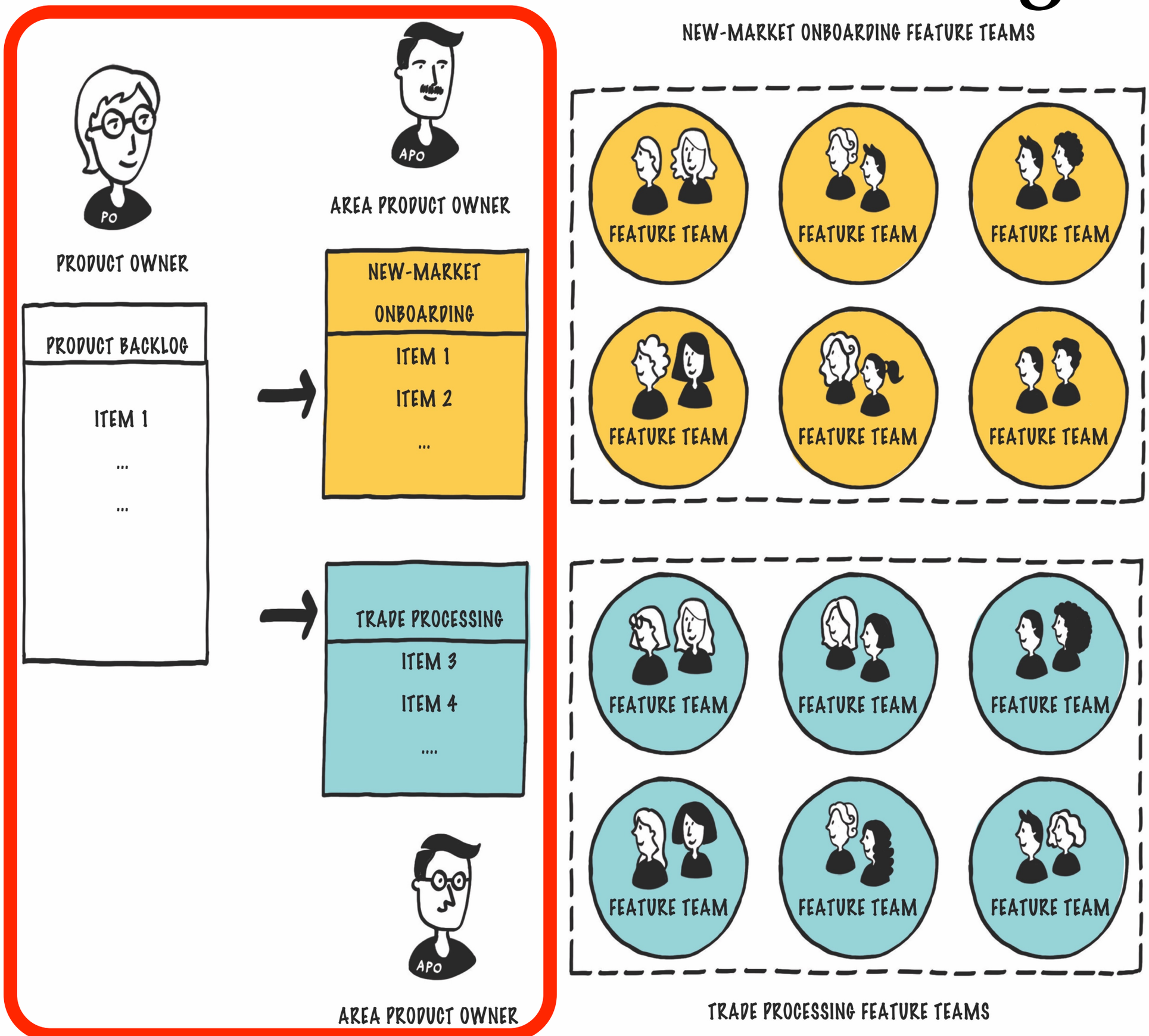
Agile Gurukul all rights reserved.

LeSS Huge: Role of PO?



Agile Gurukul all rights reserv

PO Team & POT Meetings



Agile Gurukul all rights reserv

divide

worshipping customers

not

worshipping code

Agile Gurukul all rights reserved.

COACH & group (or per team)

> start with these variables, and given this **puzzle**:

> **"How big should RAs be? What are the sub-Optimisations from adding RAs?"**

> (lower left quadrant of wall)

1. **size (breadth) of a RA** (Requirement Area)
2. **# RAs**
3. **# teams in a RA**
4. # backlogs/lists
5. size (breadth) of product definition

Agile Gurukul all rights reserved.

coach

> if system optimising goals are

> **adaptiveness** for **highest value**, at company level

> … *bigger or smaller Requirement Areas?*

Agile Gurukul all rights reserved.

therefore…

Agile Gurukul all rights reserved.

LeSS Rule(s)

Each Requirement Area has between "4-8" teams. (preference towards 8)

Agile Gurukul all rights reserved.

COACH & group

> which is more consistent with *The Goals*?

> Requirement Area **Bond Trading**: 4 teams, ***AND*** … Requirement Area **Stock Trading**: 4 teams

> … or …

> Requirement Area **Bond & Stock Trading**: 8 teams

Agile Gurukul all rights reserved.

Product Definition (again)

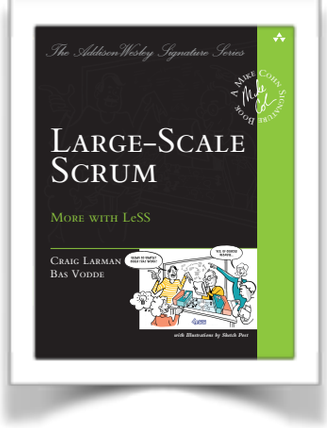
Agile Gurukul all rights reserved.

how to define a broader product? ...

Agile Gurukul all rights reserved.

Would this "Make Sense"?

Agile Gurukul all rights reserved.



Guide: Define Your Product

applying the

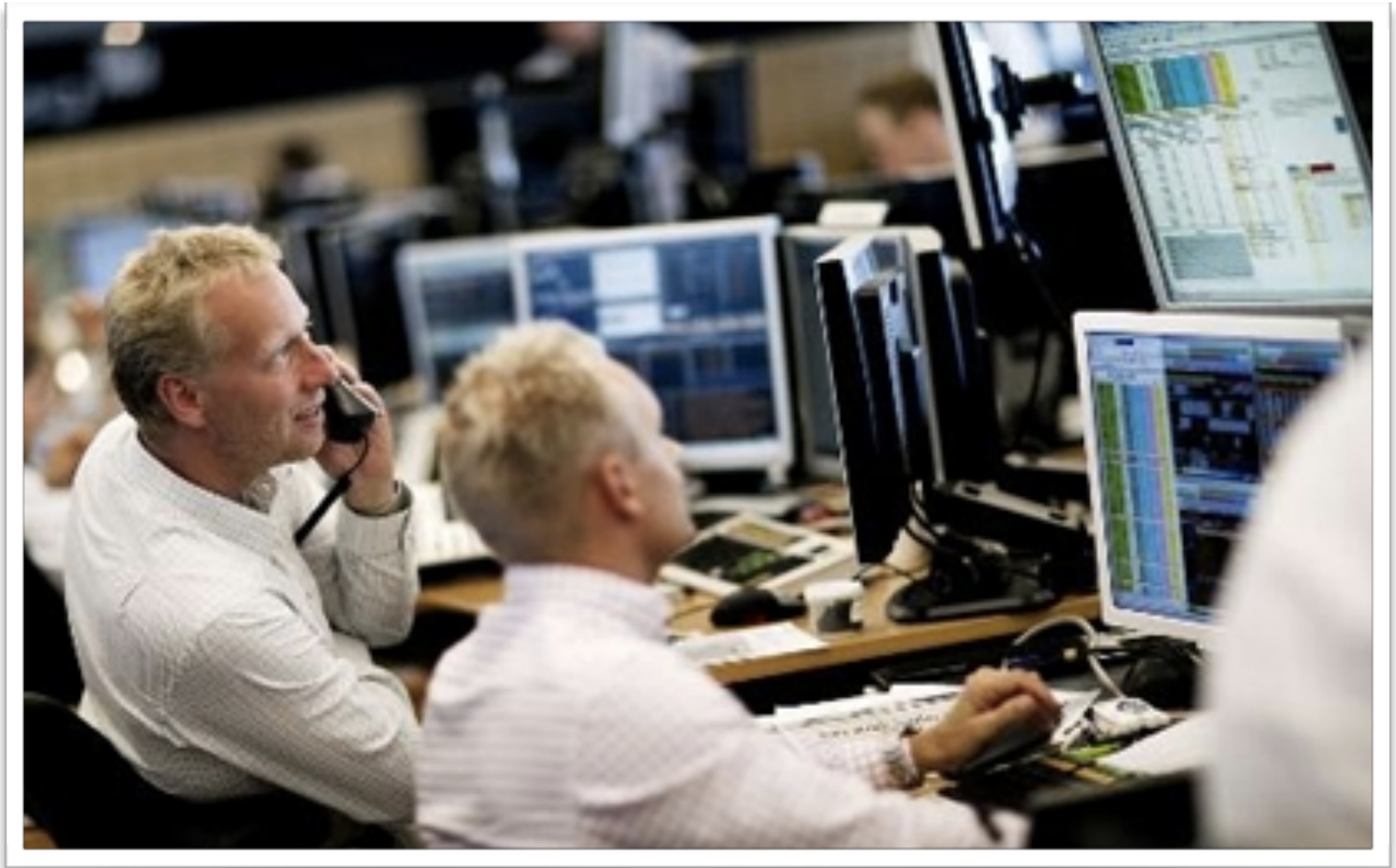
expanding &

restraining

questions…

Agile Gurukul all rights reserved.

Example: Financial Trading



Agile Gurukul all rights reserved.

what and where are the current backlogs or lists?

what are the current "product" definitions?

Agile Gurukul all rights reserved.

1. Expand Product as Broad as *Possible*

- **customer focus**: What would the end customers answer if we ask them, "What is the product?" What spans the customer journey?
- **family**: Family of similar products? Do we have meaningful components that are shared or functionality that is the same across our current products?
- **system**: Our product is part of? What problem does the product solve for end customers?

Agile Gurukul all rights reserved.

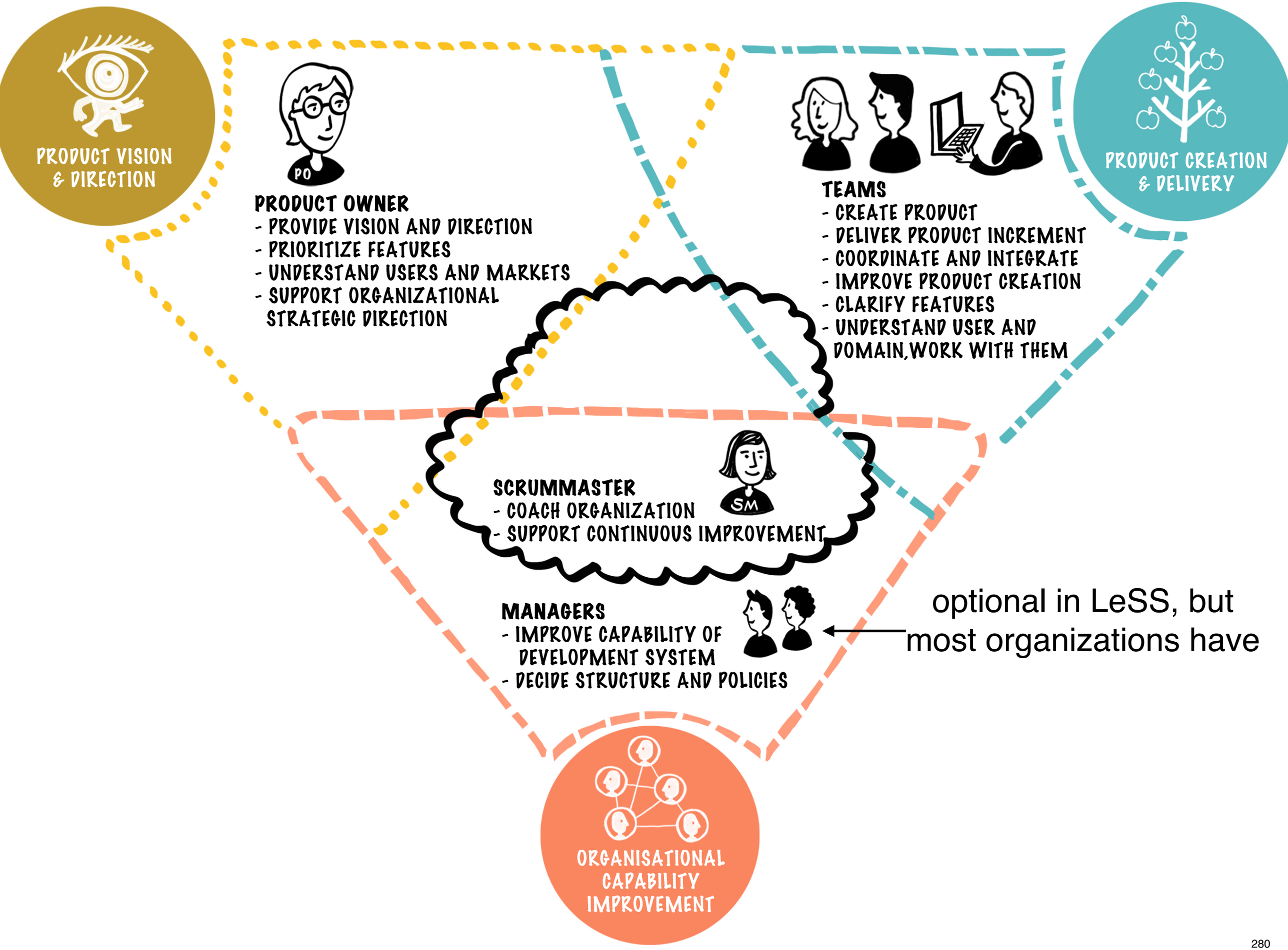
2. Restrain your Product as *Practical*

- **commonality**: What is the product vision? Who are the customers? What is the product's customer domain?
- **structural boundaries**: What development is within our company? How much structural change is *practical*?

Agile Gurukul all rights reserved.

a responsibility of **managers** in a LeSS organisation? ...

Agile Gurukul all rights reserved.


PRODUCT VISION & DIRECTION
PO
PRODUCT OWNER
- PROVIDE VISION AND DIRECTION
- PRIORITIZE FEATURES
- UNDERSTAND USERS AND MARKETS
- SUPPORT ORGANIZATIONAL STRATEGIC DIRECTION
PRODUCT CREATION & DELIVERY
TEAMS
- CREATE PRODUCT
- DELIVER PRODUCT INCREMENT
- COORDINATE AND INTEGRATE
- IMPROVE PRODUCT CREATION
- CLARIFY FEATURES
- UNDERSTAND USER AND DOMAIN,WORK WITH THEM
SM
SCRUMMASTER
- COACH ORGANIZATION
- SUPPORT CONTINUOUS IMPROVEMENT
MANAGERS
- IMPROVE CAPABILITY OF DEVELOPMENT SYSTEM
- DECIDE STRUCTURE AND POLICIES
optional in LeSS, but most organizations have
ORGANISATIONAL CAPABILITY IMPROVEMENT

Expanding Product Definition

> sometimes "as broad as **ideal**" isn't immediately possible

> a driver for **continuous improvement** by

MANAGERS:

"What prevents expanding the product definition?"

managers: IMPROVING THE SYSTEM

Agile Gurukul all rights reserved.

coach & group: identify cases:

> max “50” person product group?

> smaller LeSS framework

> huge product group?

> LeSS Huge framework

Agile Gurukul all rights reserved.

expert on product

> sketch a **“block architecture” diagram** of

  > major **software & hardware components**

  > the **broader context** that “stuff” is within

  > **don’t** do much **explaining**; focus on sketching

  > maybe use **stickie notes** for each component

Agile Gurukul all rights reserved.

expert on product

- **if teams are organised around the components** (which is common), **write for each component**:
  - the name of the site or sites
  - # developers
    - are there actually “sub component” teams?
  - # component testers
  - # of other people in noteworthy related roles (DBAs? ,,,)
- # of other people (by roles), not attached to a component
  - the name of the site or sites
  - e.g. feature teams, Product Managers, UX, HI, BAs, architects, system engineers, system testers, DBAs, …

Agile Gurukul all rights reserved.

coach

> what and where are the current backlogs or lists?

> what are the current “product” definitions?

Agile Gurukul all rights reserved.

coach, for each product

> apply the expanding & restraining questions (*see next pages again*) to create an ***initial*** product definition "as broad as ***practical***"

Agile Gurukul all rights reserved.

1. Expand Product as Broad as *Possible*

- **customer focus**: What would the end customers answer if we ask them, "What is the product?" What spans the customer journey?
- **family**: Family of similar products? Do we have meaningful components that are shared or functionality that is the same across our current products?
- **system**: Our product is part of? What problem does the product solve for end customers?

Agile Gurukul all rights reserved.

2. Restrain your Product as *Practical*

- **commonality**: What is the product vision? Who are the customers? What is the product's customer domain?
- **structural boundaries**: What development is within our company? How much structural change is practical?

Agile Gurukul all rights reserved.

coach, for each product

> if a complex adoption case, **consider predictable work flows through the components**, to identify likely groupings into feature teams

Agile Gurukul all rights reserved.

coach

> as needed, discuss

> **incremental LeSS Huge adoption** vs "all-at-once" smaller LeSS framework

> **Feature Team Adoption Maps**

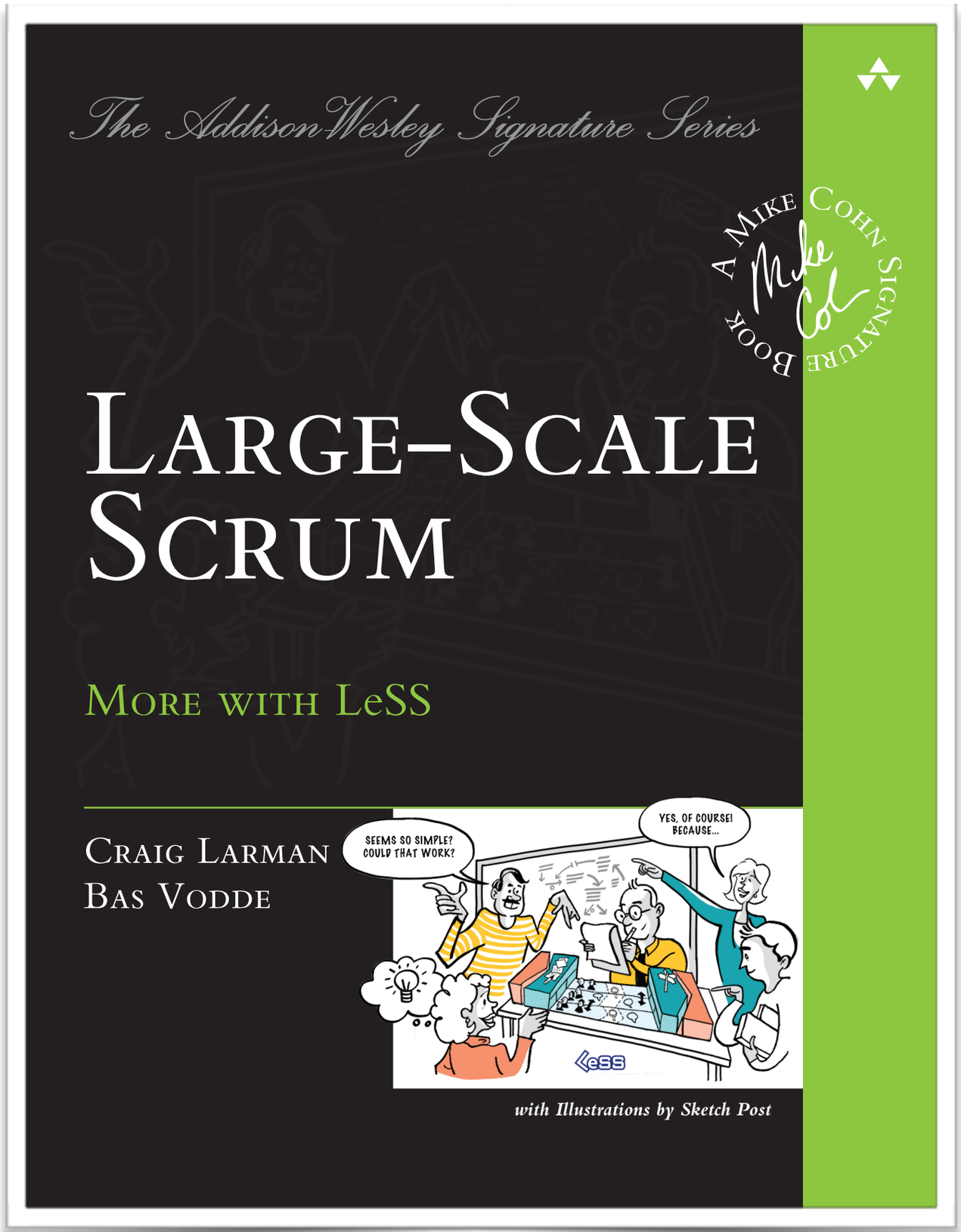
> **& Extended Component Teams**

> **Expanding Definition of Done**

Agile Gurukul all rights reserved.

broadening the product definition will touch on organisational **politics** and "Larman's Laws", due to desires by *Product Managers, Dev Managers* (etc) for **status quo**

Agile Gurukul all rights reserved.



1 More with LeSS 1
2 LeSS 5

LeSS Structure

3 Adoption 41
4 Organize by Customer Value 67
5 Management 111
6 ScrumMasters 133

LeSS Product

7 Product 155
8 Product Owner 171
9 Product Backlog 197
10 Definition of Done 231

LeSS Sprint

11 Product Backlog Refinement 249
12 Sprint Planning 275
13 Coordination and Integration 285
14 Review & Retrospective 313

Agile Gurukul all rights reserved.

Portfolio Management & Product Definition

Agile Gurukul all rights reserved.

“portfolio management”?

for…

products, programs, projects, “value streams”

Agile Gurukul all rights reserved.

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. size (breadth) of product/**etc** definition ("etc" = program, project, "value-stream")
6. # of products/**etc**

Agile Gurukul all rights reserved.

coach

> … *is there a relationship between* ***narrow*** *products/programs/value-streams and the* ***apparent*** *need for* ***portfolio management****?*

Agile Gurukul all rights reserved.

coach

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

Agile Gurukul all rights reserved.

in LeSS,

programs & **projects**

are **eliminated**

Agile Gurukul all rights reserved.

eliminating programs/projects

->

elimination of

"**program/project** portfolio management"

Agile Gurukul all rights reserved.

in LeSS,

so-called "value streams" are eliminated

Agile Gurukul all rights reserved.

eliminating
"value streams"
->
elimination of
"**value-stream**
portfolio management"

Agile Gurukul all rights reserved.

broader products

->

reduction of

"**product** portfolio

management"

Agile Gurukul all rights reserved.

necessary & real product portfolio management

vs

"artificial" portfolio management

Agile Gurukul all rights reserved.

"portfolio managers"

->

artificial manager role for **artificial** portfolio management

Agile Gurukul all rights reserved.

“Real” **Product** Portfolio Management?

the CEO is

in the room

Agile Gurukul all rights reserved.

LeSS **simplifies** or **eliminates**
the need for
"portfolio management" by
broader product definitions
& **elimination of**
programs/projects

Agile Gurukul all rights reserved.

any analysis of LeSS
that claimed
"LeSS doesn't include portfolio management solution"…
completely misses the insight

Agile Gurukul all rights reserved.

therefore…

Agile Gurukul all rights reserved.

LeSS Rule(s)

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.

Agile Gurukul all rights reserved.

Local Optimisation Cognitive Bias

Q: “Why do you have **narrow** products, programs, projects, “component products”, or value streams?”

A: “Because it’s **best, and most efficient**.”

Agile Gurukul all rights reserved.

Descaling with LeSS

remove

local Optimisation in
product definition

that comes from:
narrow products,
programs, projects,
value streams (& their
org design elements)

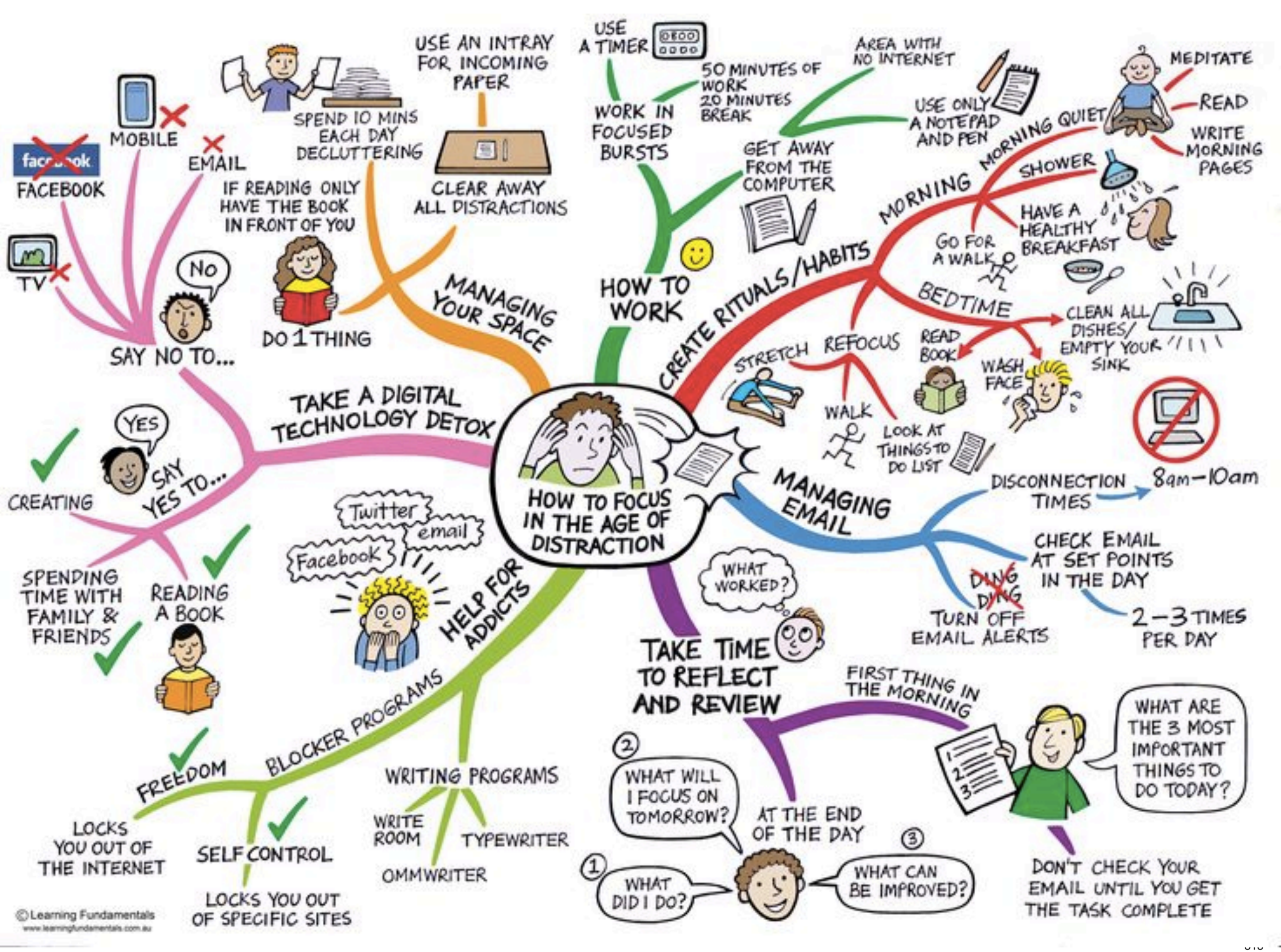
→

broader product definition

Agile Gurukul all rights reserved.

Reflections

Agile Gurukul all rights reserved.


HOW TO FOCUS IN THE AGE OF DISTRACTION
TAKE A DIGITAL TECHNOLOGY DETOX
SAY NO TO...
No
MOBILE
EMAIL
FACEBOOK
TV
SAY YES TO...
Yes
CREATING
SPENDING TIME WITH FAMILY & FRIENDS
READING A BOOK
MANAGING YOUR SPACE
SPEND 10 MINS EACH DAY DECLUTTERING
USE AN INTRAY FOR INCOMING PAPER
CLEAR AWAY ALL DISTRACTIONS
IF READING ONLY HAVE THE BOOK IN FRONT OF YOU
DO 1 THING
HOW TO WORK
USE A TIMER
WORK IN FOCUSED BURSTS
50 MINUTES OF WORK 20 MINUTES BREAK
GET AWAY FROM THE COMPUTER
AREA WITH NO INTERNET
USE ONLY A NOTEPAD AND PEN
CREATE RITUALS/HABITS
MORNING
MORNING QUIET
MEDITATE
READ
WRITE MORNING PAGES
SHOWER
HAVE A HEALTHY BREAKFAST
GO FOR A WALK
BEDTIME
CLEAN ALL DISHES/ EMPTY YOUR SINK
READ BOOK
WASH FACE
STRETCH
REFOCUS
WALK
LOOK AT THINGS TO DO LIST
MANAGING EMAIL
DISCONNECTION TIMES
8am–10am
CHECK EMAIL AT SET POINTS IN THE DAY
2–3 TIMES PER DAY
TURN OFF EMAIL ALERTS
DING
DING
HELP FOR ADDICTS
Twitter
email
Facebook
BLOCKER PROGRAMS
FREEDOM
LOCKS YOU OUT OF THE INTERNET
SELF CONTROL
LOCKS YOU OUT OF SPECIFIC SITES
WRITING PROGRAMS
WRITE ROOM
TYPEWRITER
OMMWRITER
TAKE TIME TO REFLECT AND REVIEW
WHAT WORKED?
AT THE END OF THE DAY
1
WHAT DID I DO?
2
WHAT WILL I FOCUS ON TOMORROW?
3
WHAT CAN BE IMPROVED?
FIRST THING IN THE MORNING
WHAT ARE THE 3 MOST IMPORTANT THINGS TO DO TODAY?
DON'T CHECK YOUR EMAIL UNTIL YOU GET THE TASK COMPLETE
© Learning Fundamentals
www.learningfundamentals.com.au

team: standing: round robin

> *each person*: most **noteworthy** or **interesting** idea, since last?

Agile Gurukul all rights reserved.

individual

> *each person*: write it on a post-it and *post it*

Agile Gurukul all rights reserved.

Day 3

Agile Gurukul all rights reserved.

Agenda

- ~~Introduction~~
- ~~Principles~~
- ~~Overview~~
- ~~Organised by Customer Value~~

- ~~Product~~
- ~~Definition of Done~~
- ~~Product Owner and Product Backlog~~

- Sprint Planning, Review Retrospective
- Integration and Coordination
- Management and Orientation
- Adoption
- Scrum Master

Agile Gurukul all rights reserved.

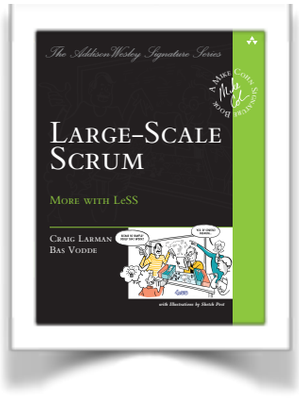
Organizational Structure

Where are We?

1. Opening Topics
2. System Optimisation, not Local Optimisation
3. **Organizational Structure**
4. LeSS Overview

Organise by Customer Value: Feature Teams

what are we about to learn?



Guide: Getting Started

0. Educate Everyone

1. Define product

2. Define ‘done’

3. Have appropriately-structured teams

4. Only the Product Owner provides work for teams

5. Keep “managers” away from teams

defining feature teams…

Harvard Business Review

LEADING TEAMS

The New New Product Development Game

by Hirotaka Takeuchi and Ikujiro Nonaka

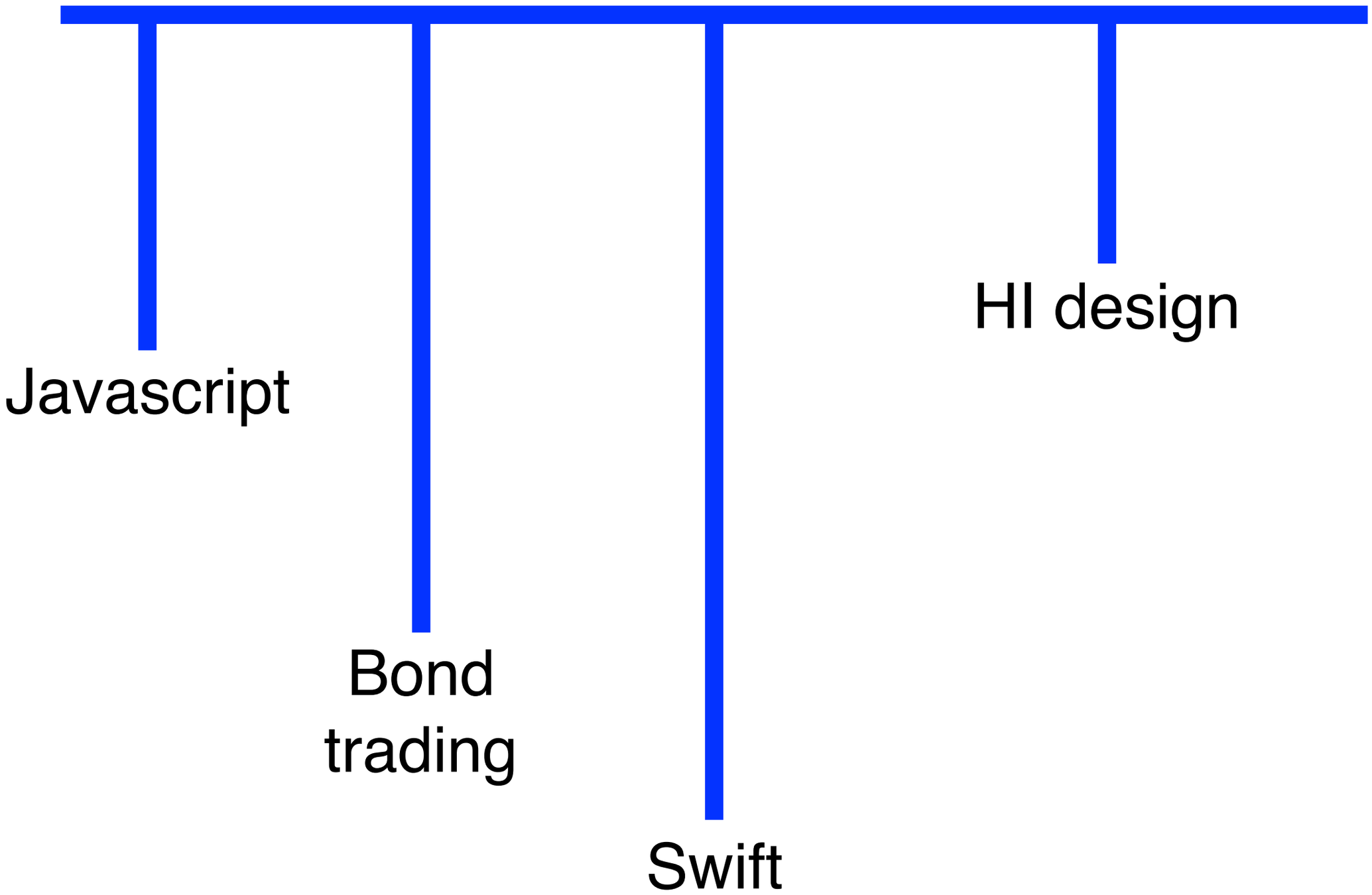
FROM THE JANUARY 1986 ISSUE

Moving the scrum downfield

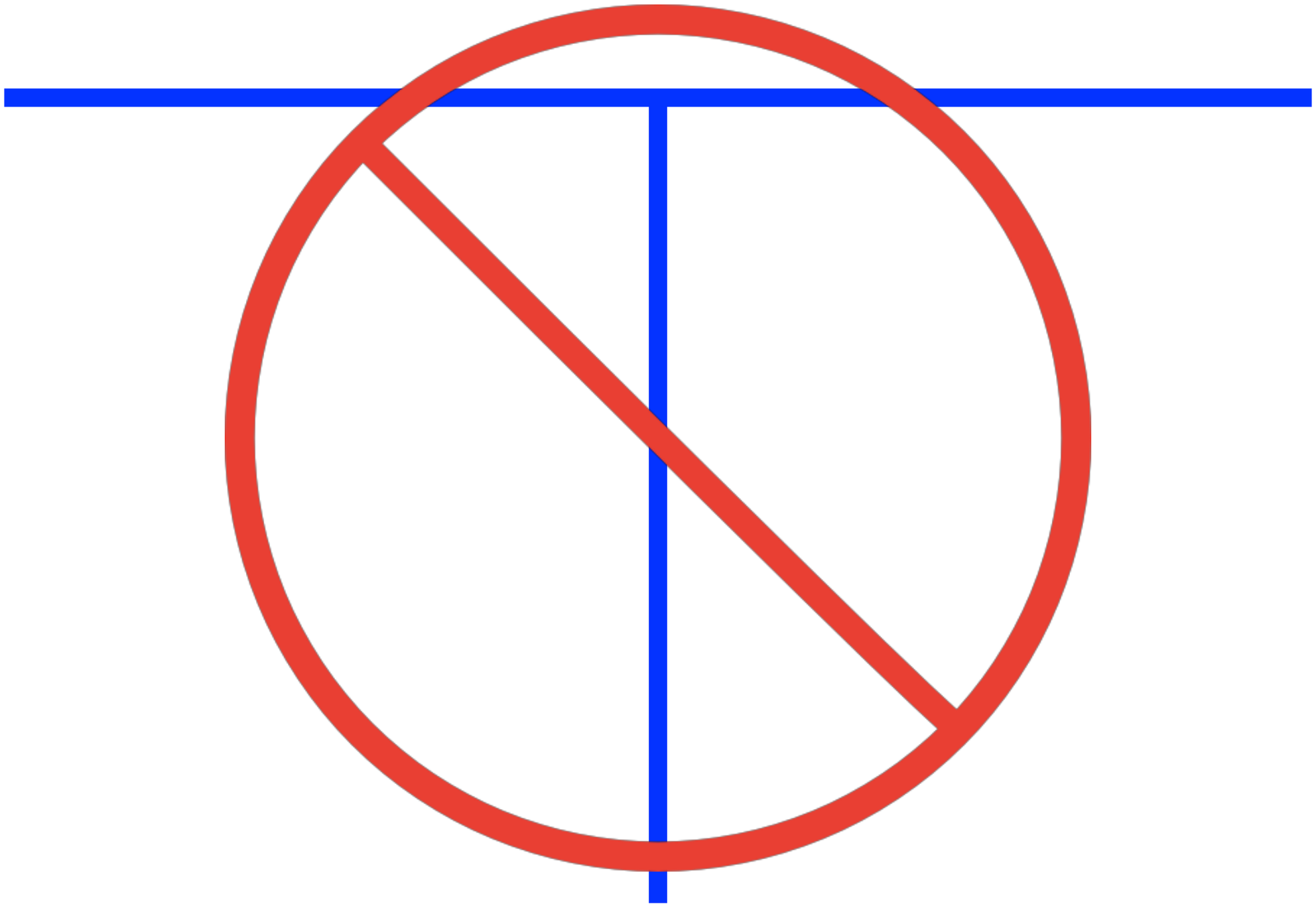
From interviews with organization members from the CEO to young engineers, we learned that leading companies show six characteristics in managing their new product development processes:

1. Built-in instability
2. Self-organizing project teams
3. Overlapping development phases
4. "Multilearning"
5. Subtle control
6. Organizational transfer of learning

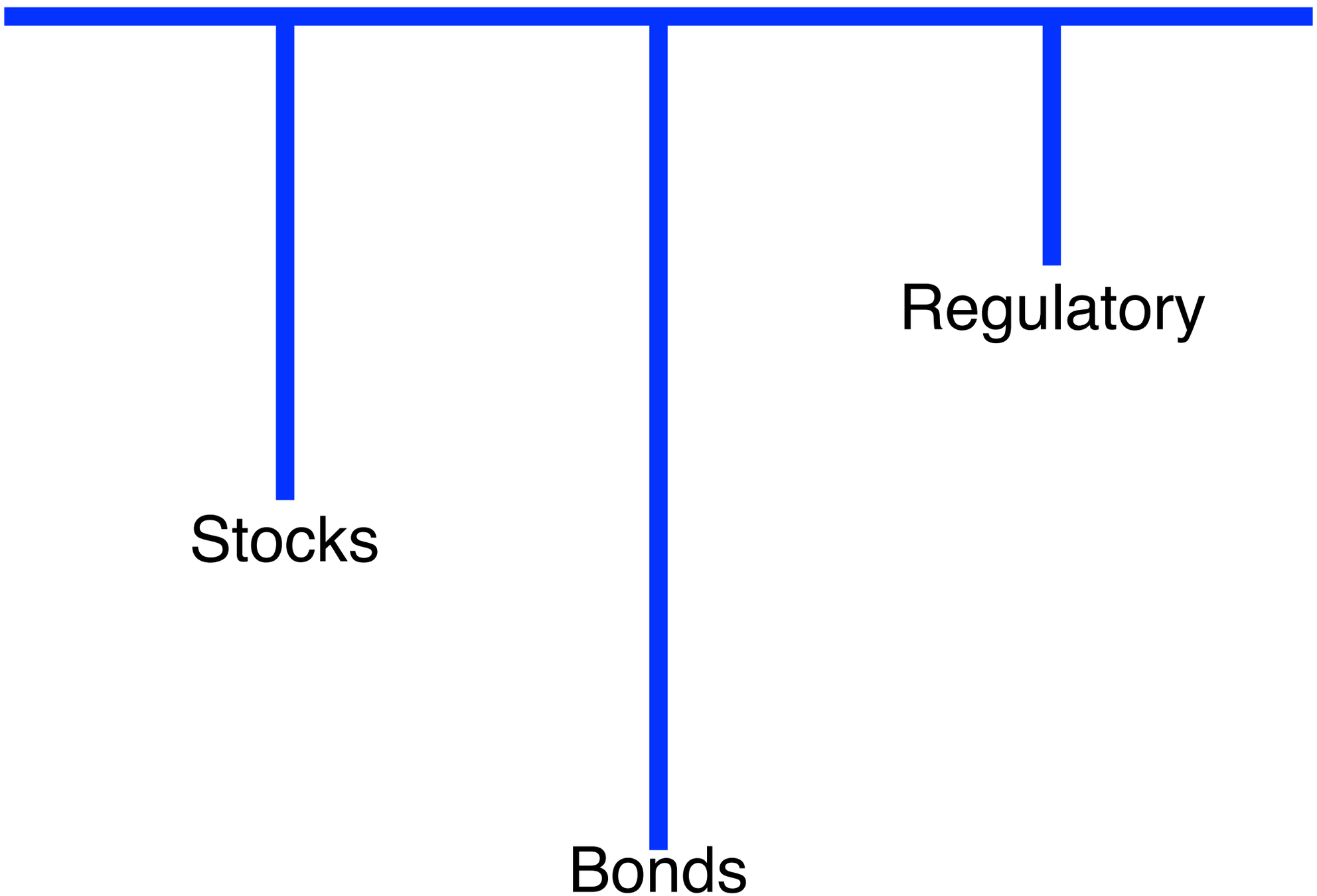
"**Multilearning**" Team Member



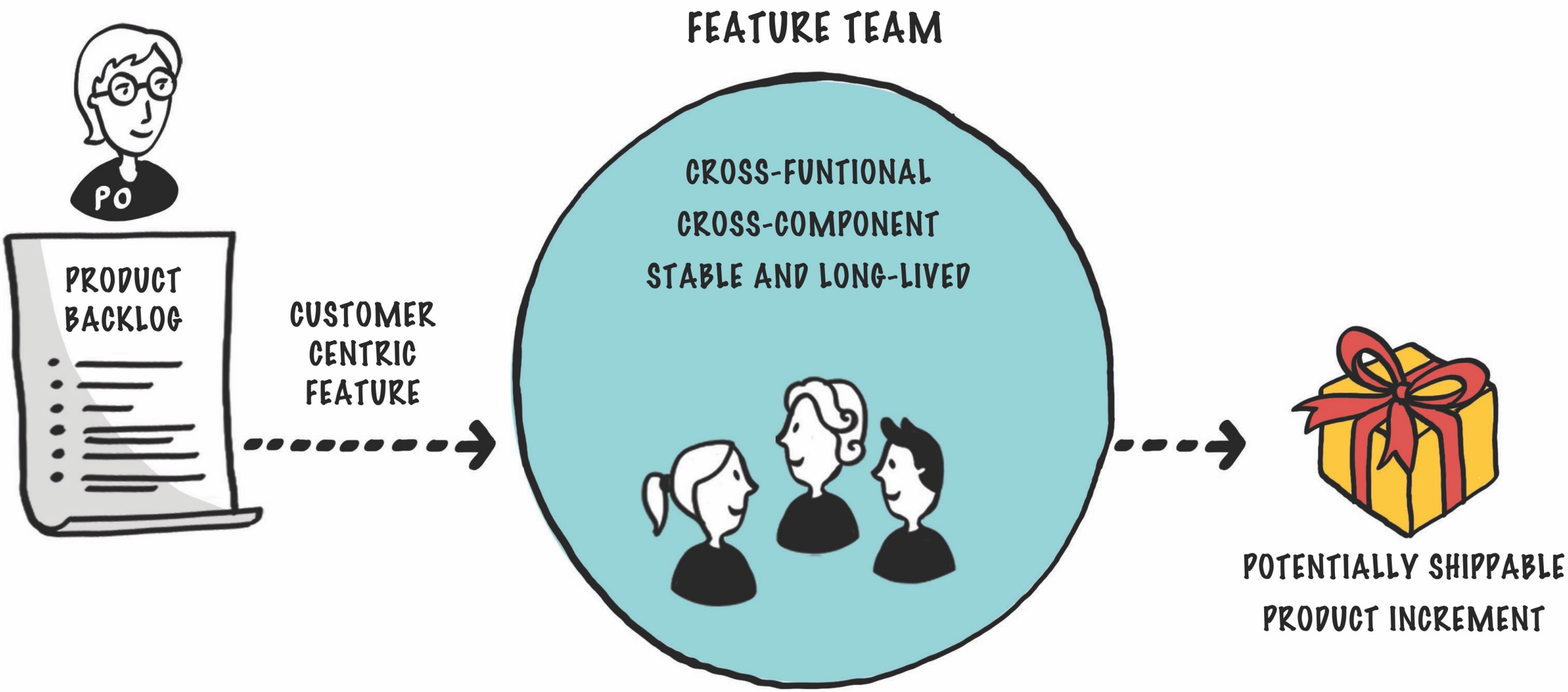
… not merely “T shaped”; only 1 deep skill



… and "Multilearning" **TEAM**



“Multilearning” **TEAM**



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

Harvard Business Review

LEADING TEAMS

The New New Product Development Game

by Hirotaka Takeuchi and Ikujiro Nonaka

FROM THE JANUARY 1986 ISSUE

Moving the scrum downfield

From interviews with organization members from the CEO to young engineers, we learned that leading companies show six characteristics in managing their new product development processes:

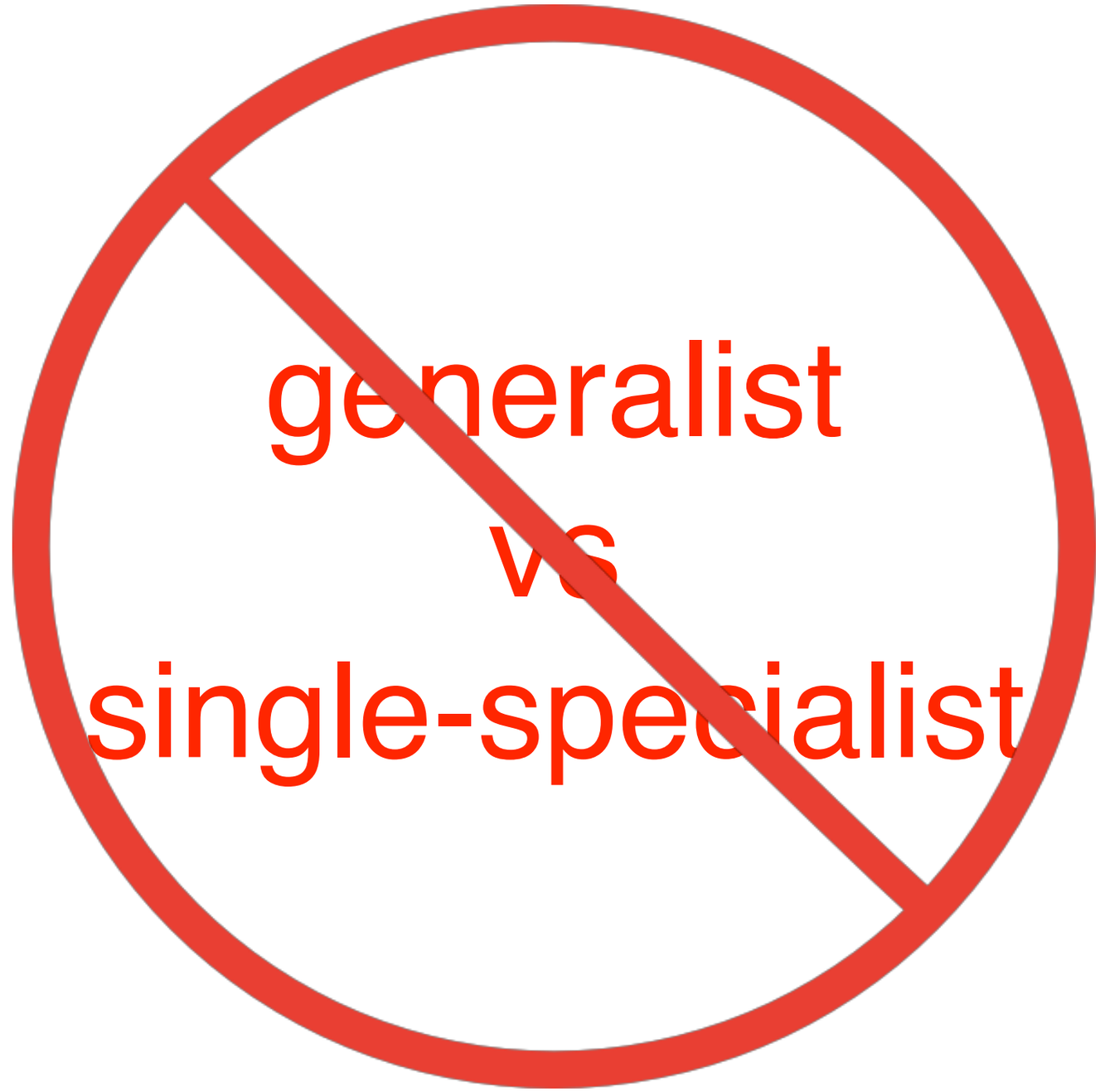
1. Built-in instability
2. Self-organizing project teams
3. Overlapping development phases
4. "Multilearning"
5. Subtle control
6. Organizational transfer of learning

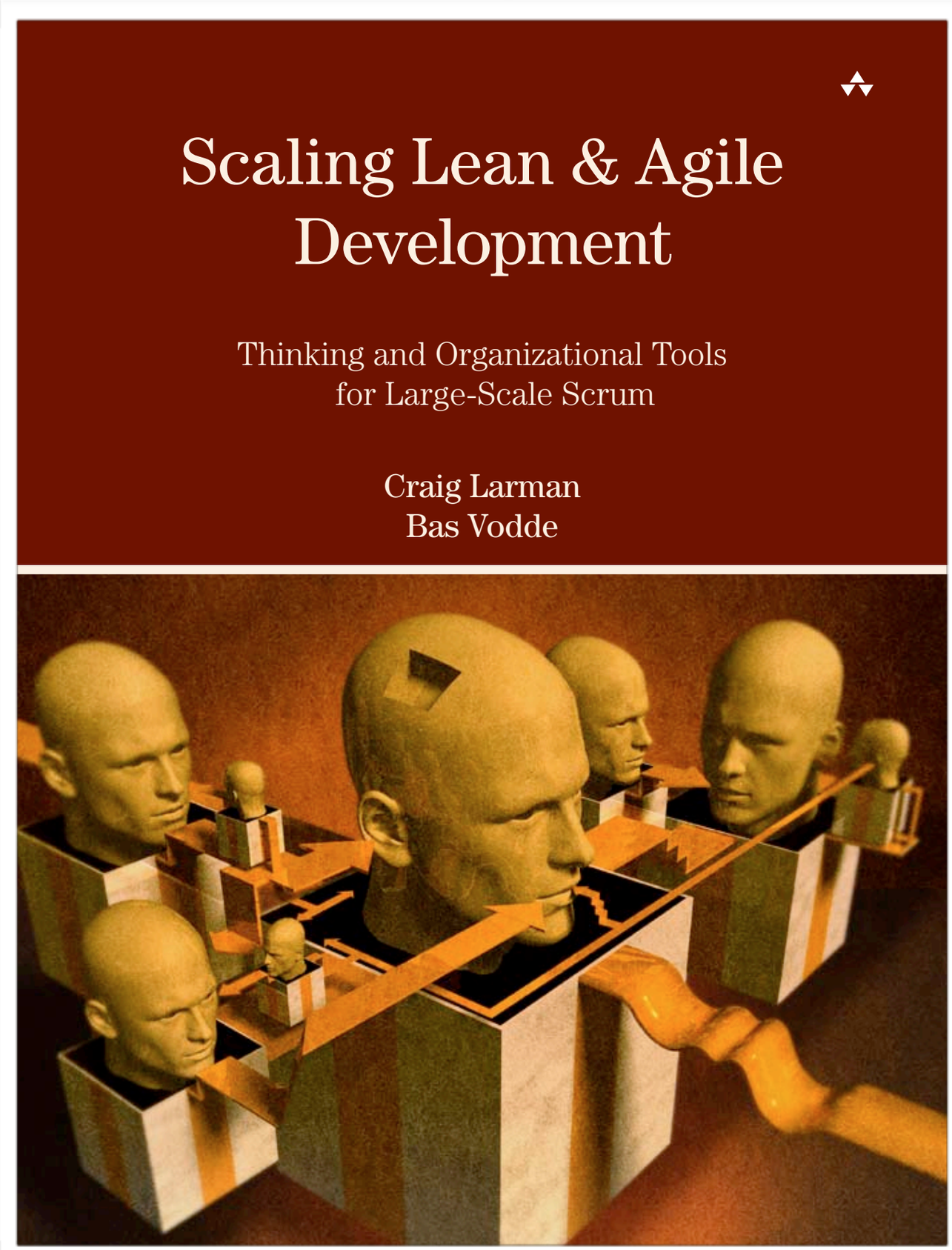
Multilearning & Multiskilled

primary

secondary

tertiary





Thinking Tools

2. Systems Thinking
3. Lean Thinking
4. Queueing Theory
5. False Dichotomies
6. Be Agile

Organizational Tools

7. Feature Teams
8. Teams
9. Requirement Areas
10. Organization
11. Large-Scale Scrum

No ***Single***-Specialist Title

Scrum Guide:

"Scrum recognises no titles for Development Team members, regardless of the work being performed by the person"

BIG Idea

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

e.g. **Product Developer**

Not a Team of *Single*-Specialists

Scrum Guide:

"Scrum recognises no sub-teams in the Development Team, regardless of domains that need to be addressed like testing, architecture, operations, or business analysis"

Not a Team of *Single*-Specialists

a Team in Scrum?

"UX analyst, Java developer, tester, BA, ..."

Not a Component or Partial Solution Team

Scrum Guide:

"teams **have all competencies** needed to accomplish the work **without depending on others** not part of the team"

"… with **all the skills** as a team necessary to create a product Increment"

a Feature Team is **full stack**… works across **all** code/components in a "shared code" model

therefore…

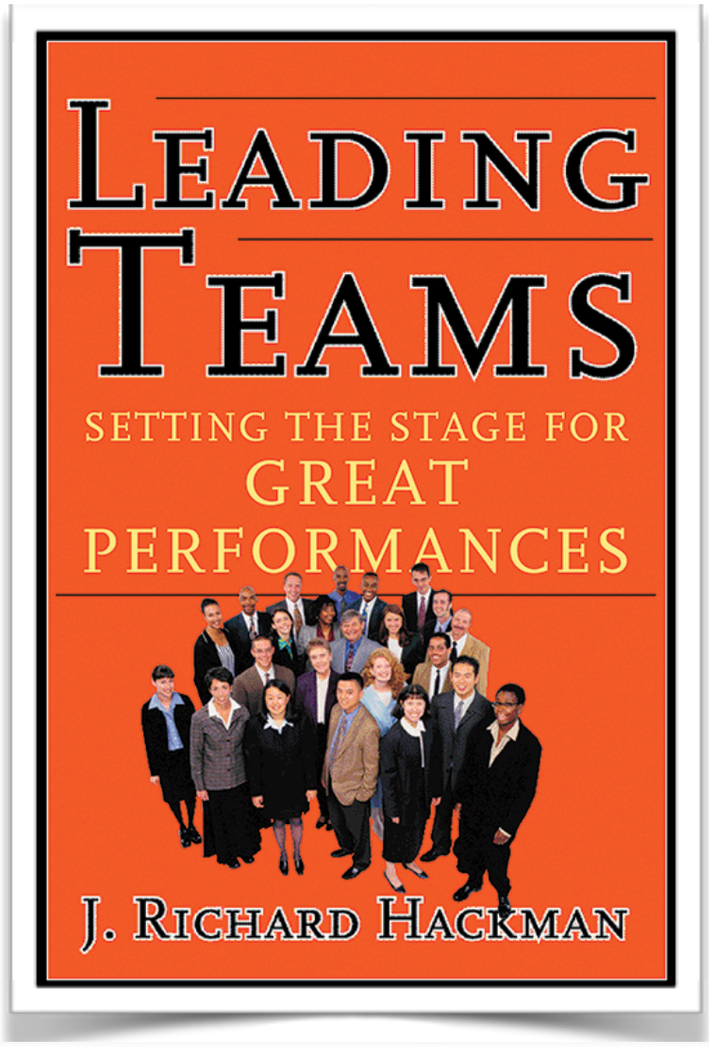
LeSS Rule(s)

The majority of the teams are customer-focused **feature teams**

LeSS Rule(s)

Structure the organisation using **real teams** as the organizational **building block**

"Real" Teams?



J. Richard Hackman

Edgar Pierce Professor of Social and Organizational Psychology

Department of Psychology, 33 Kirkland St., Harvard University, Cambridge, MA 02138

Contact

LeSS Rule(s)

Each *real* team is

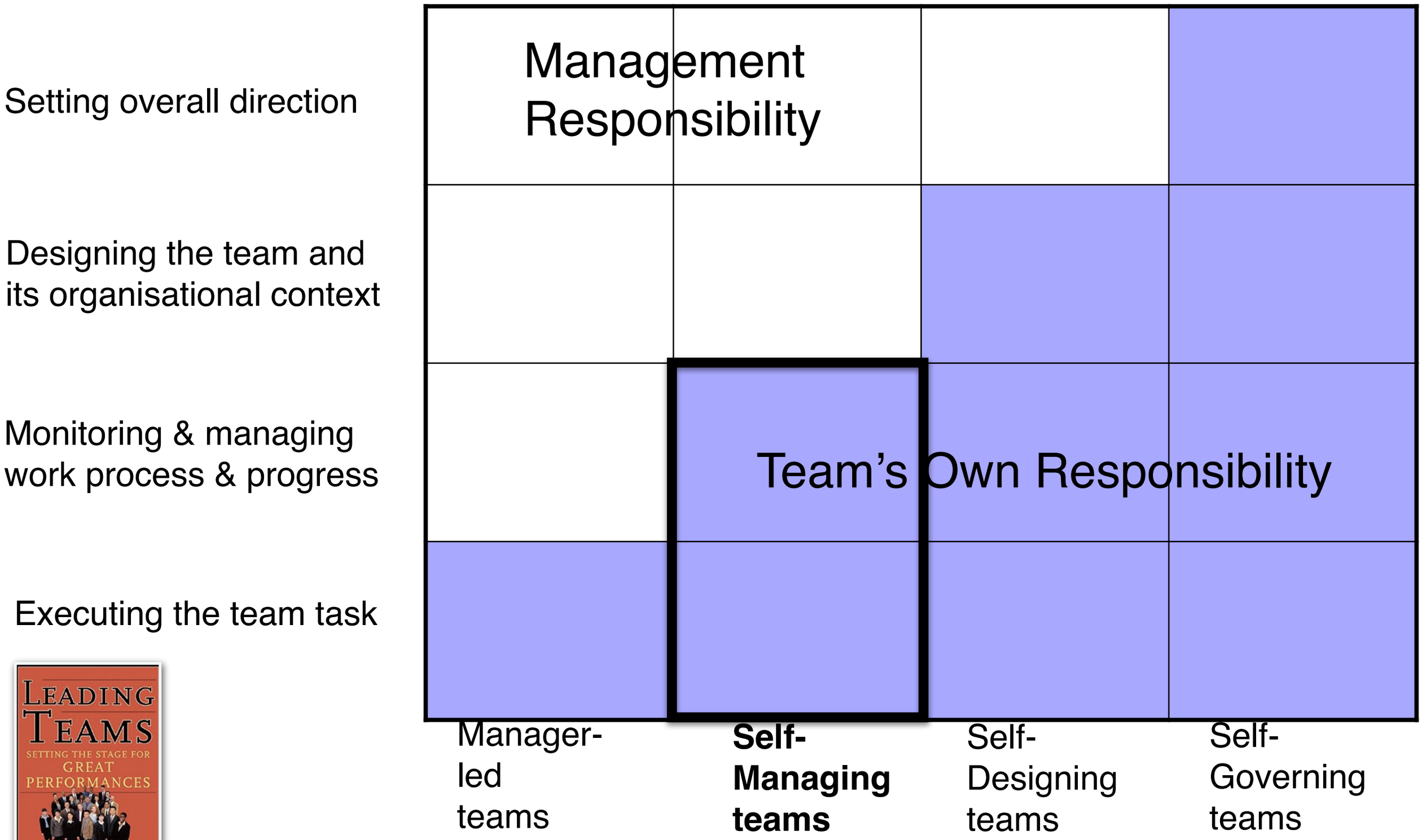
(1) **self-managing**

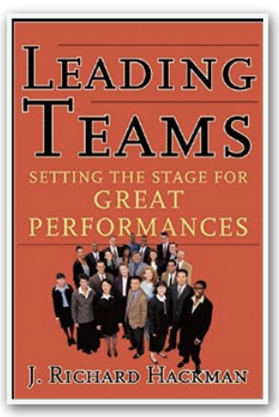
(2) cross-functional

(3) co-located

(4) long-lived

Self-Managing & Authority Matrix





team lead?

51% voting?

“self-organising?”

“Must Cover in CSM” - Ken’s Direction to CSTs

“**Self-managing teams** are extremely productive. When they work closely with the customer to derive the best solution to a need, they and the customer are even more productive.”

LeSS Rule(s)

Each *real* team is

(1) self-managing

(2) cross-functional

(3) co-located

(4) long-lived

Co-Located: Same TABLE!

LeSS Rule(s)

Each *real* team is

(1) self-managing

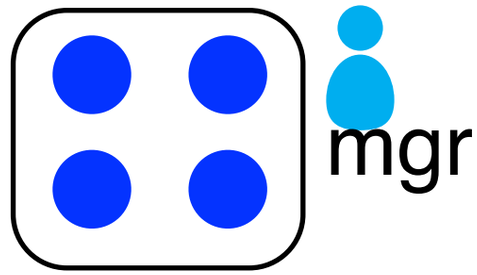
(2) cross-functional

(3) co-located

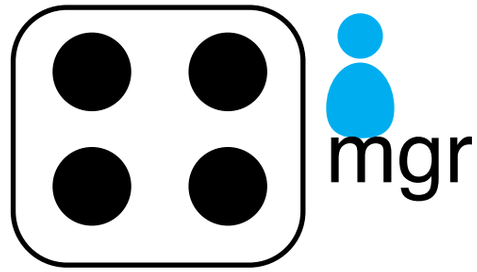
(4) **long-lived**

adopting feature teams…

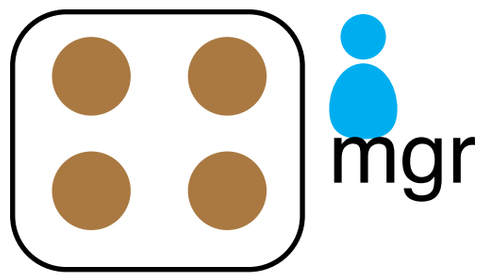
analysts and/or
UX/UI designers



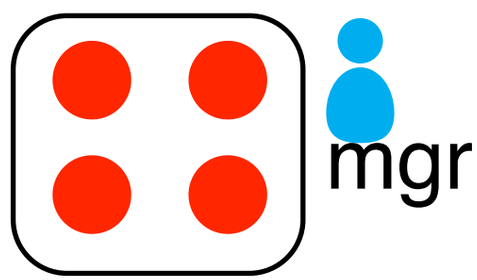
DBAs



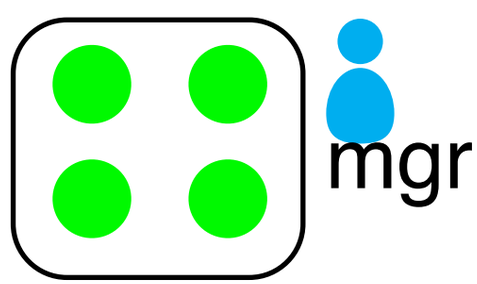
architects



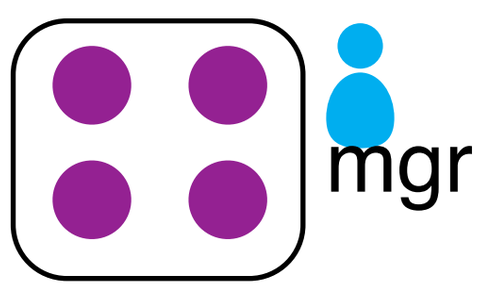
component-1
programmers



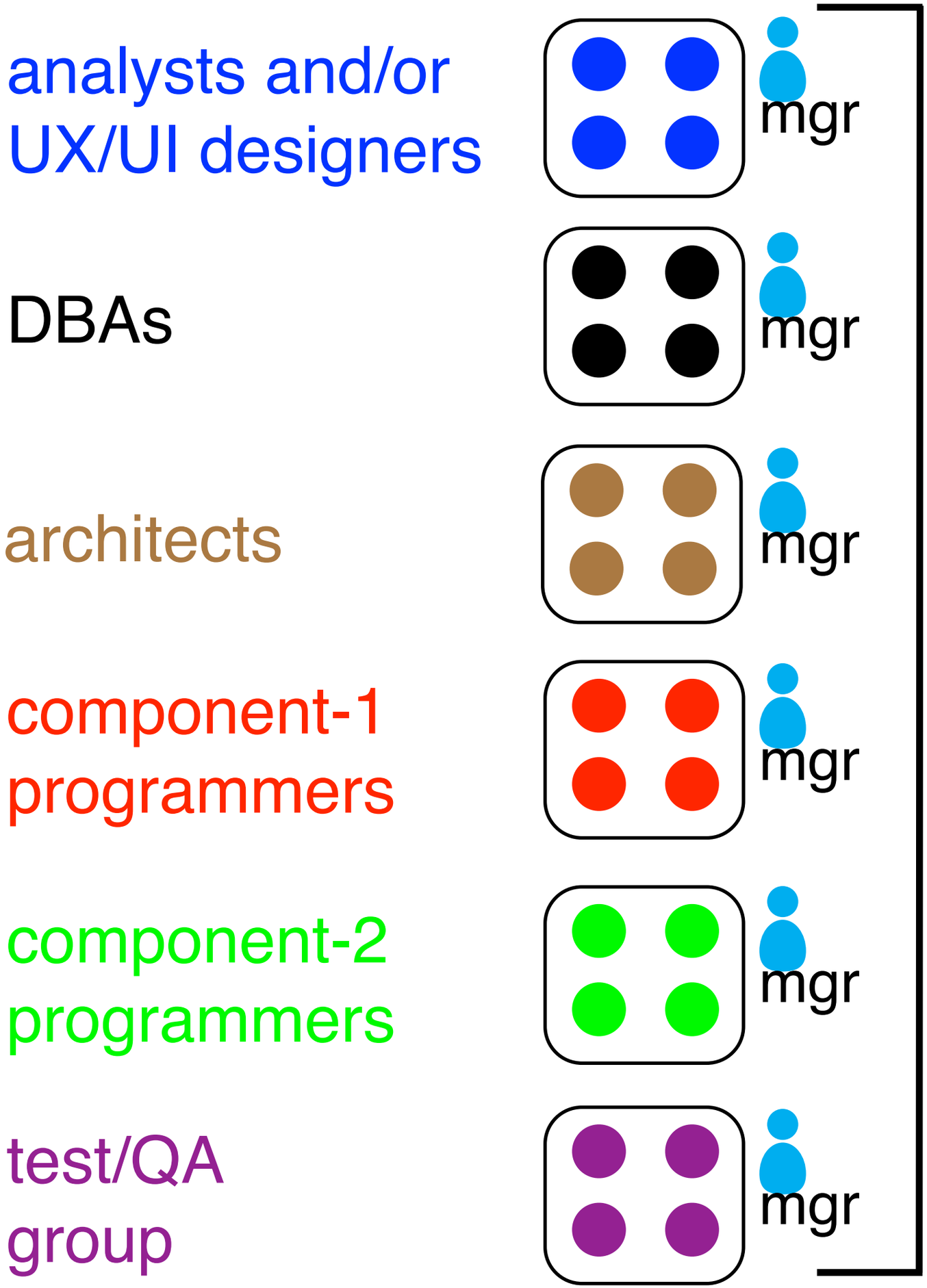
component-2
programmers



test/QA
group

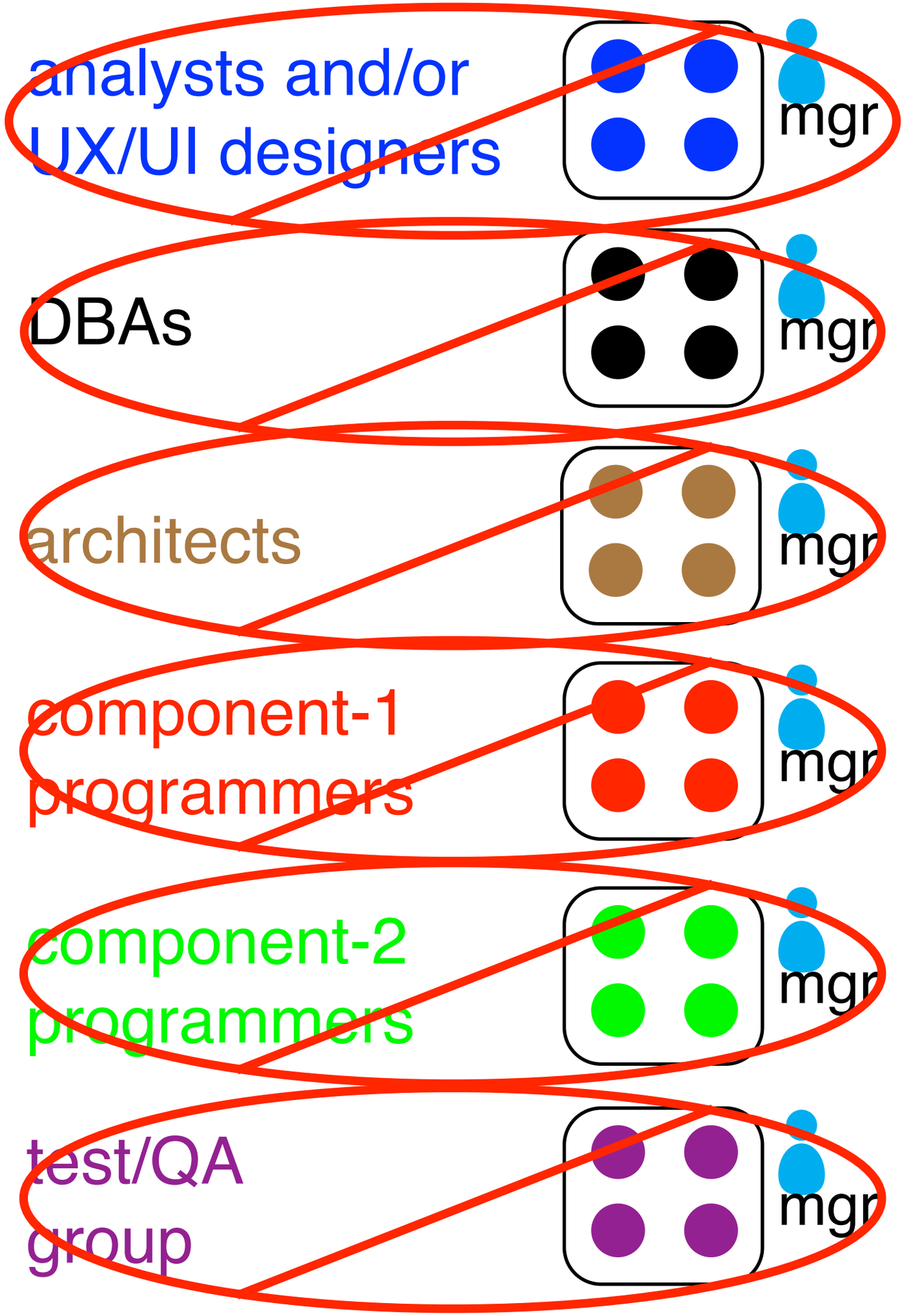


a likely traditional large-scale organisational structure before adopting Scrum



a cross-functional team in Scrum

spans all functions



single-specialist groups are dissolved

(and thus the functional & component manager roles are eliminated)

"It is difficult to get a man to understand something, when his salary depends upon his **not** understanding it!"

— Upton Sinclair

Guide: Job Safety, but not Role Safety

Job safety

& salary safety

work redesign &
job re-training

BIG Idea

in LeSS many traditional manager positions are *eliminated* (not "redefined") — project & program managers, functional managers, component team managers, PMO, etc — and any few remaining manager positions *have* ***nothing to do with teams or development or tracking*** *(etc)*, but with improving the system (e.g., closing sites, eliminating traditional budgeting, eliminating outsourcing, eliminating tracking/reporting).

defining the teams…?

“let me show you the org chart of my decision for the new feature teams”

Self-Designing Teams Workshop

How to Form Teams in Large-Scale Scrum? A Story of Self-Designing Teams

5 April 2013

Craig Larman

Ahmad Fahmy

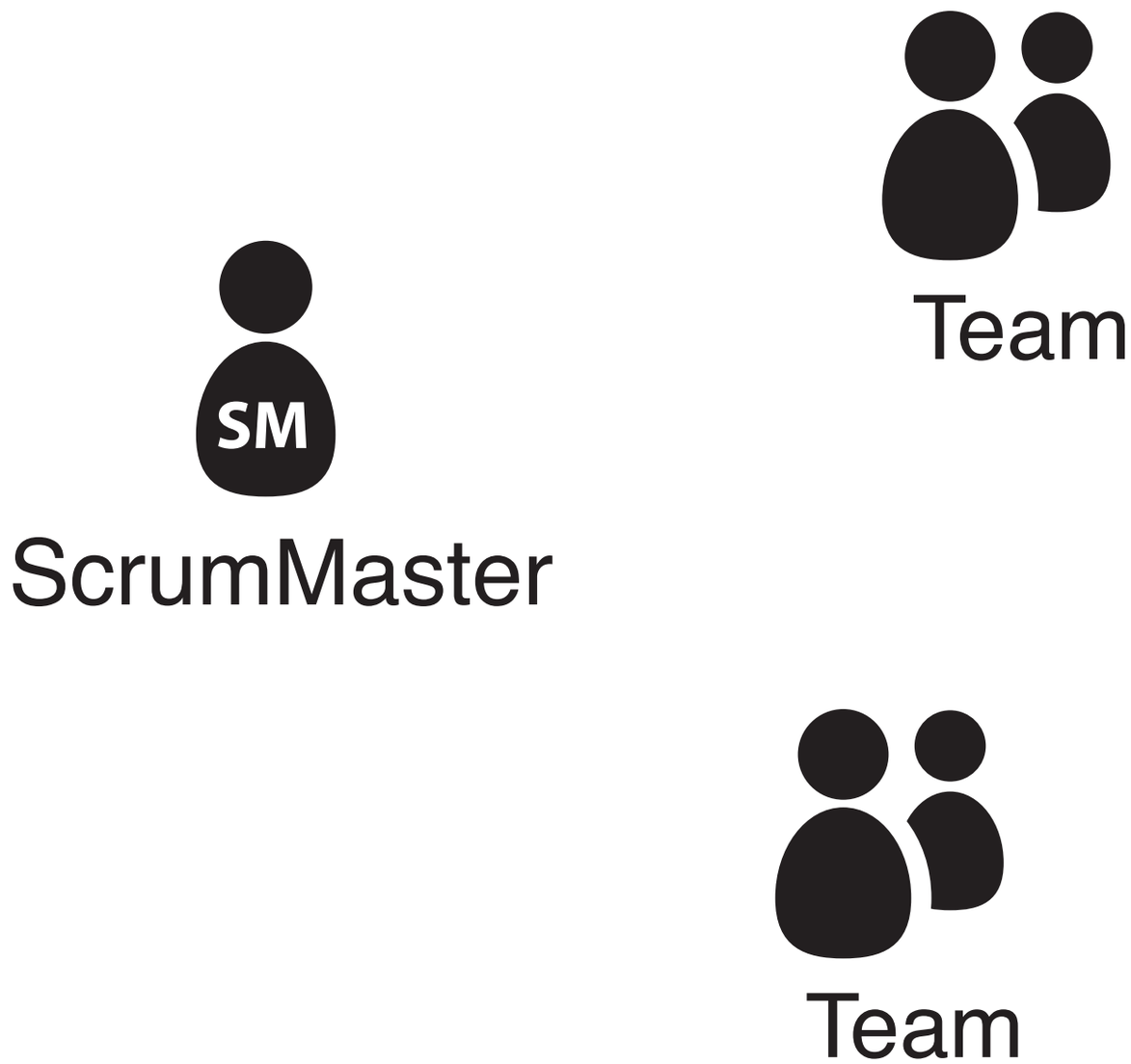
How long does it take an organization to reorganize in order to adopt Scrum? Three hours. Really.

Self-Designing Teams Workshop

Scrum Masters…?

Connecting to Scrum Masters?

Rule: 1 Scrum Master can serve 1-3 teams



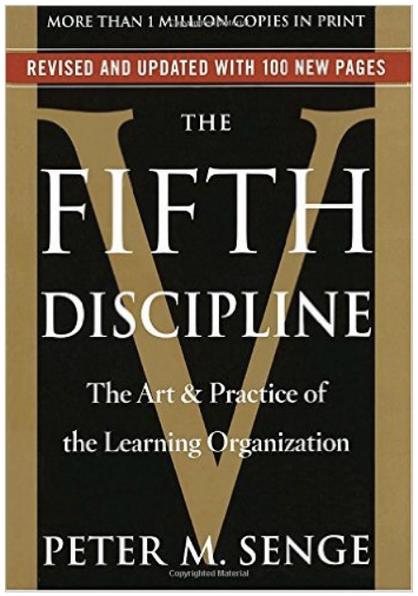
LeSS Rule(s)

Scrum Master is a dedicated full-time role

Scrum Master is
another name for…

project manager
agile project manager
team manager
coordinator
tracker/status provider
Jira updater
note-taker

a “bad” ex-project manager *may* make a good Scrum Master ;)



source of future
senior (true) *leaders*?

your **Scrum Masters,**
and **Product Owner**

(optional)
connecting to
line managers

span if required?
80-100 to 1

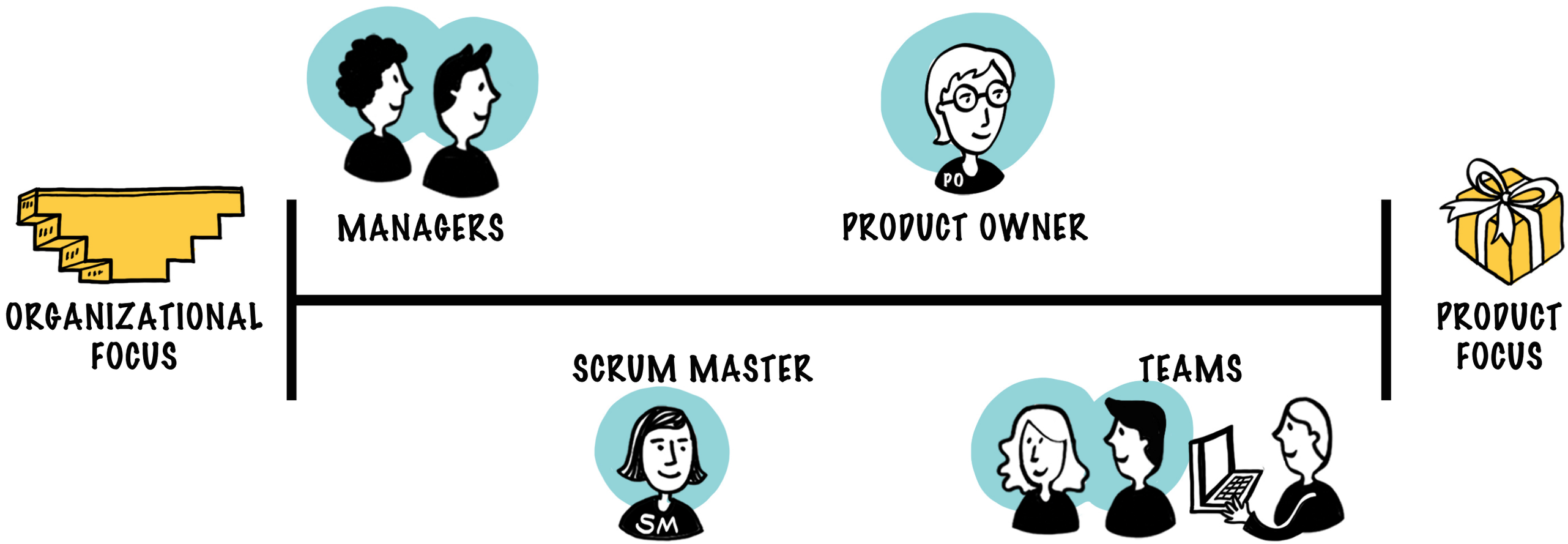
Connecting to Line Managers?

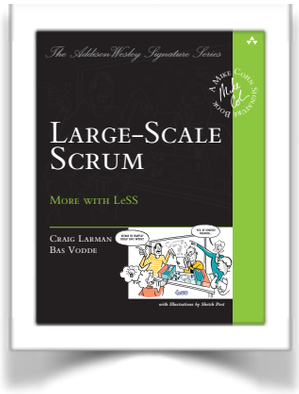
coach & group

> new roles for

- ex-functional-team managers
- ex-component-team managers
- ex-project & program managers
- ex-team-leads
- ex-team-managers
- ex-team-level “Product Owners”
- ex-analysts/specifiers/BAs/requirement engineers
- ex-UX analysts
- ex-UI designers
- ex-architects, ex-system engineers
- ex-DBAs

Focus of Different Roles





Guide: Getting Started

0. Educate Everyone

1. Define product

2. Define ‘done’

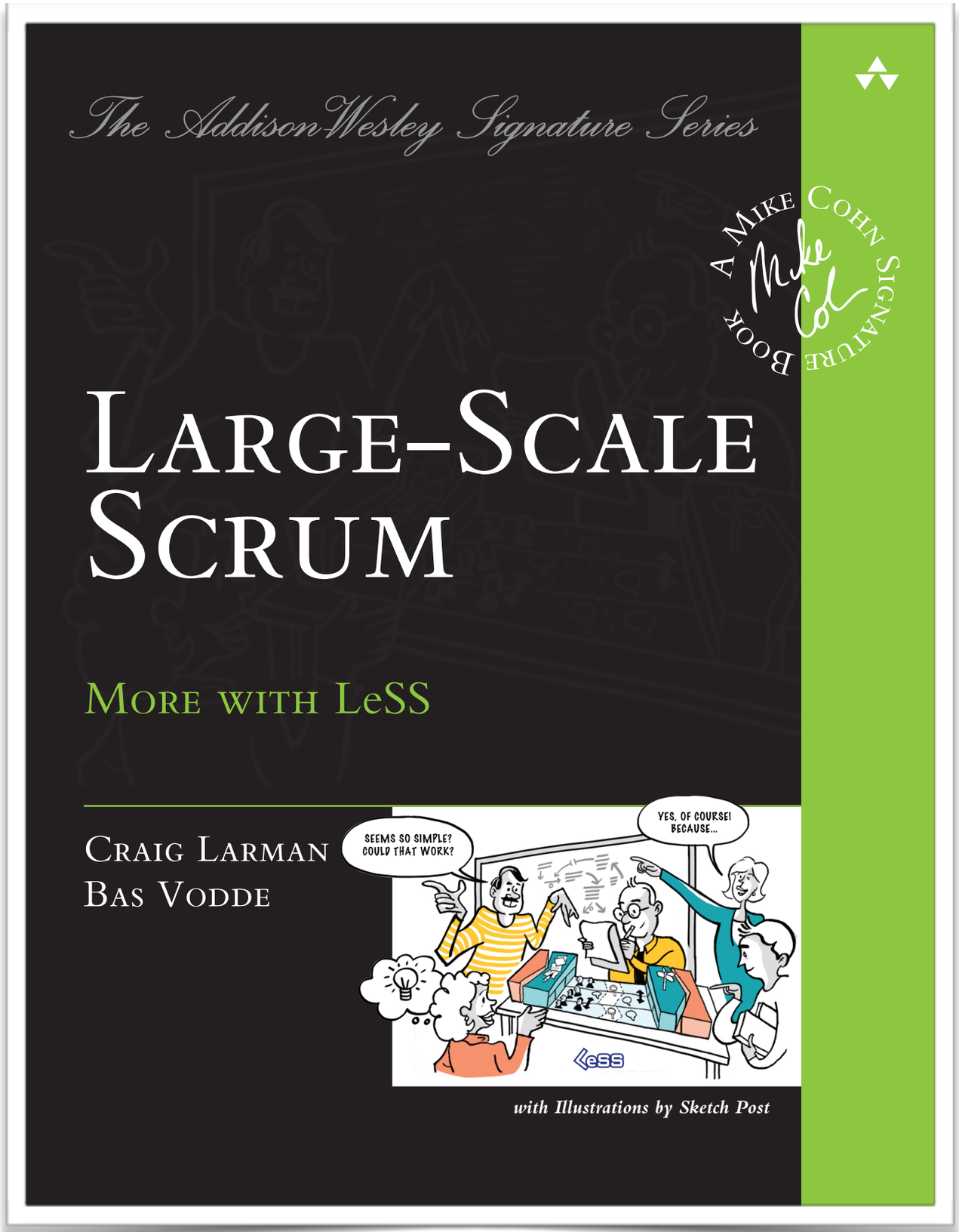
3. Have appropriately-structured teams

4. Only the Product Owner provides work for teams

5. Keep “managers” away from teams

most of the points in this section are from regular one-team Scrum (not LeSS)

if they are surprising, and you are in a "Scrum" group, that's interesting ;)



1 More with LeSS 1
2 LeSS 5

LeSS Structure

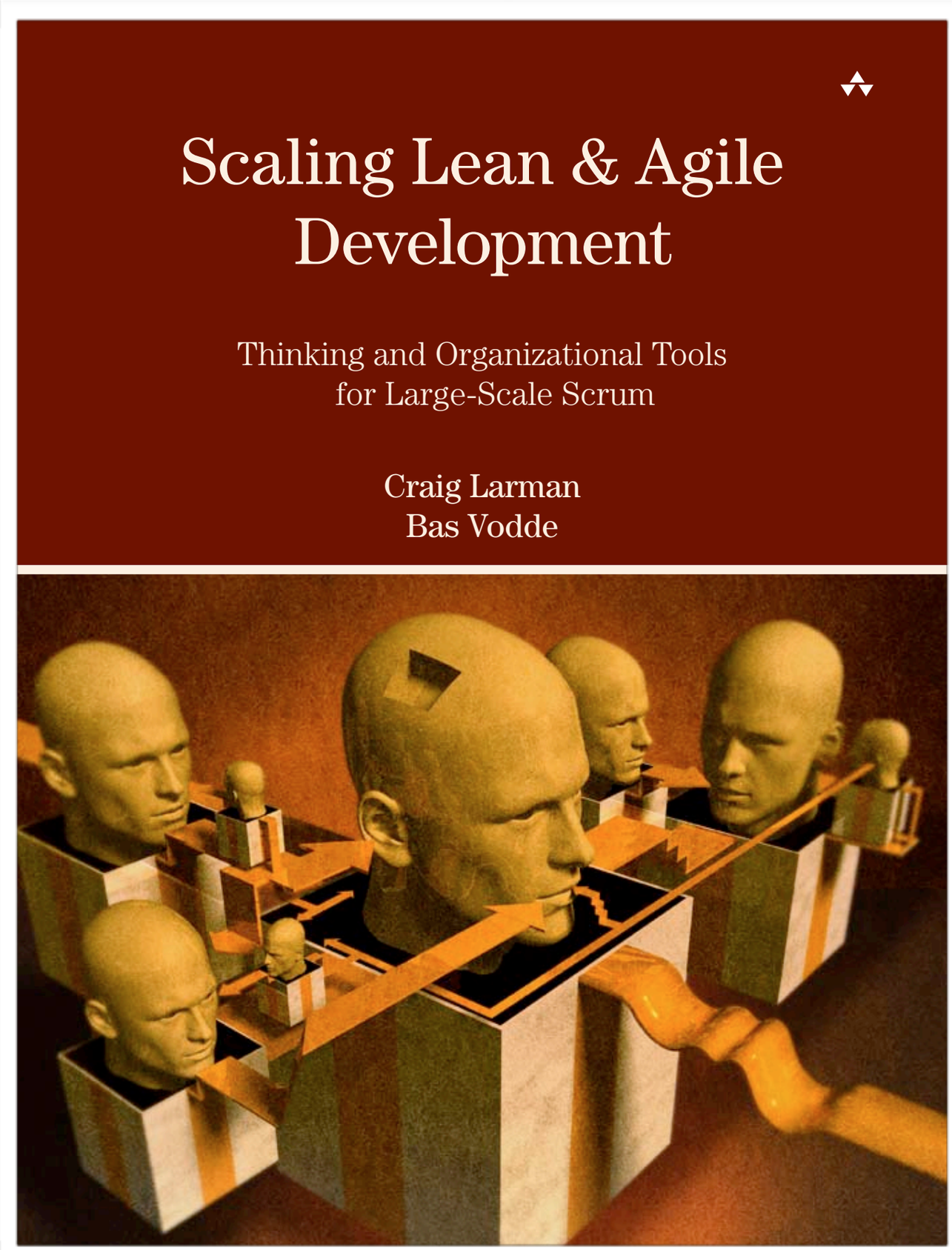
3 Adoption 41
4 Organize by Customer Value 67
5 Management 111
6 ScrumMasters 133

LeSS Product

7 Product 155
8 Product Owner 171
9 Product Backlog 197
10 Definition of Done 231

LeSS Sprint

11 Product Backlog Refinement 249
12 Sprint Planning 275
13 Coordination and Integration 285
14 Review & Retrospective 313



Thinking Tools

2. Systems Thinking
3. Lean Thinking
4. Queueing Theory
5. False Dichotomies
6. Be Agile

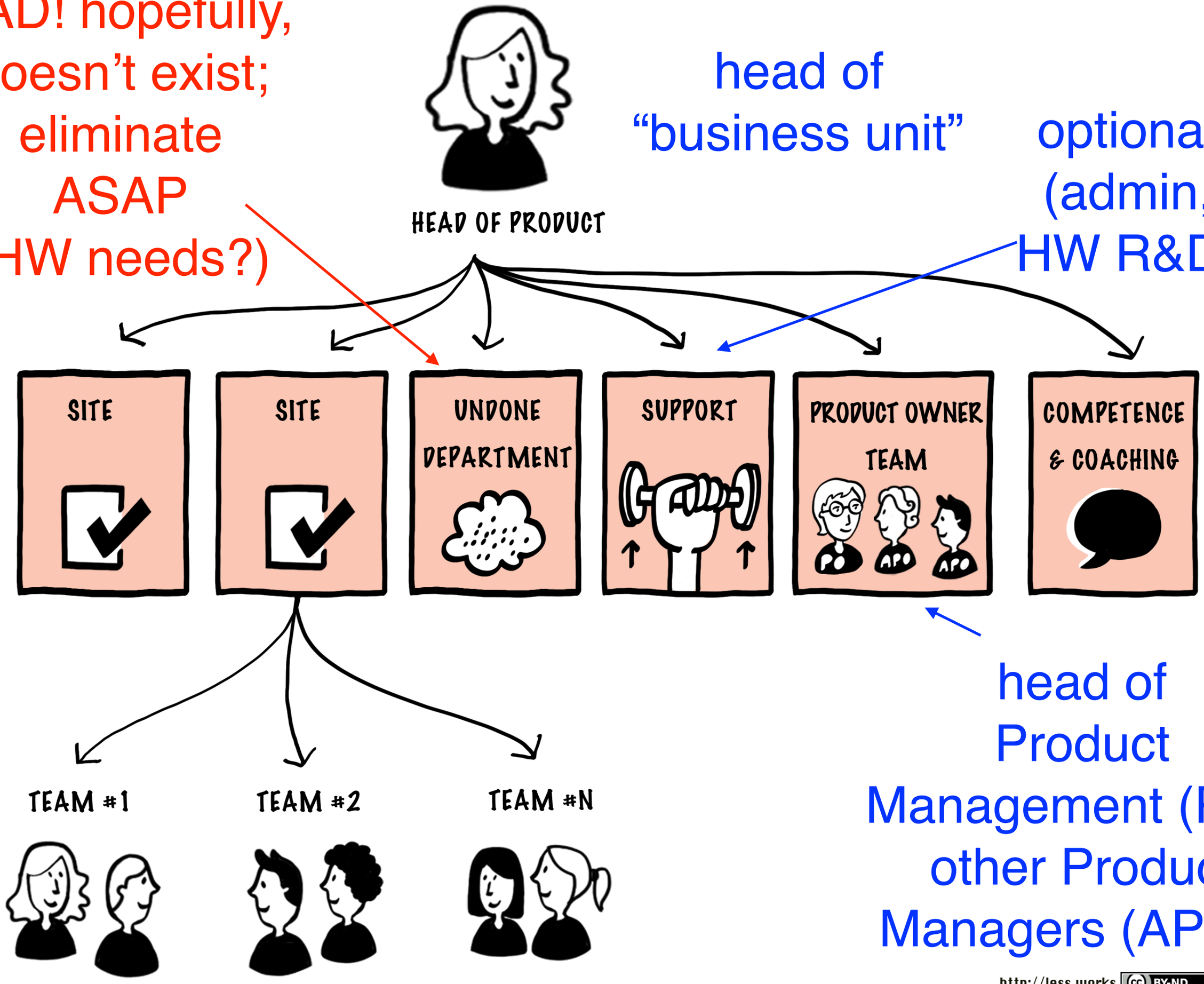
Organizational Tools

7. Feature Teams
8. Teams
9. Requirement Areas
10. Organization
11. Large-Scale Scrum

team work

Organizational Structure for LeSS Huge

BAD! hopefully, doesn't exist; eliminate ASAP (HW needs?)



head of "business unit"

optional (admin, HW R&D)

head of Product Management (PO), other Product Managers (APOs)

http://less.works CC BY-ND

why are Requirement Areas *not* associated with an “official” org structure group

Larman's Laws of Organizational Behavior

why so much?...

Lean-but

Scrum-but

Kanban-but

DevOps-but

AnyChangeIdea-but

one reason…

CIO FROM IDG

INSIDER Sign In | Register

FEATURE

Cargo Cult Methodology: How Agile Can Go Terribly, Terribly Wrong

another…

"It is difficult to get a man to understand something, when his salary depends upon his **not** understanding it!"

— Upton Sinclair

Larman's 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", "religious", 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, reinforcing (2) and (3).**

5. **Culture follows structure (or behavior/mindset follows system)**

larmanslaws.org

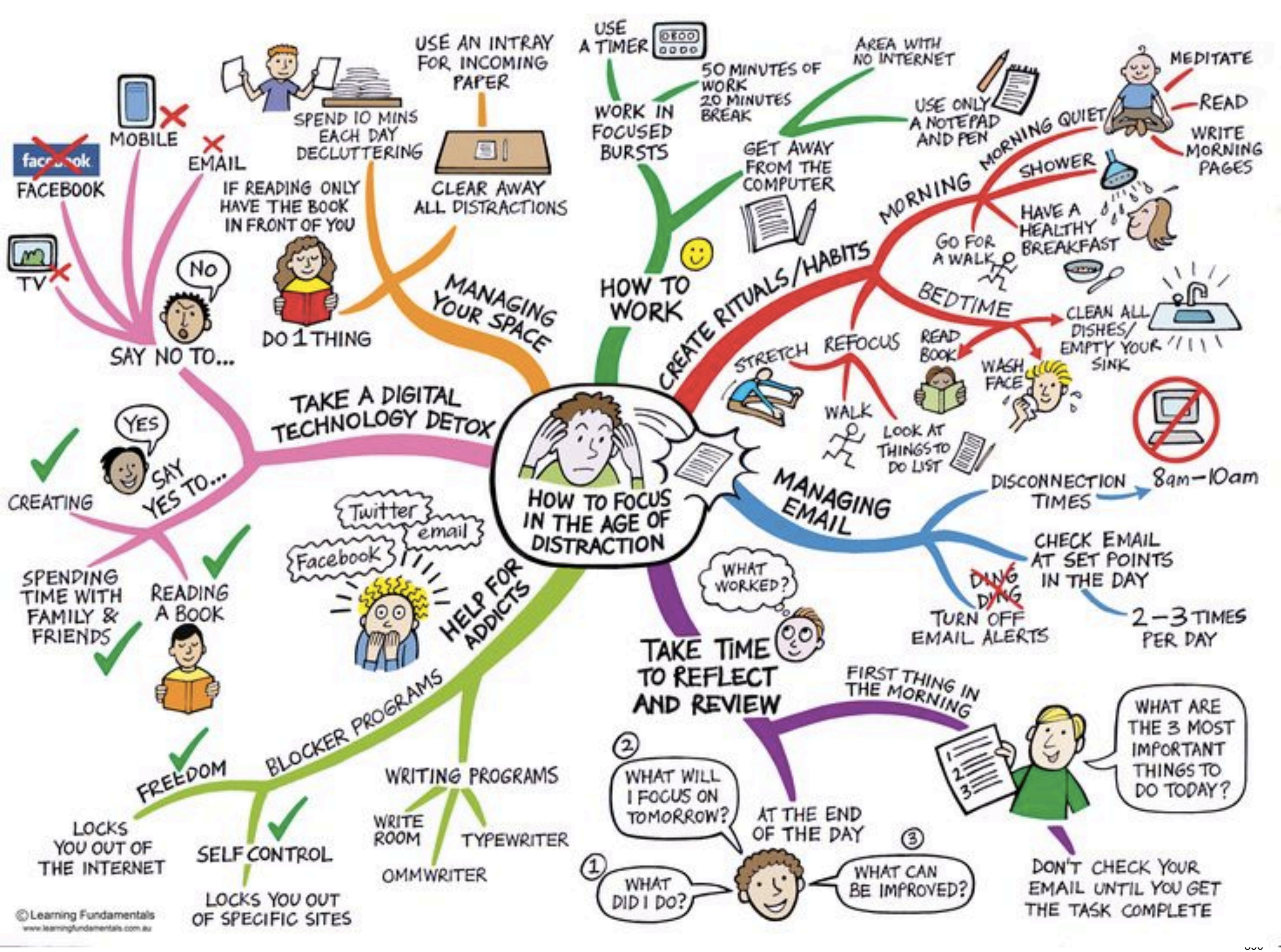
;)

Reflections

team: standing: round robin

> *each person*: most **noteworthy** or **interesting** idea, since last?

> *each person*: then, write it on a post-it and *post it*


HOW TO FOCUS IN THE AGE OF DISTRACTION
TAKE A DIGITAL TECHNOLOGY DETOX
SAY NO TO...
No
MOBILE
EMAIL
FACEBOOK
TV
SPENDING TIME WITH FAMILY & FRIENDS
CREATING
SAY YES TO...
Yes
READING A BOOK
MANAGING YOUR SPACE
SPEND 10 MINS EACH DAY DECLUTTERING
USE AN INTRAY FOR INCOMING PAPER
CLEAR AWAY ALL DISTRACTIONS
IF READING ONLY HAVE THE BOOK IN FRONT OF YOU
DO 1 THING
HOW TO WORK
USE A TIMER
WORK IN FOCUSED BURSTS
50 MINUTES OF WORK 20 MINUTES BREAK
GET AWAY FROM THE COMPUTER
AREA WITH NO INTERNET
USE ONLY A NOTEPAD AND PEN
CREATE RITUALS/HABITS
MORNING
MORNING QUIET
MEDITATE
READ
WRITE MORNING PAGES
SHOWER
HAVE A HEALTHY BREAKFAST
GO FOR A WALK
BEDTIME
CLEAN ALL DISHES/ EMPTY YOUR SINK
READ BOOK
WASH FACE
STRETCH
REFOCUS
WALK
LOOK AT THINGS TO DO LIST
MANAGING EMAIL
DISCONNECTION TIMES
8am–10am
CHECK EMAIL AT SET POINTS IN THE DAY
2–3 TIMES PER DAY
TURN OFF EMAIL ALERTS
DING
DING
HELP FOR ADDICTS
Twitter
email
Facebook
BLOCKER PROGRAMS
FREEDOM
LOCKS YOU OUT OF THE INTERNET
SELF CONTROL
LOCKS YOU OUT OF SPECIFIC SITES
WRITING PROGRAMS
WRITE ROOM
TYPEWRITER
OMMWRITER
TAKE TIME TO REFLECT AND REVIEW
WHAT WORKED?
FIRST THING IN THE MORNING
WHAT ARE THE 3 MOST IMPORTANT THINGS TO DO TODAY?
DON'T CHECK YOUR EMAIL UNTIL YOU GET THE TASK COMPLETE
AT THE END OF THE DAY
1
WHAT DID I DO?
2
WHAT WILL I FOCUS ON TOMORROW?
3
WHAT CAN BE IMPROVED?
© Learning Fundamentals
www.learningfundamentals.com.au

LeSS

Introduction ;)

Where are We?

1. Opening Topics
2. System Optimisation, not Local Optimisation
3. Organizational Structure
4. **LeSS Overview**

now that we've discovered LeSS for ourselves via "why"…

summarize **what**…

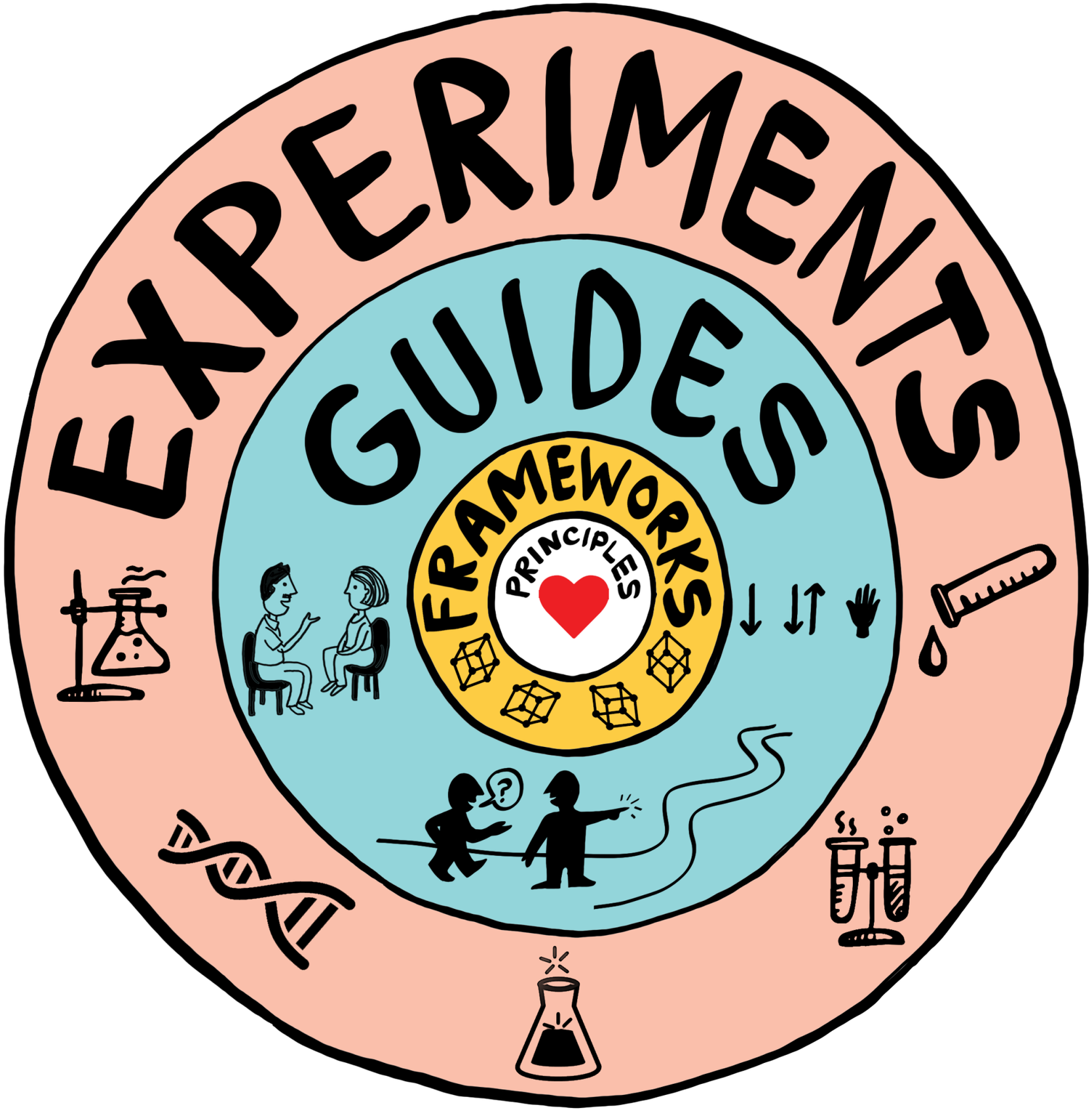
LeSS

Overview

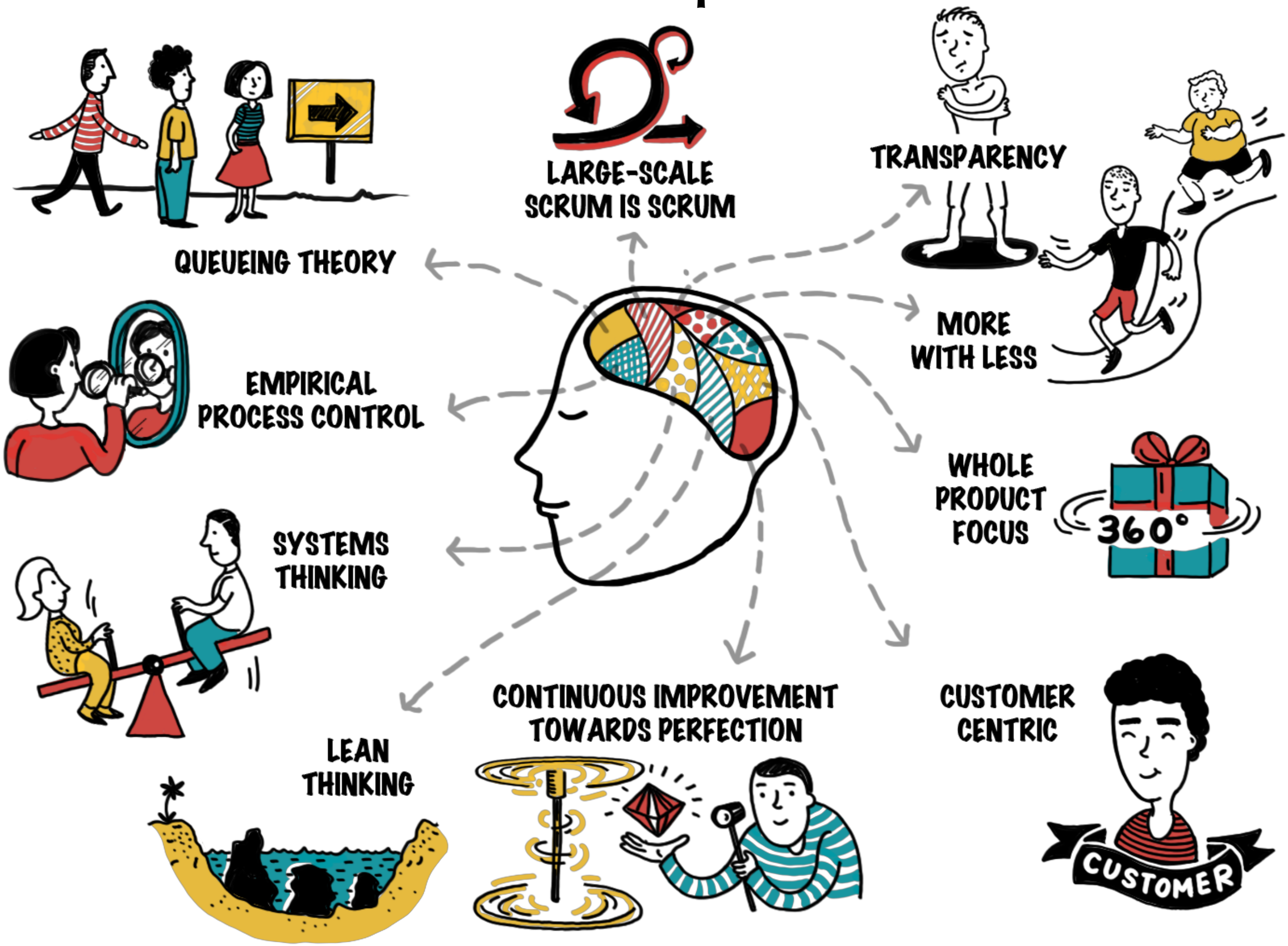
some (or much) of this is repetition for you, but is together for a **summary**

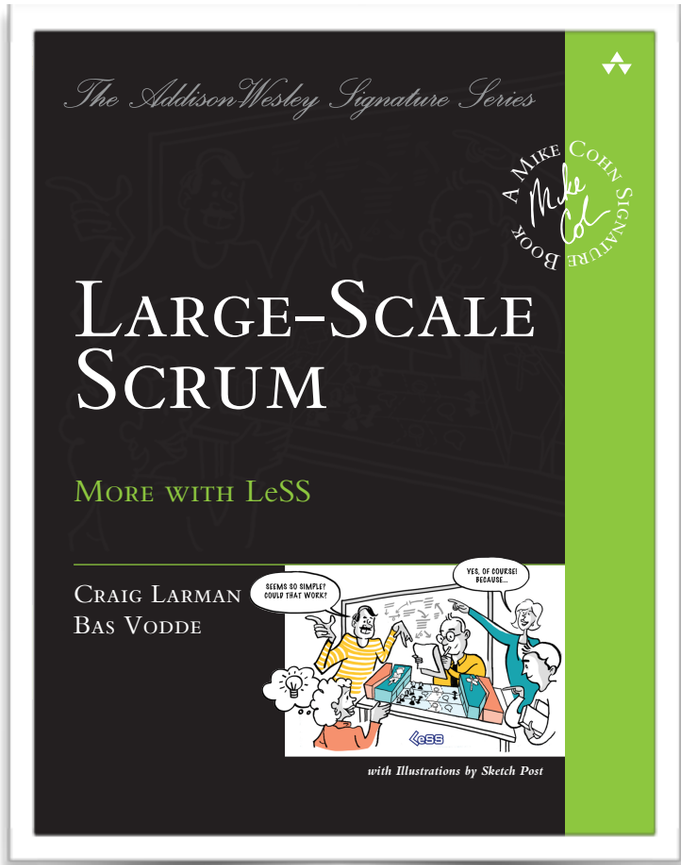
(optional) preparation: at end of section, you will be **sketching and teaching** "all" of its ideas with others, **without referring to notes** 😉

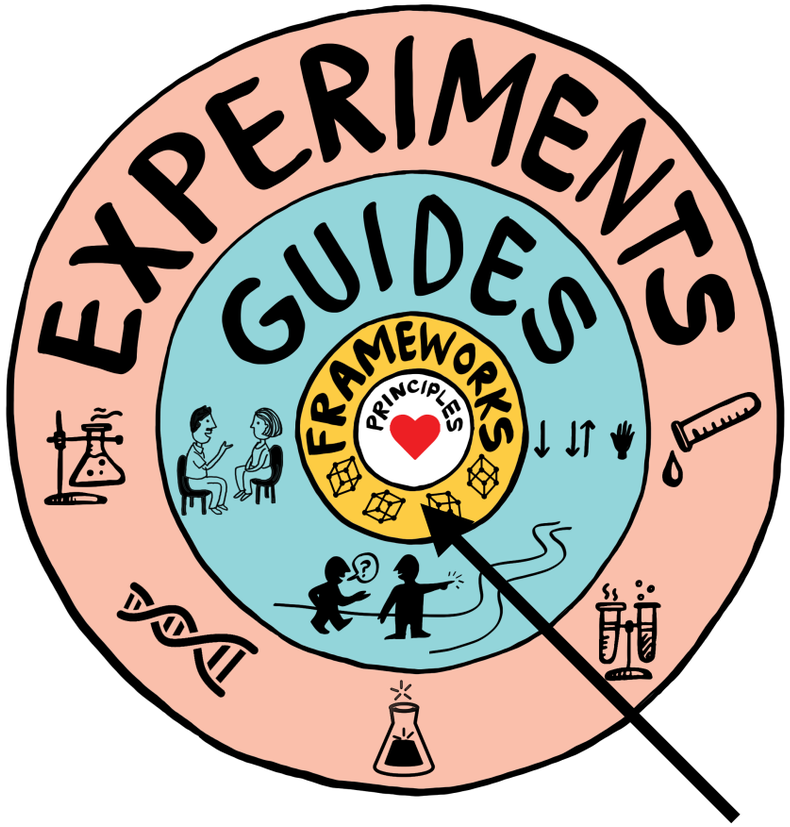
LeSS Complete Picture


EXPERIMENTS
GUIDES
FRAMEWORKS
PRINCIPLES

Principles


QUEUEING THEORY
LARGE-SCALE SCRUM IS SCRUM
TRANSPARENCY
EMPIRICAL PROCESS CONTROL
MORE WITH LESS
WHOLE PRODUCT FOCUS
360°
SYSTEMS THINKING
CONTINUOUS IMPROVEMENT TOWARDS PERFECTION
CUSTOMER CENTRIC
LEAN THINKING
CUSTOMER

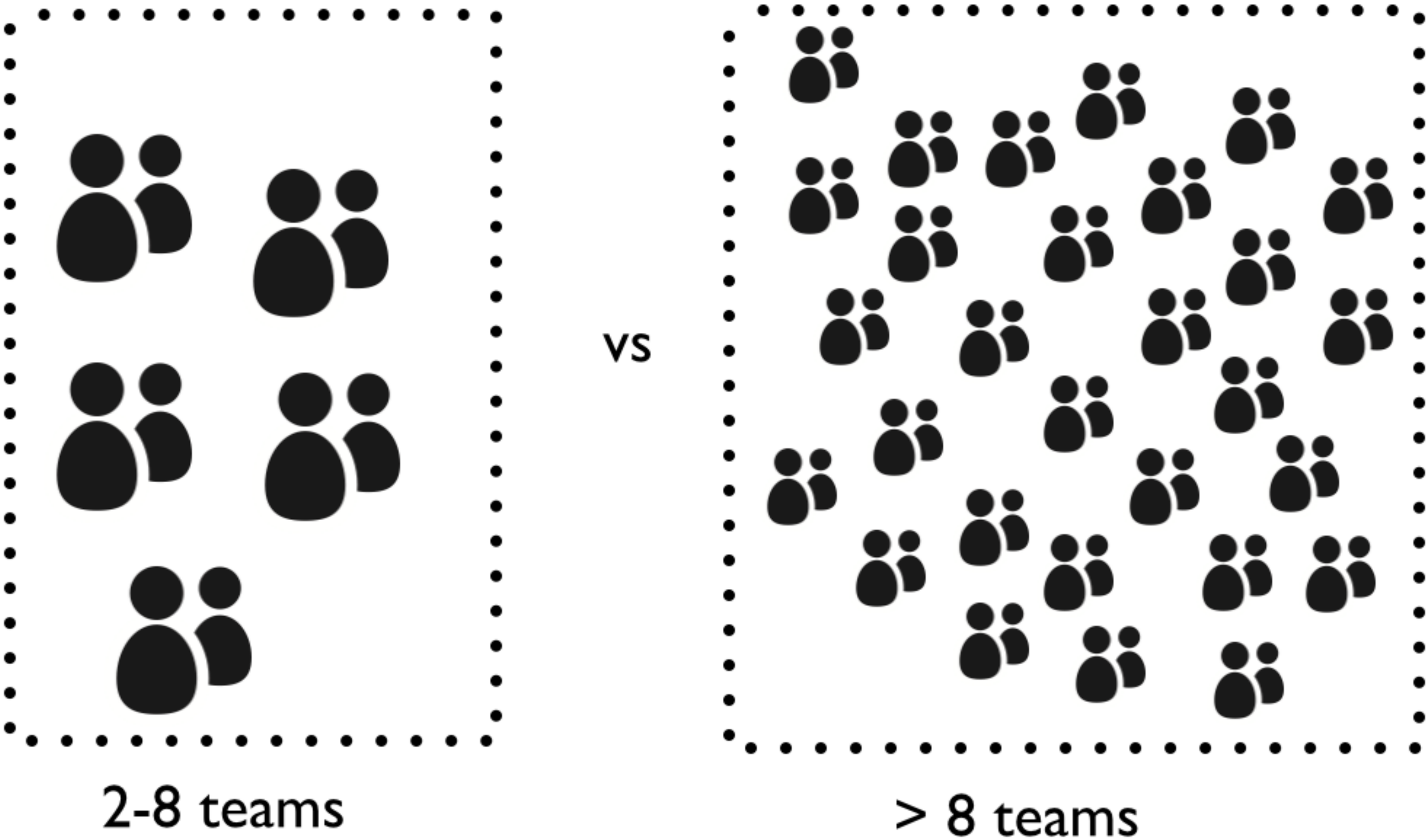




the LeSS rules
define the 2 frameworks;
STUDY THEM & KNOW WHY

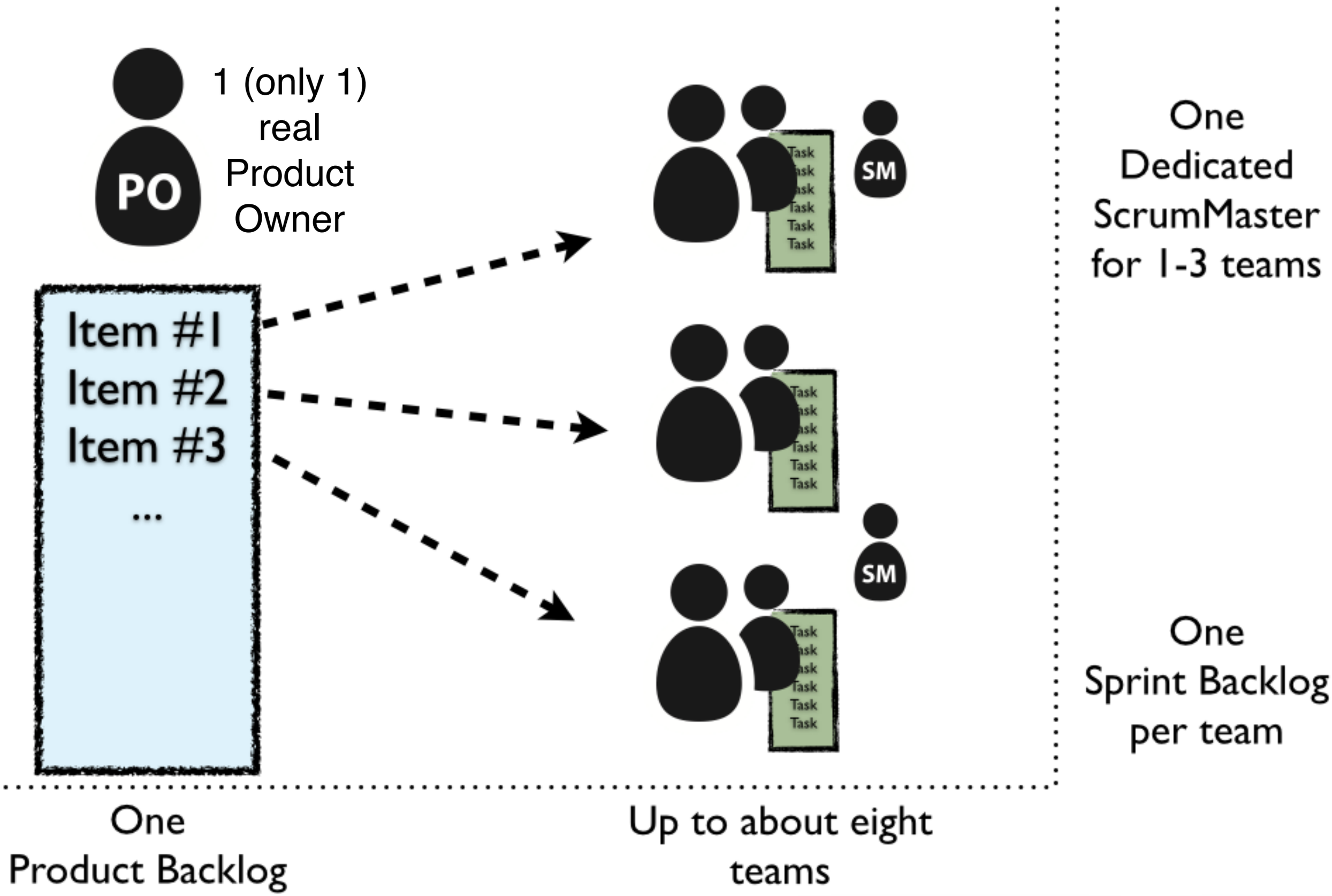
LeSS Introduction

2 Frameworks: LeSS & LeSS Huge

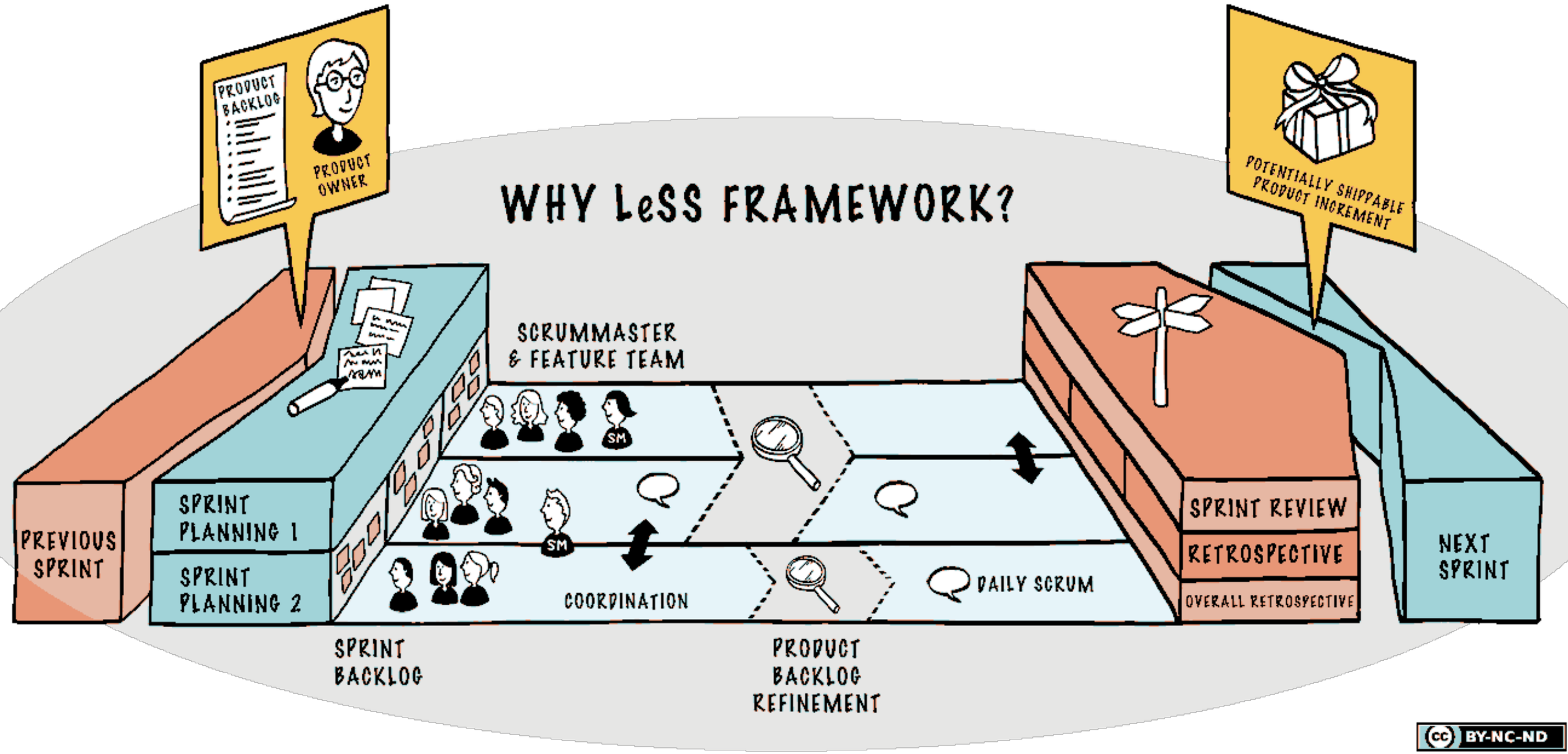


smaller LeSS framework...

(smaller) LeSS Framework



1 Common Sprint, 1 Shippable Product

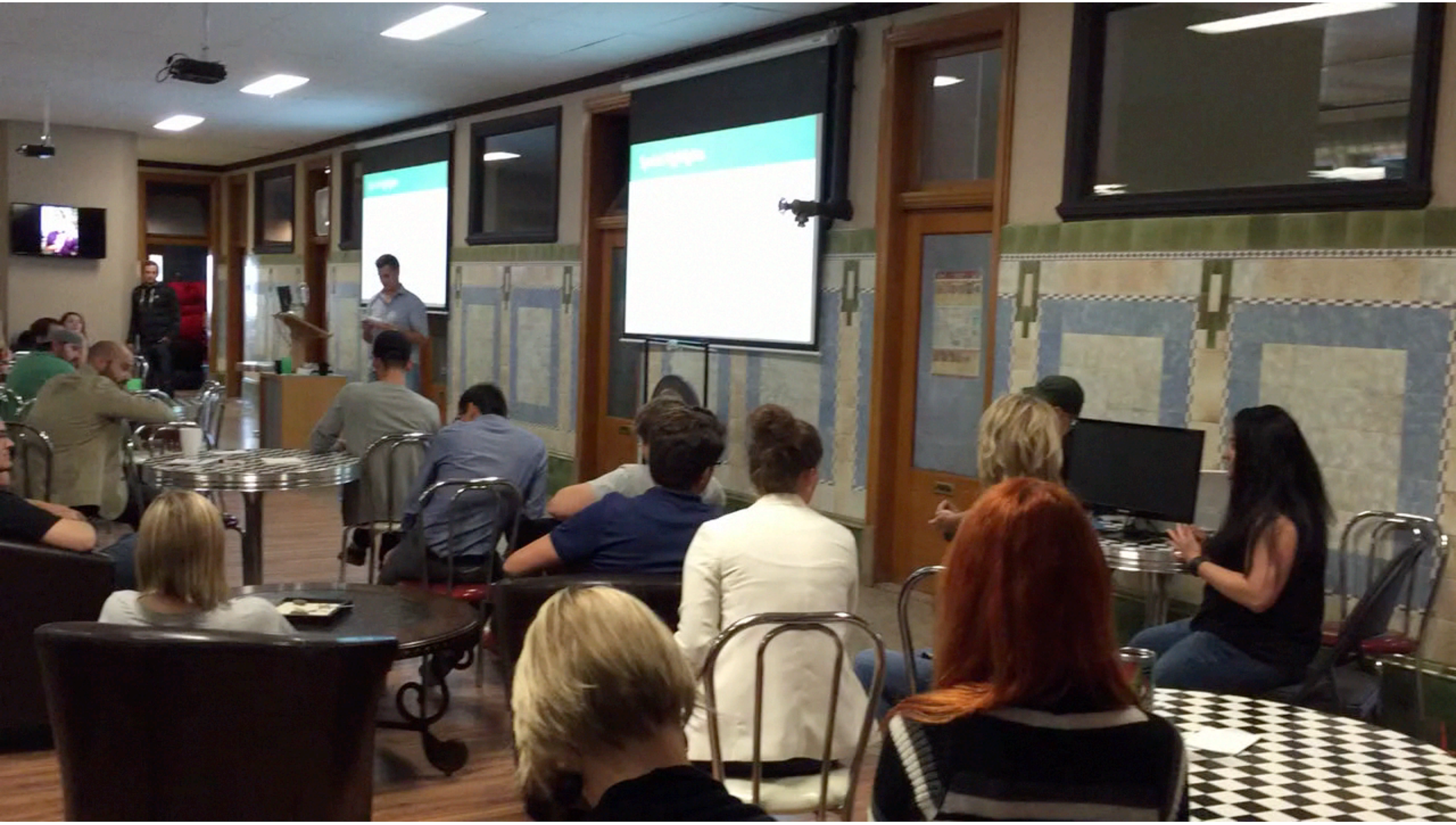


BY-NC-ND

Guide: Sprint Planning One

Guide: Sprint Review Bazaar

Sprint Review “Bazaar” Q&A; Next Steps



team: standing (or all together)

> **why** is there one **common Sprint** (versus asynchronous Sprints for each Team)?

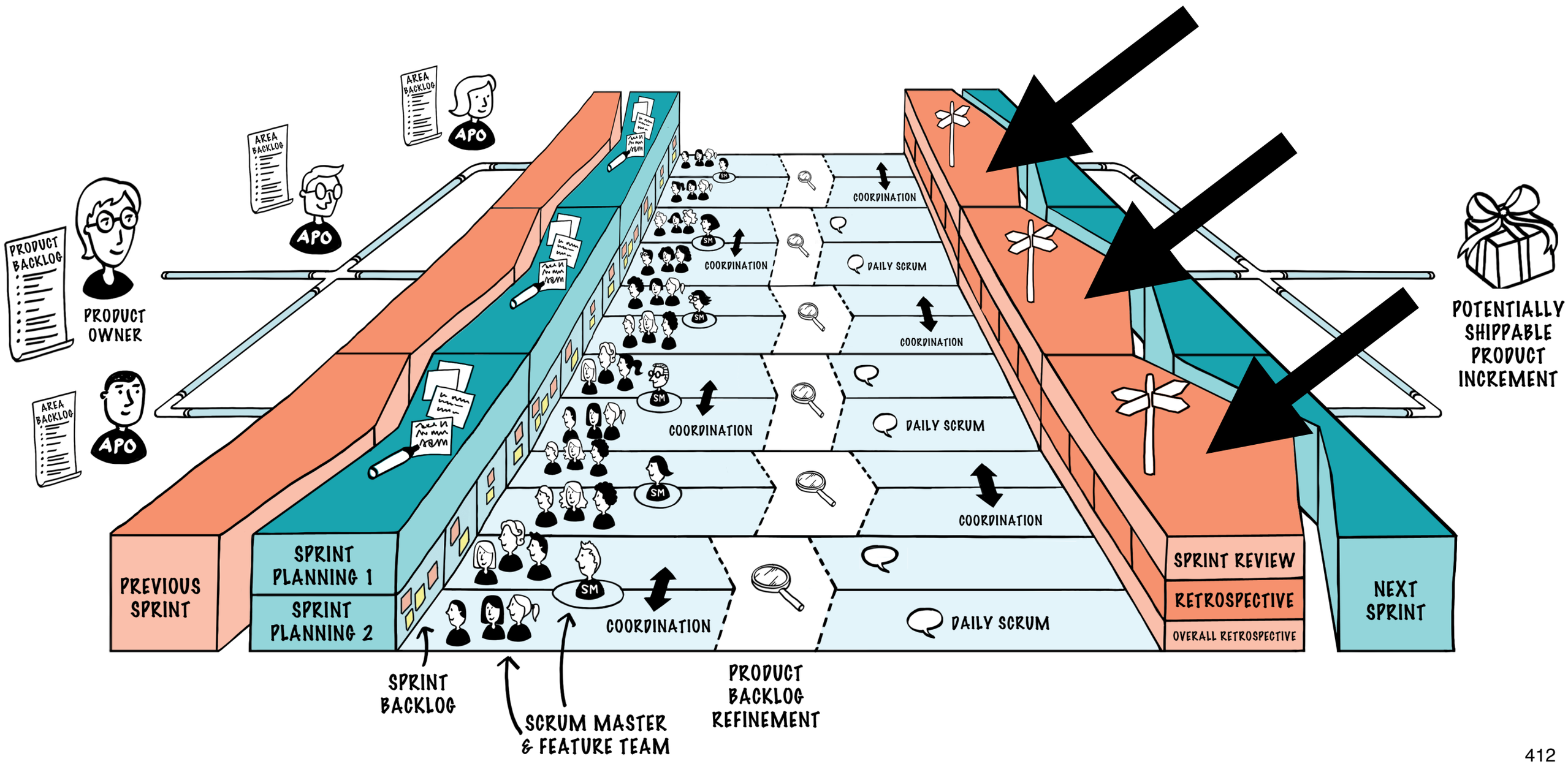
coach: review

Adoption

- **narrow & deep**
  - not broad & shallow
  - (*smaller LeSS FW*): “50” people, 1 product, 1 or 2 sites, “many” months
- **top-down** & **bottom up**
- **volunteering; don’t push**

LeSS Huge…

LeSS Huge Sprint: “Stack of LeSS”



Adoption: LeSS Huge

- not (or rarely) all-at-once
- typical approach:
  - **1 Requirement Area (RA) at a time**
    - **after "N" serial RA adoptions, may go parallel (e.g., 2 at a time)**

Guide: Tools

- no “agile” tools, no “ticket” tools
  - no **Jira**, **Version One**, “**Rally**”, etc.
- **Product Backlog** tool?
  - only use a (cloud) **spreadsheet** and **wiki** (e.g. Fitness) for the Product Backlog
- **Sprint Backlog** tool?
  - only “**cards on a wall**” — no digital tools

(optional)

individual

> briefly review this module

(optional)

pairs or triplets: standing: flipchart

> without referring to notes, one of the people teach the ideas in this section to the others, by… talking and sketching the ideas

(optional)

why LeSS? …

(optional) preparation: at end of section, you will doing a teach-back exercise 😉

Why LeSS?

- organizational design for **global systems Optimisation** for
- **adaptiveness** based on learning, to change direction towards **newly-discovered highest value**

Why LeSS?

> and, at scale …

> ***fucking* transparency**

> **whole-product focus**

> **empirical control**

Why LeSS? Occupational Psychology

owning versus renting

“barely sufficient”

-> more with **less**

shu - ha - ri

守破離

Rules Prescription & **Shu** - Ha - Ri

“barely sufficient methodology”

(optional)

pairs/triplets: standing: wall/ flipchart

> “teacher”… sketch and explain “why LeSS?”

Adoption

LeSS Adoption

(**optional**) team

> discuss:

> **“We want to do agile. To focus and accelerate the change, managers have objectives or bonuses or implicit rewards associated with an ‘agile adoption’ ”.**

(**optional**) team

> sketch a systems model, considering this scenario puzzle:

  > **We want to do agile. To focus and accelerate the change, managers have objectives or bonuses or implicit rewards associated with an "agile adoption".**

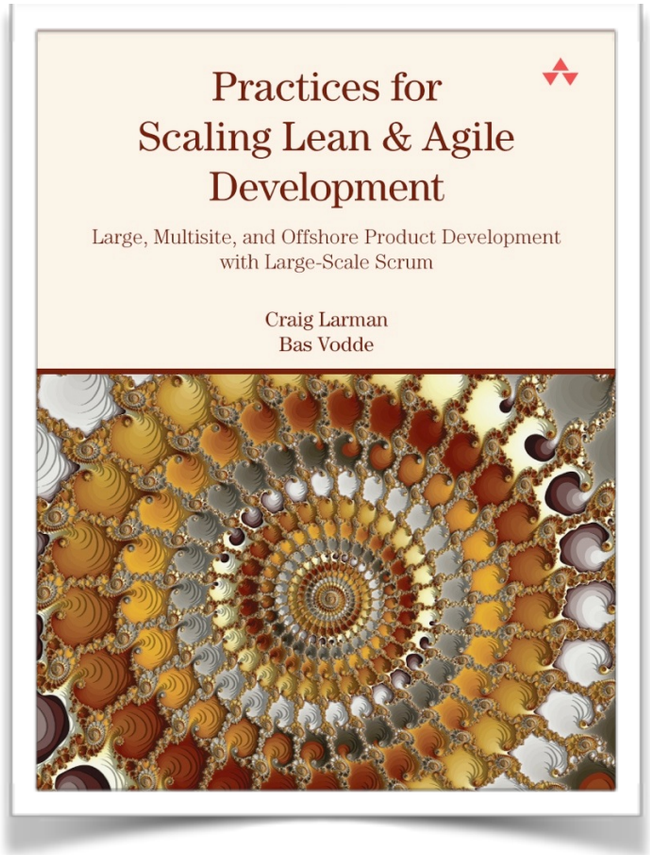
> start with these variables verbatim (some may already be in your model)

1. **strength of carrots/sticks to "meet plan"**("plan" is the "agile adoption goals")
2. **degree of "fear"**
3. **gap between true situation and "plan"** ("plan" in this case is for "agile adoption")
4. **pressure to deliver and "go faster"** ("deliver" in this case is to apparently "deliver the agile change")
5. **transparency**
6. **degree the *real* change requires elimination of *existing* mgr & single-specialist positions/groups, career paths, & financial & HR processes & policies**
7. **degree of *opaquely* re-labeling, re-defining, "gaming", and "changing the change", similar to status quo**
8. **degree of *visibly* "changing the change" for "pragmatic customization for local concerns", similar to status quo** (e.g. "let's create our own *agile cookbook*")
9. **agility to adapt early based on understanding real situation**
10. **degree of *real* improvement from *apparently* adopting the "change"**

Incentives work

… but not in the way naively assumed

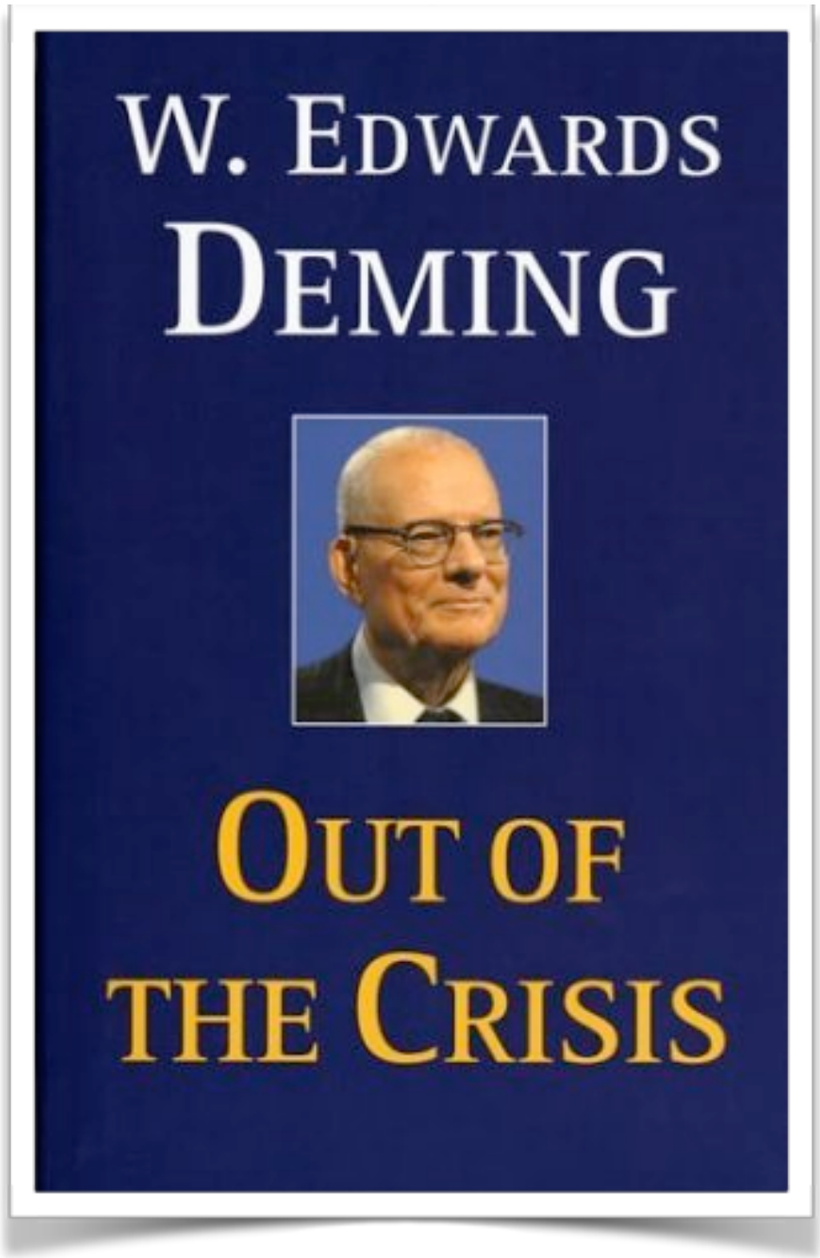
“Avoid… Agile adoption targets or rewards”



Action Tools

2. Large-Scale Scrum
3. Test
4. Product Management
5. Planning
6. Coordination
8. Requirements
9. Design & Architecture
10. Legacy Code
11. Continuous Integration
12. Inspect & Adapt
13. Multisite
14. Offshore
15. Contracts

Total Quality Management (TQM)



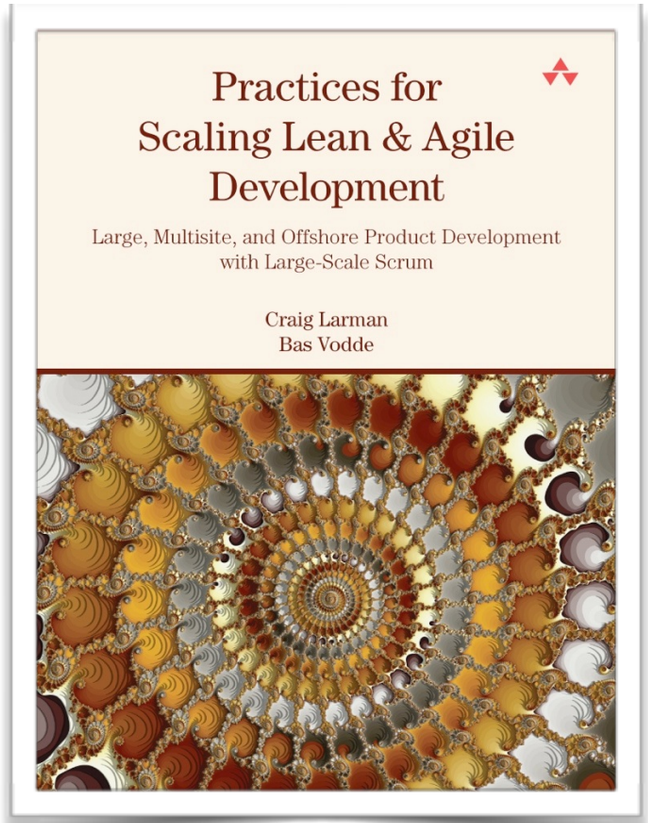
(point #11 for managers)

"Eliminate management by objective.

Eliminate management by numbers, numerical goals.

Substitute leadership."

“Avoid… Agile Transformation Team”



Action Tools

2. Large-Scale Scrum
3. Test
4. Product Management
5. Planning
6. Coordination
8. Requirements
9. Design & Architecture
10. Legacy Code
11. Continuous Integration
12. Inspect & Adapt
13. Multisite
14. Offshore
15. Contracts

preparation: at end of section, you will be sharing “all” of its ideas with others, without referring to notes 😉

Experiment

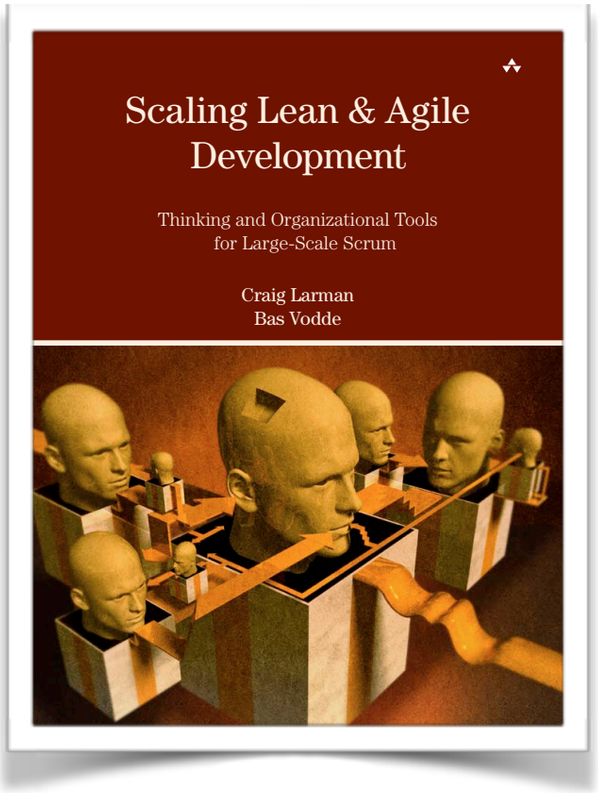
think & act like a **politician**,
not like an engineer

Experiment

think & act like a **psychologist**, not like an engineer

Experiment: Building Interest

- give "LeSS 1 or 3" **book** to key people
- book **clubs**
- send LeSS **video** links to people
- start with **why**
- find internal senior-manager **champion**
- find & grow **allies**
- just **talk directly with senior managers**
- **external expert** talk—"you're never a prophet in own land"
- hold & promote **events** to build interest:
  - LeSS Practitioner, LeSS for Executives, Less LeSS



Experiment: Building Interest?

- 1 "proof of concept" feature team, working on major high-value end-to-end features, but still surrounded by the traditional organisation for the existing product
- will this **clearly & definitely** build interest? …

team: standing

> **complications** of introducing 1 feature team while surrounded by a large traditional organisation?

  > e.g. 5 component teams with private code, misc single-function teams (BA, HI, Test, …)

> **per definition** the 1 **feature team** is doing

  > shared/open code across entire product

  > all functional activities, e.g. analysis, HI design, integration, all testing

insight:
2 diametrically opposed organisational designs in place at same time

what could go wrong? ;)

"Get 1 scrum team working in a scaling context, first" can be naive, and often won't work, because of the reasons you identify

therefore…

all-at-once

"flip the system"

How Big Can "All-at-Once" Be Successful?

maximum of

"50" team members

… and let us know how you make it work bigger ;)

LeSS Rule(s)

For the product group, establish the complete LeSS structure "at the start"; this is vital for a LeSS adoption.

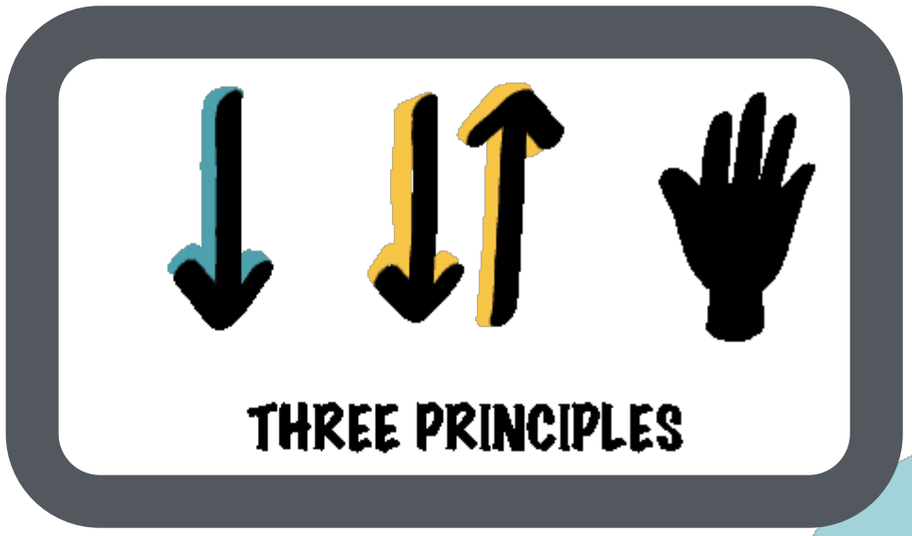
focus on **why**

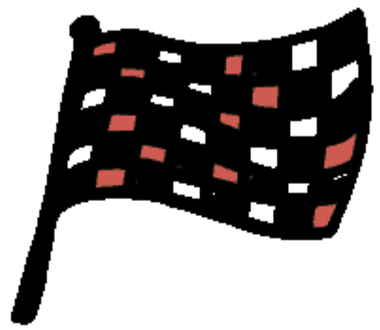
Experiment

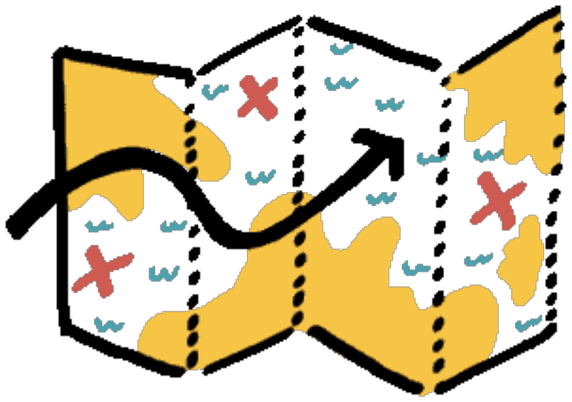
“informed consent” kickoff

gentle

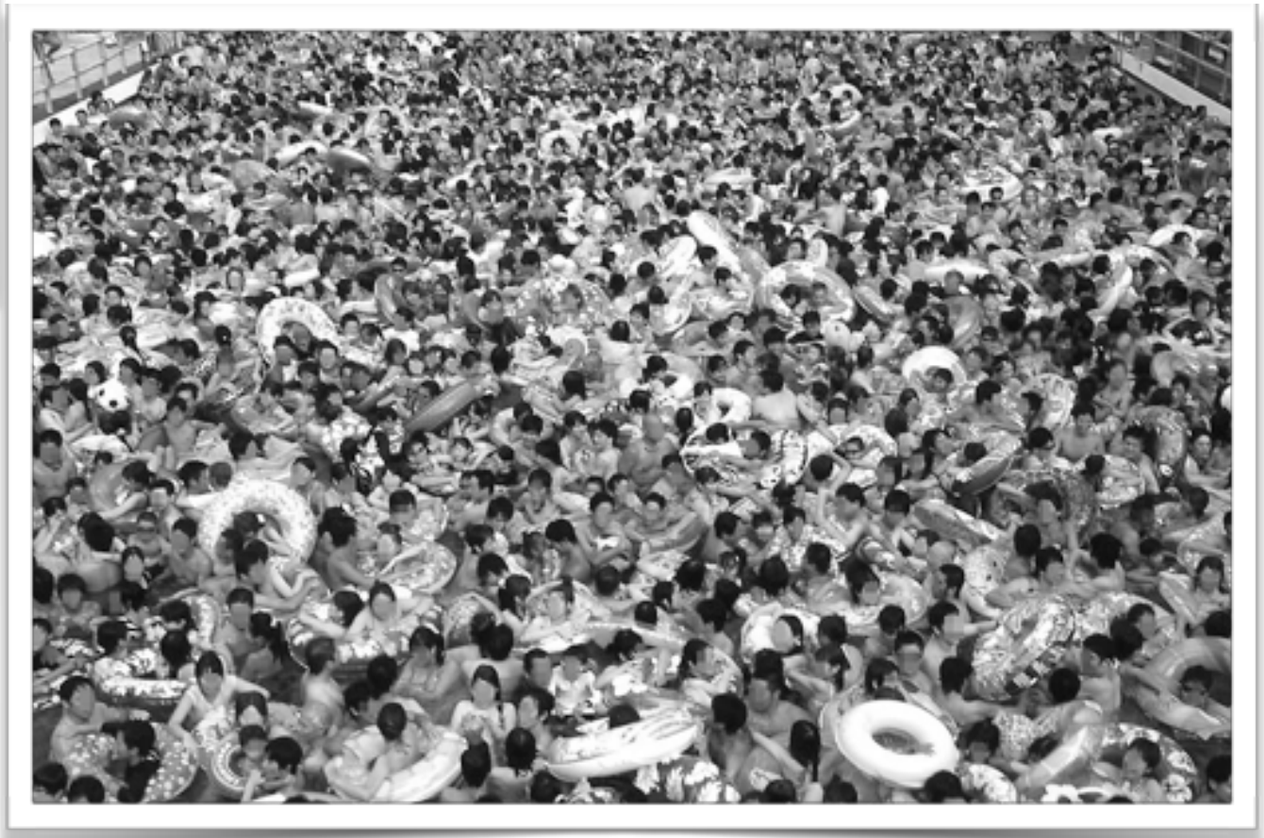
Guide: Three Principles







Adoption: **Deep & Narrow** over Broad & Shallow



Scope of First Adoption

"50" team members

1 product

preferably 1 site

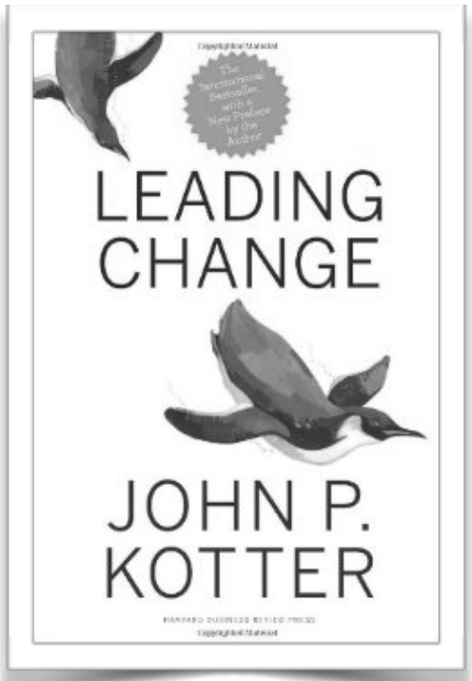
"several" months before another

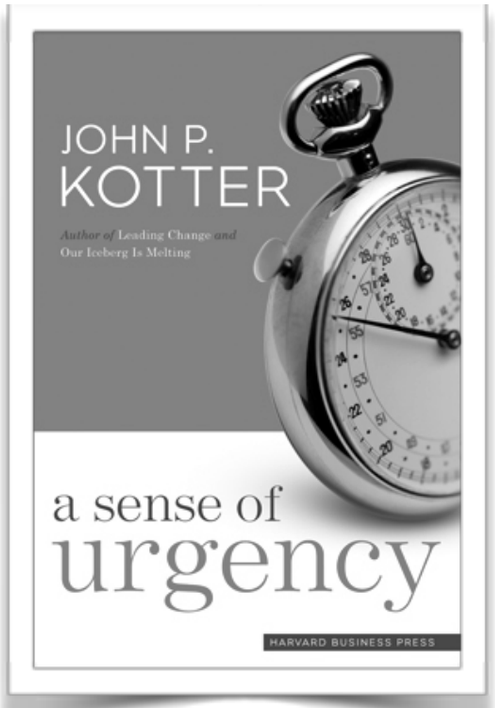
Adoption: **Top-Down** & **Bottom-Up**

3. Use Volunteering

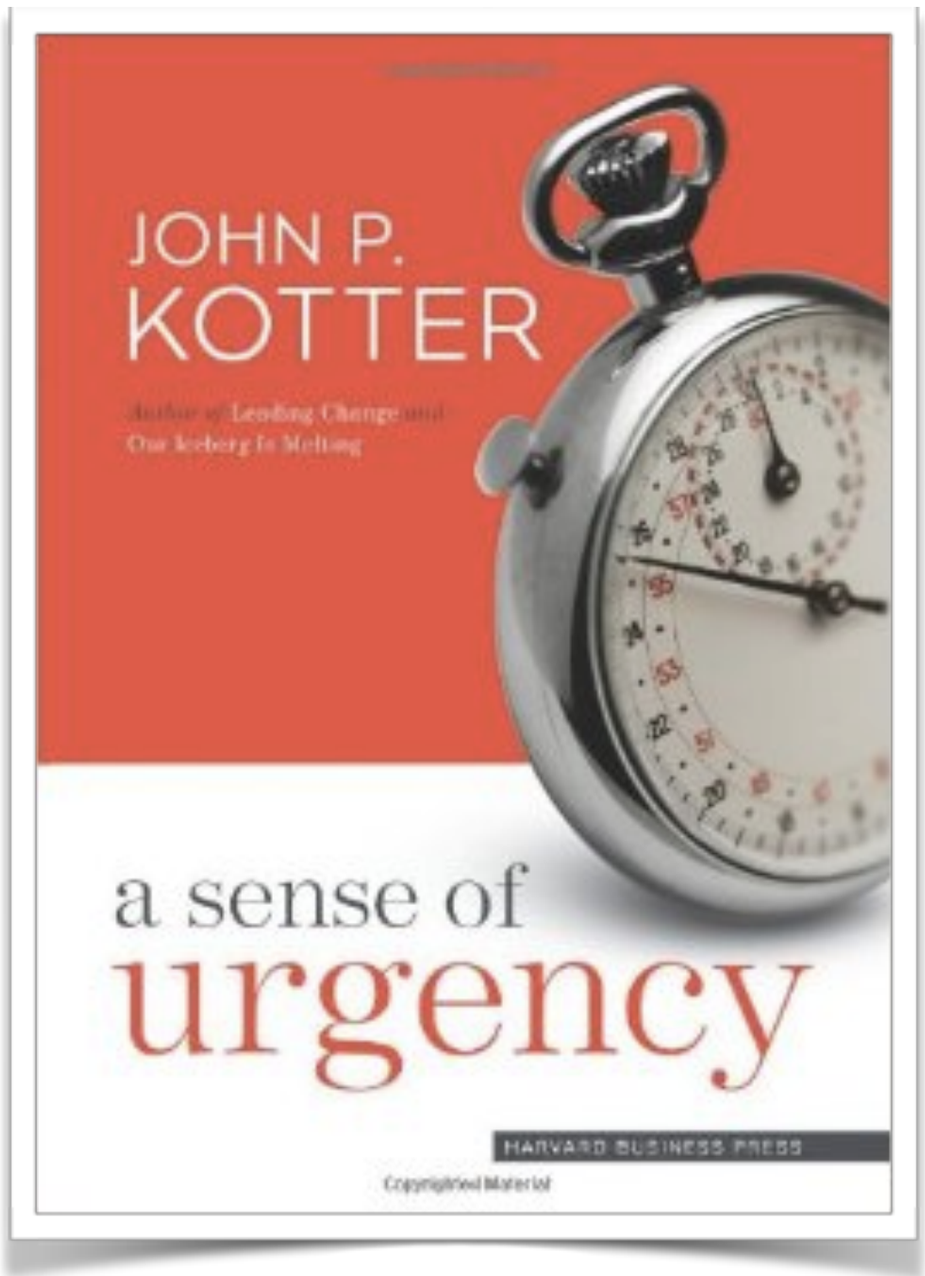
versus

Dr. J. Kotter on Resistance to Change



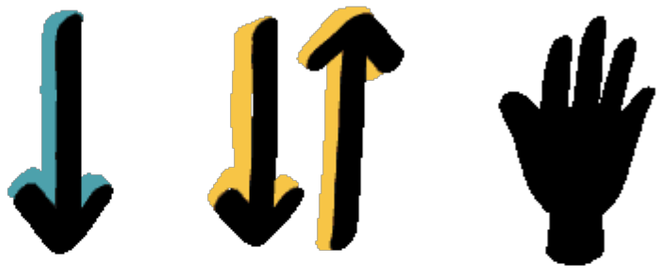


Dr. Kotter…



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

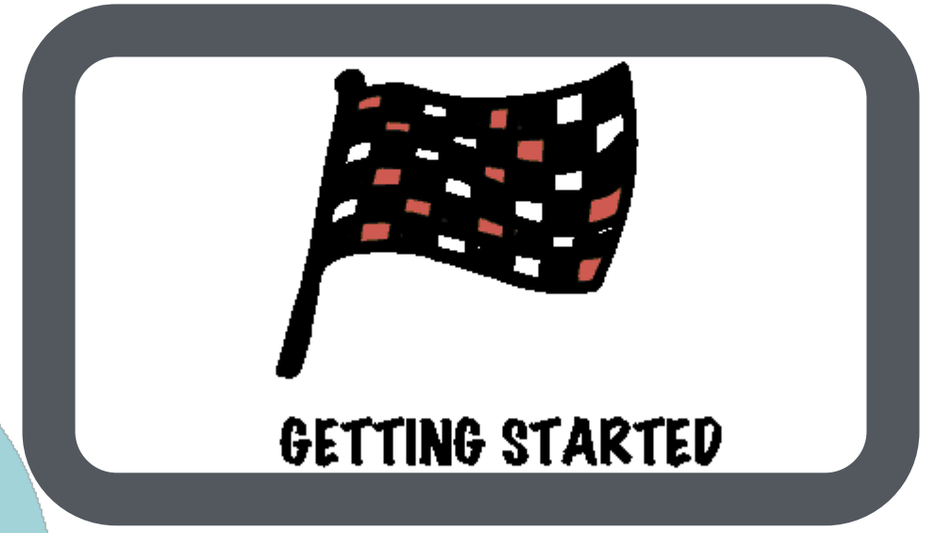
Guide: Getting Started



THREE PRINCIPLES

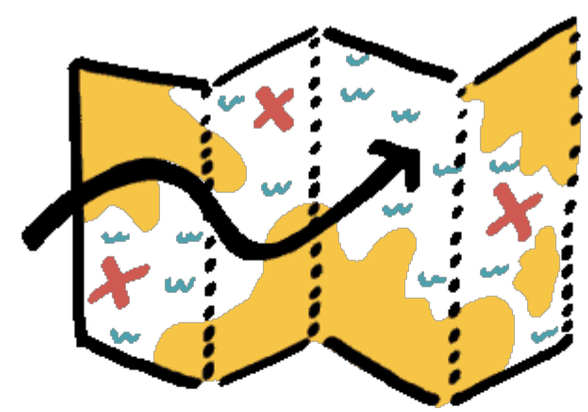
COACHING

ADOPTION



GETTING STARTED

CONTINUOUS IMPROVEMENT



FEATURE TEAM
ADOPTION MAP

BY-NC-ND

Guide: Getting Started

0. Educate *Everyone*

1. Define product

2. Define ‘done’

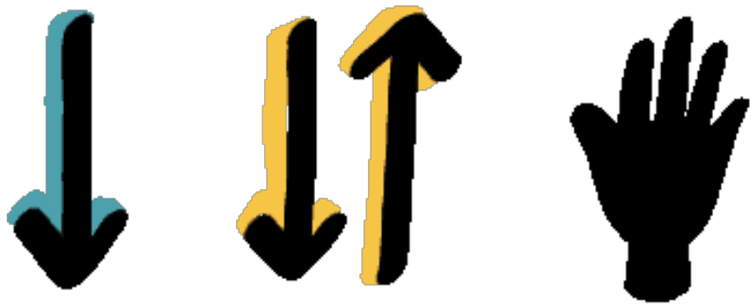
3. Have appropriately-structured teams

4. Only the Product Owner provides work for teams

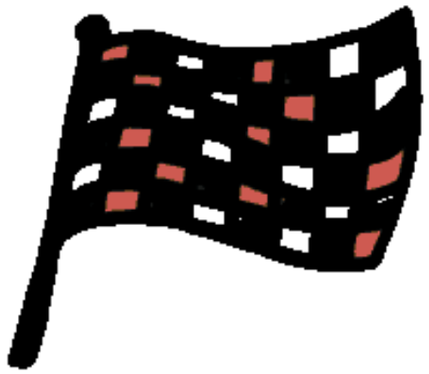
5. Keep “managers” away from teams

0. Educate Everyone

- focus on **why**, not what
- readings
- educate **all together** (not role)
- courses: Scrum, LeSS



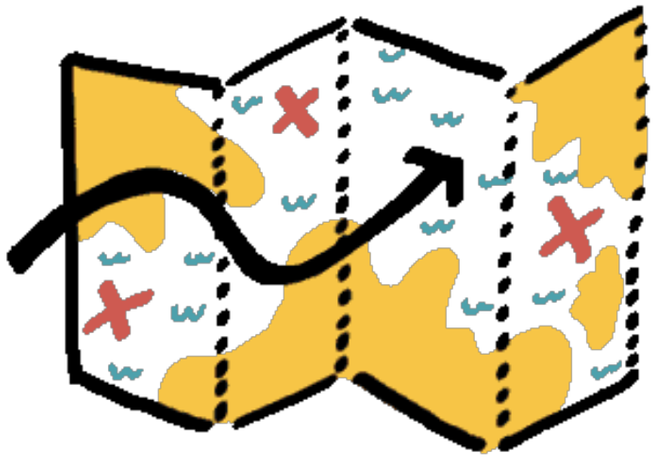
THREE PRINCIPLES



GETTING STARTED

COACHING

ADOPTION



FEATURE TEAM
ADOPTION MAP

CONTINUOUS IMPROVEMENT

Experiment: **Only Mob Programming for 3+ Months**

Guide: Getting Started

0. Educate *Everyone*

1. Define product

2. Define ‘done’

3. Have appropriately-structured teams

4. **Only the Product Owner provides work for teams**

5. **Keep “managers” away from teams**

BIG Idea

in LeSS many traditional manager positions are *eliminated* (not "redefined") — project & program managers, functional managers, component team managers, PMO, etc — and any few remaining manager positions *have* ***nothing to do with teams or development or tracking*** *(etc)*, but with improving the system (e.g., closing sites, eliminating traditional budgeting, eliminating outsourcing, eliminating tracking/reporting).

Guide: Job Safety, but not Role Safety

Job safety

& salary safety

job re-training

Experiment: Impediments Service

supporting managers receive impediments from feature teams, and improve the system by removing them ASAP

Experiment: Long Prep, Strong Start

prepare 2+ months

prepare for **shippable & impressive** in **first** Sprint

why?...

shipping speaks louder than words

“think like a politician”

Guide: Start Early or Messy with a *Temporary Fake Product Owner*

> why?

> the generalisation is

> if chance of “political blowback from messy start, don’t expose it to the outside”

> large-scale change is often fraught with “**organisational antibodies**” attacking the **change virus**, so “think like a politician”

BIG Idea

if the adoption is done seriously, all the hidden weaknesses & problems will be exposed and everything will appear "worse" for many months

BIG Idea

the **transparency**
emphasised in Scrum is
unlikely "nice" things; will
be
painful transparency
of years of **organisational**
debt & **technical debt**

BIG Idea

seriously consider all possible
problems, weaknesses,
failures, costs, & effort & time
in the adoption…
and **MULTIPLY BY TEN**

not joking or exaggerating

Guide: Ship at Least Every Sprint

Ship
at Least
Every
Sprint

LeSS **Huge** adoptions are **incremental**, not "all-at-once"

If you can realise a *successful real* LeSS adoption any other way, great!

Let us know how ;)

Guide: Tools

- no “agile” tool, no “ticket” tools
  - no **Jira**, **Version One**, **“Rally”**, etc.
- **Product Backlog** tool?
  - only use a (cloud) **spreadsheet** and **wiki** for the Product Backlog
- **Sprint Backlog** tool?
  - only “**cards on a wall**” — no digital tools

individual

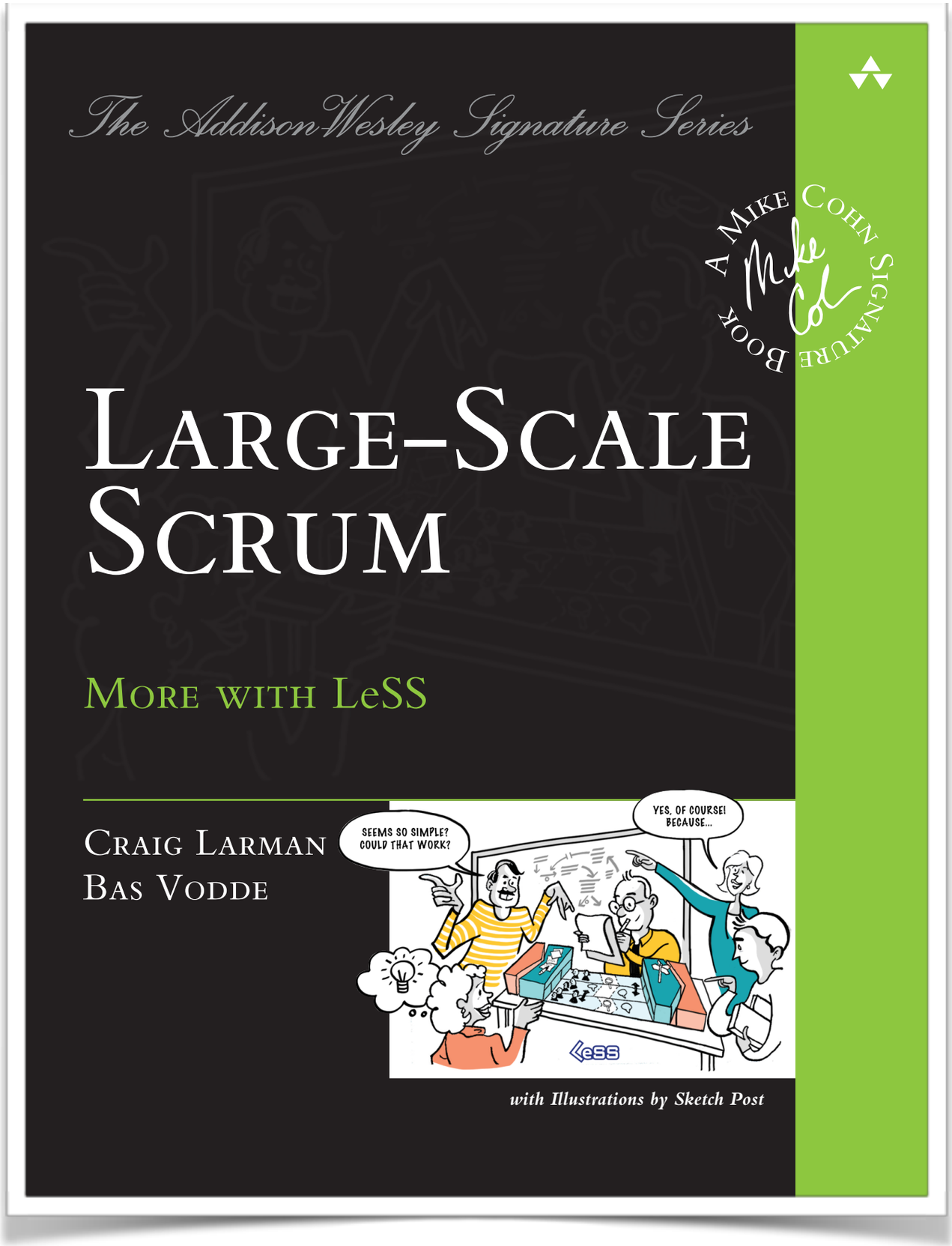
> briefly review this module

pair or team:

> **teach back** exercise

> miming, pair talking, …

> please **sit** when done



1 More with LeSS 1
2 LeSS 5

LeSS Structure

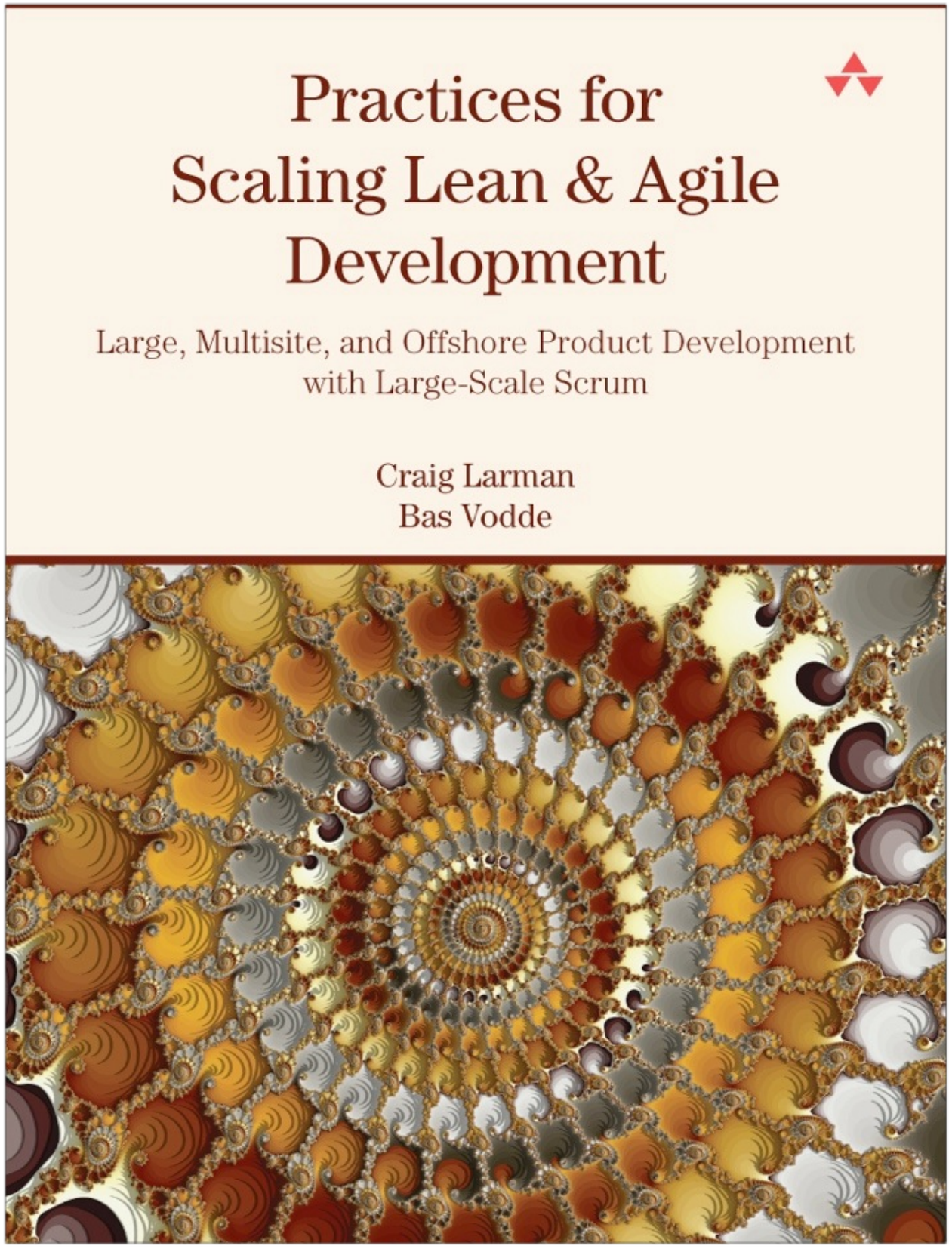
3 Adoption 41
4 Organize by Customer Value 67
5 Management 111
6 ScrumMasters 133

LeSS Product

7 Product 155
8 Product Owner 171
9 Product Backlog 197
10 Definition of Done 231

LeSS Sprint

11 Product Backlog Refinement 249
12 Sprint Planning 275
13 Coordination and Integration 285
14 Review & Retrospective 313

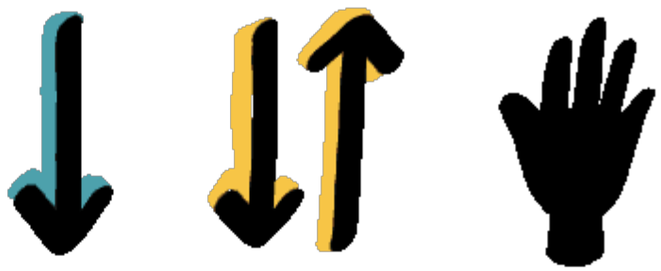


Action Tools

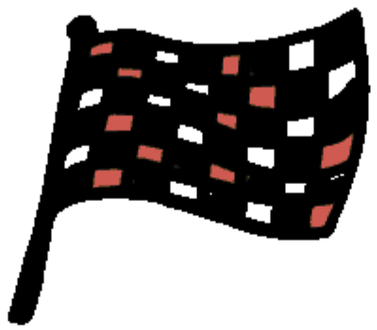
2. Large-Scale Scrum
3. Test
4. Product Management
5. Planning
6. Coordination
8. Requirements
9. Design & Architecture
10. Legacy Code
11. Continuous Integration
12. Inspect & Adapt
13. Multisite
14. Offshore
15. Contracts

Feature Team Adoption Map

Guide: Feature Team Adoption Map



THREE PRINCIPLES



GETTING STARTED

COACHING

ADOPTION

FEATURE TEAM
ADOPTION MAP

CONTINUOUS IMPROVEMENT

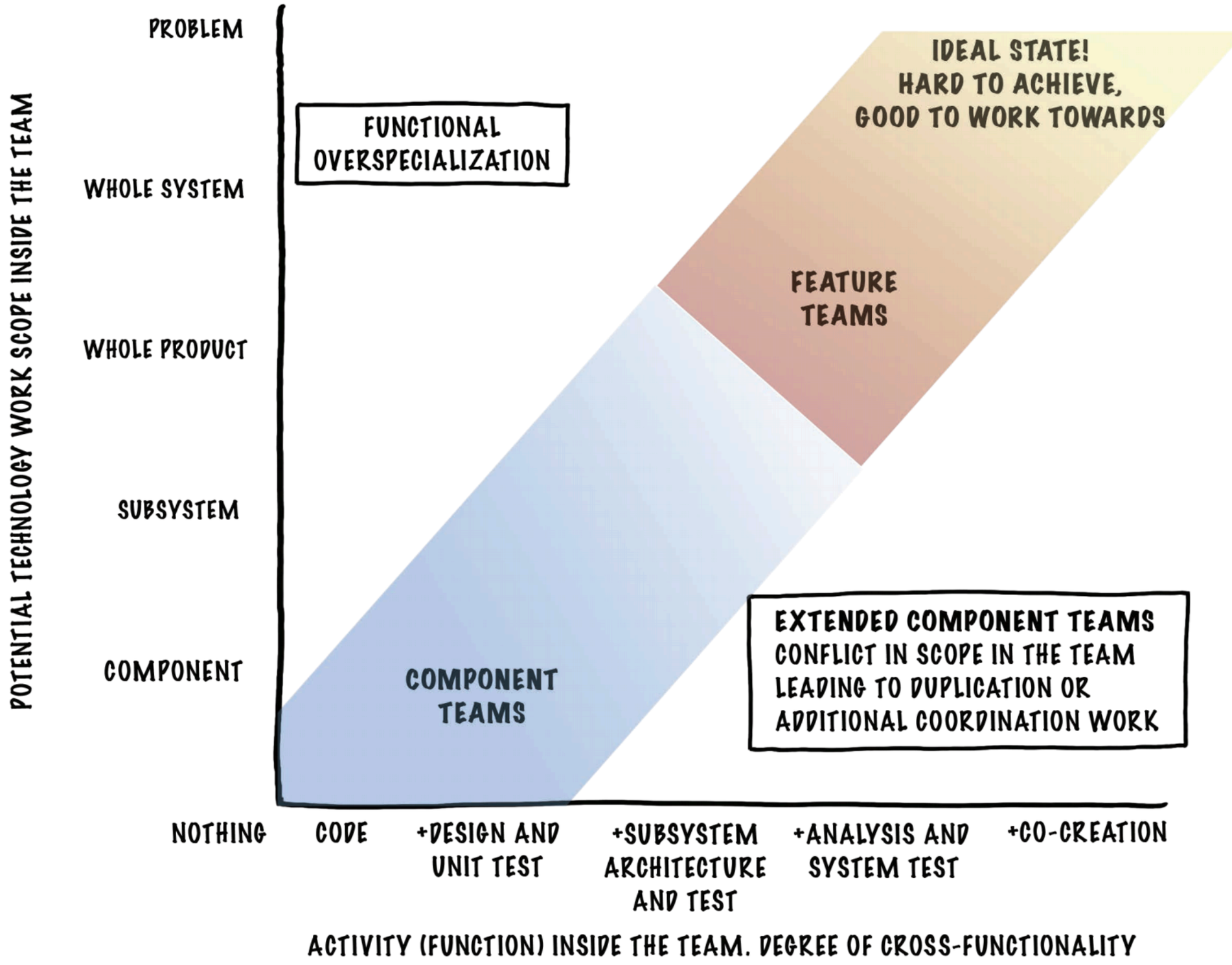
so far, we have assumed the initial creation of "complete" feature teams

but sometimes…

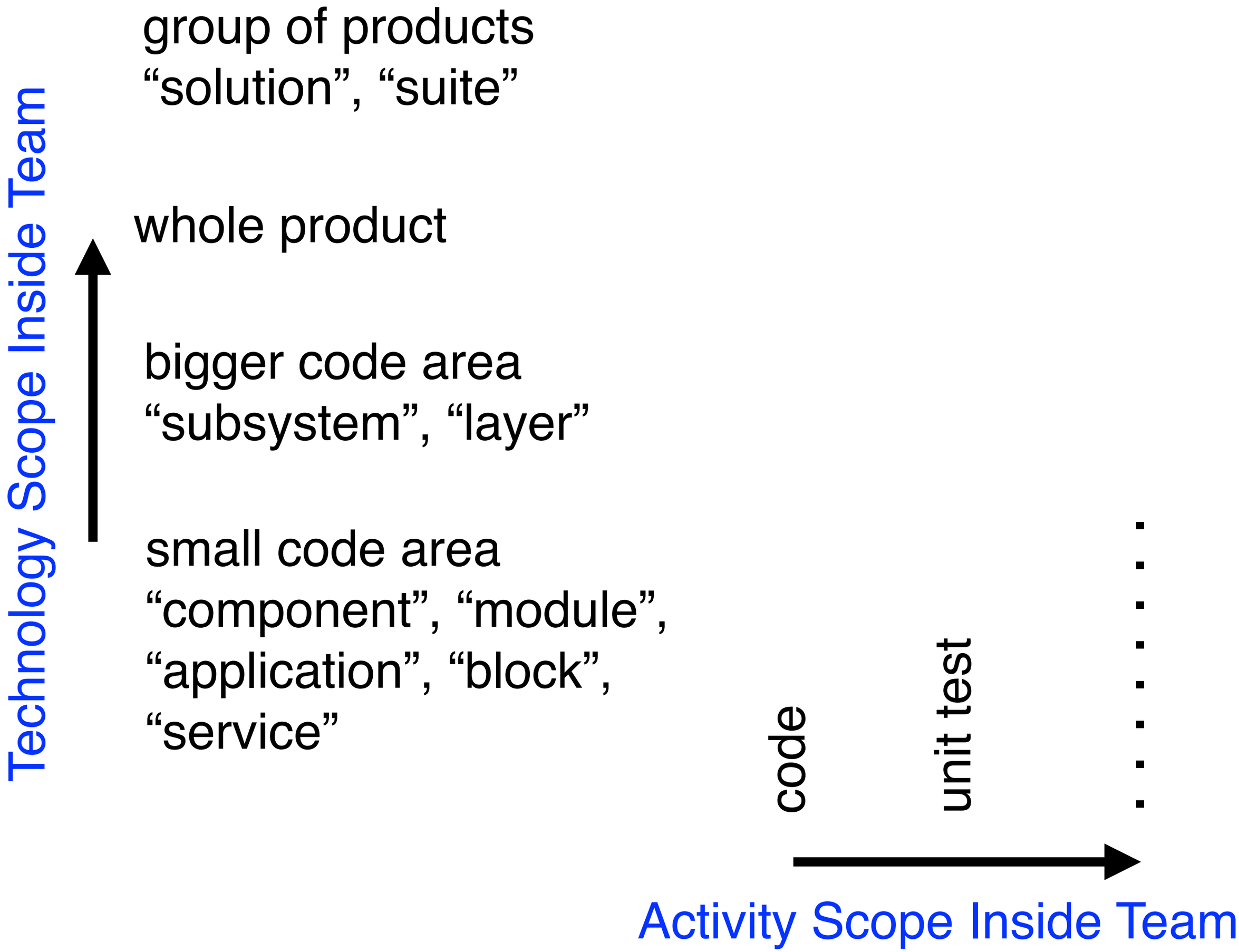
Incremental Feature Team Adoption

- extreme multi-site specialisations
- politics related to group structures
- a full-stack feature involves ‘20’ components and therefore ‘20’ developers
- extreme or very disparate technologies (“COBOL + JavaScript”)
- initially-imperfect “done” due to constraints
- LeSS Huge (many of the above issues)

when feature-team adoption must be **incremental**, analyse with a **Feature-Team Adoption Map** …



http://less.works BY-ND


group of products
“solution”, “suite”
whole product
bigger code area
“subsystem”, “layer”
small code area
“component”, “module”,
“application”, “block”,
“service”
Technology Scope Inside Team
code
unit test
. . . . . .
Activity Scope Inside Team

coach

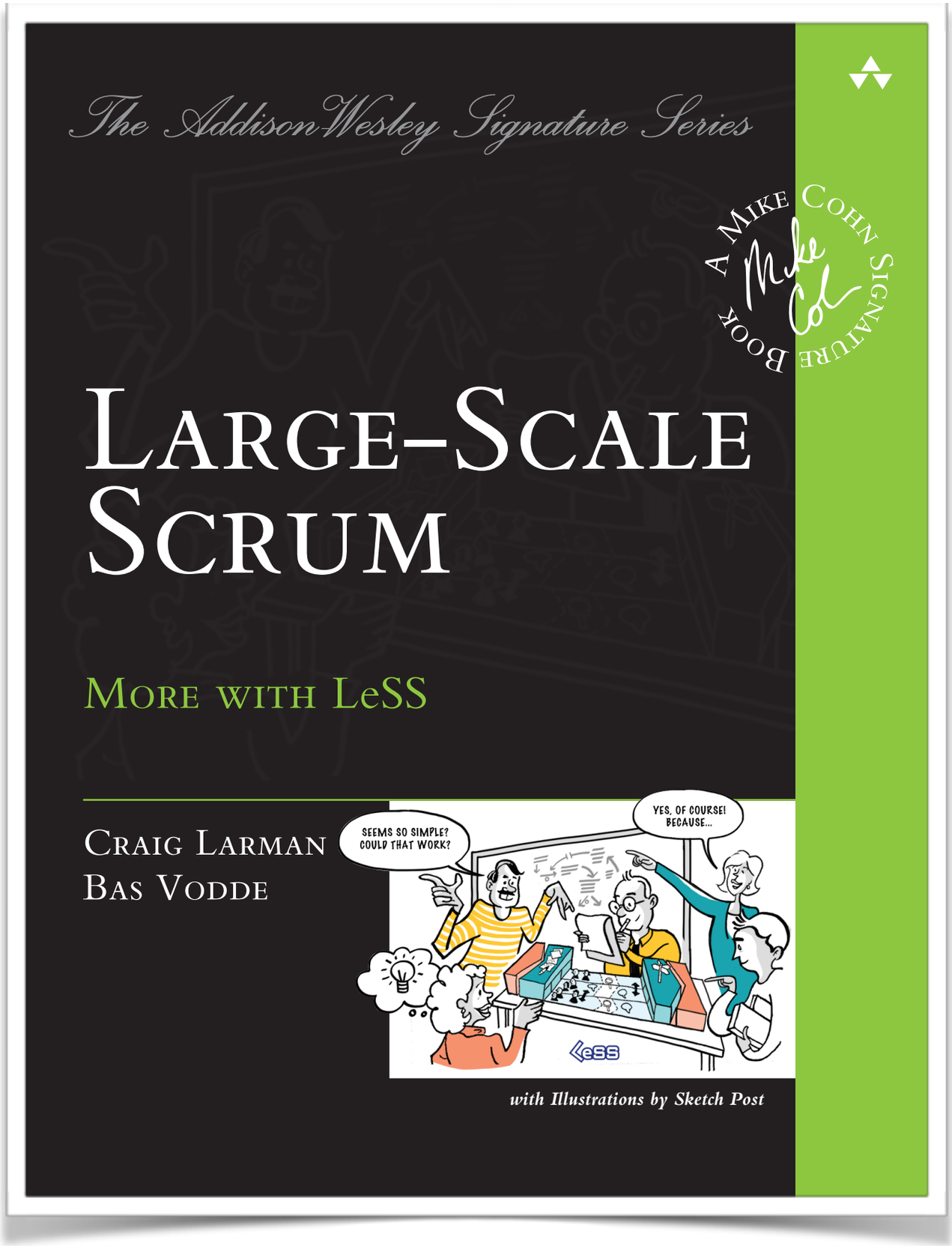
> demonstrate creating a feature-team adoption map, for an incremental adoption case

> mark where “potentially shippable”

> current state?

> next state?

> improvement experiments?



1 More with LeSS 1
2 LeSS 5

LeSS Structure

3 Adoption 41
4 Organize by Customer Value 67
5 Management 111
6 ScrumMasters 133

LeSS Product

7 Product 155
8 Product Owner 171
9 Product Backlog 197
10 Definition of Done 231

LeSS Sprint

11 Product Backlog Refinement 249
12 Sprint Planning 275
13 Coordination and Integration 285
14 Review & Retrospective 313

Why LeSS?

team

- sketch a systems model, given this:
  - **“large, detailed, framework for scaling, with many claimed best practices” is pushed onto a group by senior managers or consultants**
  - start with these variables
    - what is the ‘driving’ variable in this scenario?
    - degree of feeling of ownership and engagement by hands-on people in their processes & structures, and in improving them
    - degree of acceptance of specious arguments (e.g. argument by “best practices”, “gurus”, “sacred texts”, “they do it”, …)
    - degree of fear if dissent
    - degree of public dissent
    - degree of private dissent
    - degree of unnecessary or inappropriate processes & practices

Agile & Scrum: Original Messages

“barely sufficient”

“empirical process control”

team

> sketch a systems model, given this:

> **“large, detailed, framework for scaling, with many claimed best practices” is pushed onto a group by senior managers or consultants, and <u>then the group is invited to tailor it down</u>.**

> in addition to the prior variables, include at least

> what is the ‘driving’ variable in this scenario?

> degree of explicit or implicit goal to remain similar to status quo

> expectation to “get our money’s worth”

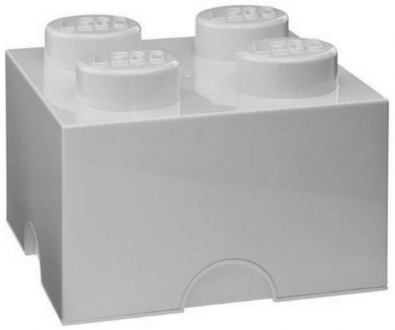
> expertise by the group to customize the framework

> similarity of the full framework to status quo

> degree of desire to shift blame “to the framework”, when problems

More with *less*

Build your framework up from a few simple core elements, based on from “why”



don’t tailor it down

why not just advise

"think & experiment"?

(i.e., zero prescription)

shu - ha - ri

守破離

Rules Prescription & **Shu** - Ha - Ri

“barely sufficient methodology”

Why “Rules”? **Shu** & Focus on Creating…

- global systems Optimisation for:
  - highest value, adaptiveness
- transparency
- whole-product focus
- empirical process control

one person per team

> sketch and explain
"prescriptiveness & LeSS rules"

Why LeSS? (part 1, our biases)

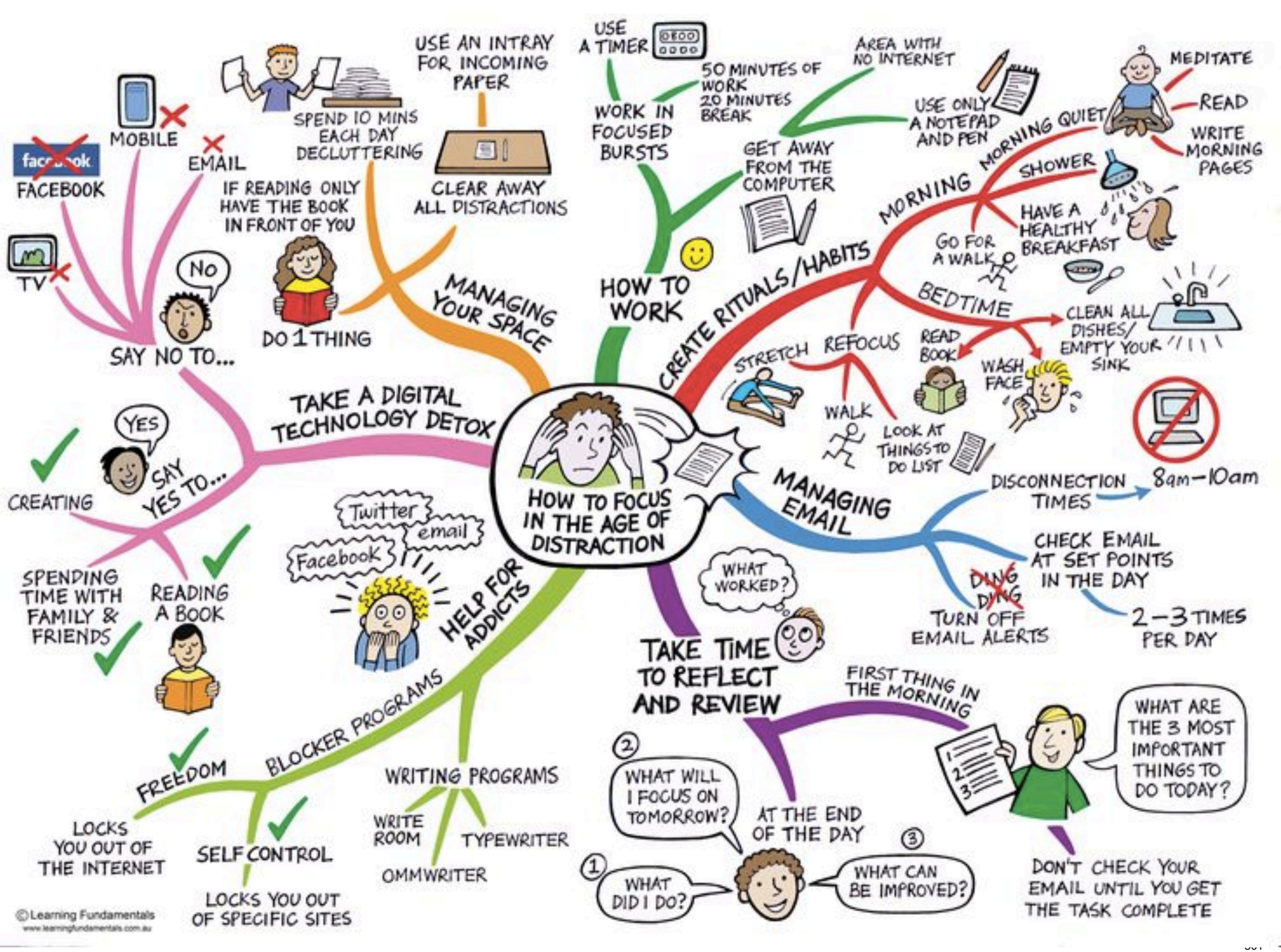
- global systems Optimisation for
  - deliver highest customer value
  - adaptiveness (“turn on a dime, for a dime”)
- transparency
- whole-product focus
- empirical process control

Why LeSS? (part 2, our biases)

We don't want **more roles**, as that	leads to **less responsibility** to the Teams.
We don't want **more artifacts**, as that	leads to a **greater distance** between Teams and customers.
We don't want **more dedicated analysts**, as that	leads to a **greater distance** between Teams and customers, **more handoff** problems, and **less engagement & empathy**.
We don't want **more supplied process & "best practices" & "renting"**, as that	leads to **less learning & team ownership** of process & **engaged improving**.

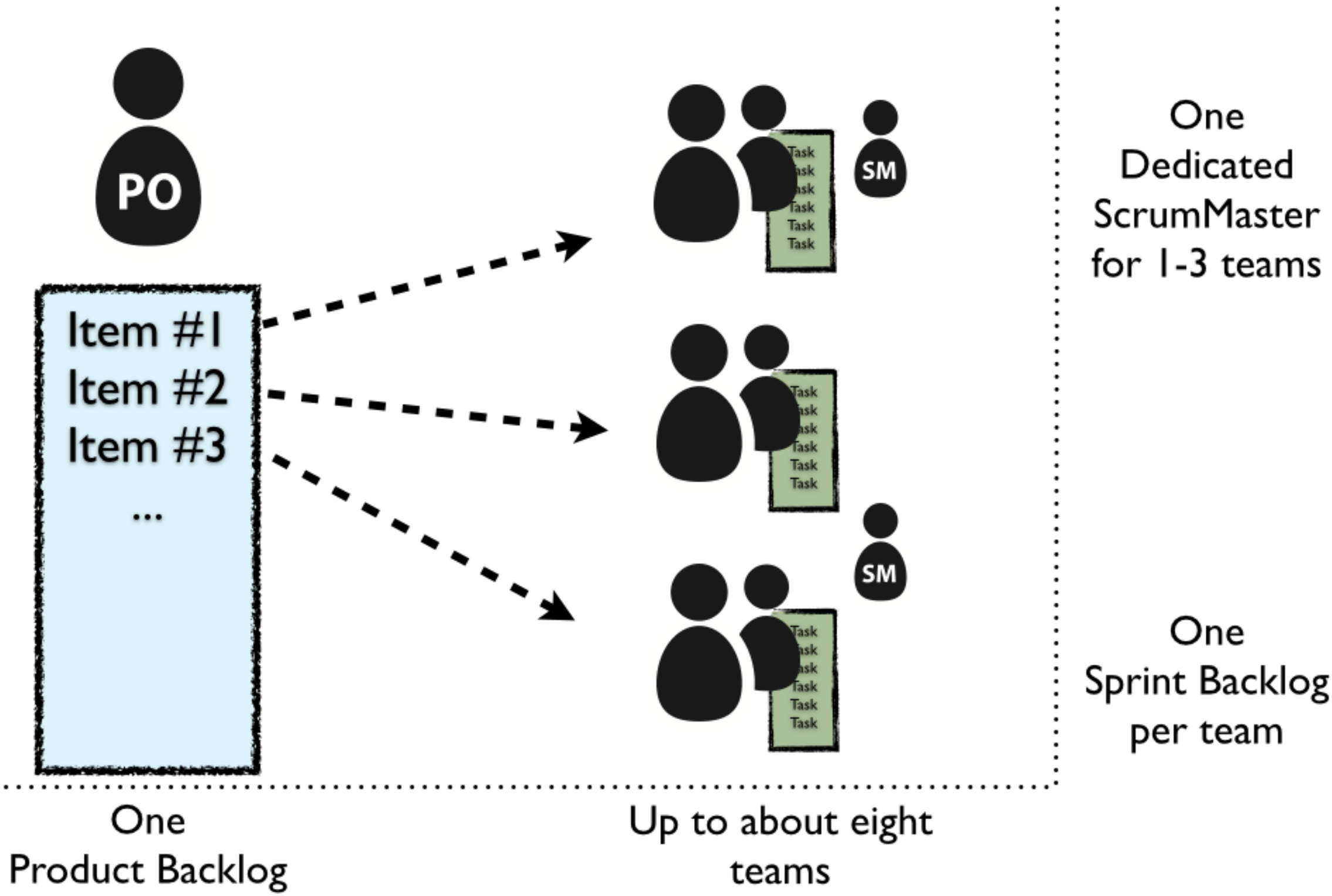
Why LeSS? (part 3, our biases)

We want **more responsible** Teams	by having **less roles**.
We want **more customer-focused** Teams building useful products	by having **less artifacts**.
We want more **customer-focused empathetic Teams that deeply understand requirements**	by **less dedicated analysts**.
We want **more Team ownership of process & meaningful work**	by having l**ess supplied processes & “best practices”**.


HOW TO FOCUS IN THE AGE OF DISTRACTION
TAKE A DIGITAL TECHNOLOGY DETOX
SAY NO TO...
No
MOBILE
EMAIL
FACEBOOK
TV
SAY YES TO...
Yes
CREATING
SPENDING TIME WITH FAMILY & FRIENDS
READING A BOOK
MANAGING YOUR SPACE
SPEND 10 MINS EACH DAY DECLUTTERING
USE AN INTRAY FOR INCOMING PAPER
CLEAR AWAY ALL DISTRACTIONS
IF READING ONLY HAVE THE BOOK IN FRONT OF YOU
DO 1 THING
HOW TO WORK
USE A TIMER
WORK IN FOCUSED BURSTS
50 MINUTES OF WORK 20 MINUTES BREAK
GET AWAY FROM THE COMPUTER
AREA WITH NO INTERNET
USE ONLY A NOTEPAD AND PEN
CREATE RITUALS/HABITS
MORNING
MORNING QUIET
MEDITATE
READ
WRITE MORNING PAGES
SHOWER
HAVE A HEALTHY BREAKFAST
GO FOR A WALK
BEDTIME
CLEAN ALL DISHES/ EMPTY YOUR SINK
READ BOOK
WASH FACE
REFOCUS
STRETCH
WALK
LOOK AT THINGS TO DO LIST
MANAGING EMAIL
DISCONNECTION TIMES
8am–10am
CHECK EMAIL AT SET POINTS IN THE DAY
2–3 TIMES PER DAY
TURN OFF EMAIL ALERTS
DING
DING
HELP FOR ADDICTS
Twitter
email
Facebook
BLOCKER PROGRAMS
FREEDOM
LOCKS YOU OUT OF THE INTERNET
SELF CONTROL
LOCKS YOU OUT OF SPECIFIC SITES
WRITING PROGRAMS
WRITE ROOM
TYPEWRITER
OMMWRITER
TAKE TIME TO REFLECT AND REVIEW
WHAT WORKED?
FIRST THING IN THE MORNING
WHAT ARE THE 3 MOST IMPORTANT THINGS TO DO TODAY?
DON'T CHECK YOUR EMAIL UNTIL YOU GET THE TASK COMPLETE
AT THE END OF THE DAY
1
WHAT DID I DO?
2
WHAT WILL I FOCUS ON TOMORROW?
3
WHAT CAN BE IMPROVED?
© Learning Fundamentals
www.learningfundamentals.com.au

LeSS Sprint

(smaller) LeSS Framework



Preparation Meetings

Preparation Meetings

- **Educate everyone**
- **Define Product**
- **DoD meeting**
- Feature-Team Adoption Analysis (e.g., with a Map)
- **Self-Designing Teams meeting**
- **Community kickoff meetings**
- **Initial Product Backlog refinement**
- Current-Architecture Learning Workshop
- Agile Modelling Design/Architecture Workshop

Self-Designing Teams Workshop

How to Form Teams in Large-Scale Scrum? A Story of Self-Designing Teams

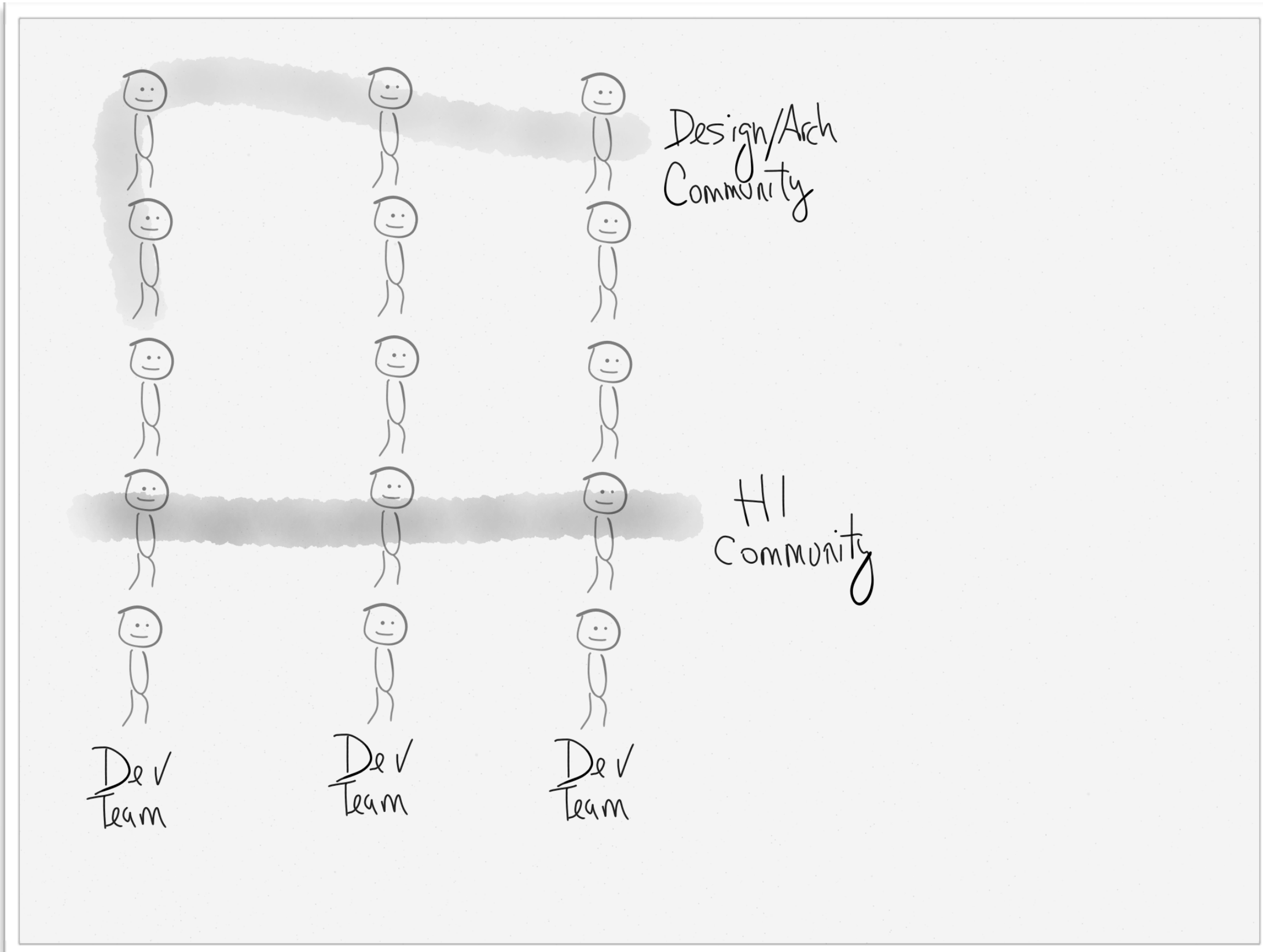
5 April 2013

Craig Larman

Ahmad Fahmy

How long does it take an organization to reorganize in order to adopt Scrum? Three hours. Really.

Community Kickoff meetings

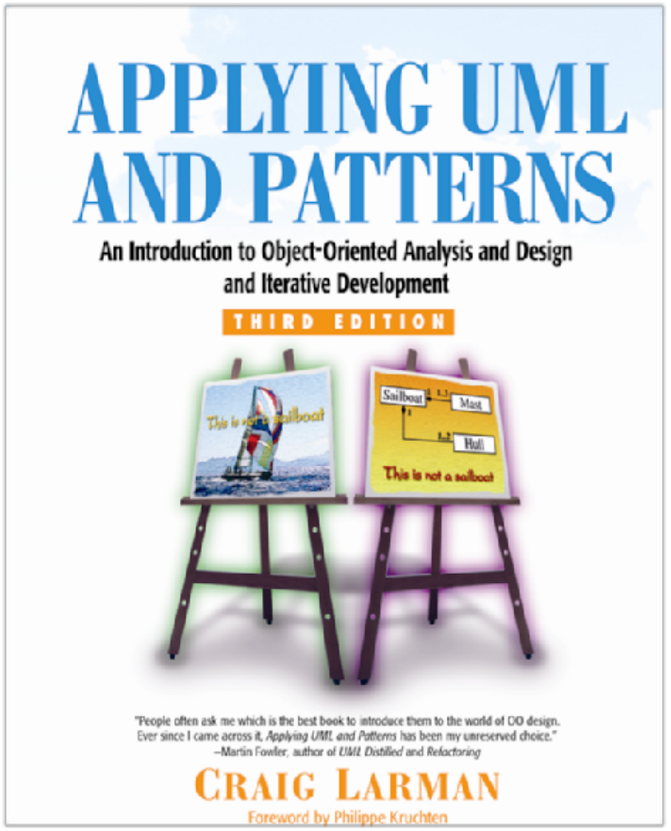


Guide: Initial Product Backlog Refinement

create shared understanding

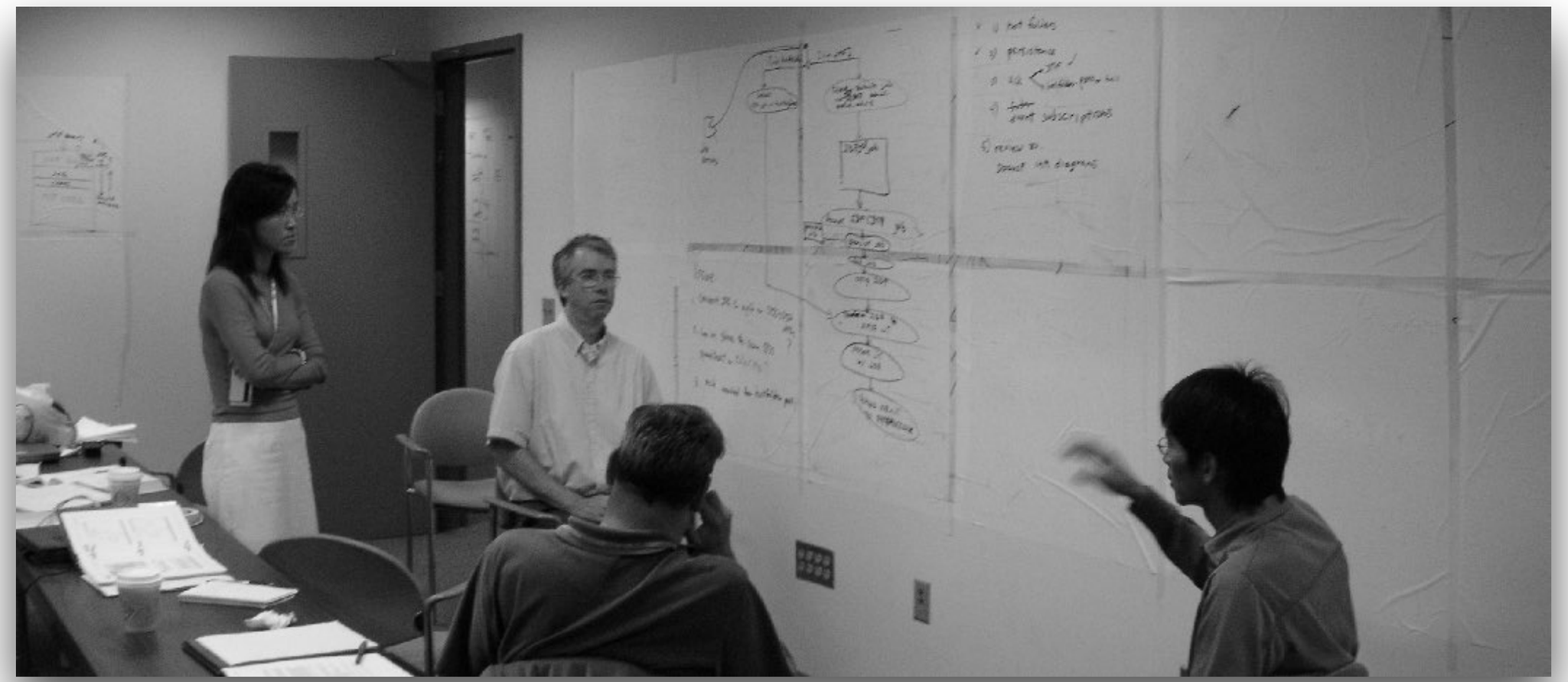
useful activities?

Guide: Current-Architecture Workshop

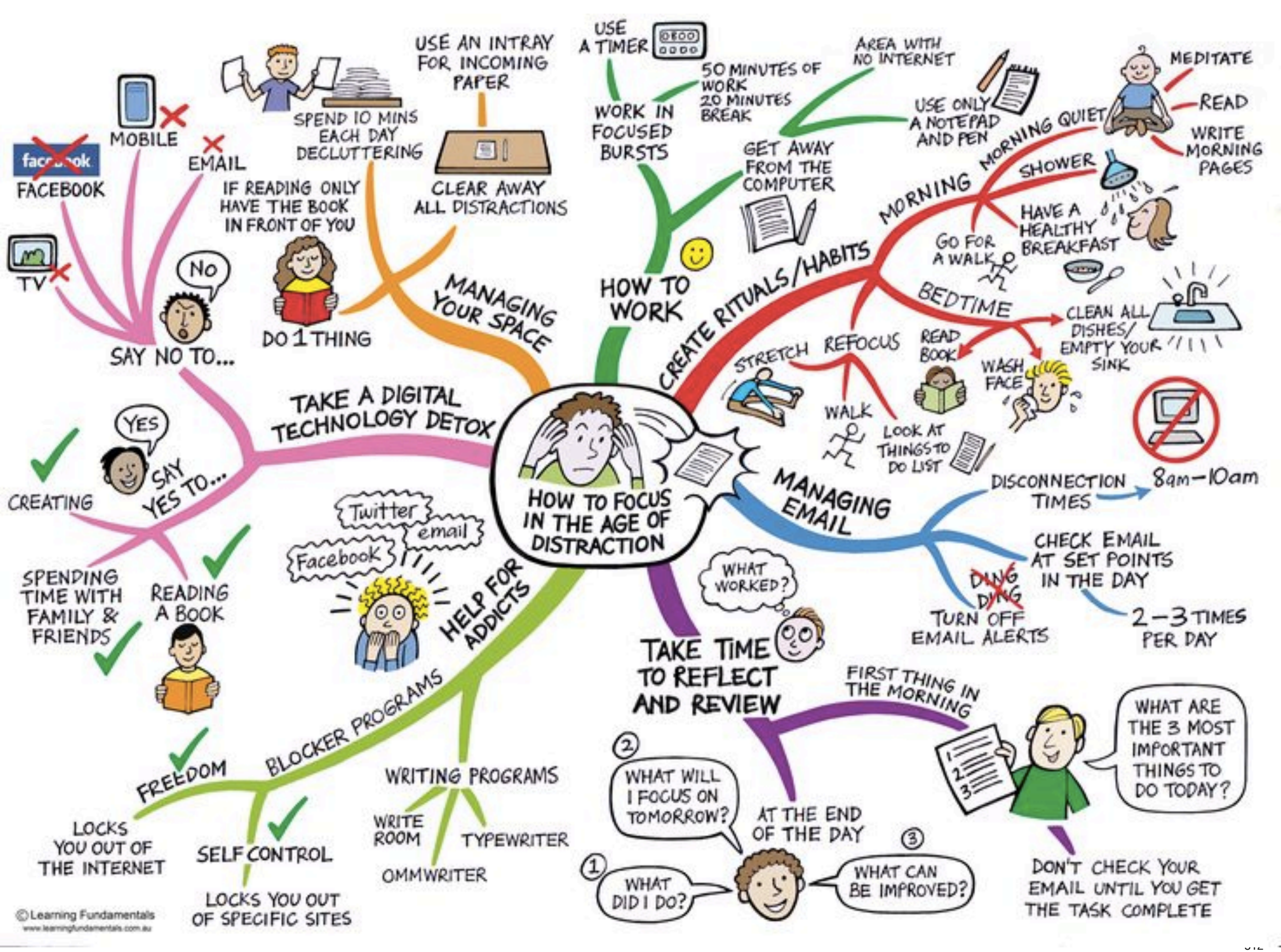


Chapter 39: Documenting Architecture

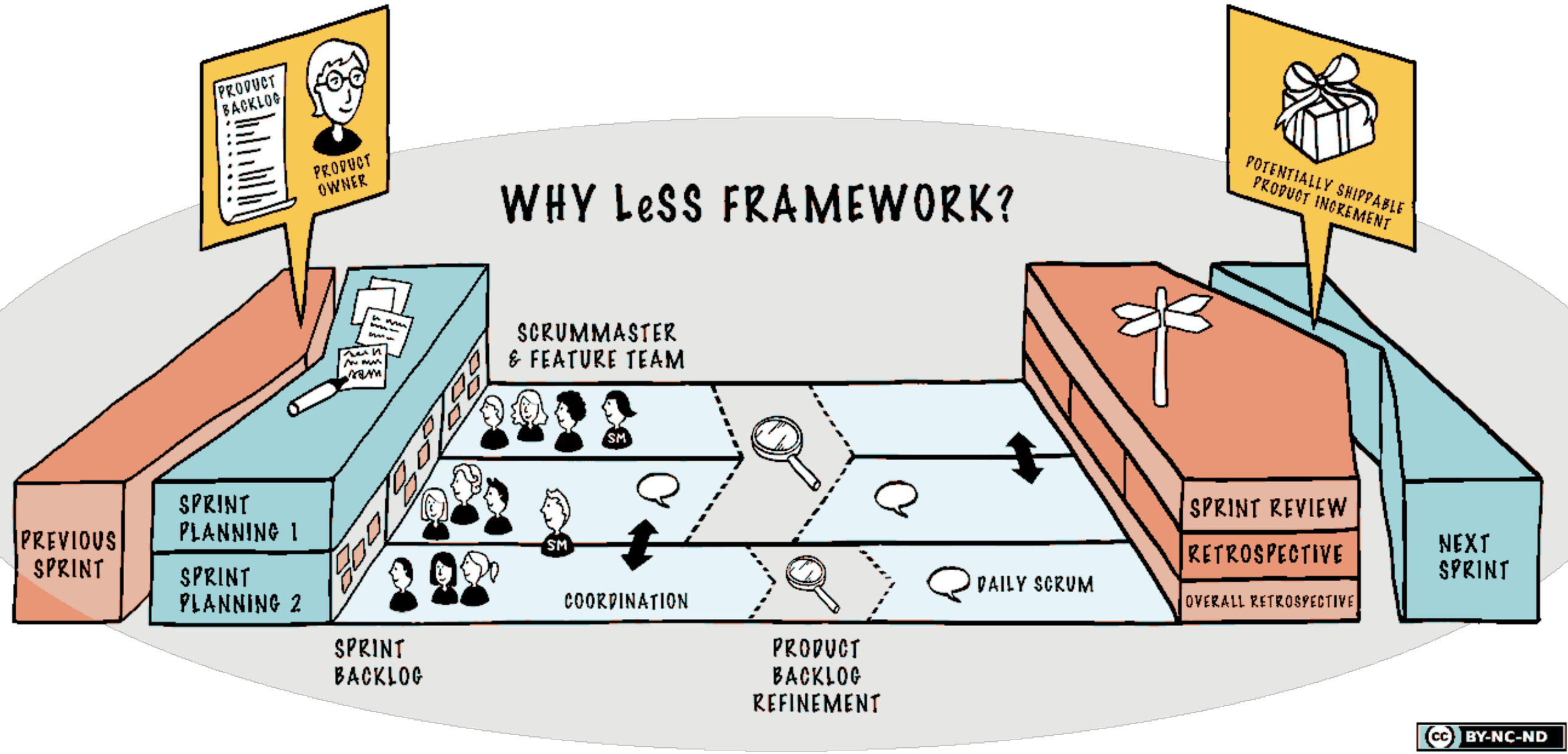
Guide: Multi-Team Design Workshop



coach: questions?

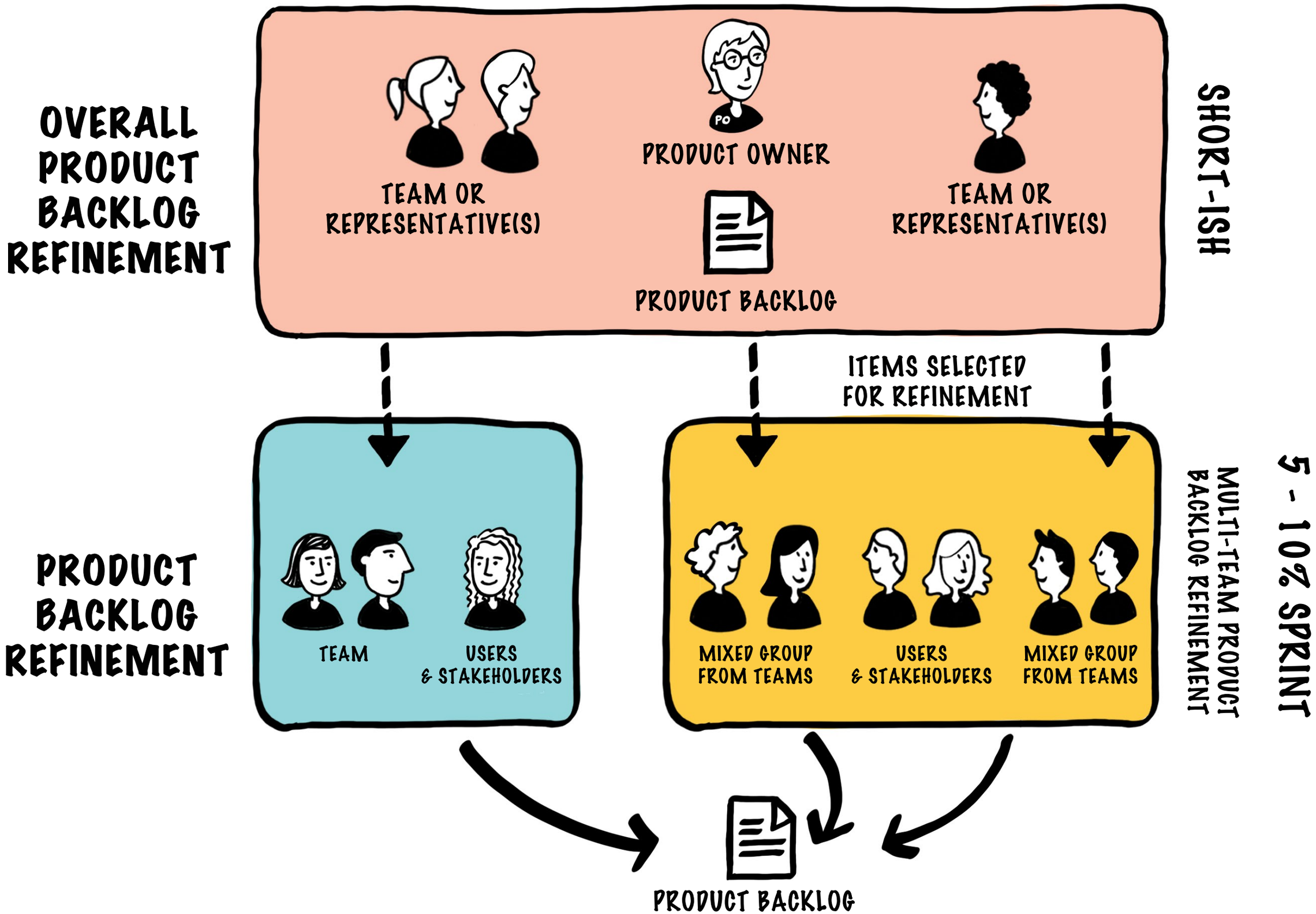

HOW TO FOCUS IN THE AGE OF DISTRACTION
TAKE A DIGITAL TECHNOLOGY DETOX
SAY NO TO...
No
MOBILE
EMAIL
FACEBOOK
TV
SAY YES TO...
Yes
CREATING
SPENDING TIME WITH FAMILY & FRIENDS
READING A BOOK
MANAGING YOUR SPACE
SPEND 10 MINS EACH DAY DECLUTTERING
USE AN INTRAY FOR INCOMING PAPER
CLEAR AWAY ALL DISTRACTIONS
IF READING ONLY HAVE THE BOOK IN FRONT OF YOU
DO 1 THING
HOW TO WORK
USE A TIMER
WORK IN FOCUSED BURSTS
50 MINUTES OF WORK 20 MINUTES BREAK
GET AWAY FROM THE COMPUTER
AREA WITH NO INTERNET
USE ONLY A NOTEPAD AND PEN
CREATE RITUALS/HABITS
MORNING
MORNING QUIET
MEDITATE
READ
WRITE MORNING PAGES
SHOWER
HAVE A HEALTHY BREAKFAST
GO FOR A WALK
BEDTIME
CLEAN ALL DISHES/ EMPTY YOUR SINK
READ BOOK
WASH FACE
STRETCH
REFOCUS
WALK
LOOK AT THINGS TO DO LIST
MANAGING EMAIL
DISCONNECTION TIMES
8am–10am
CHECK EMAIL AT SET POINTS IN THE DAY
2–3 TIMES PER DAY
TURN OFF EMAIL ALERTS
HELP FOR ADDICTS
Twitter
email
Facebook
BLOCKER PROGRAMS
FREEDOM
LOCKS YOU OUT OF THE INTERNET
SELF CONTROL
LOCKS YOU OUT OF SPECIFIC SITES
WRITING PROGRAMS
WRITE ROOM
TYPEWRITER
OMMWRITER
TAKE TIME TO REFLECT AND REVIEW
WHAT WORKED?
FIRST THING IN THE MORNING
WHAT ARE THE 3 MOST IMPORTANT THINGS TO DO TODAY?
DON'T CHECK YOUR EMAIL UNTIL YOU GET THE TASK COMPLETE
AT THE END OF THE DAY
1 WHAT DID I DO?
2 WHAT WILL I FOCUS ON TOMORROW?
3 WHAT CAN BE IMPROVED?
© Learning Fundamentals
www.learningfundamentals.com.au

LeSS Product Backlog Refinement


WHY LeSS FRAMEWORK?
PRODUCT BACKLOG
PRODUCT OWNER
POTENTIALLY SHIPPABLE PRODUCT INCREMENT
SCRUMMASTER & FEATURE TEAM
SM
SM
PREVIOUS SPRINT
SPRINT PLANNING 1
SPRINT PLANNING 2
COORDINATION
DAILY SCRUM
SPRINT REVIEW
RETROSPECTIVE
OVERALL RETROSPECTIVE
NEXT SPRINT
SPRINT BACKLOG
PRODUCT BACKLOG REFINEMENT

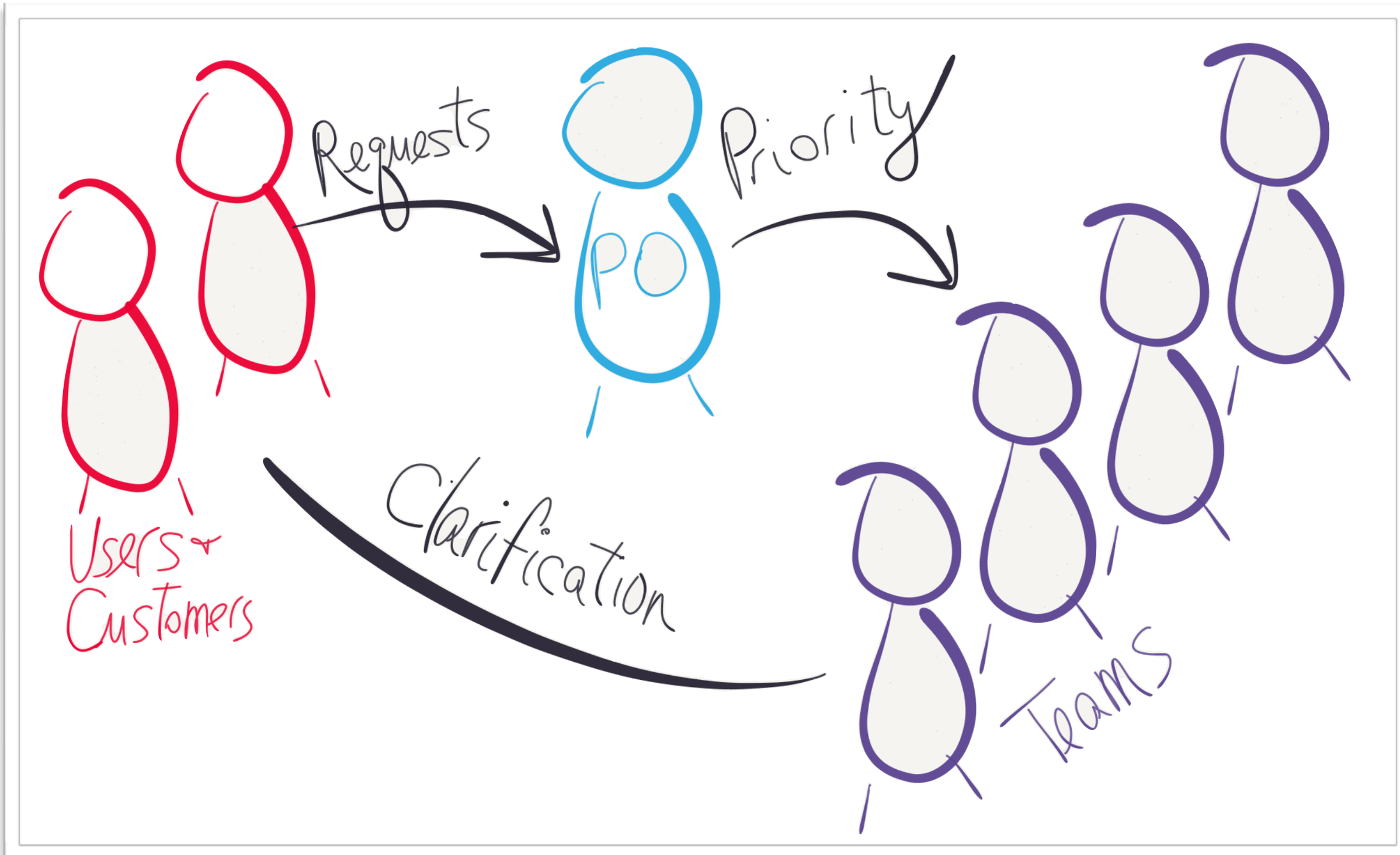
BY-NC-ND

LeSS PRODUCT BACKLOG REFINEMENT



Guide: Multi-Team PBR

Clarification vs Prioritisation



Teams emphasise learning **customer domains** (not just tech domains)

Scrum.org | Blog

MAIN SITE ABOUT US SUBSCRIBE

Insights from Scrum.org's community of experts

Who is the Professional Scrum Product Owner

Many Product Owners currently do not really have any ownership with the product they are managing. The label has changed but the behaviour hasn't. They aren't empowered to make decision for the product and they can't say NO to the stakeholders. Basically they're just a proxy for the real Product Owner. Or they're just focusing on requirements engineering, writing perfect requirements. They are still expected to write perfect User Stories. The way requirements are written has changed, but the perspective towards requirements hasn't.

Many people thought that Product Owner is just another name for a Business Analyst.

1. Product Owner is an entrepreneur
2. Product Owner is an innovator
3. Product Owner is a systems thinker
4. Product Owner is a Mini CEO

PO or PO team creating any specifications, documents, designs, mockups, wireframes, ... and handing them off to teams

Types of PBR

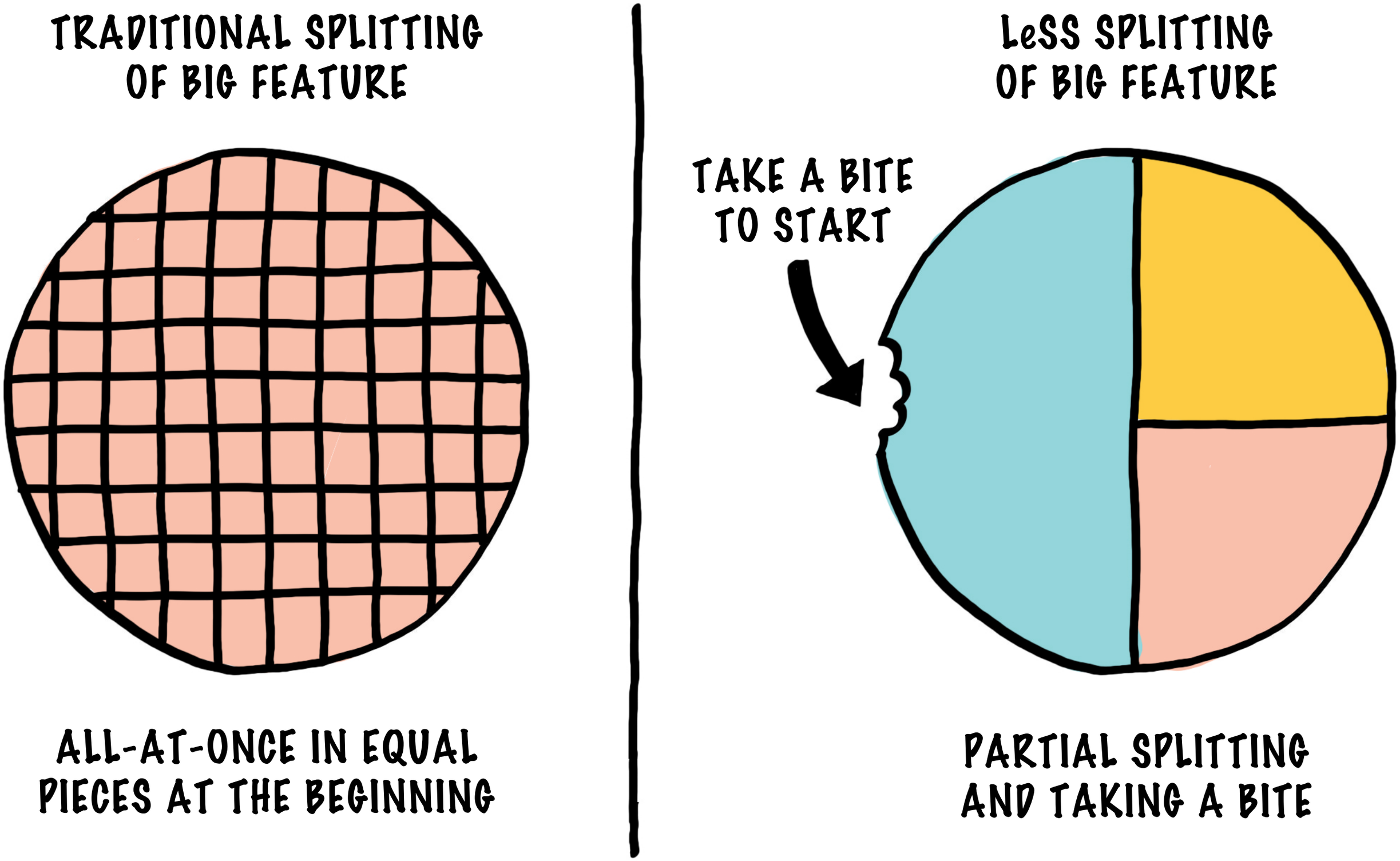
	Overall PBR	Multi-team PBR	Single-team PBR	Initial PBR
members from	all teams	2+ teams	1 team	all teams
includes Product Owner?	definitely	depends	rarely	definitely
includes customers/users?	rarely	probably	probably	definitely
select which teams work on which items?	yes (prefer *set of items with group of teams*)	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

Initial PBR is often the first time that **hidden problems** from prior separate specialist groups are surfaced…

and so can be "hard" ;)

Estimation synchronisation

Guide: Take a Bite

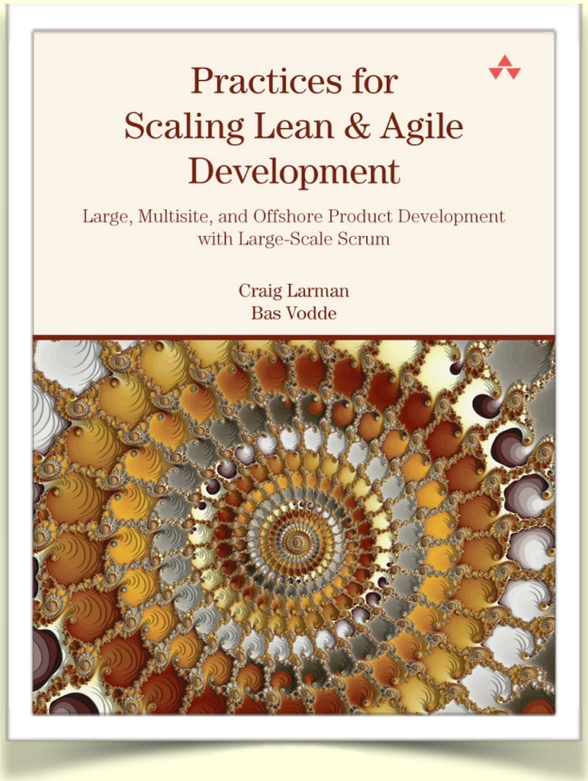


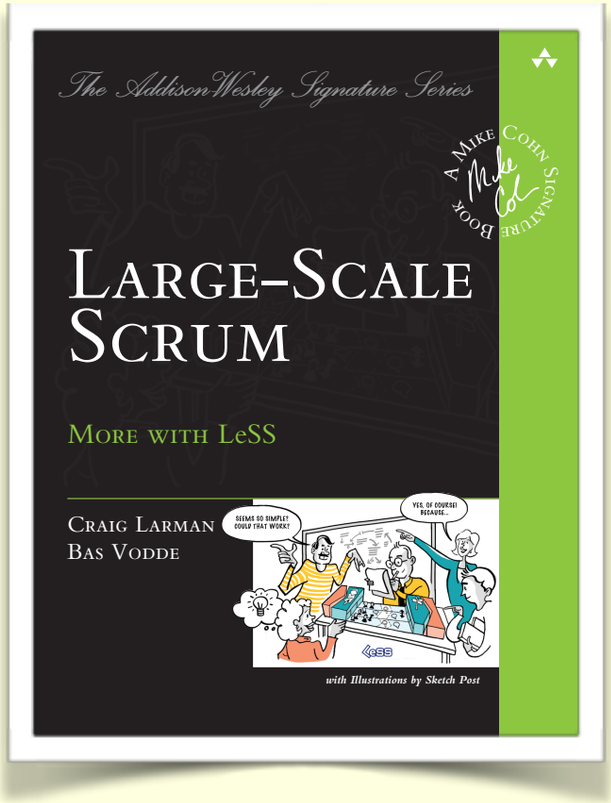
Guide: Splitting

Use cases	the major work flows or use cases
Scenario	a specific sequence of steps
Data part	subset of the data elements
Type	Varying types of kinds of things
Integration	integration between existing (or non-existing) elements
Configuration	varying configurations, e.g., OS or browser

I/O channel	different interfaces, e.g., GUI or command line
Data format	XML, …
Role or persona	e.g., novice or power user
Non-functional	e.g., moderate vs high throughput
Operation	system operation, e.g., HTTP GET
Stub	working with a fake first

individual:



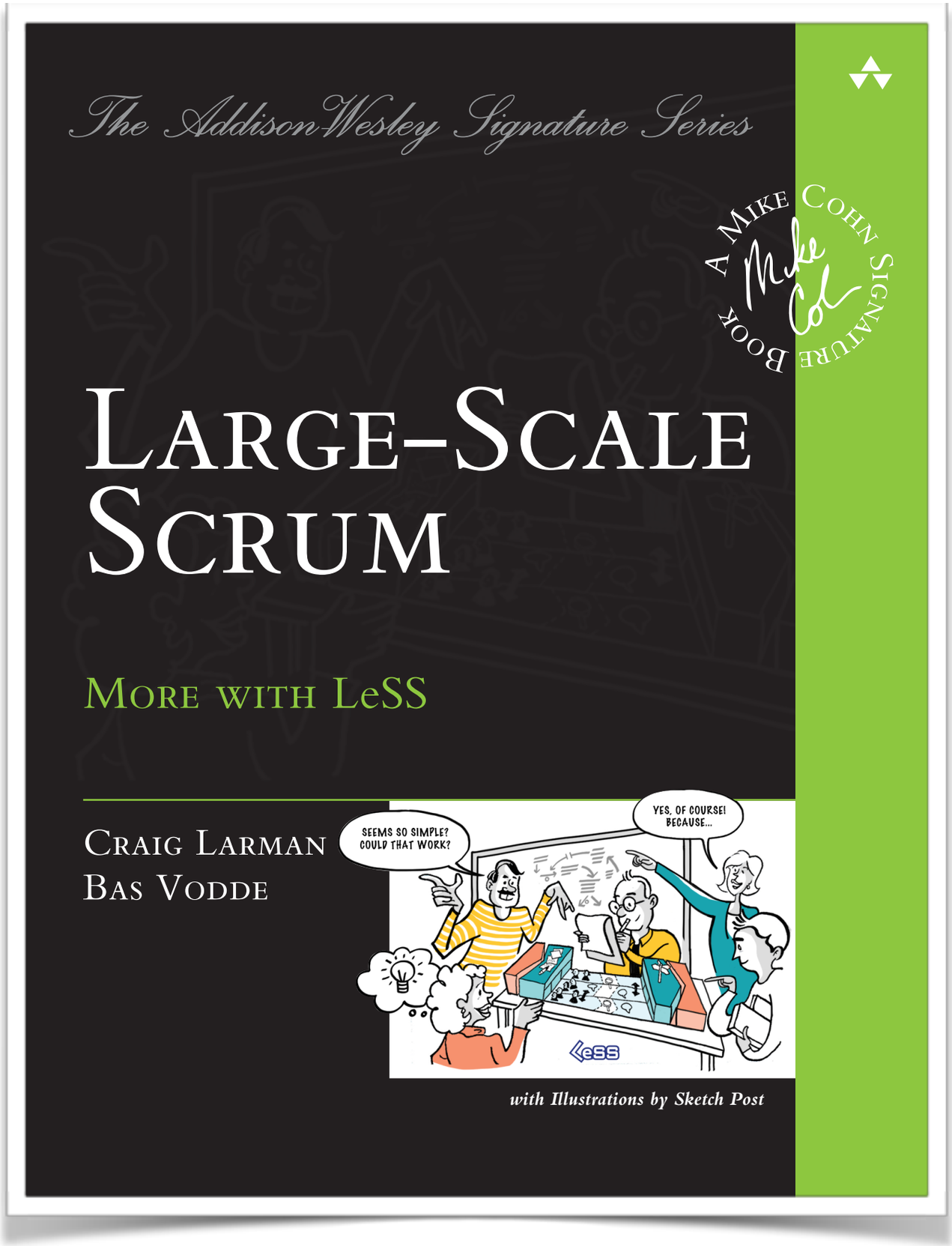


> scan the section on splitting big items

(optional) coach:

> find a candidate giant item

> demonstrate splitting it



1 More with LeSS 1
2 LeSS 5

LeSS Structure

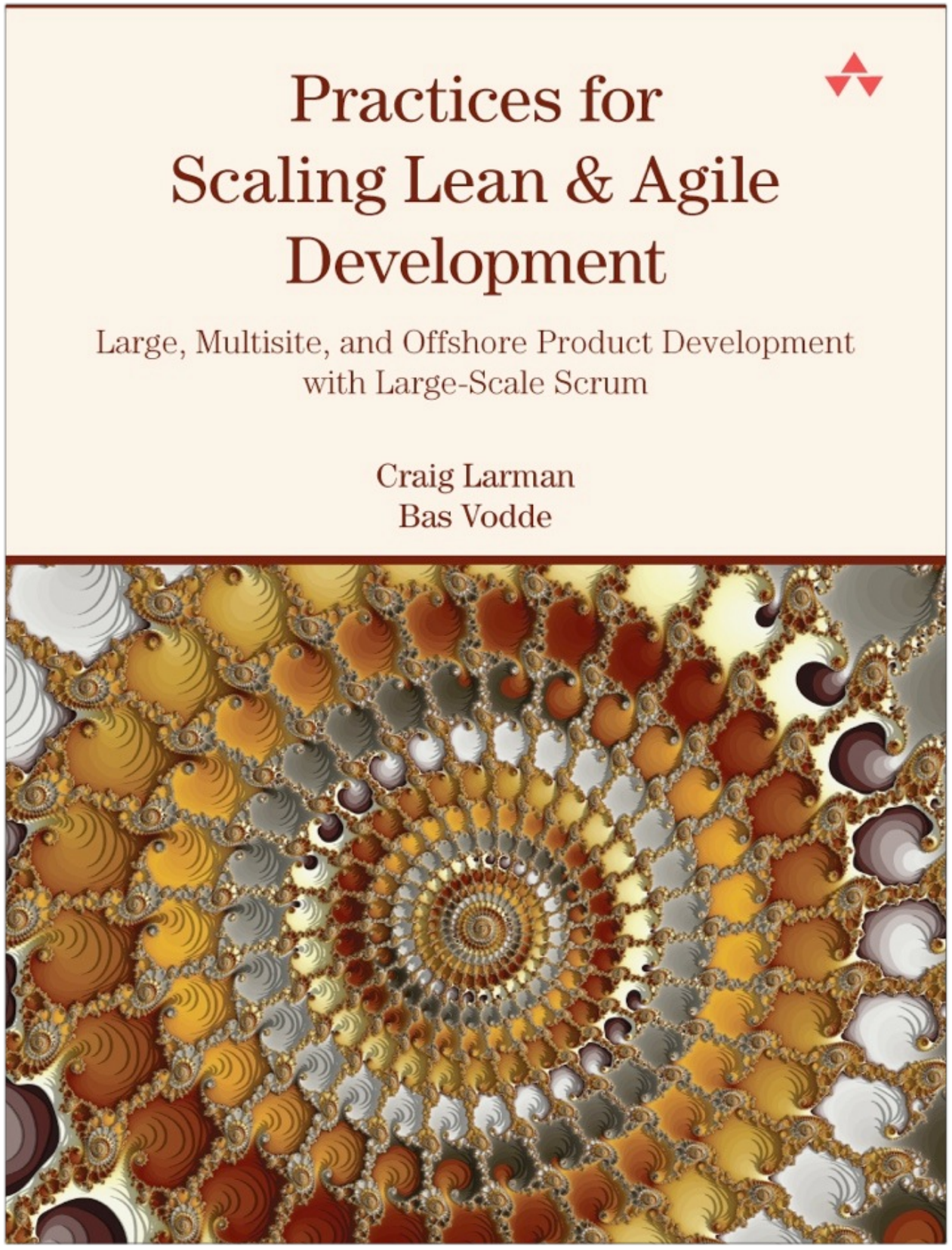
3 Adoption 41
4 Organize by Customer Value 67
5 Management 111
6 ScrumMasters 133

LeSS Product

7 Product 155
8 Product Owner 171
9 Product Backlog 197
10 Definition of Done 231

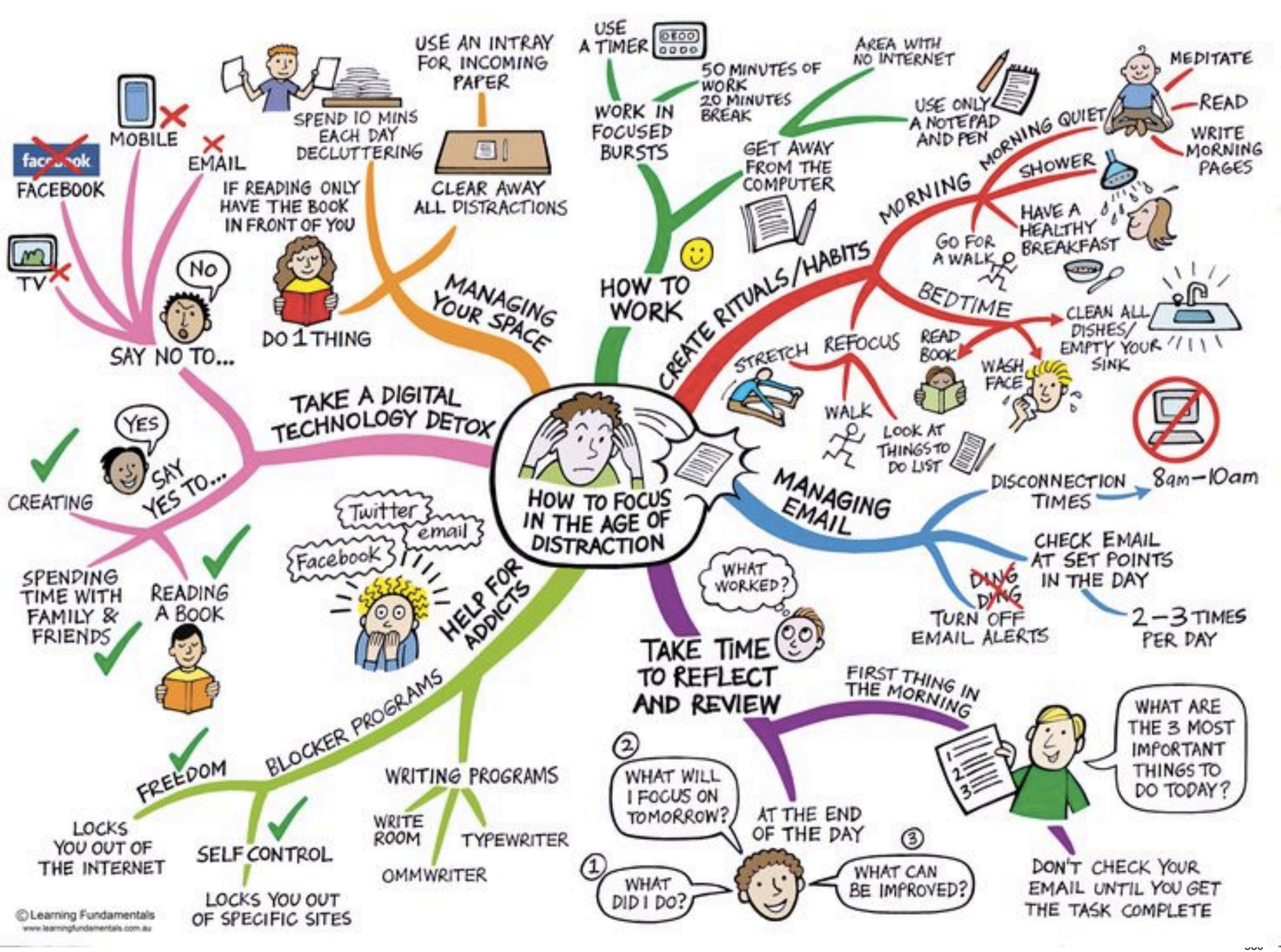
LeSS Sprint

11 Product Backlog Refinement 249
12 Sprint Planning 275
13 Coordination and Integration 285
14 Review & Retrospective 313



Action Tools

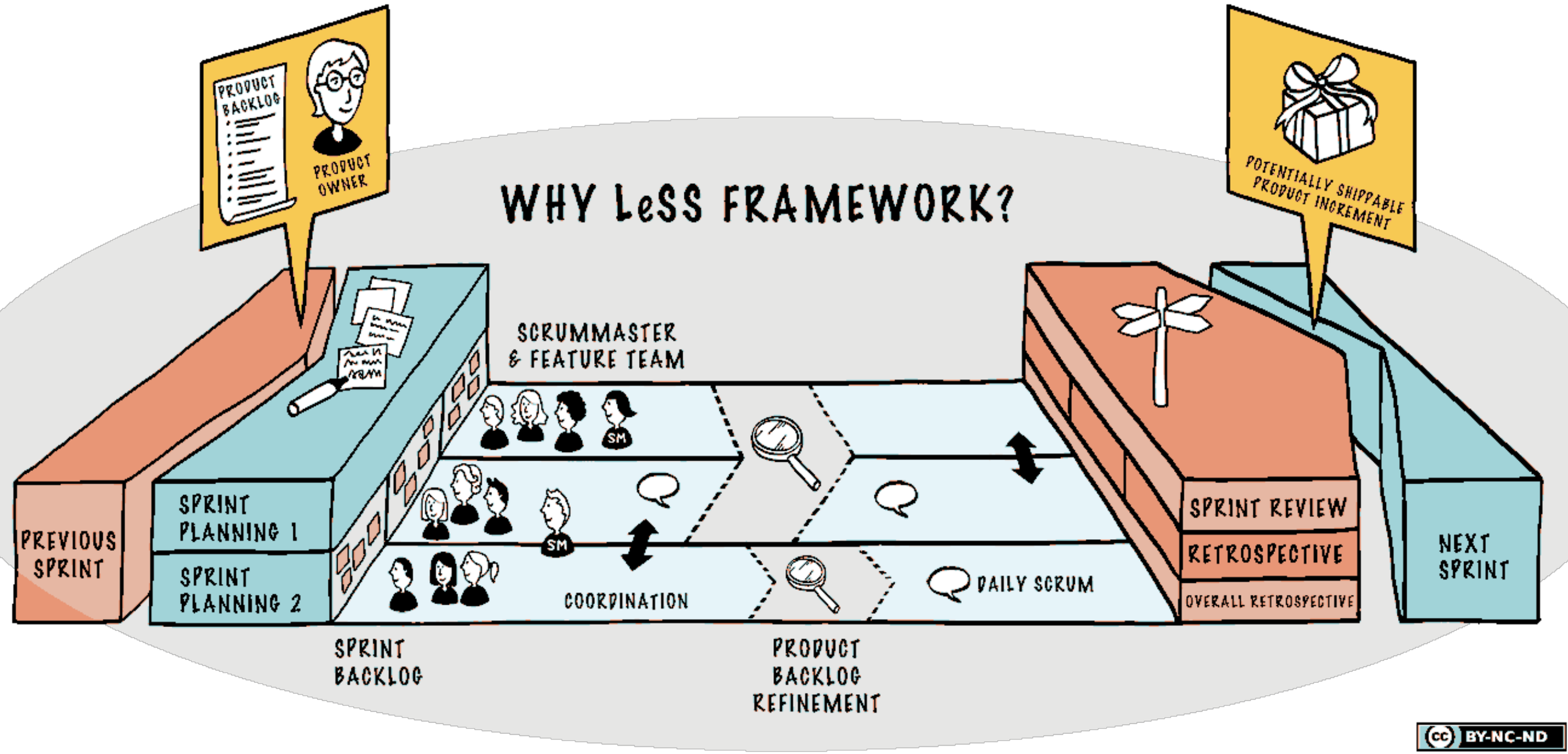
2. Large-Scale Scrum
3. Test
4. Product Management
5. Planning
6. Coordination
8. Requirements
9. Design & Architecture
10. Legacy Code
11. Continuous Integration
12. Inspect & Adapt
13. Multisite
14. Offshore
15. Contracts


HOW TO FOCUS IN THE AGE OF DISTRACTION
TAKE A DIGITAL TECHNOLOGY DETOX
SAY NO TO...
No
MOBILE
EMAIL
FACEBOOK
TV
CREATING
SAY YES TO...
Yes
SPENDING TIME WITH FAMILY & FRIENDS
READING A BOOK
MANAGING YOUR SPACE
SPEND 10 MINS EACH DAY DECLUTTERING
USE AN INTRAY FOR INCOMING PAPER
CLEAR AWAY ALL DISTRACTIONS
IF READING ONLY HAVE THE BOOK IN FRONT OF YOU
DO 1 THING
HOW TO WORK
USE A TIMER
WORK IN FOCUSED BURSTS
50 MINUTES OF WORK 20 MINUTES BREAK
GET AWAY FROM THE COMPUTER
AREA WITH NO INTERNET
USE ONLY A NOTEPAD AND PEN
CREATE RITUALS/HABITS
MORNING
MORNING QUIET
MEDITATE
READ
WRITE MORNING PAGES
SHOWER
HAVE A HEALTHY BREAKFAST
GO FOR A WALK
BEDTIME
CLEAN ALL DISHES/ EMPTY YOUR SINK
READ BOOK
WASH FACE
STRETCH
REFOCUS
WALK
LOOK AT THINGS TO DO LIST
MANAGING EMAIL
DISCONNECTION TIMES
8am–10am
CHECK EMAIL AT SET POINTS IN THE DAY
2–3 TIMES PER DAY
TURN OFF EMAIL ALERTS
DING
DING
HELP FOR ADDICTS
Twitter
email
Facebook
BLOCKER PROGRAMS
FREEDOM
LOCKS YOU OUT OF THE INTERNET
SELF CONTROL
LOCKS YOU OUT OF SPECIFIC SITES
WRITING PROGRAMS
WRITE ROOM
TYPEWRITER
OMMWRITER
TAKE TIME TO REFLECT AND REVIEW
WHAT WORKED?
FIRST THING IN THE MORNING
WHAT ARE THE 3 MOST IMPORTANT THINGS TO DO TODAY?
DON'T CHECK YOUR EMAIL UNTIL YOU GET THE TASK COMPLETE
AT THE END OF THE DAY
1
WHAT DID I DO?
2
WHAT WILL I FOCUS ON TOMORROW?
3
WHAT CAN BE IMPROVED?
© Learning Fundamentals
www.learningfundamentals.com.au

LeSS Sprint Planning

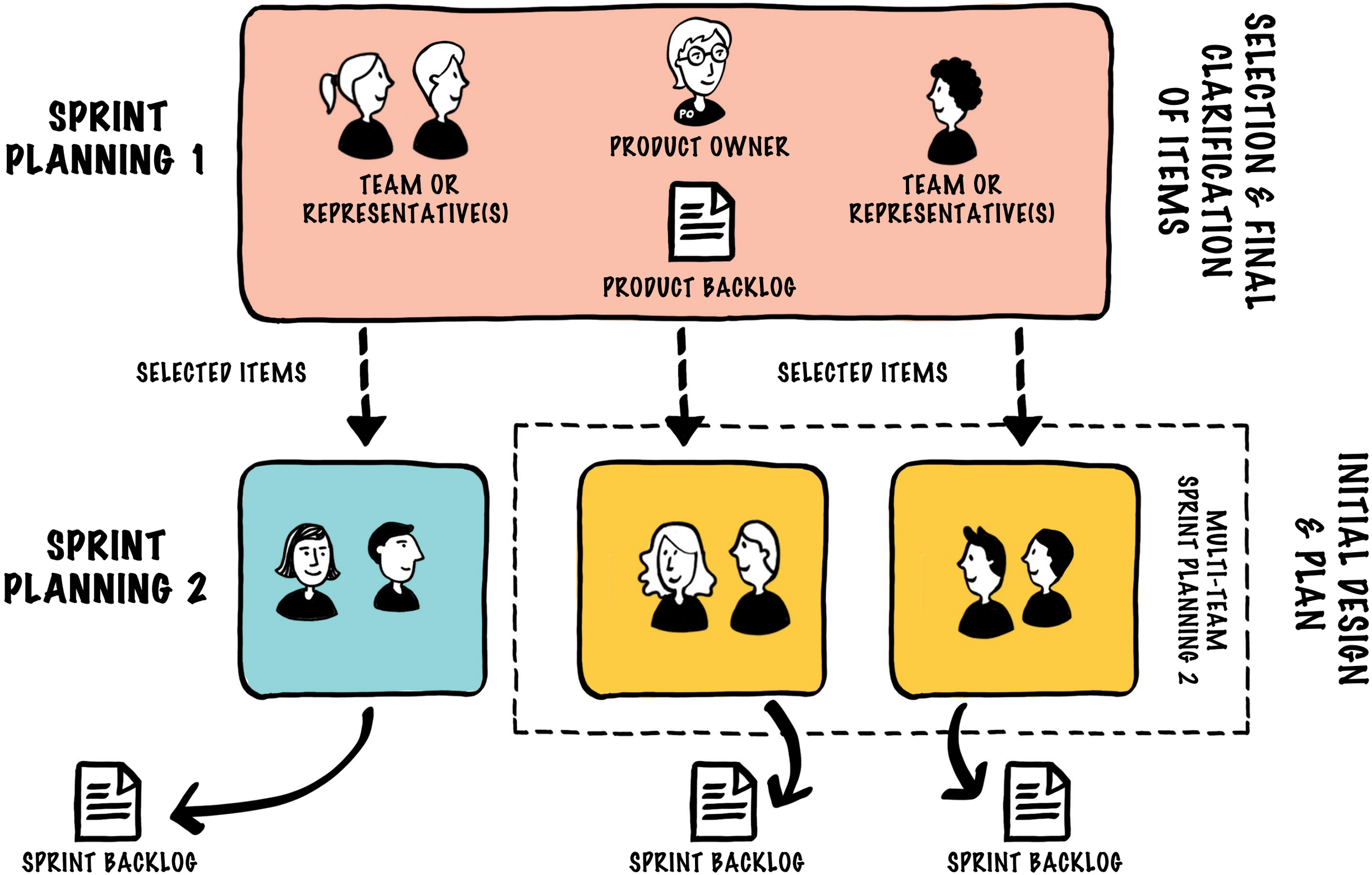
Sprint Planning & LeSS Principles

- whole-product focus
- empirical process control & continuous improvement
- more with less
  - dependency management? complex coordination?


WHY LeSS FRAMEWORK?
PRODUCT BACKLOG
PRODUCT OWNER
POTENTIALLY SHIPPABLE PRODUCT INCREMENT
SCRUMMASTER & FEATURE TEAM
SM
SM
PREVIOUS SPRINT
SPRINT PLANNING 1
SPRINT PLANNING 2
COORDINATION
DAILY SCRUM
SPRINT REVIEW
RETROSPECTIVE
OVERALL RETROSPECTIVE
NEXT SPRINT
SPRINT BACKLOG
PRODUCT BACKLOG REFINEMENT

BY-NC-ND

LeSS SPRINT PLANNING



LeSS Rule(s)

Sprint Planning consists of two parts: Sprint Planning One is common for all teams; 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 during the next 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.

LeSS is Scrum…

time boxes enforced

it is **'illegal'** in Scrum for a meeting **to go longer**

Guide: Sprint Planning One

class

> are there **task dependencies between teams**, in 1 product?

> in Sprint Planning, do people need to analyse and plan for "dependency management"?

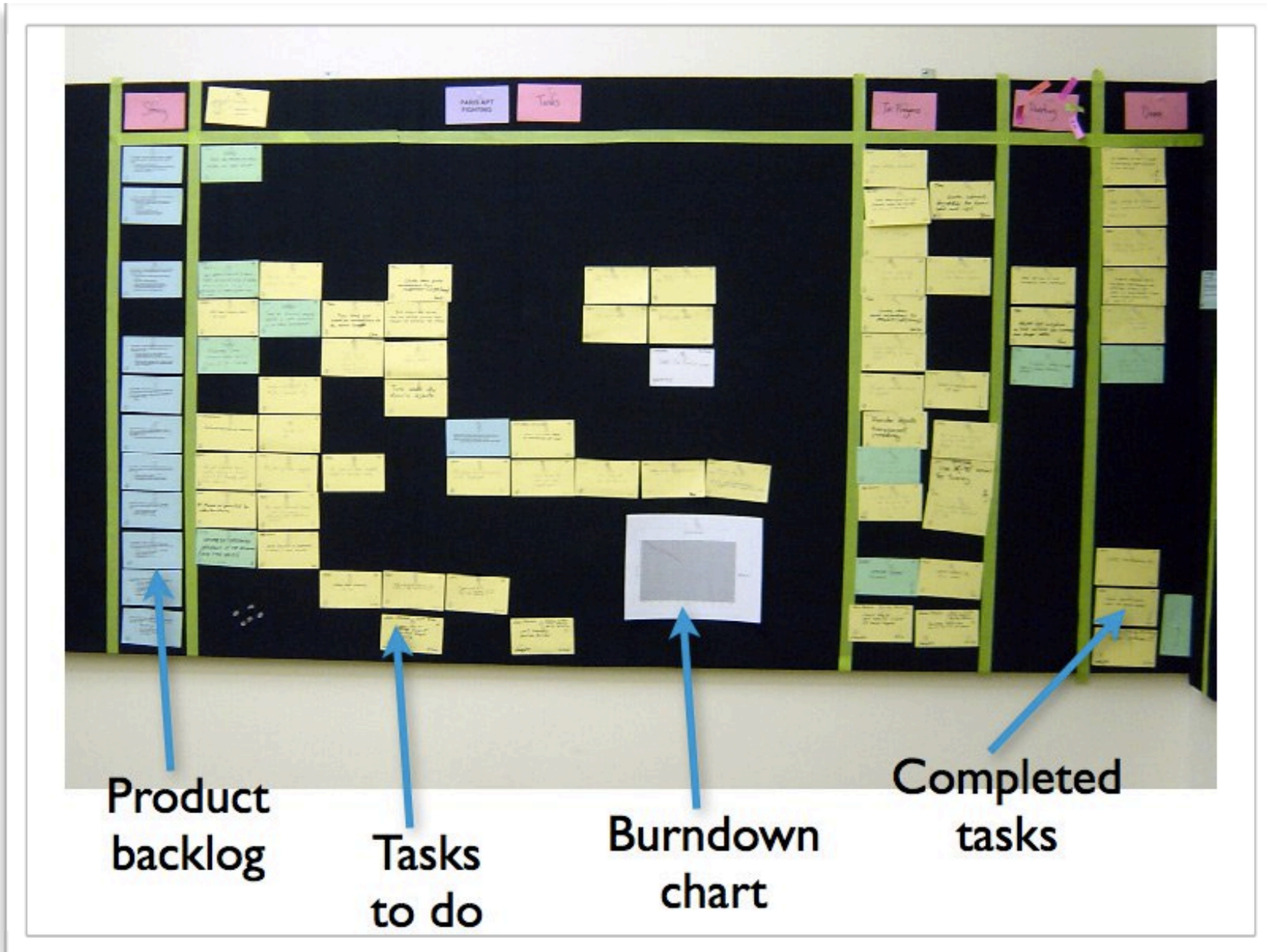
Task dependencies between teams? None exist in a LeSS group

->

shared work, opportunities to work together

Sprint Planning Two

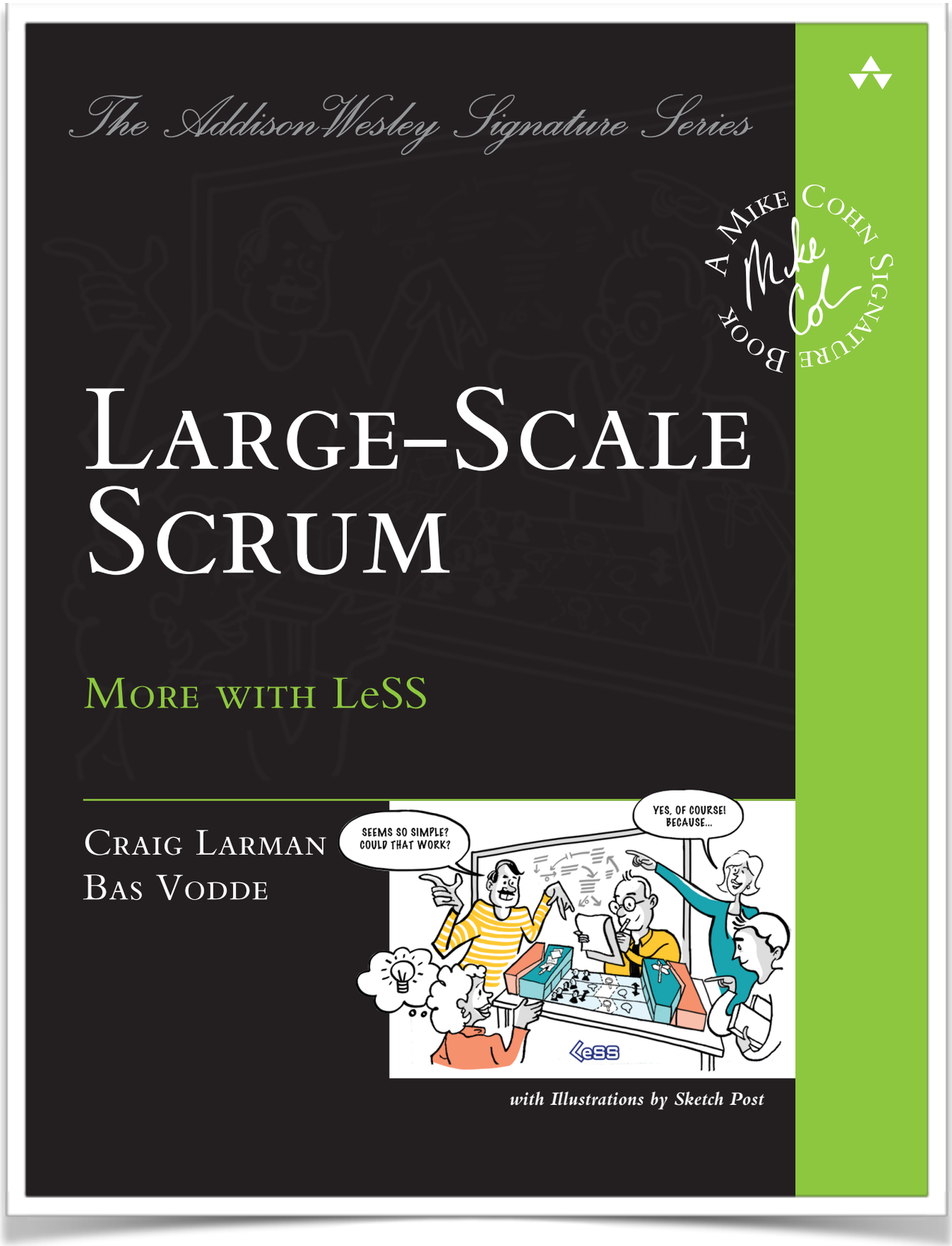
Guide: No Software Tools for Sprint Backlog



just…

cards on a wall

post-Sprint Planning recap meeting



1 More with LeSS 1
2 LeSS 5

LeSS Structure

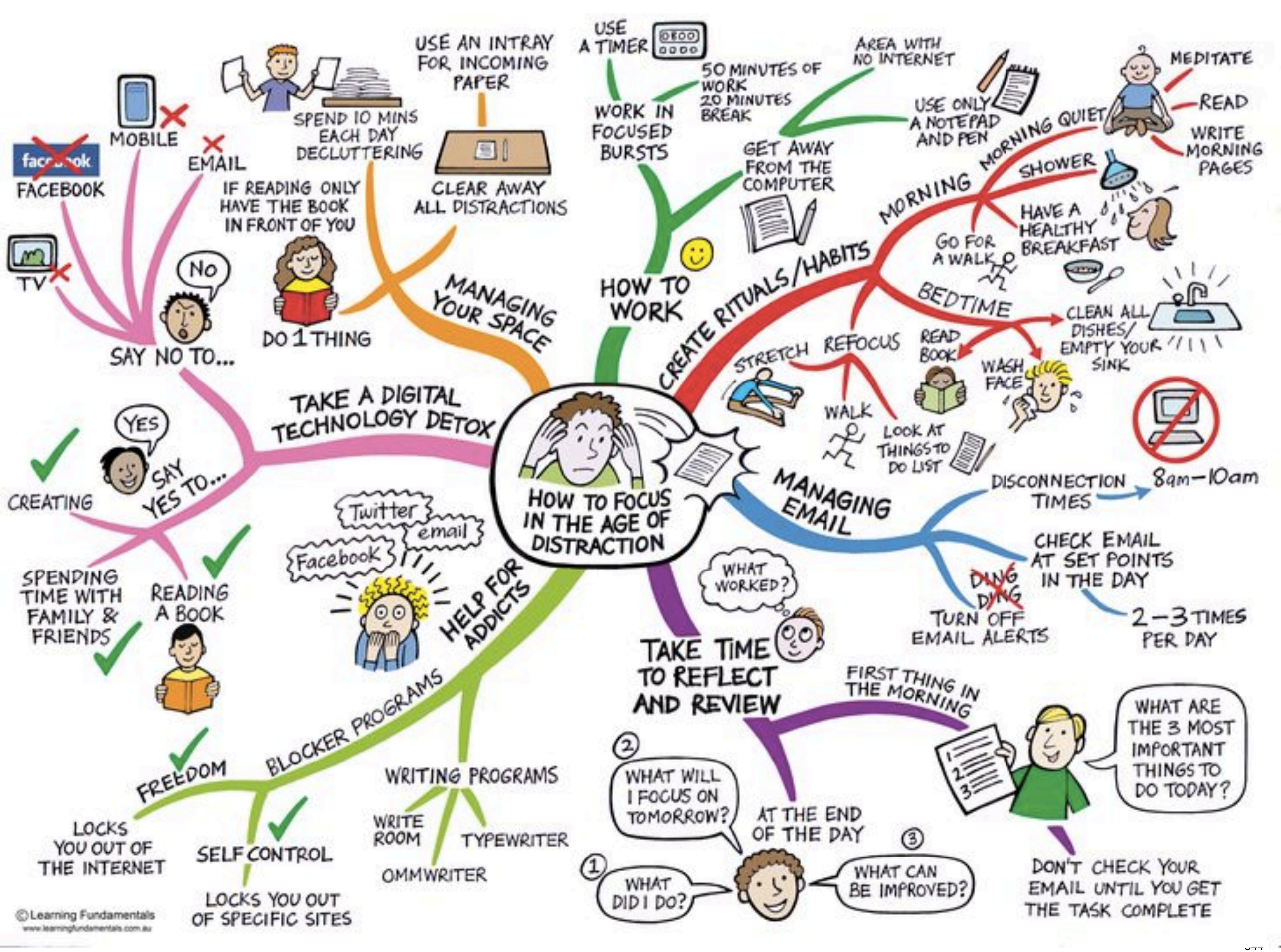
3 Adoption 41
4 Organize by Customer Value 67
5 Management 111
6 ScrumMasters 133

LeSS Product

7 Product 155
8 Product Owner 171
9 Product Backlog 197
10 Definition of Done 231

LeSS Sprint

11 Product Backlog Refinement 249
12 Sprint Planning 275
13 Coordination and Integration 285
14 Review & Retrospective 313


HOW TO FOCUS IN THE AGE OF DISTRACTION
TAKE A DIGITAL TECHNOLOGY DETOX
SAY NO TO...
No
MOBILE
EMAIL
FACEBOOK
TV
SPENDING TIME WITH FAMILY & FRIENDS
CREATING
SAY YES TO...
Yes
READING A BOOK
MANAGING YOUR SPACE
SPEND 10 MINS EACH DAY DECLUTTERING
USE AN INTRAY FOR INCOMING PAPER
CLEAR AWAY ALL DISTRACTIONS
IF READING ONLY HAVE THE BOOK IN FRONT OF YOU
DO 1 THING
HOW TO WORK
USE A TIMER
WORK IN FOCUSED BURSTS
50 MINUTES OF WORK 20 MINUTES BREAK
GET AWAY FROM THE COMPUTER
AREA WITH NO INTERNET
USE ONLY A NOTEPAD AND PEN
CREATE RITUALS/HABITS
MORNING
MORNING QUIET
MEDITATE
READ
WRITE MORNING PAGES
SHOWER
HAVE A HEALTHY BREAKFAST
GO FOR A WALK
BEDTIME
CLEAN ALL DISHES/ EMPTY YOUR SINK
READ BOOK
WASH FACE
STRETCH
REFOCUS
WALK
LOOK AT THINGS TO DO LIST
MANAGING EMAIL
DISCONNECTION TIMES
8am–10am
CHECK EMAIL AT SET POINTS IN THE DAY
2–3 TIMES PER DAY
TURN OFF EMAIL ALERTS
DING
DING
HELP FOR ADDICTS
Twitter
email
Facebook
BLOCKER PROGRAMS
FREEDOM
LOCKS YOU OUT OF THE INTERNET
SELF CONTROL
LOCKS YOU OUT OF SPECIFIC SITES
WRITING PROGRAMS
WRITE ROOM
TYPEWRITER
OMMWRITER
TAKE TIME TO REFLECT AND REVIEW
WHAT WORKED?
FIRST THING IN THE MORNING
WHAT ARE THE 3 MOST IMPORTANT THINGS TO DO TODAY?
DON'T CHECK YOUR EMAIL UNTIL YOU GET THE TASK COMPLETE
AT THE END OF THE DAY
1
WHAT DID I DO?
2
WHAT WILL I FOCUS ON TOMORROW?
3
WHAT CAN BE IMPROVED?
© Learning Fundamentals
www.learningfundamentals.com.au

Coordination & Integration

coach & group:

> "It's Tuesday 2pm. I (a Developer) see a coordination problem. The Scrum of Scrums meeting is tomorrow at 11 am."

therefore…

LeSS Rule(s)

Prefer **decentralised** and **informal** coordination over centralised coordination.

LeSS Rule(s)

Cross-team coordination is decided by the teams.

We Observe…

> coordination techniques need to be especially **situational** and **customisable** in large groups

> "one size fits all" — not advised

the
most **advanced**
coordination technique
in LeSS?...

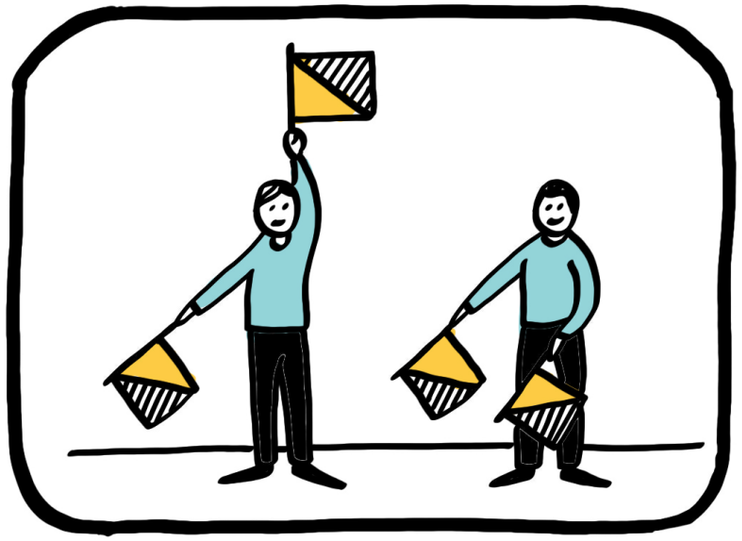
Guide: Just Talk (for Sprint delivery)

The problem with large-scale coordination isn't *what* coordination technique to use, but knowing there's a ***need to coordinate***, and ***who*** to talk with.

how to solve
“when” & **“who”**?

Guide: Communicate in Code

use **the code** to tell you **when** there's a **need** to coordinate, and **who** to talk with



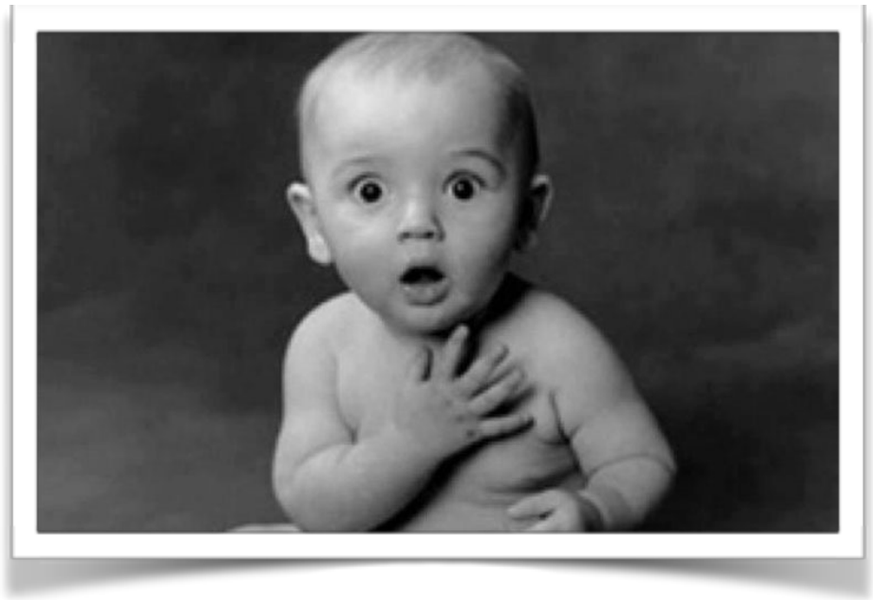
COMMUNICATE IN CODE

Guide: Communicate in Code. **How**?

> "social coding" tools such as GitHub or GitLab

> plugins that tell you who worked on the code, and initiates chats

> and most important, integrate continuously…



the surprising

meaning of

continuous integration

continuous integration

means to

integrate continuously

it's a developer behaviour

continuous integration

means to

have a **build server**

therefore…

Guide: Integrate Continuously

> use **continuous integration of the code** to tell you **when** there's a **need** to coordinate, and **who** to talk with

barriers to integration

are

barriers to communication

are

barriers to coordination

Guide: Communities

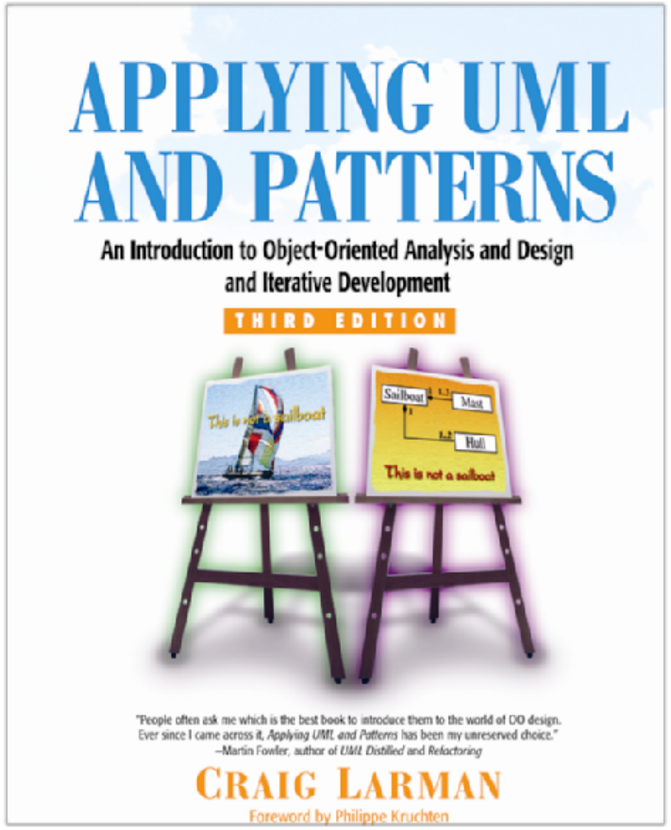
> people from different teams participate for a cross-team concern, e.g. Architecture

> *(see next section)*

Guide: Multi-Team Design Workshop

… with agile modelling

Guide: Current-Architecture Workshop



Chapter 39: Documenting Architecture

… Videos, …

Guide: Component Mentors

> regular feature-team member

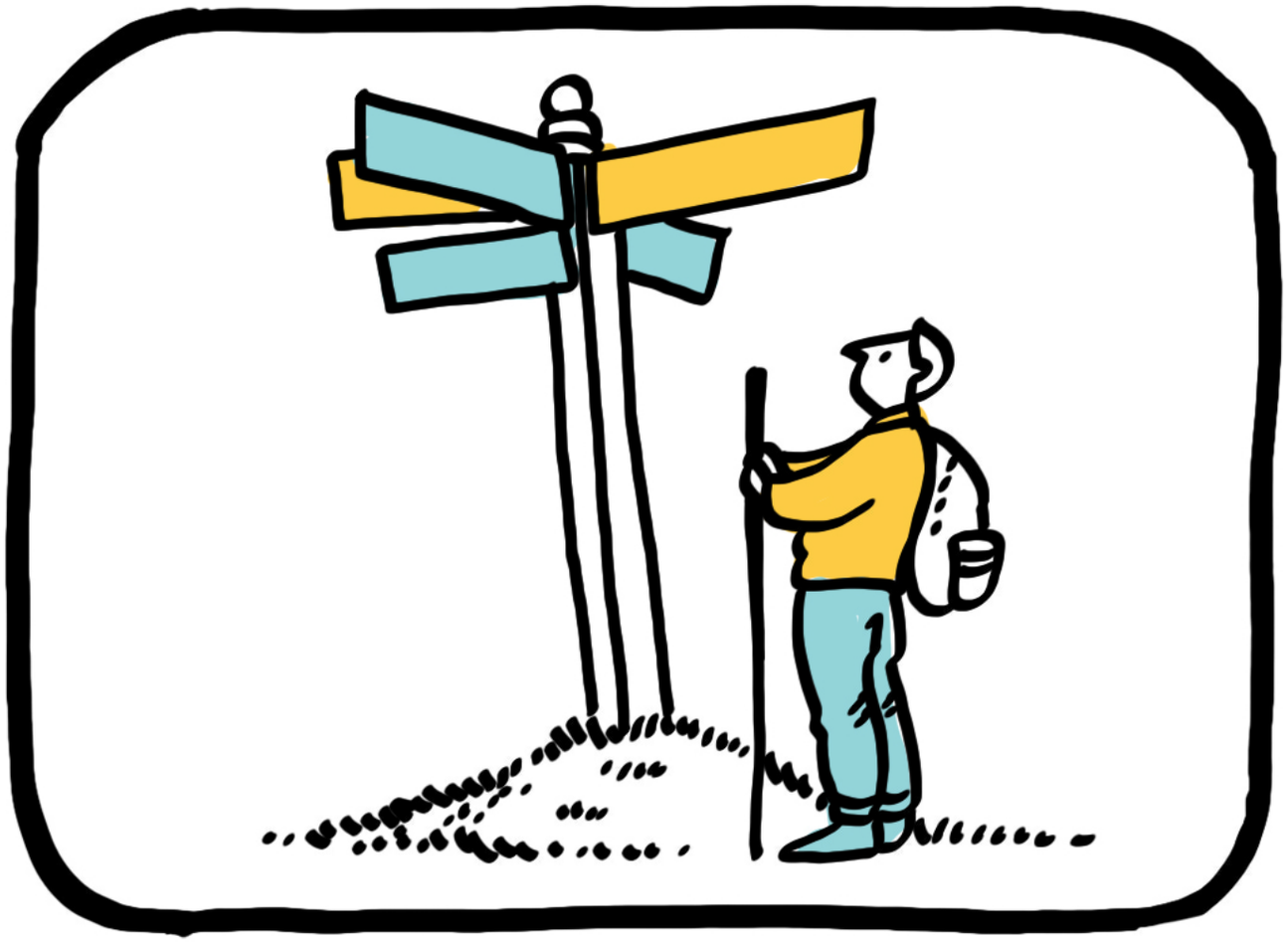
> does NOT approve other's code commits; is not a "committer gate"

COMPONENT MENTOR

code reviews
quality control

barriers to integration
are barriers to
coordination

Guide: Traveler



TRAVELER

Guide: Maybe Don't do Scrum of Scrums

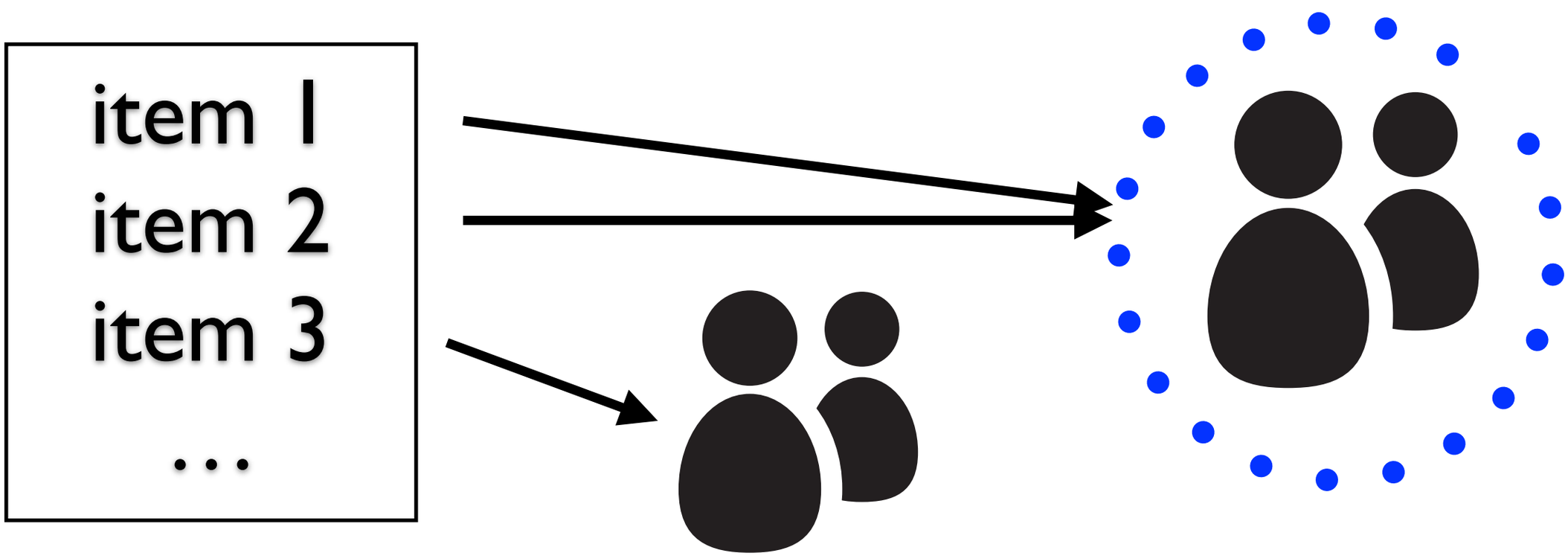
Guide: Leading Team

> start work on a **big** item (or family of items)

> start solo

> as more teams join, they educate them

> not “discovery team” or “designer team” that “discovers” and then hands off to real teams



Guide: Leading Team

> coordinating with external parties, especially related to integration & specification

Guide: Open Space

Experiment:
Rotate Infrastructure Tasks Across Feature Teams

- build system, etc.
- slowly rotate

- NB: no separate specialist infrastructure/tools groups

Experiment: Rotate “Surprises” Across Feature Teams

Key Elements of "**DevOps**"

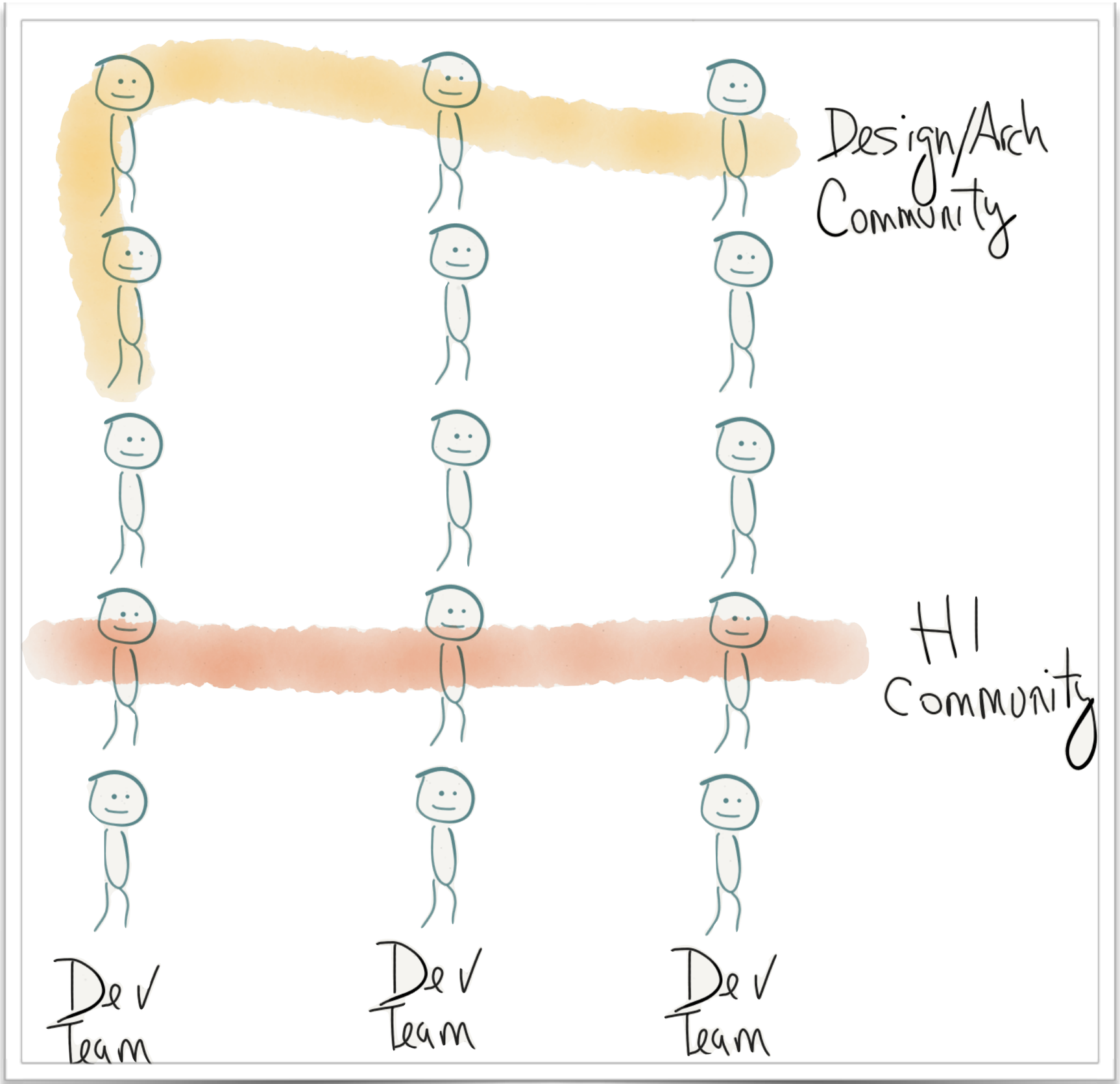
- Rotate Infrastructure Tasks Across Feature Teams
- Rotate "Surprises" Across Feature Teams

Guides in LeSS: Coordination

1. **Just Talk**
2. **Communicate in Code**
3. **Integrate Continuously**
4. **Communities**
5. **Multi-Team Design Workshops**
6. **Current-Architecture Workshops**
7. **Component Mentor**
8. **Travelers**
9. **Maybe Don't Do SoS**
10. **Leading Team**
11. **Open Space**
12. Scouts
13. Cross-Team Meetings
14. Mix & Match

Communities

Communities (for Coordination)



Communities? For…

- learning
- speculative solutions
- cross-team agreements
  - they can't make decisions for teams, but can make proposals that teams decide to adopt

communities do **not** do “the work”

Tips for Good Communities

- have a **community coordinator** with passion for the concern and desire to cultivate a strong community
  - is an **active hands-on practitioner**
- actively try to **recruit** participation from most teams
- focus on **concrete** problem-solving goal
- has **agreed how** they work and make decisions
- might have a **Scrum Master** who helps it work
- are strongly **encouraged** within the organization

How to **Kill** Communities

- there is **no or bad** community coordinator
- holds frequent meetings just for the sake of it; **blah blah** meetings
- holds meetings “at lunch”
- has members that are **not in** feature teams
- are considered **secondary** and participation is downgraded because “we’re too busy to participate”

beware **fake**

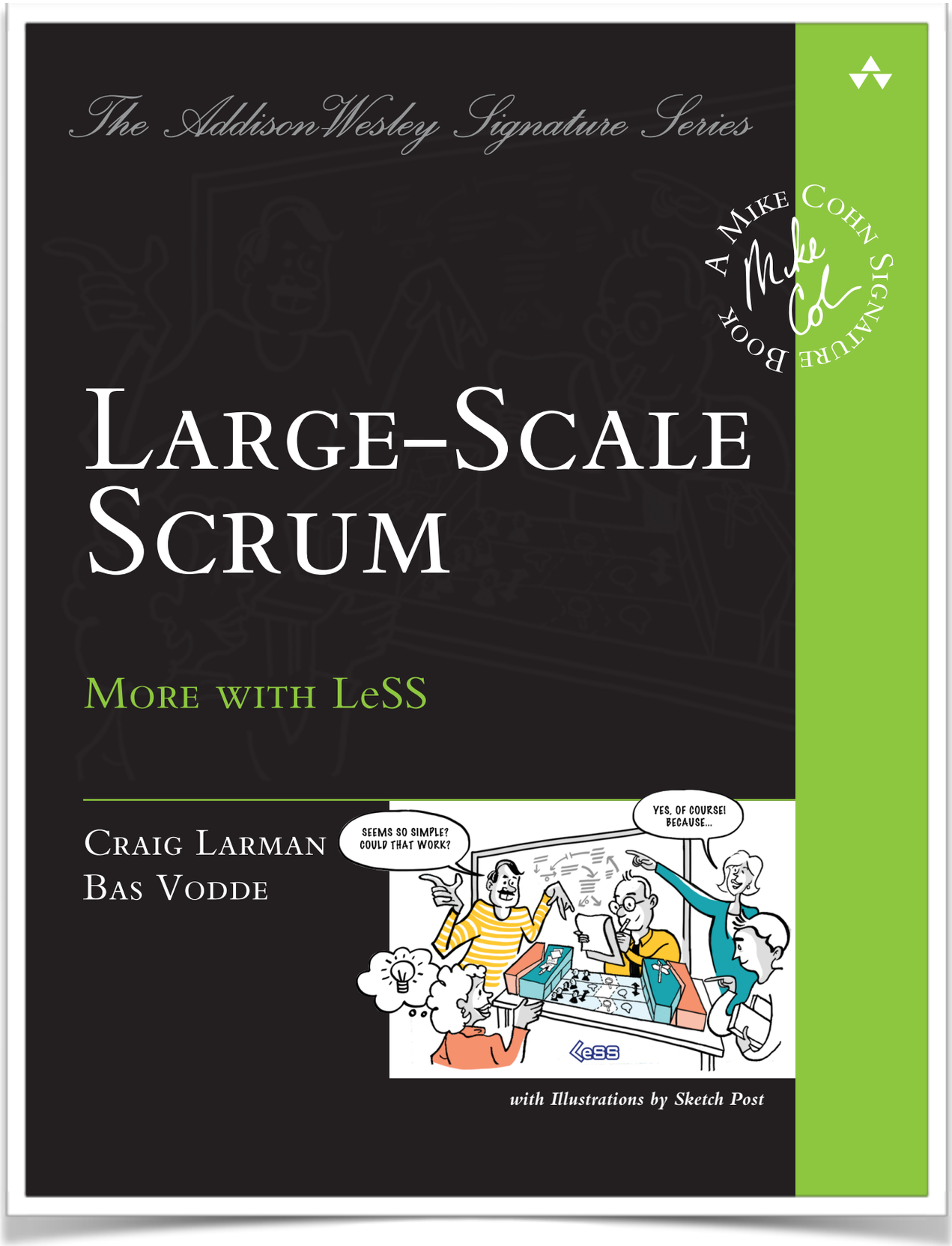
communities!

Minimal Recommended Communities?

- **UX & HI**
- **Architecture**
- **Test**
- **Security**
- sometimes: **Safety**

Variants

- cross-product communities
- site communities



1 More with LeSS 1
2 LeSS 5

LeSS Structure

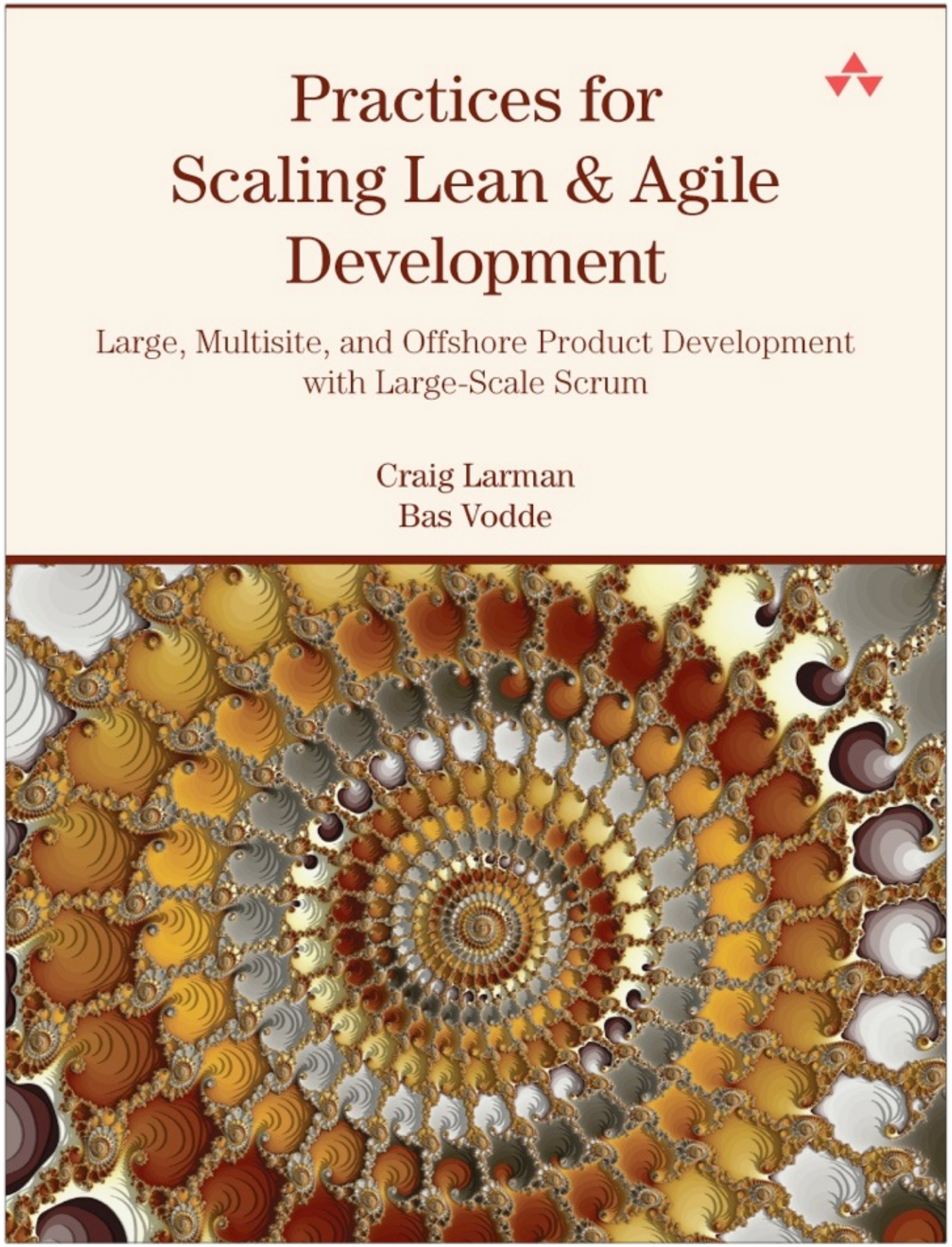
3 Adoption 41
4 Organize by Customer Value 67
5 Management 111
6 ScrumMasters 133

LeSS Product

7 Product 155
8 Product Owner 171
9 Product Backlog 197
10 Definition of Done 231

LeSS Sprint

11 Product Backlog Refinement 249
12 Sprint Planning 275
13 Coordination and Integration 285
14 Review & Retrospective 313

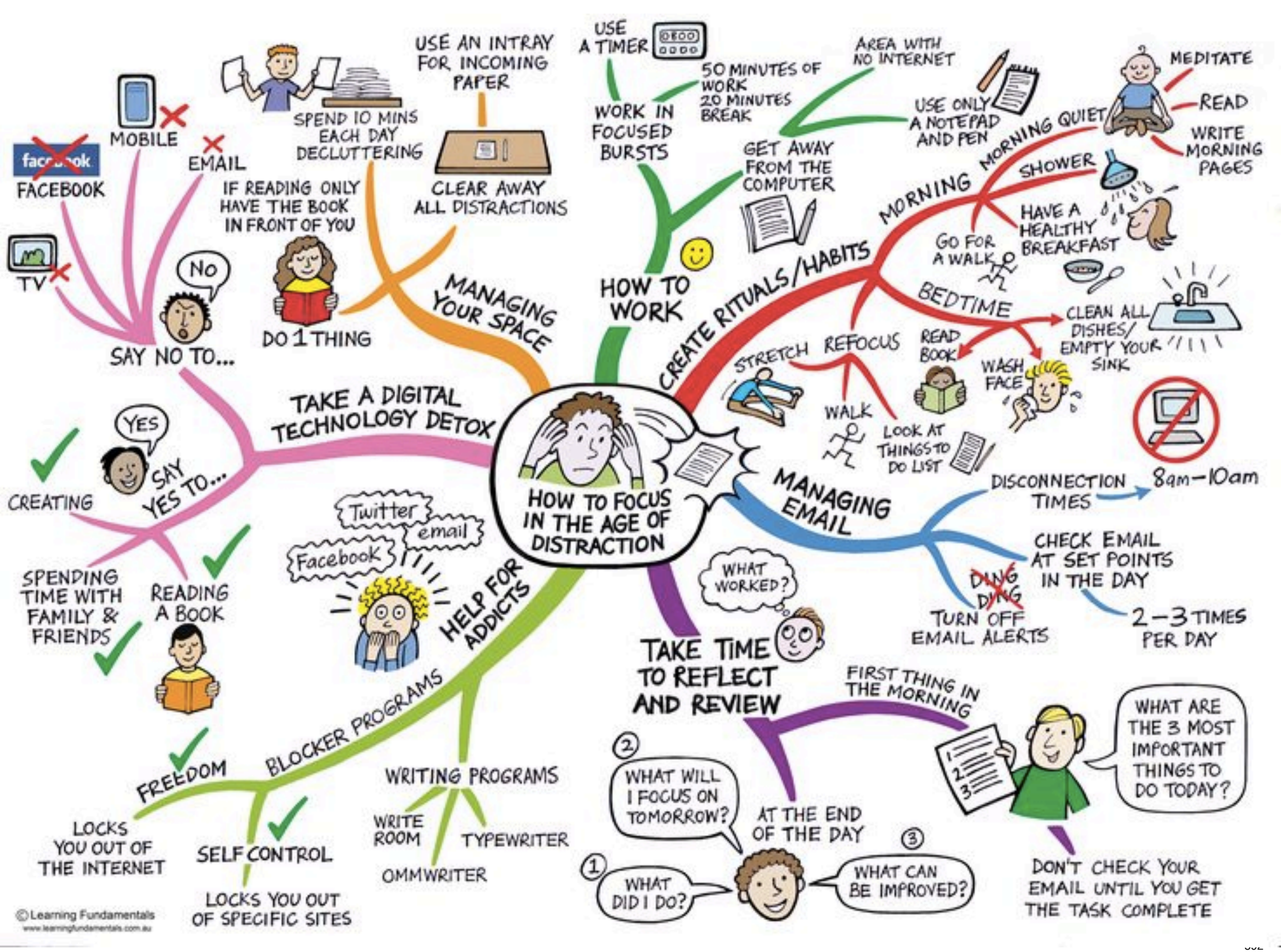


Action Tools

2. Large-Scale Scrum
3. Test
4. Product Management
5. Planning
6. Coordination
8. Requirements
9. Design & Architecture
10. Legacy Code
11. Continuous Integration
12. Inspect & Adapt
13. Multisite
14. Offshore
15. Contracts

individual

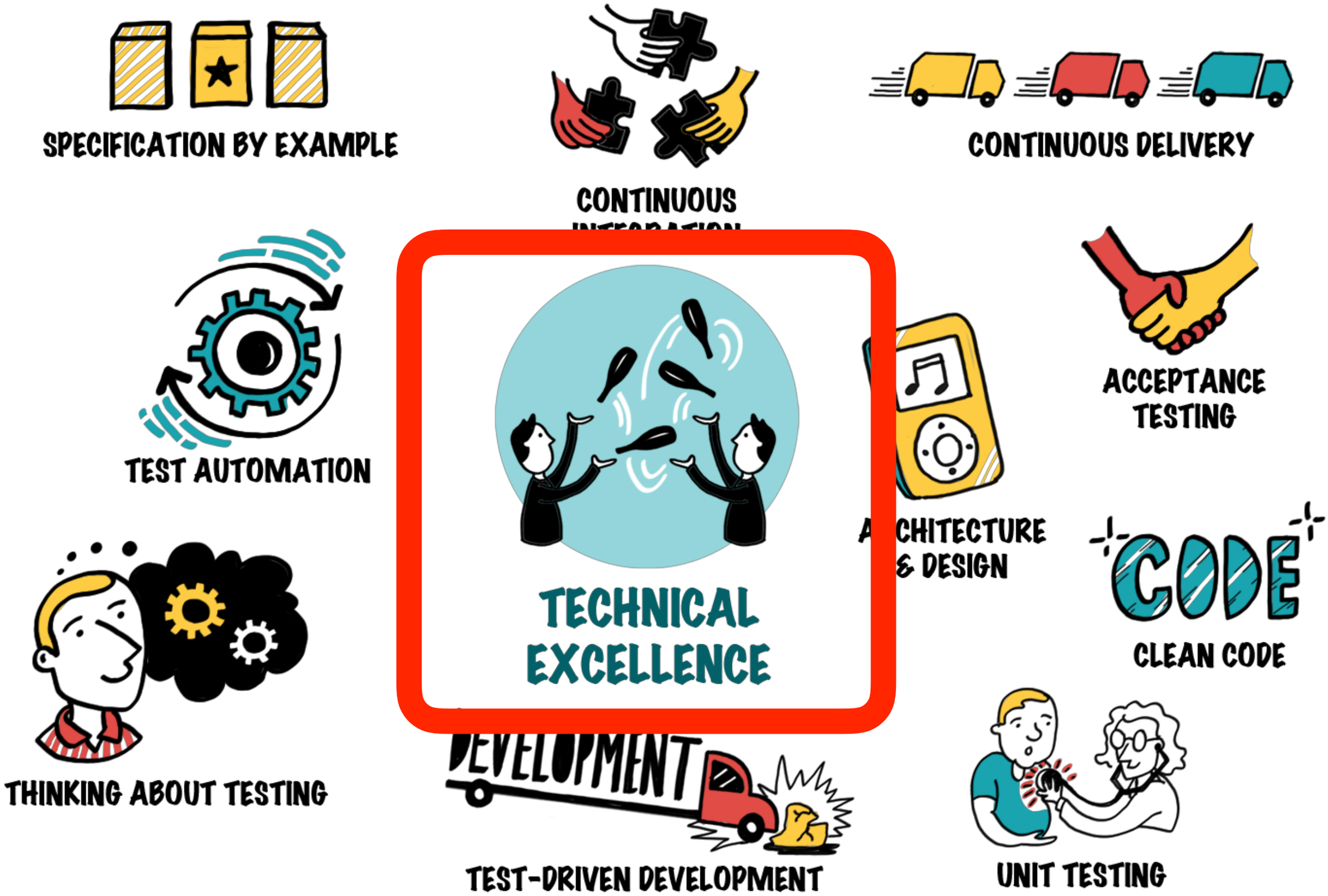
> review Coordination & Communities sections


HOW TO FOCUS IN THE AGE OF DISTRACTION
TAKE A DIGITAL TECHNOLOGY DETOX
SAY NO TO...
No
MOBILE
EMAIL
FACEBOOK
TV
SAY YES TO...
Yes
CREATING
SPENDING TIME WITH FAMILY & FRIENDS
READING A BOOK
MANAGING YOUR SPACE
SPEND 10 MINS EACH DAY DECLUTTERING
USE AN INTRAY FOR INCOMING PAPER
CLEAR AWAY ALL DISTRACTIONS
IF READING ONLY HAVE THE BOOK IN FRONT OF YOU
DO 1 THING
HOW TO WORK
USE A TIMER
WORK IN FOCUSED BURSTS
50 MINUTES OF WORK 20 MINUTES BREAK
GET AWAY FROM THE COMPUTER
AREA WITH NO INTERNET
USE ONLY A NOTEPAD AND PEN
CREATE RITUALS/HABITS
MORNING
MORNING QUIET
MEDITATE
READ
WRITE MORNING PAGES
SHOWER
HAVE A HEALTHY BREAKFAST
GO FOR A WALK
BEDTIME
CLEAN ALL DISHES/ EMPTY YOUR SINK
READ BOOK
WASH FACE
STRETCH
REFOCUS
WALK
LOOK AT THINGS TO DO LIST
MANAGING EMAIL
DISCONNECTION TIMES
8am–10am
CHECK EMAIL AT SET POINTS IN THE DAY
2–3 TIMES PER DAY
TURN OFF EMAIL ALERTS
DING
DING
HELP FOR ADDICTS
Twitter
email
Facebook
BLOCKER PROGRAMS
FREEDOM
LOCKS YOU OUT OF THE INTERNET
SELF CONTROL
LOCKS YOU OUT OF SPECIFIC SITES
WRITING PROGRAMS
WRITE ROOM
TYPEWRITER
OMMWRITER
TAKE TIME TO REFLECT AND REVIEW
WHAT WORKED?
FIRST THING IN THE MORNING
WHAT ARE THE 3 MOST IMPORTANT THINGS TO DO TODAY?
DON'T CHECK YOUR EMAIL UNTIL YOU GET THE TASK COMPLETE
AT THE END OF THE DAY
1
WHAT DID I DO?
2
WHAT WILL I FOCUS ON TOMORROW?
3
WHAT CAN BE IMPROVED?
© Learning Fundamentals
www.learningfundamentals.com.au

Technical Excellence

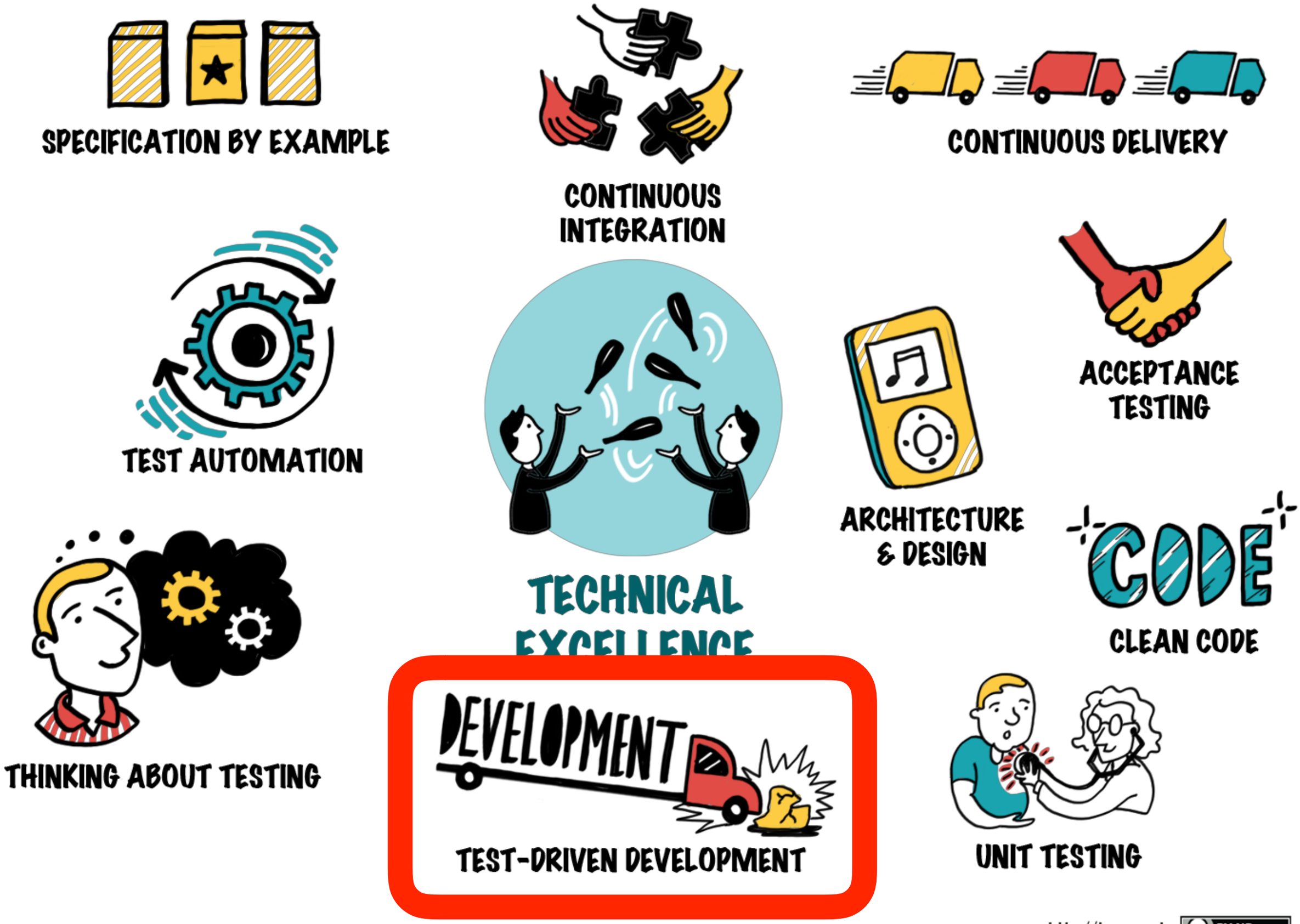
Less

… requires Technical Excellence



http://less.works CC BY-ND

IBB



http://less.works CC BY-ND

~~TDD = testing~~

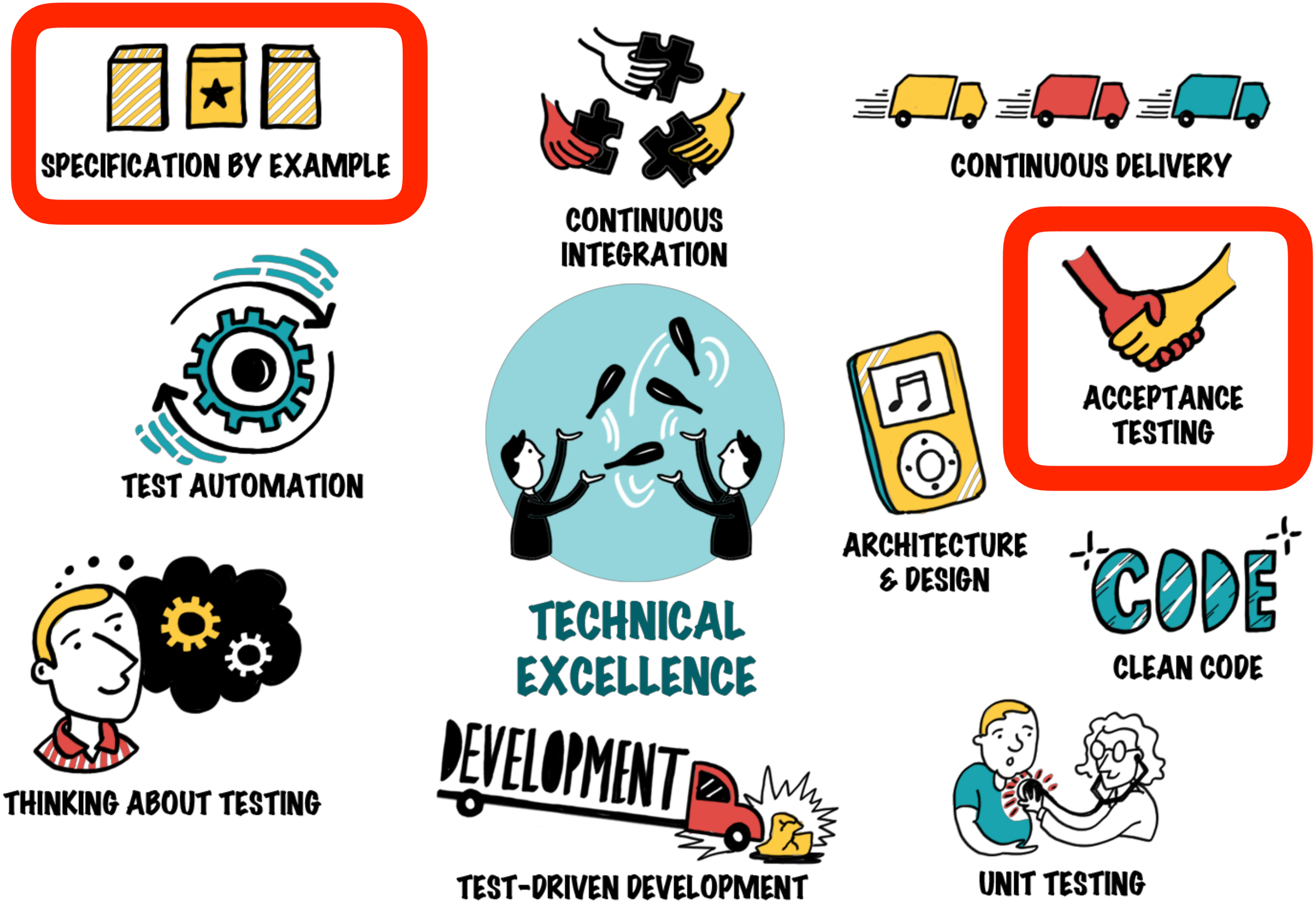
acceptance TDD

create the *Right Code*

unit TDD

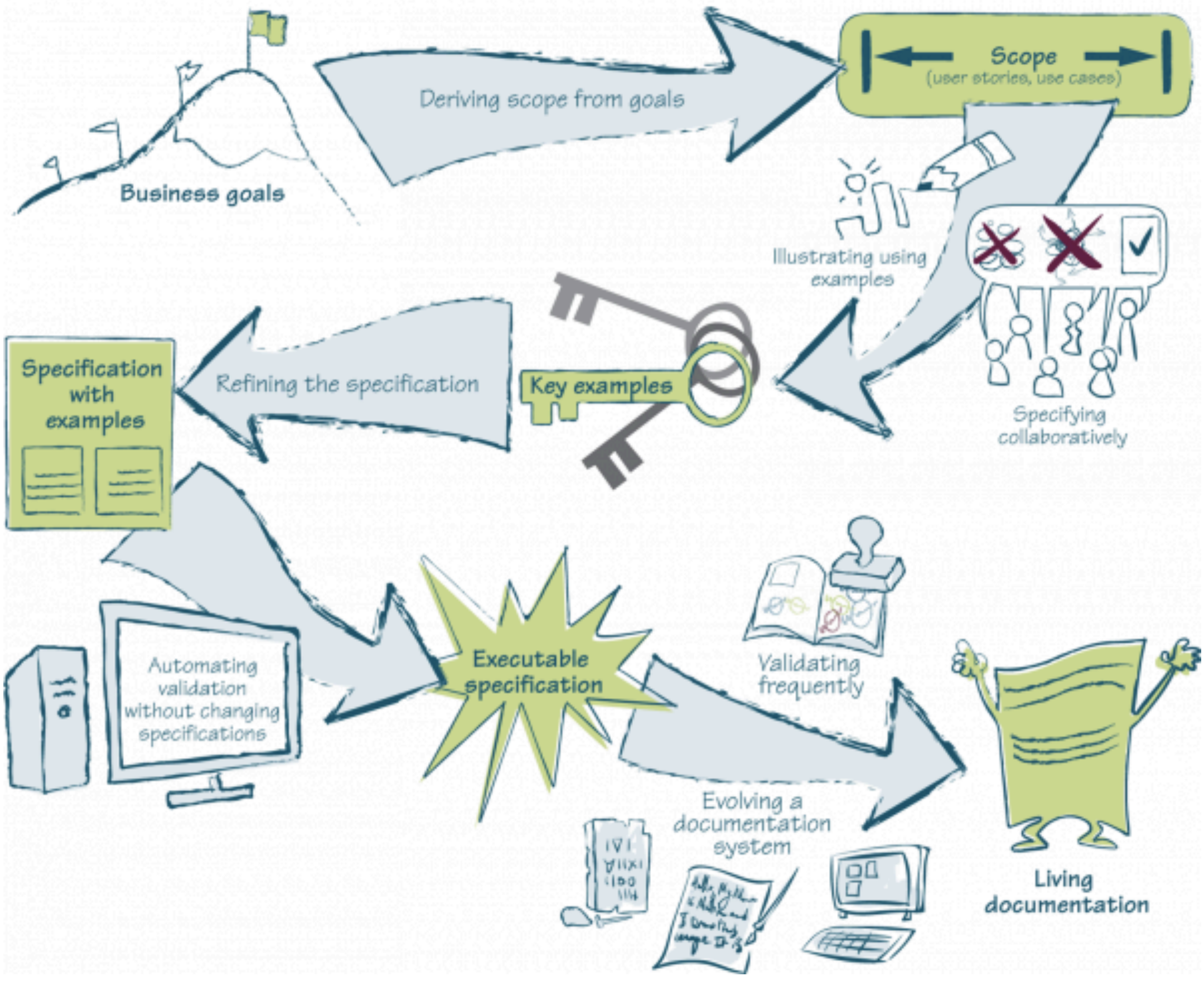
create the *Code Right*

Acceptance TDD & Specification by Example



http://less.works CC BY-ND

Key process patterns of Specification by Example



Specification by Example

includes…

ATDD/BDD

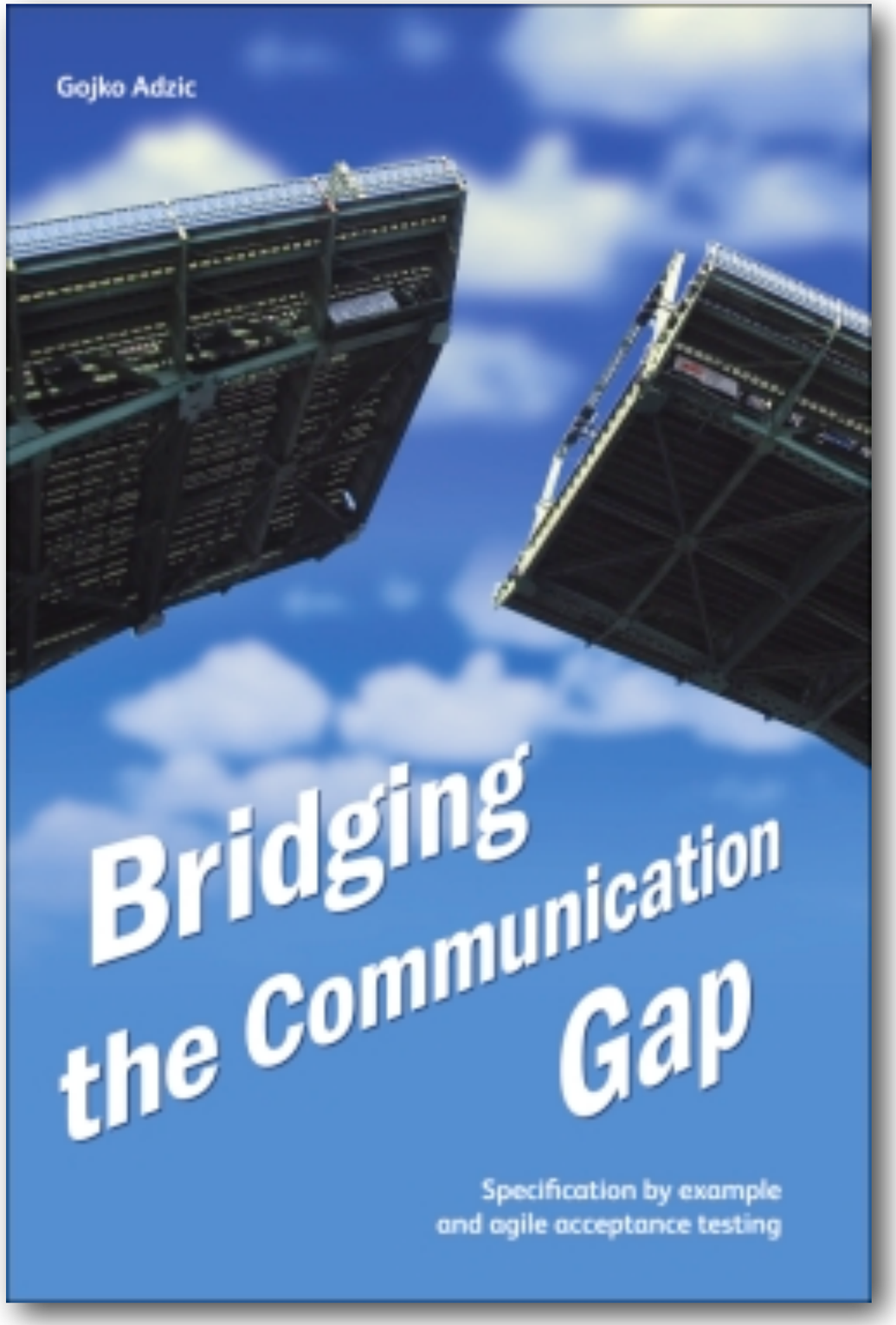
Specification by Example

***executable* "customer"**

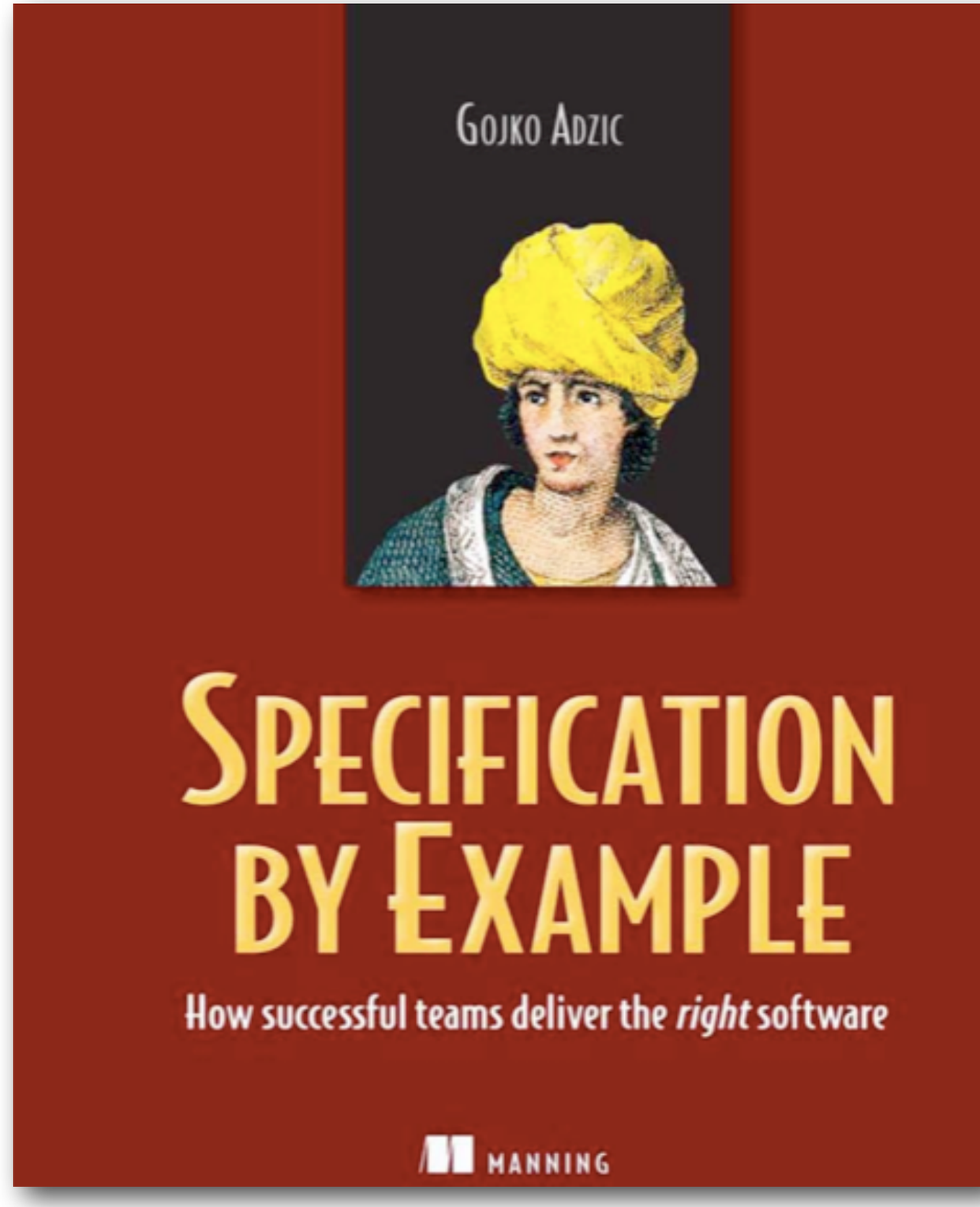
specifications, first

"living documentation"

Some of the Best…



Vol 1



Vol 2

Specification by Example

learn & define specifications, from *examples*…

"there's no such thing as **requirements analysis, business analysis, or requirements engineering**"

it's an education problem

Specification by Example

Dev state	Mic Exists	Mic Muted	Input Channel	Input Event	Mic Muted	OSD mute symbol	Table Mic mute LED	For End of mute Poster	Other OSD feedback
Out of call	T	F	R.C.	"mute"	T	Lit	Lit	NA	NA

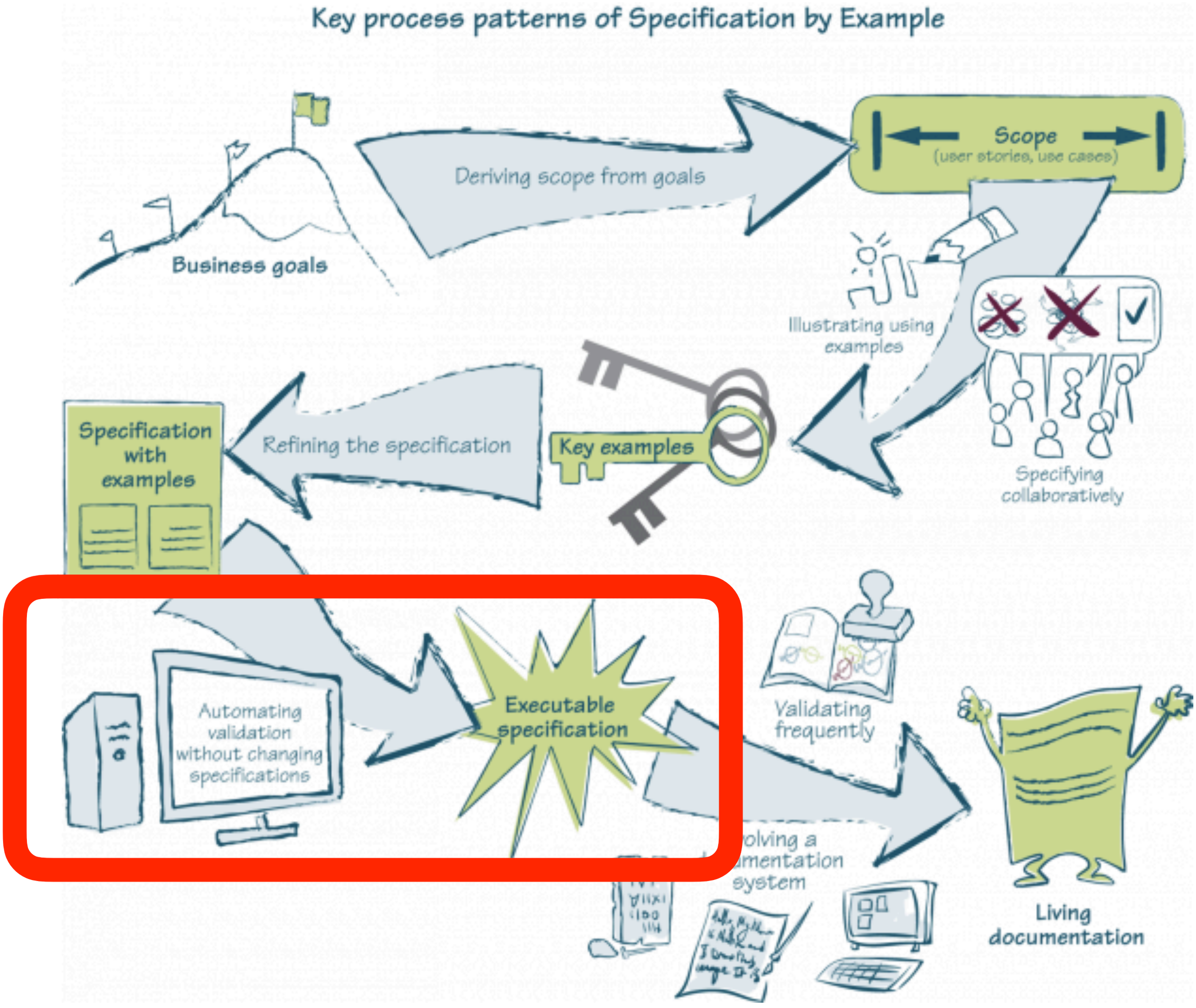
Dev state	Mic Exists	Mic Muted	Input Channel	Input Changed Event	Mic Muted	OSD mute symbol	Table Mic mute LED	For End of mute Poster	Other OSD feedback
Out of call	T	F	R.C.	"mute"	T	Lit	Lit	NA	NA
Out of call	T	T	R.C.	"mute"	F	Unlit	Unlit	NA	NA

Dev state	Mic Exists	Mic Muted	Input Channel	Input Event	Mic Muted	OSD mute symbol	Table Mic mute LED	For End of mute Poster	Other OSD feedback
Out of call	T	F	R.C.	"mute"	T	Lit	Lit	NA	NA
Out of call	T	T	R.C	"mute"	F	Unlit	Unlit	NA	NA
"	F	NA	R.C	"	NA	NA	NA	NA	Error Message
In call	T	F	R.C	"	T	Lit	Lit	Lit	NA
"	T	T	RC	"	F	Unlit	Unlit	Unlit	
"	T	T	RC	END CALL					

testing the learners!

all this happens during PBR

live example?


Key process patterns of Specification by Example
Business goals
Deriving scope from goals
Scope
(user stories, use cases)
Illustrating using examples
Specifying collaboratively
Key examples
Refining the specification
Specification with examples
Automating validation without changing specifications
Executable specification
Validating frequently
olving a
mentation
system
Living documentation

Specification by Example

***executable* "customer"**

specifications, first

Specification by Example

executable specifications "without change", with automated validation…

Specification by Example

specification with examples (real case)

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
Order Details										RefData					
currency pair	fx type	Verb	Amount	Dealt	Markup 1	Markup 2	Client Value	Value	Markup Type	Pip Fmt		Success	Total Markup	Wholesale Value	Profit
EURUSD	Spot	Buy	1,000,000	EUR	2		1.30	0.00	Pts	4		OK	2	1.3002	200.00
USDJPY	Spot	Sell	2,000,000	USD	10		110.00	0.00	Pts	2		OK	10	109.9	200,000.00
GBPUSD	Fwd	Buy	1,000,000	GBP	1	3	1.50	0.00	Pts	4		OK	4	1.5004	400.00
GBPUSD	Fwd	Sell	1,000,000	GBP	1	3	1.50	0.00	Pts	4		OK	4	1.4996	400.00
GBPUSD	Fwd	Sell	1,000,000	GBP	1	0	1.50	0.00	Pts	4		OK	1	1.4999	100.00

executable specifications “without change”,
with automated validation

Action	currency pair	fx type	Verb	Amount	Dealt	Markup 1	Markup 2	Client Value	Value	Markup Type	Error Code	Total Markup	Wholesale Value	Profit
Verify Profit Details For Order	EURUSD	Spot	Buy	1000000	EUR	2.0	0.0	1.30	0.00	Pts	0	2.0	1.3002	200.00
Verify Profit Details For Order	USDJPY	Spot	Sell	2000000	USD	10.0	0.0	110.0	0.00	Pts	0	10.0	109.9	200000.00
Verify Profit Details For Order	GBPUSD	Fwd	Buy	1000000	GBP	1.0	3.0	1.5	0.00	Pts	0	4.0	1.5004	400.00
Verify Profit Details For Order	GBPUSD	Fwd	Sell	1000000	GBP	1.0	3.0	1.5	0.00	Pts	0	4.0	1.4996	400.00
Verify Profit Details For Order	GBPUSD	Fwd	Sell	1000000	GBP	1.0	0.0	1.5	0.00	Pts	0	1.0	1.4999	100.00

REQUIREN
TEST

A Better Example Language

lessons from usability and customer-centric: *Choose a language familiar to the experts*

so, don't force them into "selfish" programmer-created languages such as "given-when-then" (unless the experts naturally use that)

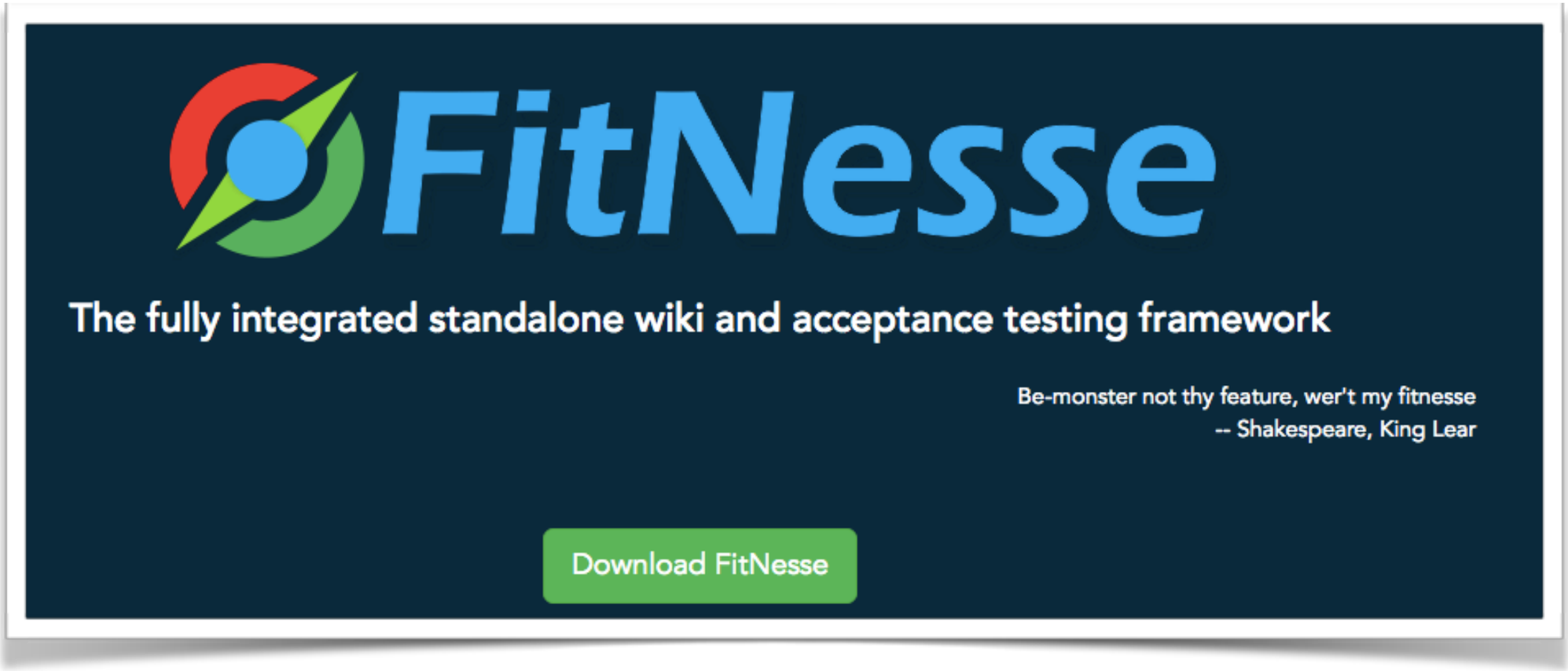
A Better Example Language

the *abstraction* of examples is preconditions & postconditions

there are an infinite number of *surface languages* for that

thinking that "given-when-then" is some magical formulation (rather than just an example of a pre- & post-condition language) is … magical thinking ;)

Tools


FitNesse
The fully integrated standalone wiki and acceptance testing framework
Be-monster not thy feature, wer't my fitnesse
-- Shakespeare, King Lear
Download FitNesse

Specification by Example

create "living documentation"

FitNesse Test Edit Add Tools

SpecSemiPrime

Semiprime Executable Specification

For a general overview, see https://en.wikipedia.org/wiki/Semiprime

semiprime numbers

A semiprime number is the product of two prime numbers.

Also called biprime, 2-almost prime, or pq number.

2 x 2
5 x 11
13 x 2
31 x 41
3 x 7
3 x 19

semi prime numbers to 100

Example semiprimes:

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

Here are the executable specs:

integer	is semiprime?
2	false
3	false
4	true

Acceptance TDD Cycle

1. Red – (create living documentation) for the next customer feature, transform all the **example specifications** into all the **executable specifications** (1-3 HOURS)
2. Green – implement feature, so that all the executable specifications are fulfilled (1-3 DAYS)

outside-in development

live example?

Unit TDD

~~TDD = testing~~

unit TDD

TDD

TDD

development method for *clean code !!!*

executable *developer* specifications, first

emergent *micro*-design

TDD

1. create failing executable spec

2. create passing executable spec (<u>simply</u>)

<u>3. !!! refactor to clean code !!!</u>

live example?

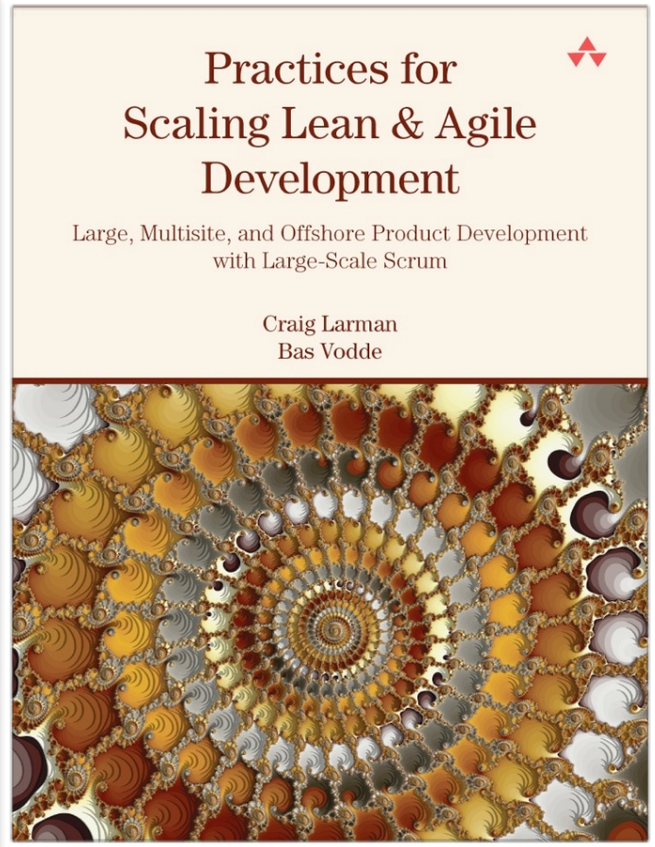
LeSS More with LeSS

Architecture & Design

Thinking About Design

Behavior-Oriented Tips

- Design workshops with agile modeling
- Just-in-Time (JIT) modeling; vary the abstraction level
- Design workshops each iteration
- Design workshops in the team rooms
- Multi-team design workshops for broader design issues
- Technical leaders teach at workshops
- Architects and system engineers are regular (feature) team members
- Question all early architectural decisions as final
- Don't comform to outdated architectural decisions
- Avoid architecture astronauts (PowerPoint architects)
- Very early, develop a walking skeleton with tracer code
- Incrementally build 'vertical' architectural slices of customer-centric features
- Do customer-centric features with major architectural impact first
- Architects clarify by programming spike solutions
- Don't let architects hand off to 'coders'
- Tiger team conquers then divides
- SAD workshops at end of "tiger phase"



Action Tools

2. Large-Scale Scrum
3. Test
4. Product Management
5. Planning
6. Coordination
8. Requirements
9. Design & Architecture
10. Legacy Code
11. Continuous Integration
12. Inspect & Adapt
13. Multisite
14. Offshore
15. Contracts

technical excellence

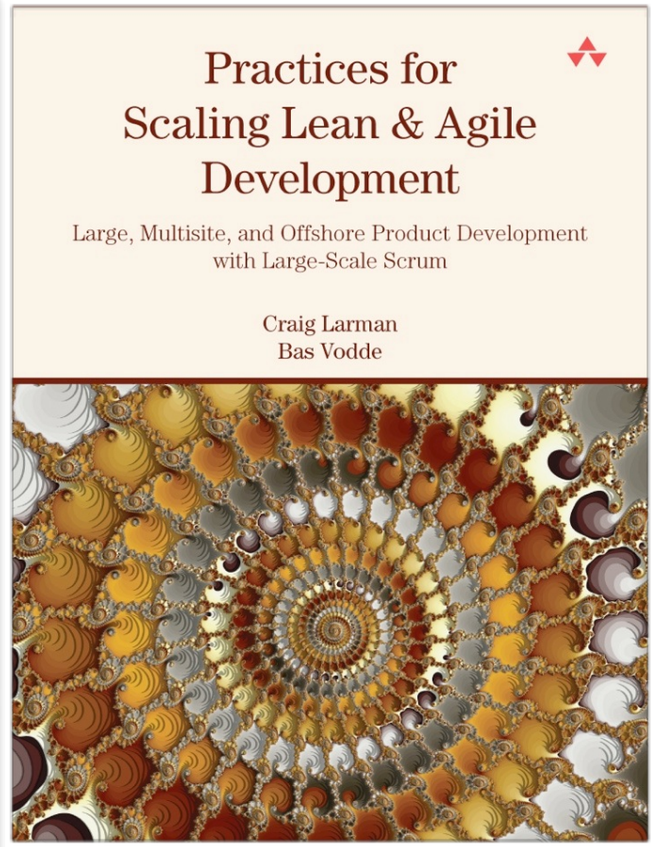
LeSS More with LeSS

Architecture & Design

Thinking About Design

Behavior-Oriented Tips

- Design workshops with agile modeling
- Just-in-Time (JIT) modeling; vary the abstraction level
- Design workshops each iteration
- Design workshops in the team rooms
- Multi-team design workshops for broader design issues
- Technical leaders teach at workshops
- Architects and system engineers are regular (feature) team members
- Question all early architectural decisions as final
- Don't comform to outdated architectural decisions
- Avoid architecture astronauts (PowerPoint architects)
- Very early, develop a walking skeleton with tracer code
- Incrementally build 'vertical' architectural slices of customer-centric features
- Do customer-centric features with major architectural impact first
- Architects clarify by programming spike solutions
- Don't let architects hand off to 'coders'
- Tiger team conquers then divides
- SAD workshops at end of "tiger phase"



Action Tools

2. Large-Scale Scrum
3. Test
4. Product Management
5. Planning
6. Coordination
8. Requirements
9. Design & Architecture
10. Legacy Code
11. Continuous Integration
12. Inspect & Adapt
13. Multisite
14. Offshore
15. Contracts

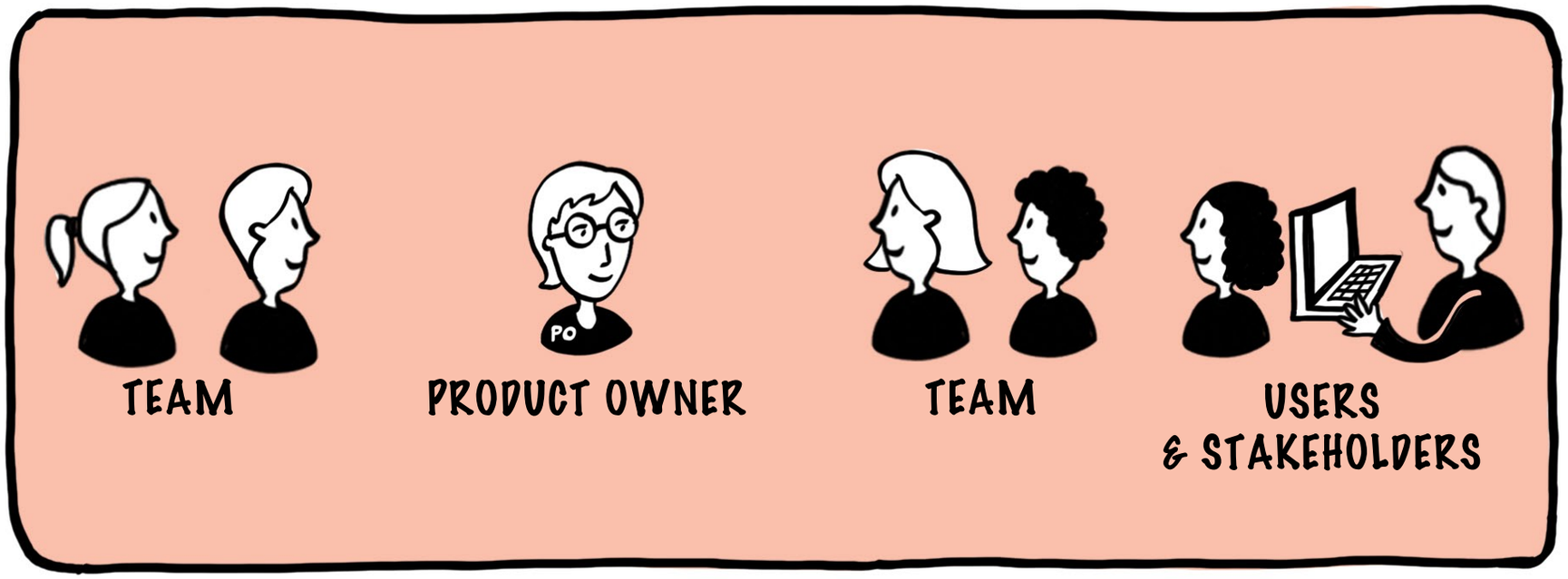
technical excellence

LeSS

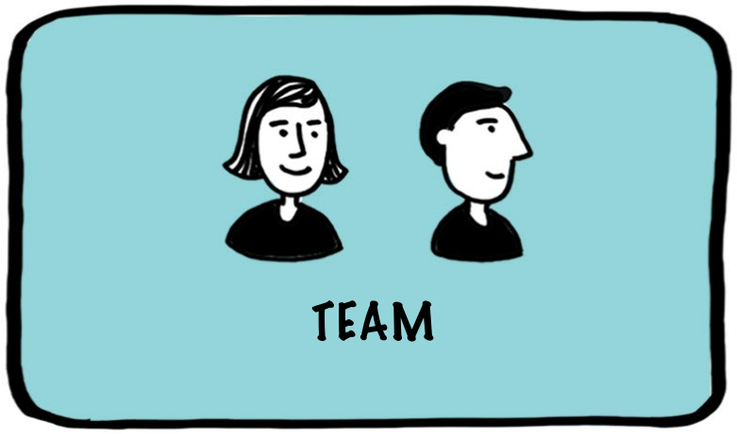
Sprint Review

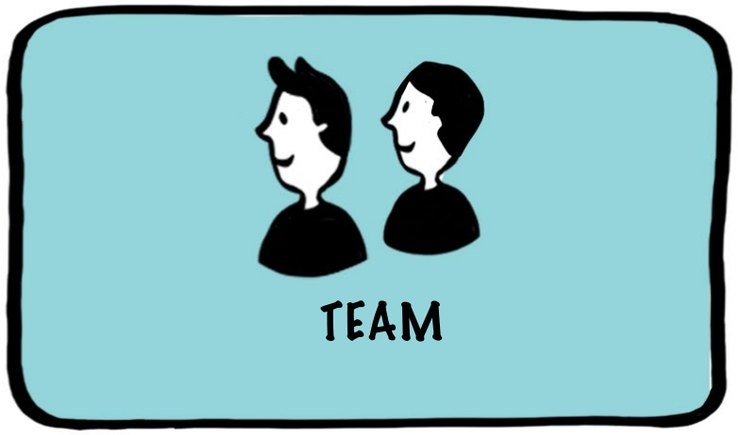
LeSS SPRINT REVIEW & RETROSPECTIVE

SPRINT REVIEW

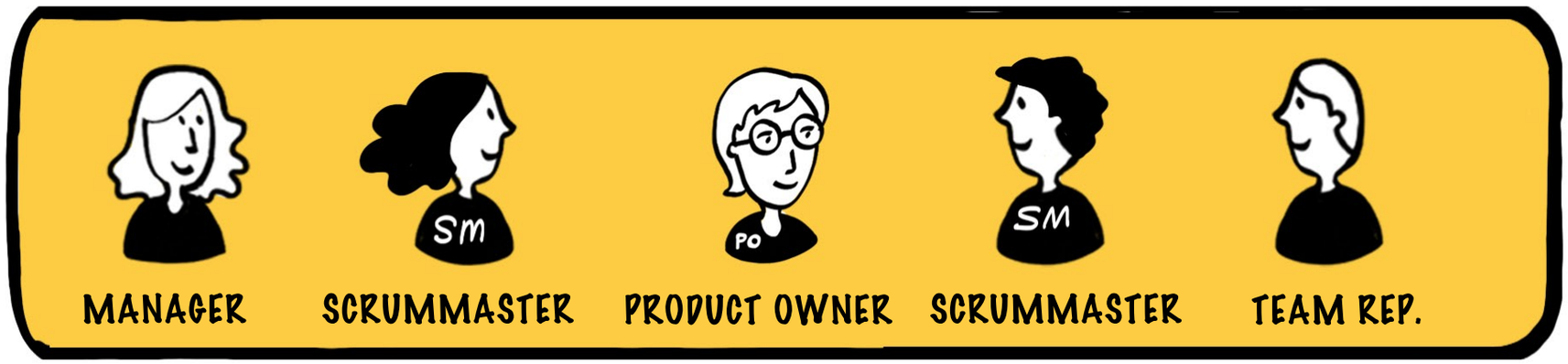


TEAM RETROSPECTIVE





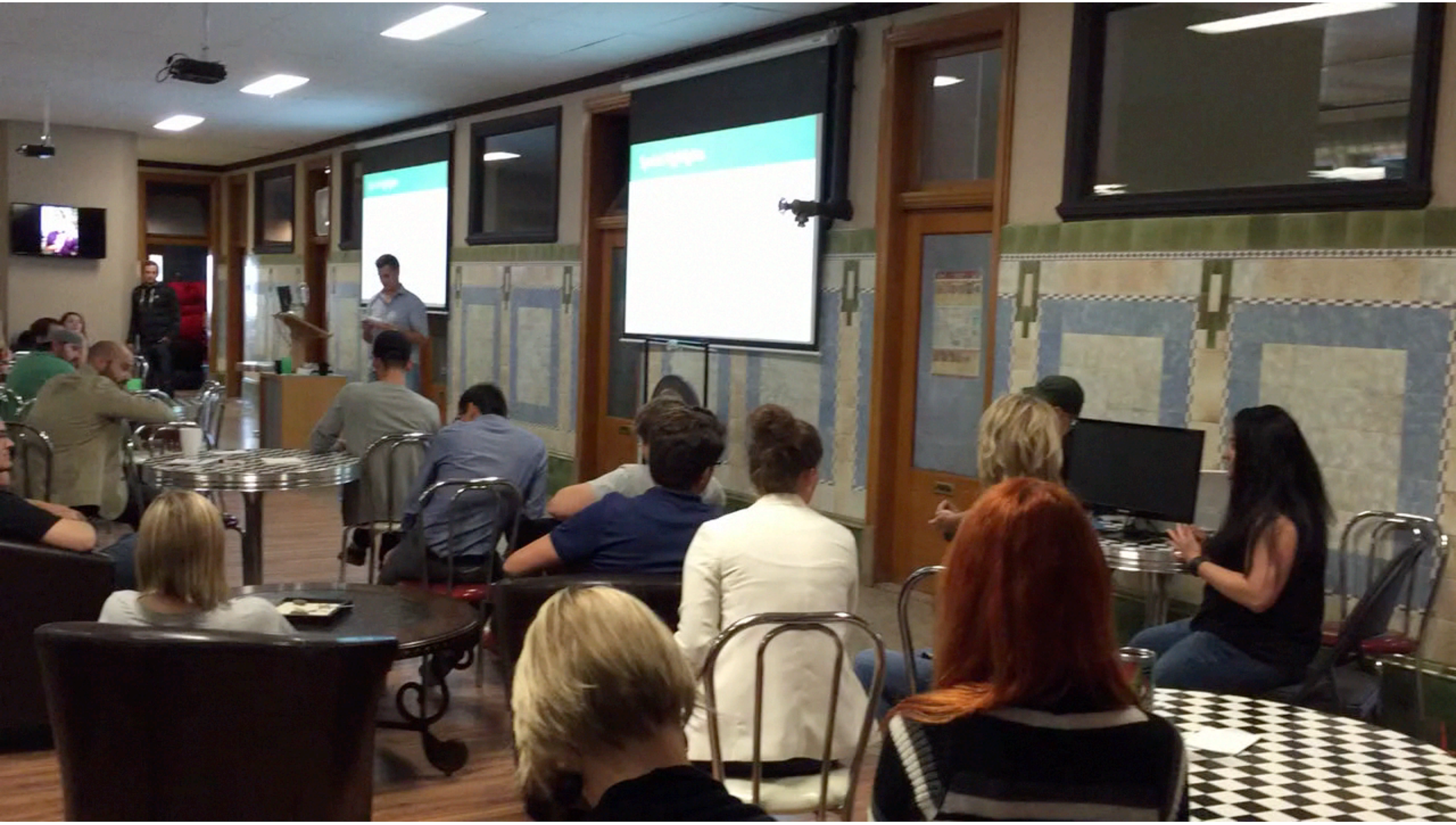
OVERALL RETROSPECTIVE



in-Sprint early item feedback

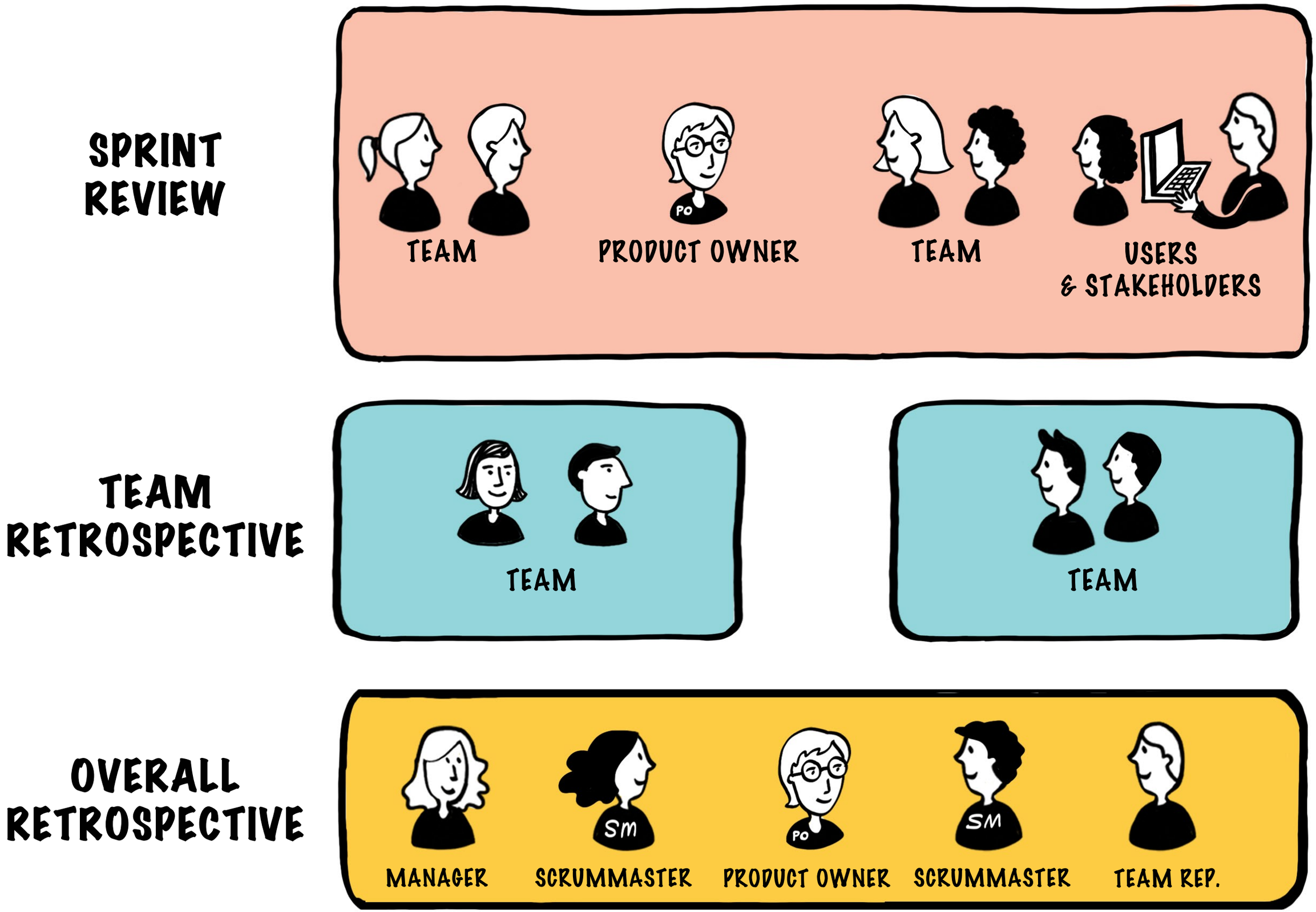
Guide: Sprint Review Bazaar

Sprint Review “Bazaar” Q&A



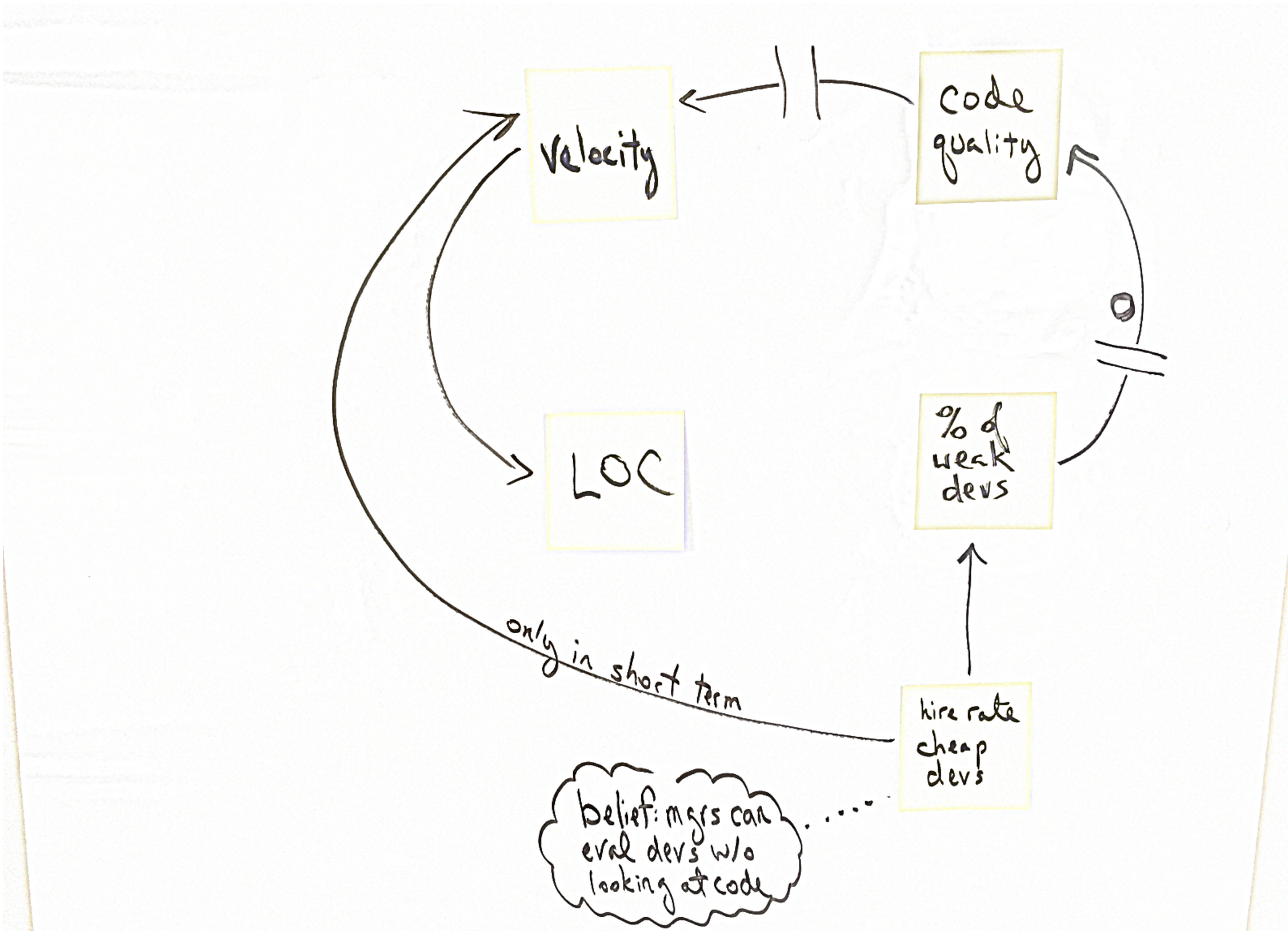
LeSS Sprint Retrospective

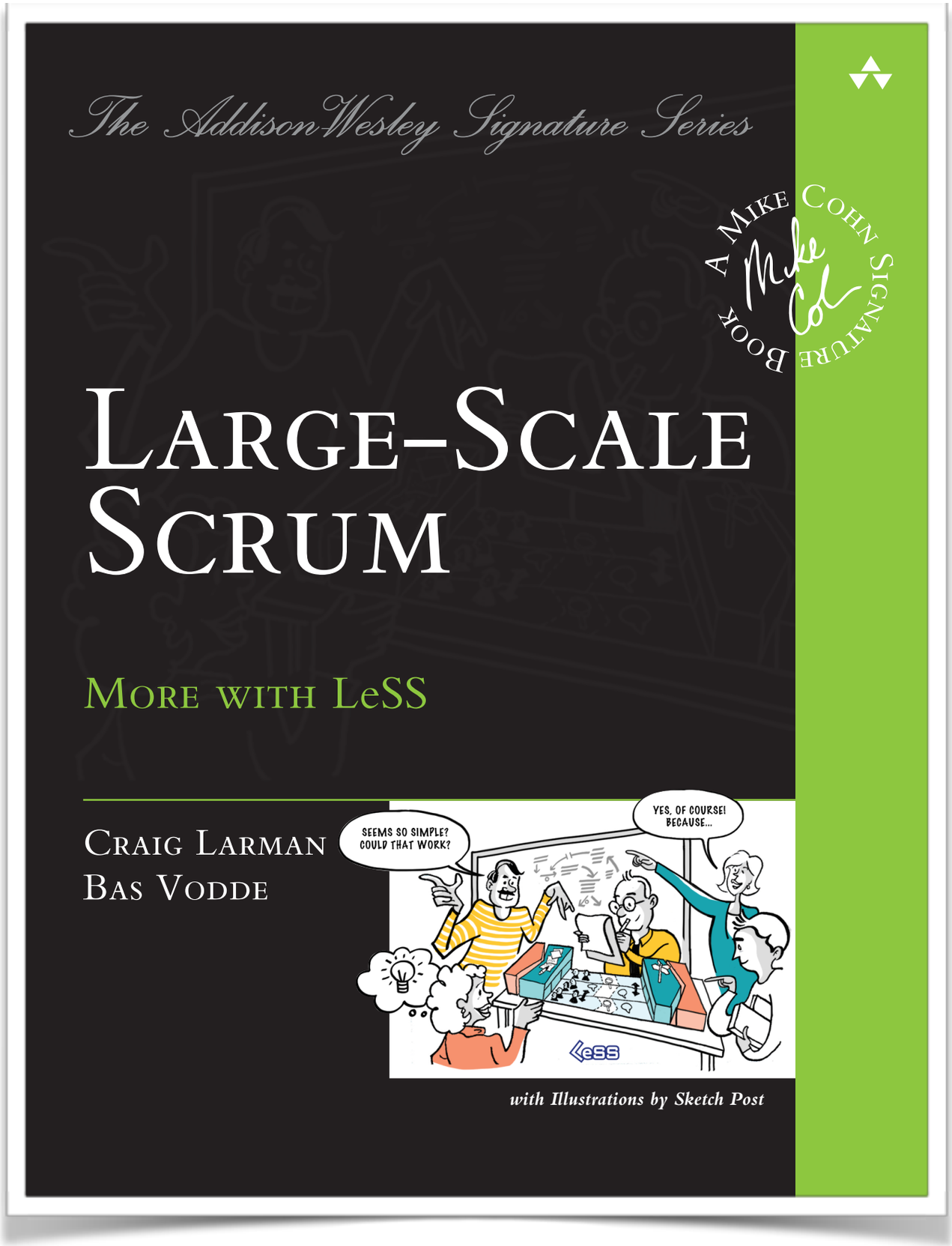
LeSS SPRINT REVIEW & RETROSPECTIVE



Guide: Overall Retrospective (multisite)

Systems Modelling





1 More with LeSS 1
2 LeSS 5

LeSS Structure

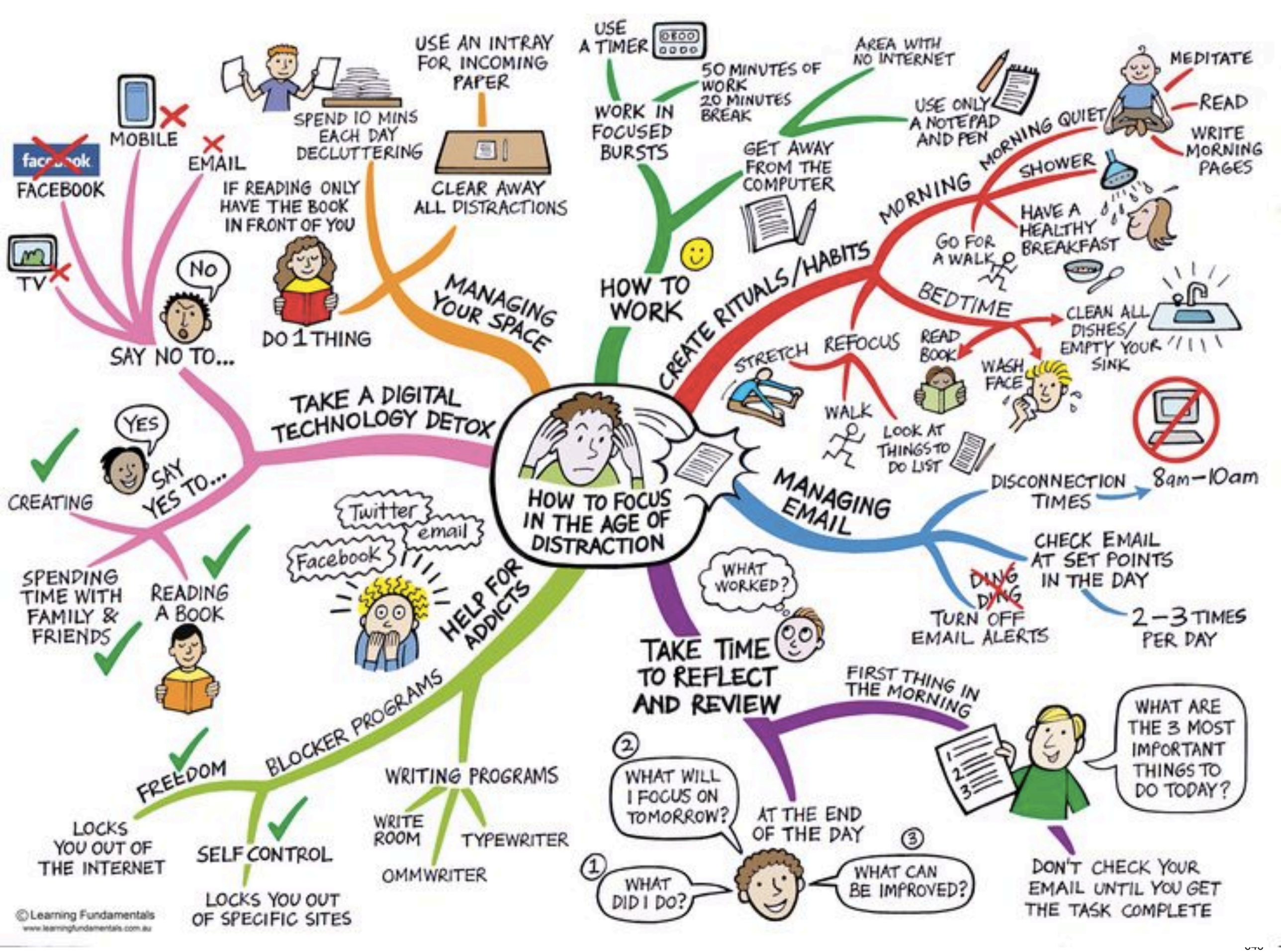
3 Adoption 41
4 Organize by Customer Value 67
5 Management 111
6 ScrumMasters 133

LeSS Product

7 Product 155
8 Product Owner 171
9 Product Backlog 197
10 Definition of Done 231

LeSS Sprint

11 Product Backlog Refinement 249
12 Sprint Planning 275
13 Coordination and Integration 285
14 Review & Retrospective 313


HOW TO FOCUS IN THE AGE OF DISTRACTION
TAKE A DIGITAL TECHNOLOGY DETOX
SAY NO TO...
No
MOBILE
EMAIL
FACEBOOK
TV
SAY YES TO...
Yes
CREATING
SPENDING TIME WITH FAMILY & FRIENDS
READING A BOOK
MANAGING YOUR SPACE
SPEND 10 MINS EACH DAY DECLUTTERING
USE AN INTRAY FOR INCOMING PAPER
CLEAR AWAY ALL DISTRACTIONS
IF READING ONLY HAVE THE BOOK IN FRONT OF YOU
DO 1 THING
HOW TO WORK
USE A TIMER
WORK IN FOCUSED BURSTS
50 MINUTES OF WORK 20 MINUTES BREAK
GET AWAY FROM THE COMPUTER
AREA WITH NO INTERNET
USE ONLY A NOTEPAD AND PEN
CREATE RITUALS/HABITS
MORNING
MORNING QUIET
MEDITATE
READ
WRITE MORNING PAGES
SHOWER
HAVE A HEALTHY BREAKFAST
GO FOR A WALK
BEDTIME
CLEAN ALL DISHES/ EMPTY YOUR SINK
READ BOOK
WASH FACE
STRETCH
REFOCUS
WALK
LOOK AT THINGS TO DO LIST
MANAGING EMAIL
DISCONNECTION TIMES
8am–10am
CHECK EMAIL AT SET POINTS IN THE DAY
2–3 TIMES PER DAY
TURN OFF EMAIL ALERTS
DING
DING
HELP FOR ADDICTS
Twitter
email
Facebook
BLOCKER PROGRAMS
FREEDOM
LOCKS YOU OUT OF THE INTERNET
SELF CONTROL
LOCKS YOU OUT OF SPECIFIC SITES
WRITING PROGRAMS
WRITE ROOM
TYPEWRITER
OMMWRITER
TAKE TIME TO REFLECT AND REVIEW
WHAT WORKED?
FIRST THING IN THE MORNING
WHAT ARE THE 3 MOST IMPORTANT THINGS TO DO TODAY?
DON'T CHECK YOUR EMAIL UNTIL YOU GET THE TASK COMPLETE
AT THE END OF THE DAY
1
WHAT DID I DO?
2
WHAT WILL I FOCUS ON TOMORROW?
3
WHAT CAN BE IMPROVED?
© Learning Fundamentals
www.learningfundamentals.com.au

Done & Undone

Done

class

> for one sample large-scale product, tasks to have a shippable product?

perfect DoD

=

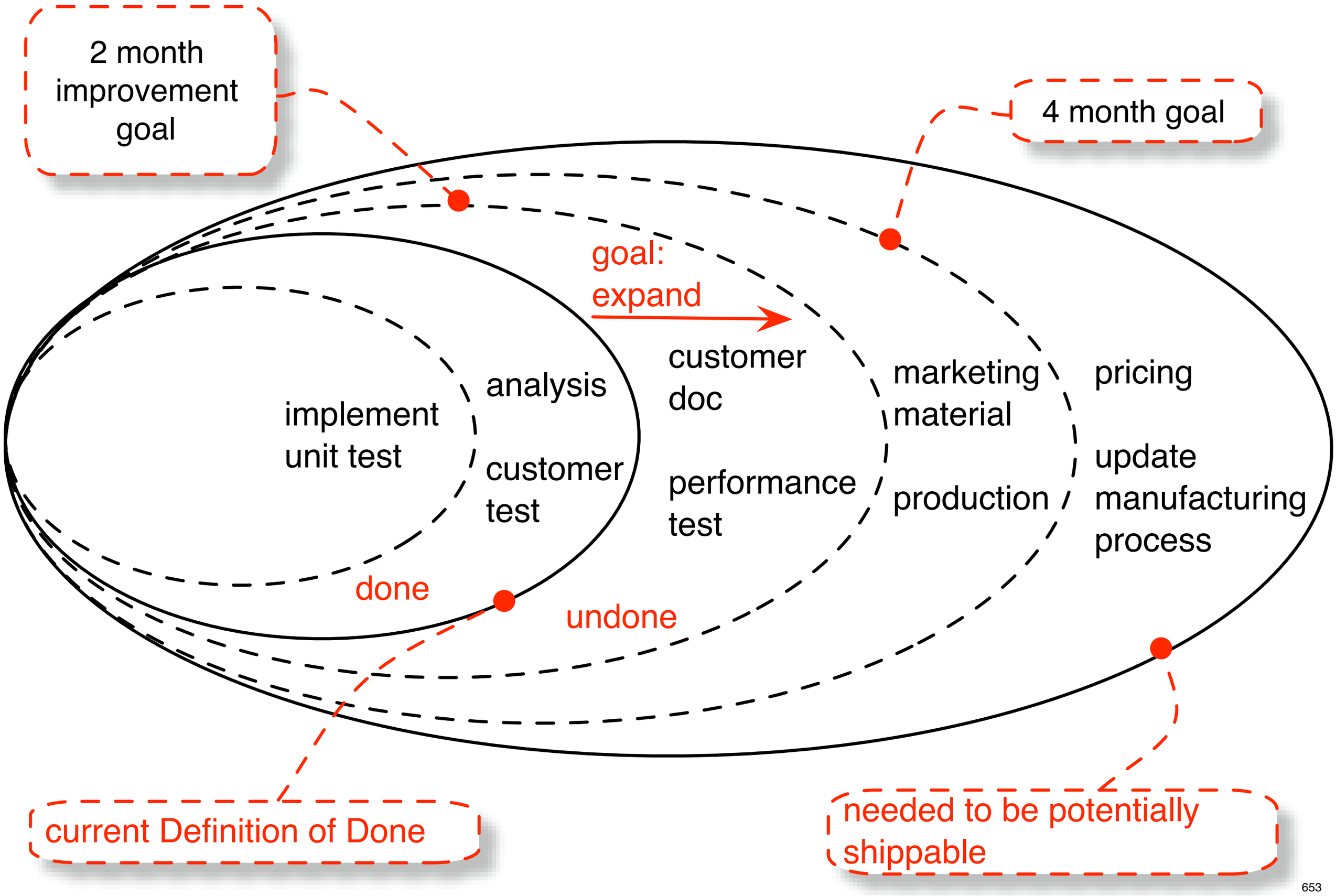
shippable DoD

=

potentially shippable

an **imperfect DoD** is especially common in large-scale groups first moving to LeSS

Improving DoD over Time



Perfect DoD

- Imperfect DoD

= **Undone Work**

DEFINITION OF DONE:

- CODED (UNIT-TESTING)
- INTEGRATED: INT. TEST, REAL ENV. (WHEN APPLICABLE)
- ACCEPTANCE TESTS PASSED (MANUAL), MOCK, STUB
- DRAFT USER DOC WRITTEN (CHANGE LOG FILE)
- RELEASED (BUILD + REL NOTES AVAILABLE ON TRACKER)

UNDONE

- PERFORMANCE TEST
- QA TEST (END TO END)
- FULL UAT

Handling Undone Work?

covered in next section

Perfection Goal

perfect DoD = shippable

no Undone Work

LeSS Rule(s)

1 Definition of Done

not for each team

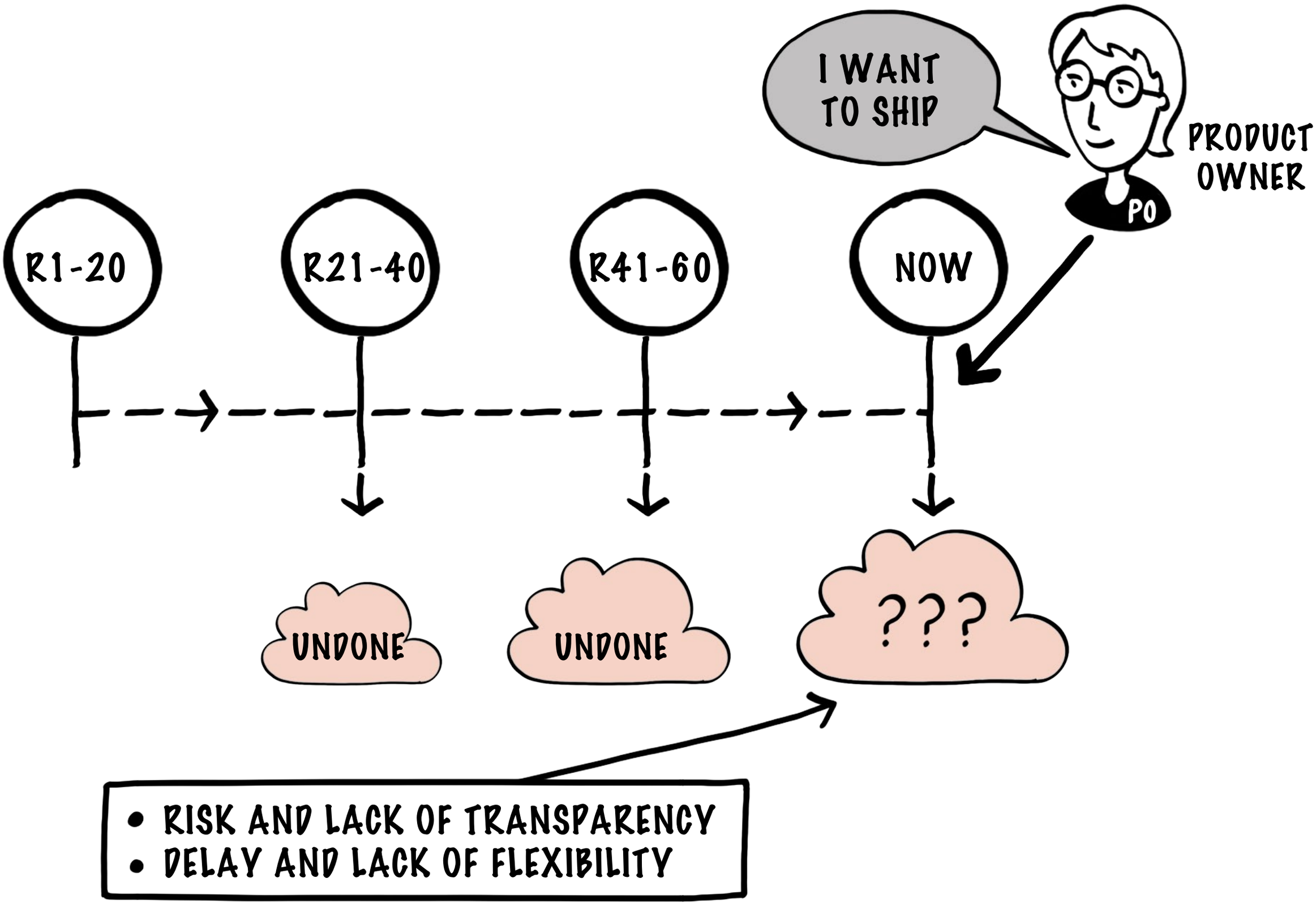
(teams can extend base version)

LeSS Adoption Tip

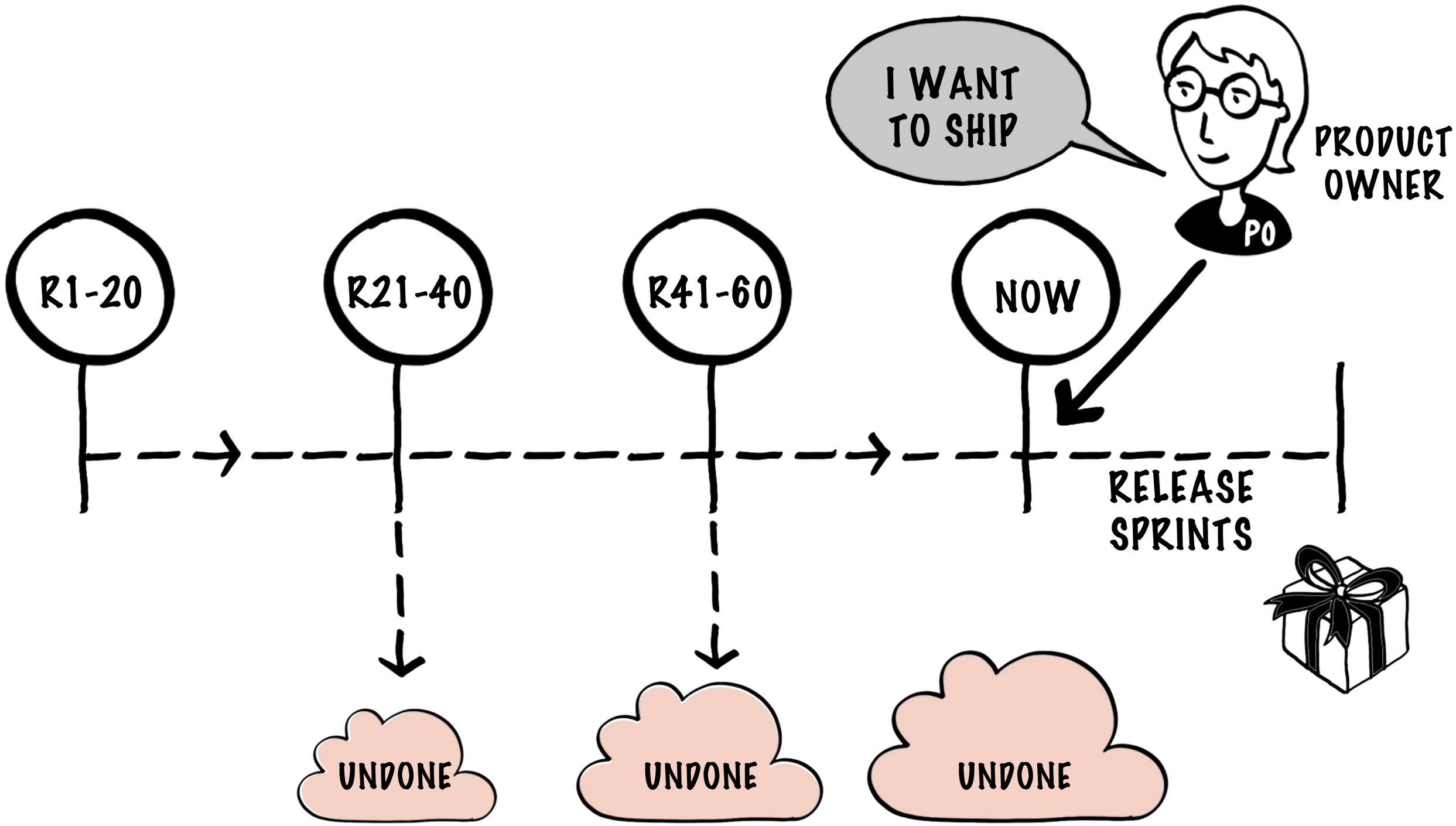
if possible, solve the problems so you can have a perfect DoD before Sprint 1

why?

Undone


I WANT TO SHIP
PO
PRODUCT OWNER
R1-20
R21-40
R41-60
NOW
UNDONE
UNDONE
???
• RISK AND LACK OF TRANSPARENCY
• DELAY AND LACK OF FLEXIBILITY

Release Sprint… by **TEAMS**



Lean Wastes in Product Development

1. **Over-production**—of intermediate, WIP, or finished things; sooner, faster, greater than demand
2. **Inventory**—intermediate, WIP, or finished things
3. **Over-processing**—& extra processes, rediscovery
4. **Handoff**—& transport
5. **Task switching**—& motion
6. **Waiting**—& delay
7. **Defects & finding/ correcting**—tasks to find & correct: test, inspect, review, modify
8. **Not using people's full potential**—working to title, not multi-skilling
9. **Knowledge/information scatter/loss**—& connection to handoff & inventory & rediscovery; communication barriers: indirection, 1-way flows

and…

to get to perfect DoD

Teams **learn by doing**

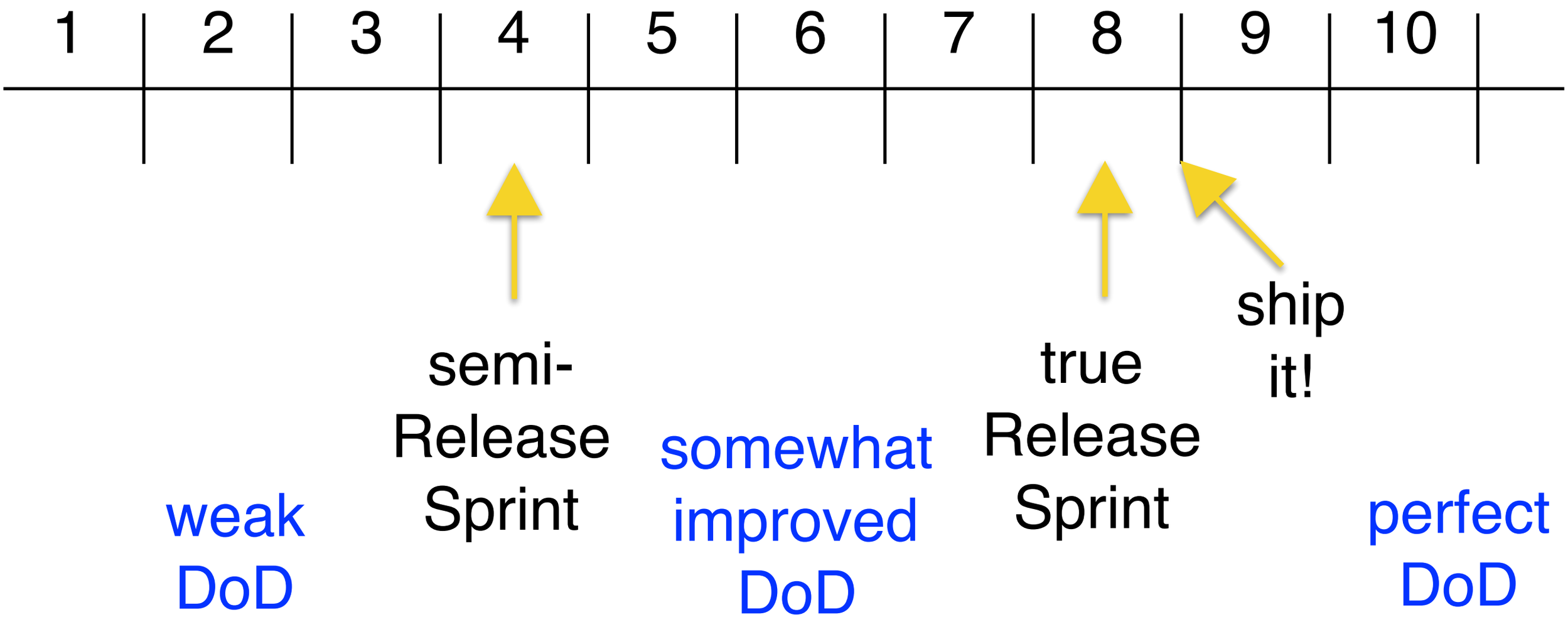
you should NOT need a **Release Sprint**; but it may be a temporary "necessary evil" during early transition to LeSS

what if there were '7' Sprints before the Release Sprint?

(a bad idea; rather, ship every Sprint)

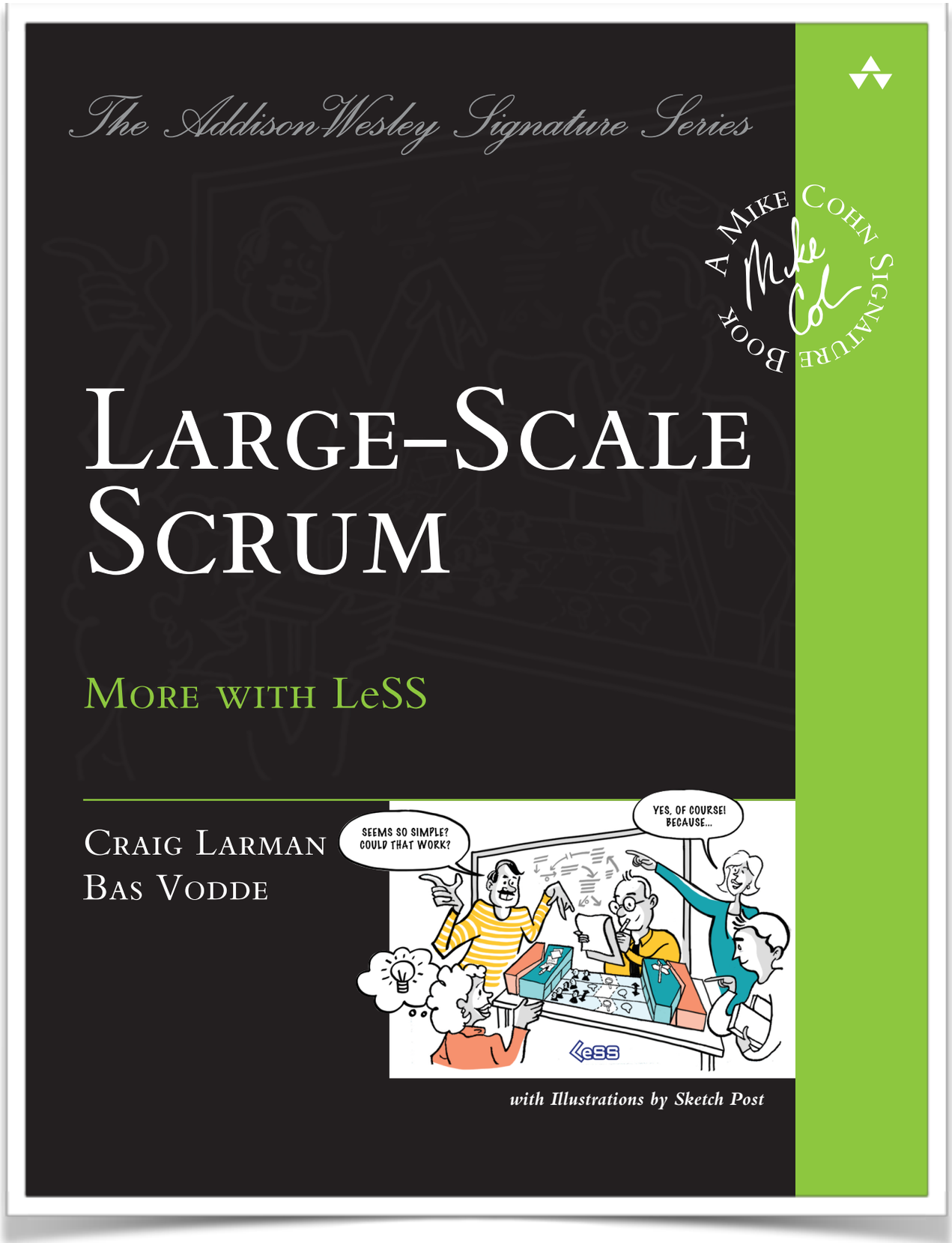
Frequent "Semi-Release Sprints"

Undone Work has:
risk
delay



get to perfect DoD

ASAP



1 More with LeSS 1
2 LeSS 5

LeSS Structure

3 Adoption 41
4 Organize by Customer Value 67
5 Management 111
6 ScrumMasters 133

LeSS Product

7 Product 155
8 Product Owner 171
9 Product Backlog 197
10 Definition of Done 231

LeSS Sprint

11 Product Backlog Refinement 249
12 Sprint Planning 275
13 Coordination and Integration 285
14 Review & Retrospective 313

coach:

> discuss questions

DevOps

since we're exploring shipping, what about DevOps? ...

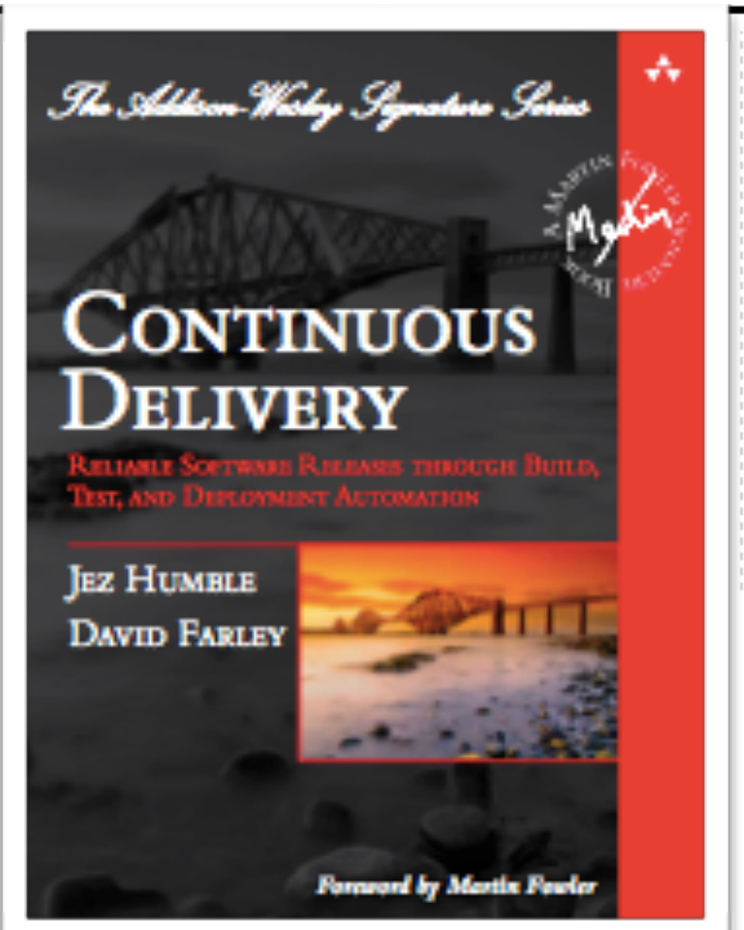
Realising "DevOps" implies…

- extending the DoD to include operations tasks
- increasing the cross-functionality of the feature teams
- dissolving and merging in the Operations group into feature teams

“level 1 support”?

DevOps thought leaders on DevOps …

BLOG PUBLICATIONS TALKS ABOUT



« On Antifragility in Systems and Organizational Architecture

Elisabeth Hendrickson Dis

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

Nor should there be "devops specialists"

Why Segregation of Duties Doesn't Work

Why are Functional Silos Problematic?

why does
“**fake DevOps**”
arise?

“DevOps team”,
“DevOps specialist”

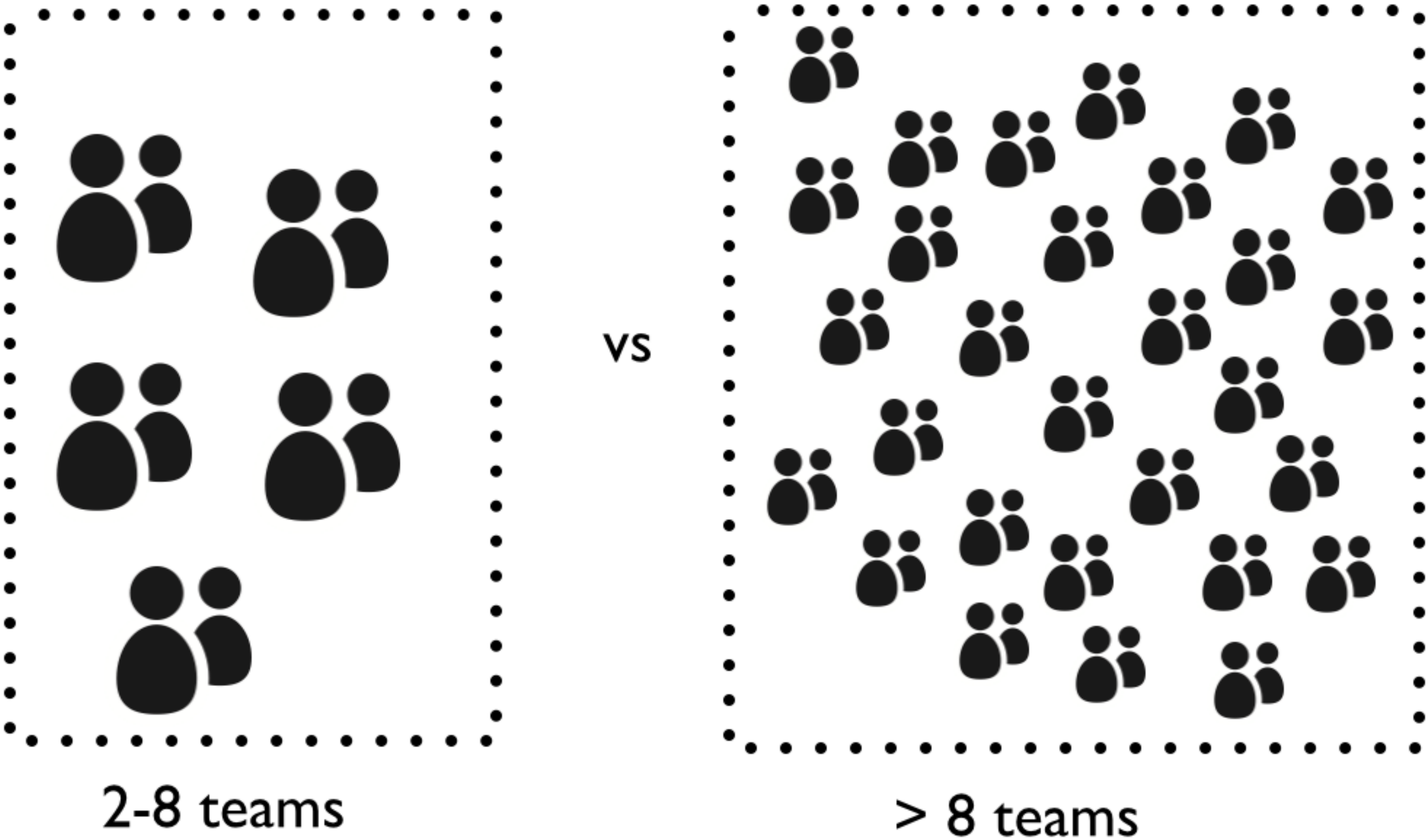
Larman's Laws of Organizational Behavior

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", "religious", and "needing pragmatic customization for local concerns" — which deflects from addressing weaknesses and manager/specialist status quo.**
4. **As a corollary to (1), if after *changing the change* some managers and single-specialists are still displaced, they become "coaches/trainers" for the change, frequently reinforcing (2) and (3).**
5. **Culture follows structure (or behavior/mindset follows system)**

LeSS Huge

LeSS Huge Framework

2 Frameworks: LeSS & LeSS Huge



team

> sketch a systems model, given:

> one Product Owner, for a product with 20 teams

> what are the noteworthy variables?

"8" is not a magic number

Requirement Areas

Item	Requirement Area	...
B	market on-boarding	
C	trade processing	
D	asset servicing	
F	market on-boarding	
...		

coach

> for some participant in a "huge" context, your possible requirement areas?

Requirement Areas are SLOWLY DYNAMIC

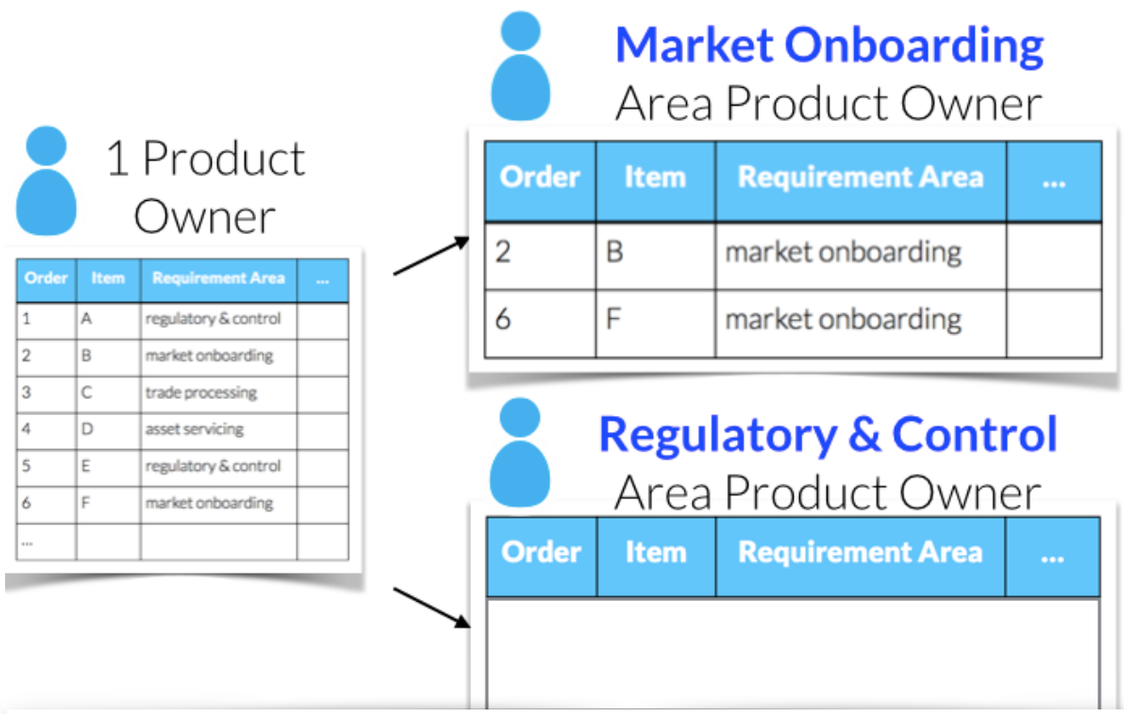
Requirement Areas Are…

> … *not* the same "domains" in Domain-Driven Design

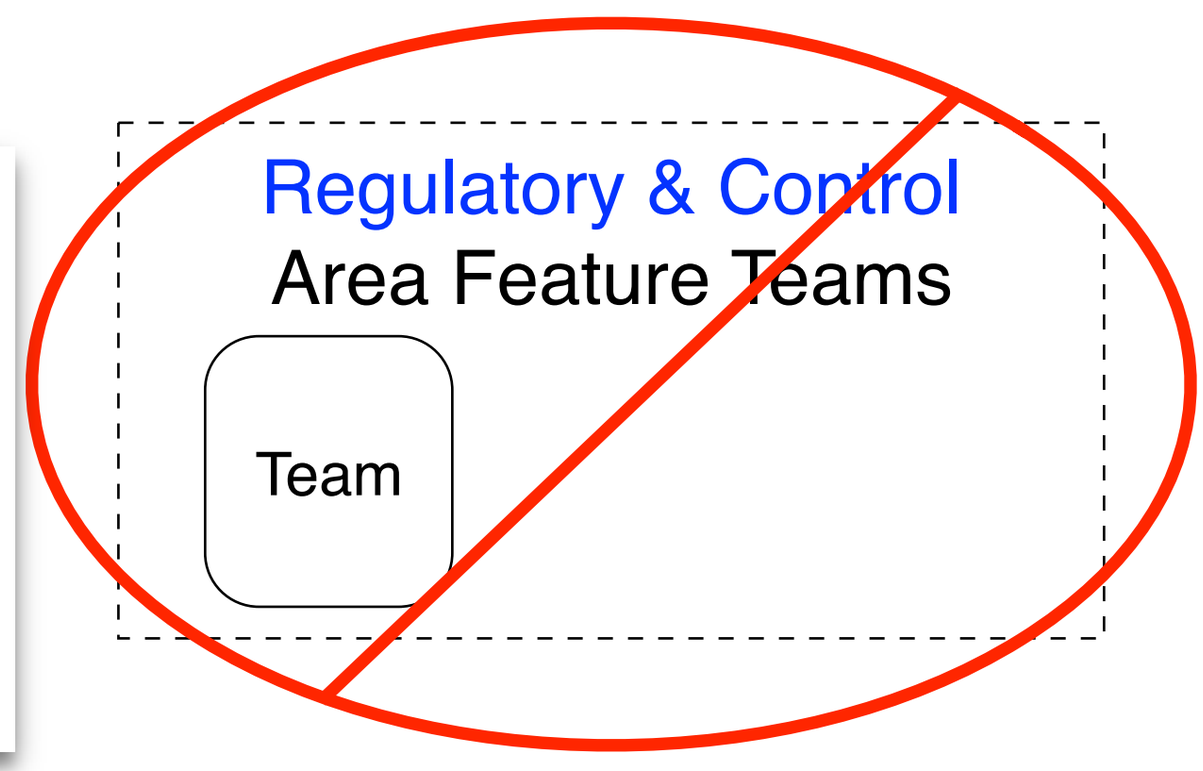
> though there can be an overlap

> Requirement Areas can be more dynamic and market driven

An Area can't have only 1 Team



Exceptions? When starting to grow a new area, that is "sure" to become big.



team

> sketch a systems model, given:

> **many small (e.g. 1 or 2-team) Requirement Areas**

> reminder: each RA has an Area Backlog, and teams are in 1 area

> what are the noteworthy variables?

LeSS Rule(s)

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

"Area Backlog"

Item	Requirement Area	...
B	market on-boarding	
F	market on-boarding	

Item	Requirement Area	...
B	market on-boarding	
C	trade processing	
D	asset servicing	
F	market on-boarding	
...		

Area Backlogs via **Views**

Item	Requirement Area	...
B	market on-boarding	
F	market on-boarding	

Item	Requirement Area	...
B	market on-boarding	
C	trade processing	
D	asset servicing	
F	market on-boarding	
...		

a VIEW for one requirement area

it is NOT a separate artifact

Area Backlogs via **Separate Artifacts**

Item	Requirement Area	...
B	market on-boarding	
F	market on-boarding	

Item	Requirement Area	...
B	market on-boarding	
C	trade processing	
D	asset servicing	
F	market on-boarding	
...		

an ARTIFACT for one requirement area

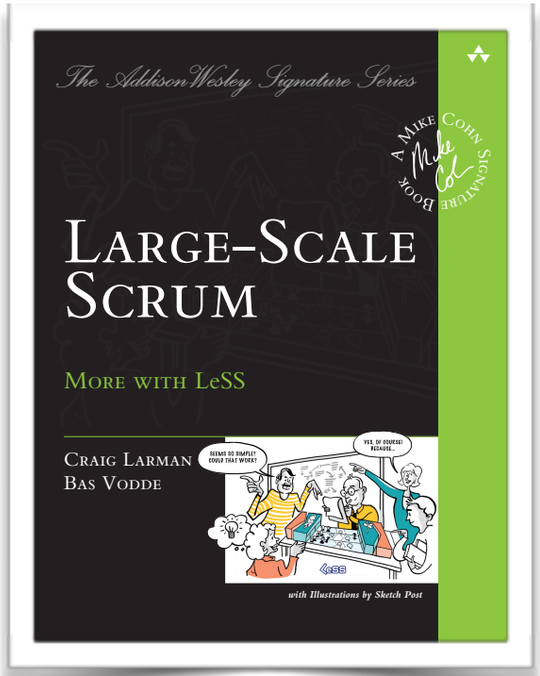
it IS a separate artifact

Area Backlogs as **views**

vs

Area Backlogs as **separate artifacts**

Area Backlogs via **Separate Artifacts**



Contents

Product Backlog in LeSS 207

- Guide: Don't "Manage Dependencies" but Minimize Constraints 208
- Guide: Take a Bite 212
- Guide: Dealing with Parents 214
- Guide: Handling Special Items 217
- Guide: Tools for Large Product Backlogs 220
- Guide: More Outcome, less Output 223

LeSS Huge 225

- Guide: Area Backlogs 225
- Guide: Three Levels Max 232
- Guide: New Area for Giant Requirement 233
- Guide: Handling Gigantic Requirements 234

"Area Product Owners"

1
"overall"
Product
Owner

Item	Requirement Area	...
B	market on-boarding	
C	trade processing	
D	asset servicing	
F	market on-boarding	
...		

Market Onboarding
Area Product Owner

Item	Requirement Area	...
B	market on-boarding	
F	market on-boarding	

Regulatory & Control
Area Product Owner

Item	Requirement Area	...

. . .

"Product Owner Team"

1
"overall"
Product
Owner

Item	Requirement Area	...
B	market on-boarding	
C	trade processing	
D	asset servicing	
F	market on-boarding	
...		

Market Onboarding
Area Product Owner

Item	Requirement Area	...
B	market on-boarding	
F	market on-boarding	

Regulatory & Control
Area Product Owner

Item	Requirement Area	...

. . .

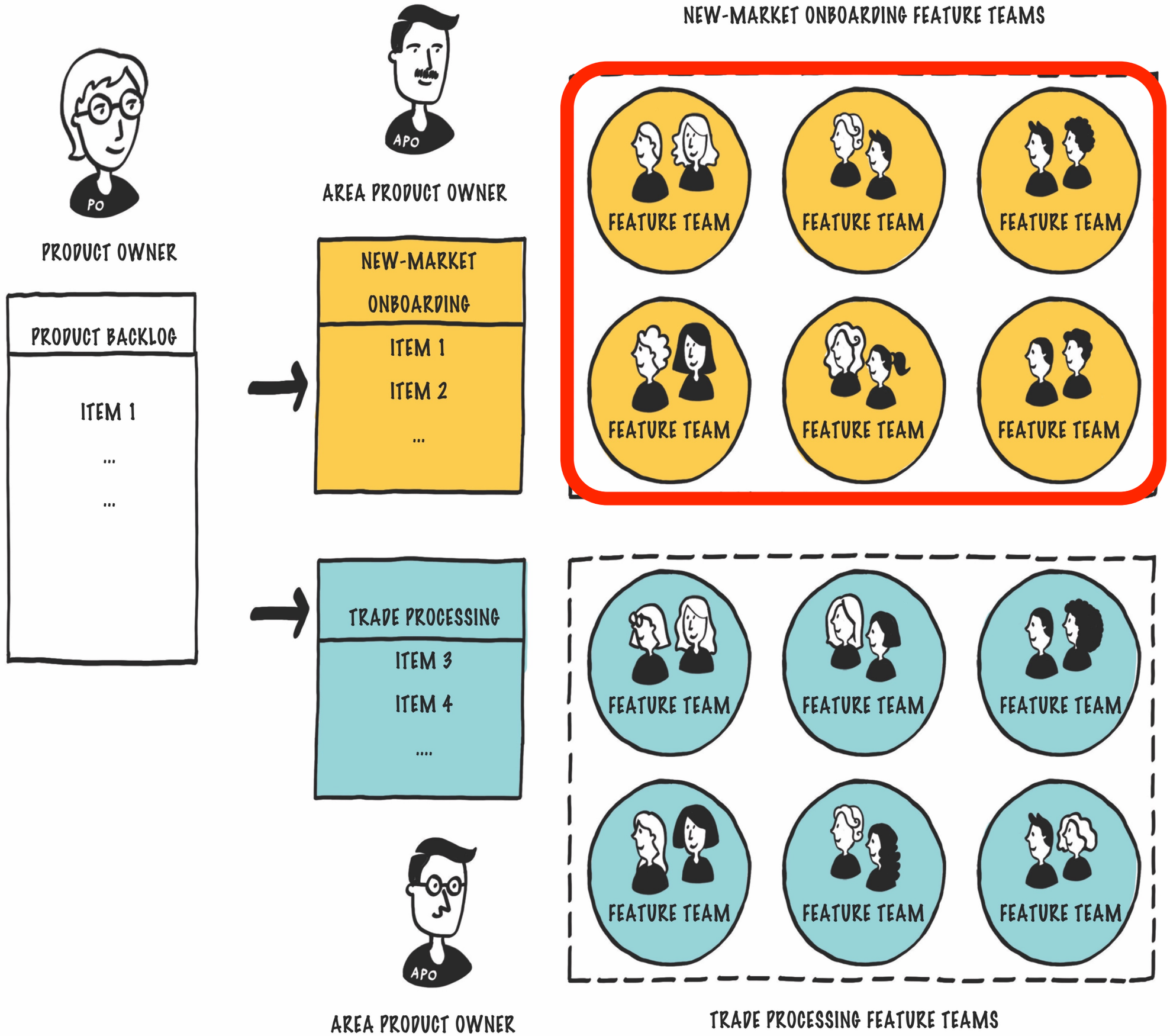
~~PO Team with analysts, UX/UI designers, architects, project managers~~

responsibilities of the Product Owner?

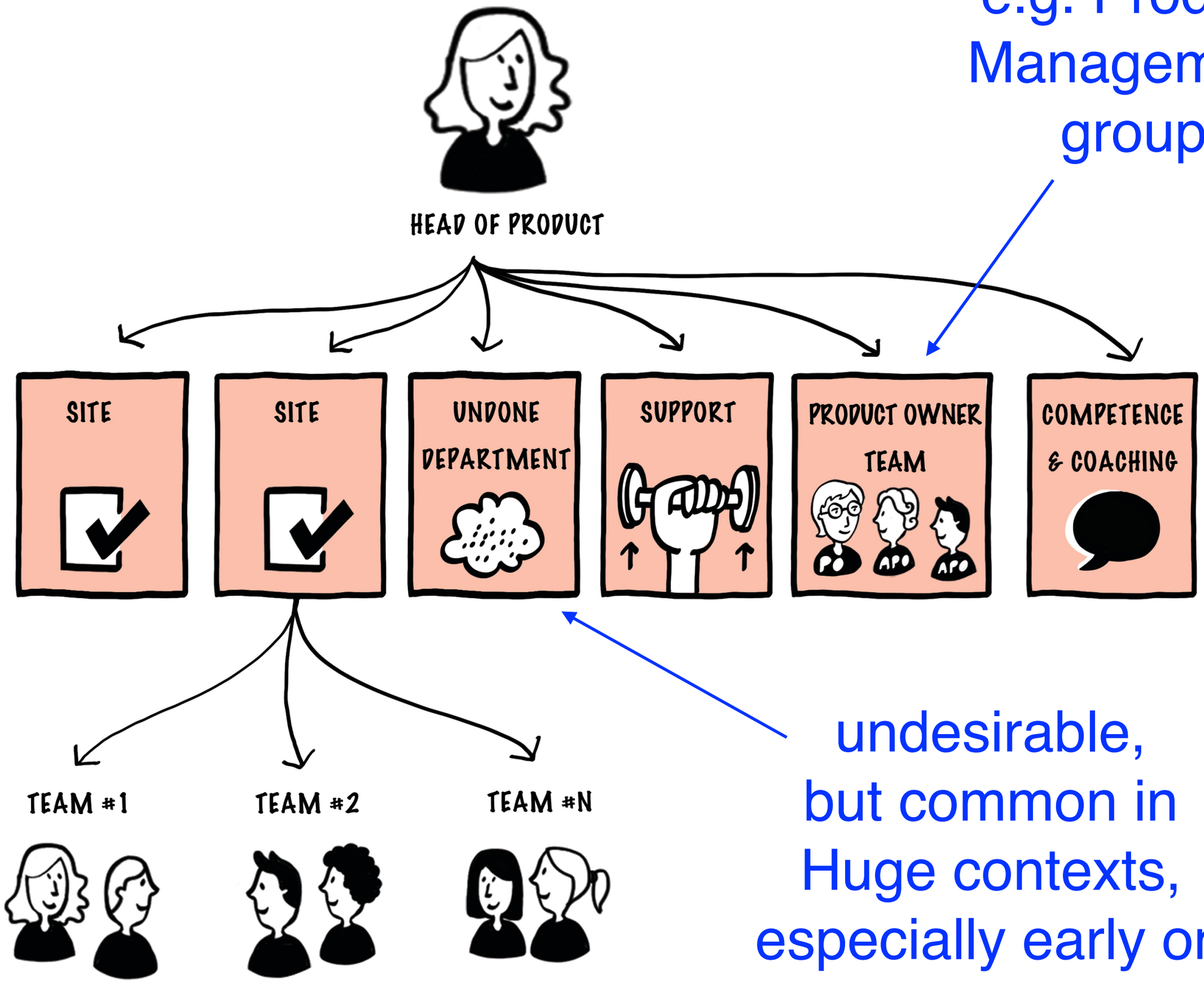
Guide: Product Owner Team Meeting

- issue: losing whole-product focus or alignment between areas in the choice of themes and items
- each Area Product Owner shares their situation and upcoming goals, and they discuss opportunities to align
- Product Owner can provide high-level guidance
- discuss the results of the previous Sprint Review meetings in each Requirement Area, as input to planning
- include some team reps for learning and feedback
- include 1 Scrum Master to support reflection and improvement

"Area Feature Teams"



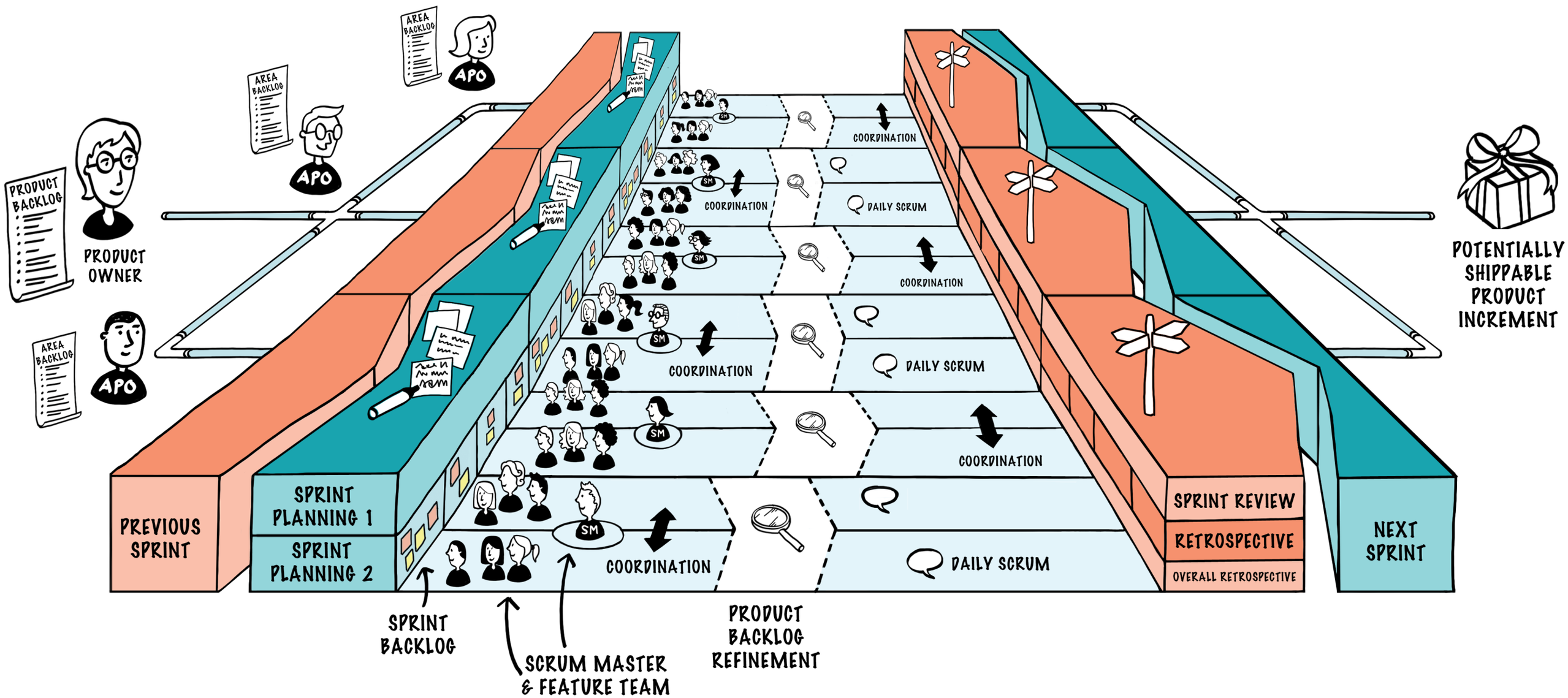
Huge Structure

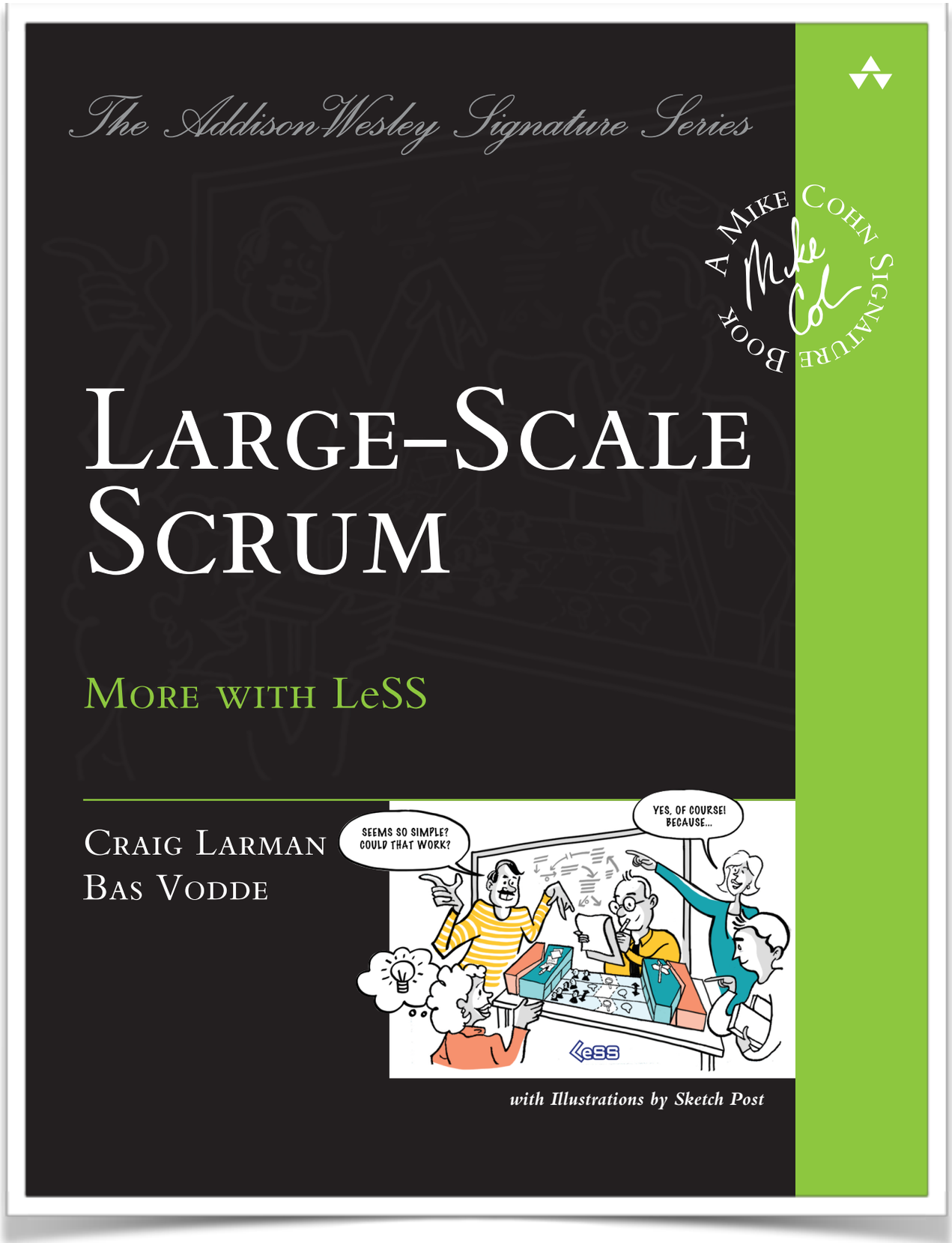


e.g. Product Management group

undesirable, but common in Huge contexts, especially early on

LeSS Huge: “Stack of LeSS”





1 More with LeSS 1
2 LeSS 5

LeSS Structure

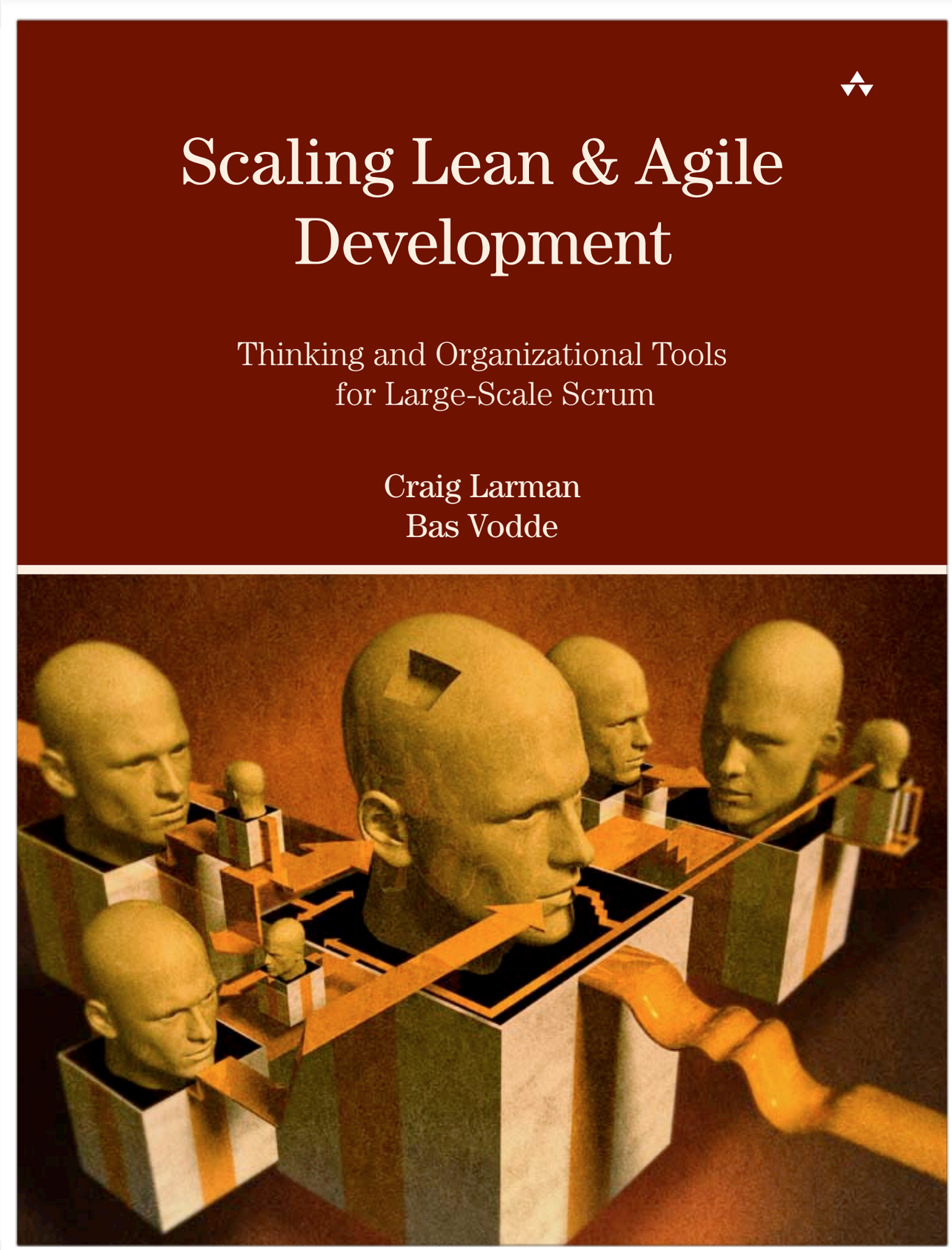
3 Adoption 41
4 Organize by Customer Value 67
5 Management 111
6 ScrumMasters 133

LeSS Product

7 Product 155
8 Product Owner 171
9 Product Backlog 197
10 Definition of Done 231

LeSS Sprint

11 Product Backlog Refinement 249
12 Sprint Planning 275
13 Coordination and Integration 285
14 Review & Retrospective 313



Thinking Tools

2. Systems Thinking
3. Lean Thinking
4. Queueing Theory
5. False Dichotomies
6. Be Agile

Organizational Tools

7. Feature Teams
8. Teams
9. Requirement Areas
10. Organization
11. Large-Scale Scrum

LeSS Huge Adoption

Rarely… Total All-at-Once

- maybe, if…
  - relatively small (e.g. 12 teams, 2 Areas)
  - lifetime short
  - single-specialization low
  - one site
- warning! do **NOT** underestimate the **massive** amount of learning and coaching required

but more common…

LeSS Rule(s)

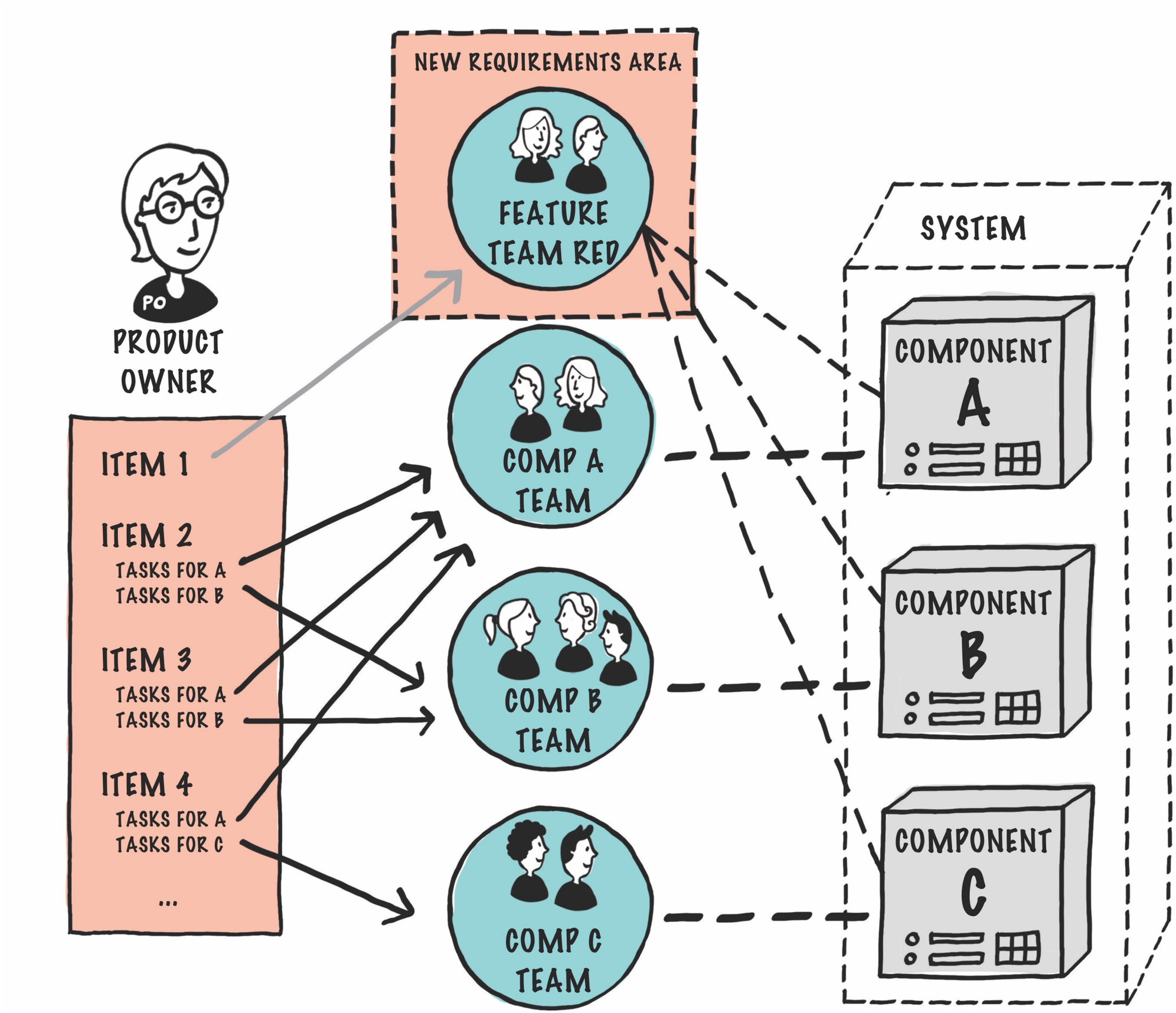
LeSS Huge adoptions, including the structural changes, are done with an **evolutionary incremental** approach.

Remember each day: LeSS Huge adoptions take months or years, infinite patience, and sense of humor.

Guide: Evolutionary Incremental Adoption

- two alternatives:
  - **focused deeper adoption at a part of the product group**
  - gradual incremental adoption over the whole product group

Guide: Parallel Organizations



Guide: Parallel Organizations

- *focused deeper adoption at a part of the product group*
- gradual, low-risk, well-suited for **huge** LeSS Huge product groups
- key drawback? takes a *long* time
- must: abandon private code
- don't: allow branching

Guide: One Requirement Area at a Time

- *focused deeper adoption at a part of the product group*
- "all at once" in only one Requirement Area
- is there some high-benefit low-risk area?
- wicked problem: the new org model exists interacting closely with the old model
- must: abandon private code

Guide: Usual Recommendation…

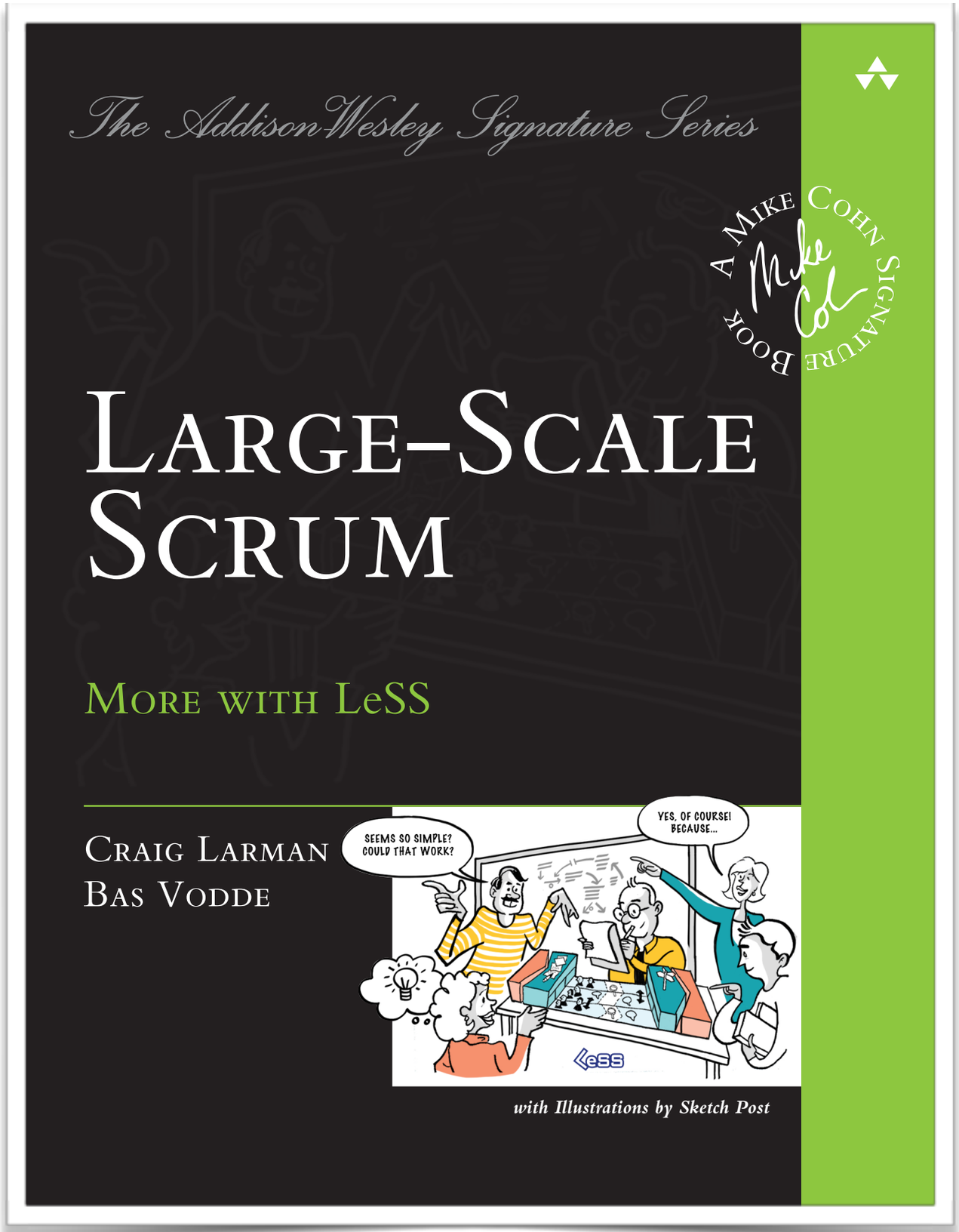
Parallel Organizations

in

One Requirement Area at a Time

Guide: Transitioning to Feature Teams

- *gradual incremental adoption over the whole product group*
- **gradually expand component team responsibility**
- use **feature-team adoption map**
- context: huge, many sites, high learning across sites required
- problems:
  - drawbacks of both feature and component teams while not giving the best benefits
  - hard to adopt customer-centric Requirement Areas when the teams are still component teams



1 More with LeSS 1
2 LeSS 5

LeSS Structure

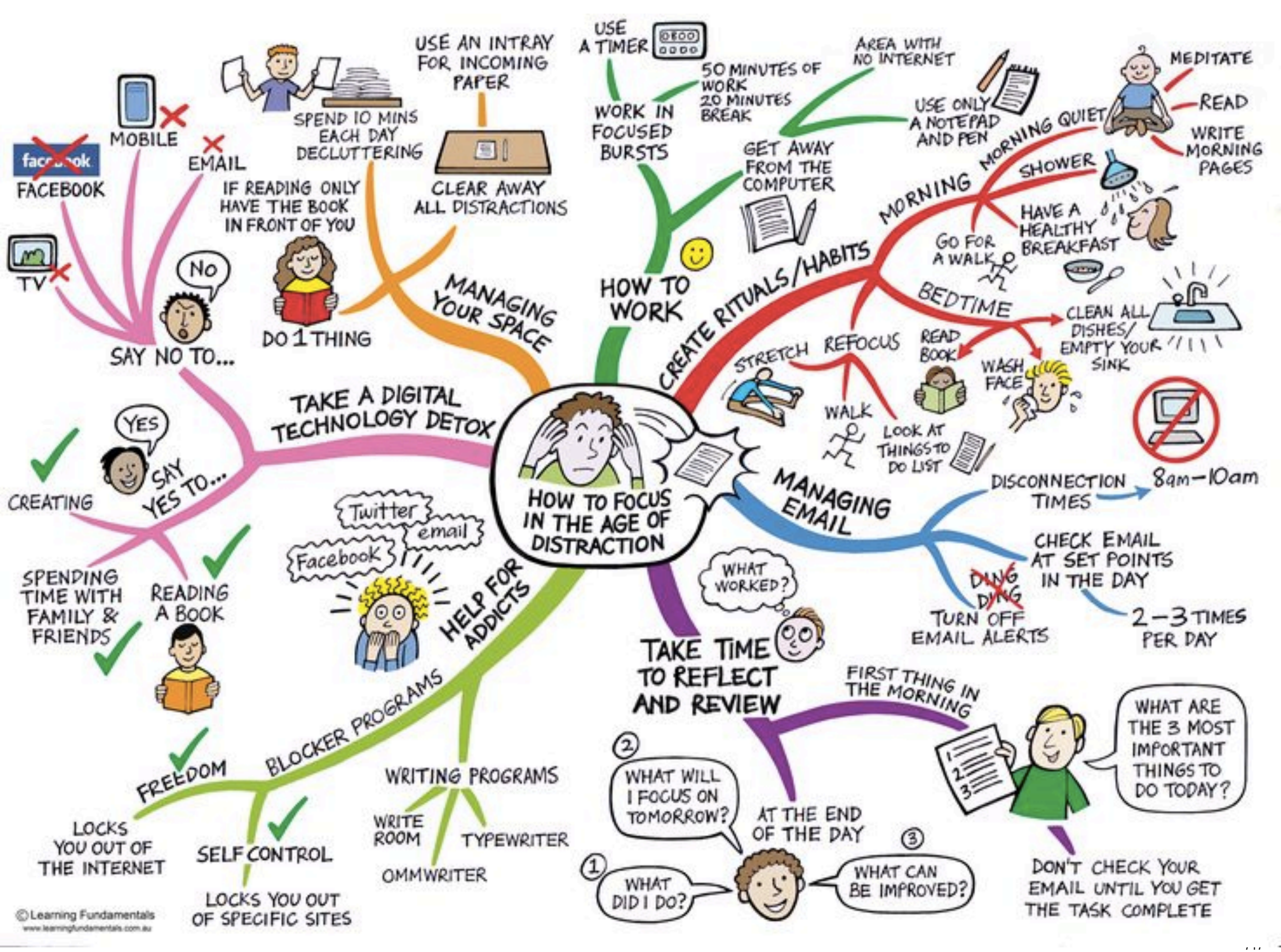
3 Adoption 41
4 Organize by Customer Value 67
5 Management 111
6 ScrumMasters 133

LeSS Product

7 Product 155
8 Product Owner 171
9 Product Backlog 197
10 Definition of Done 231

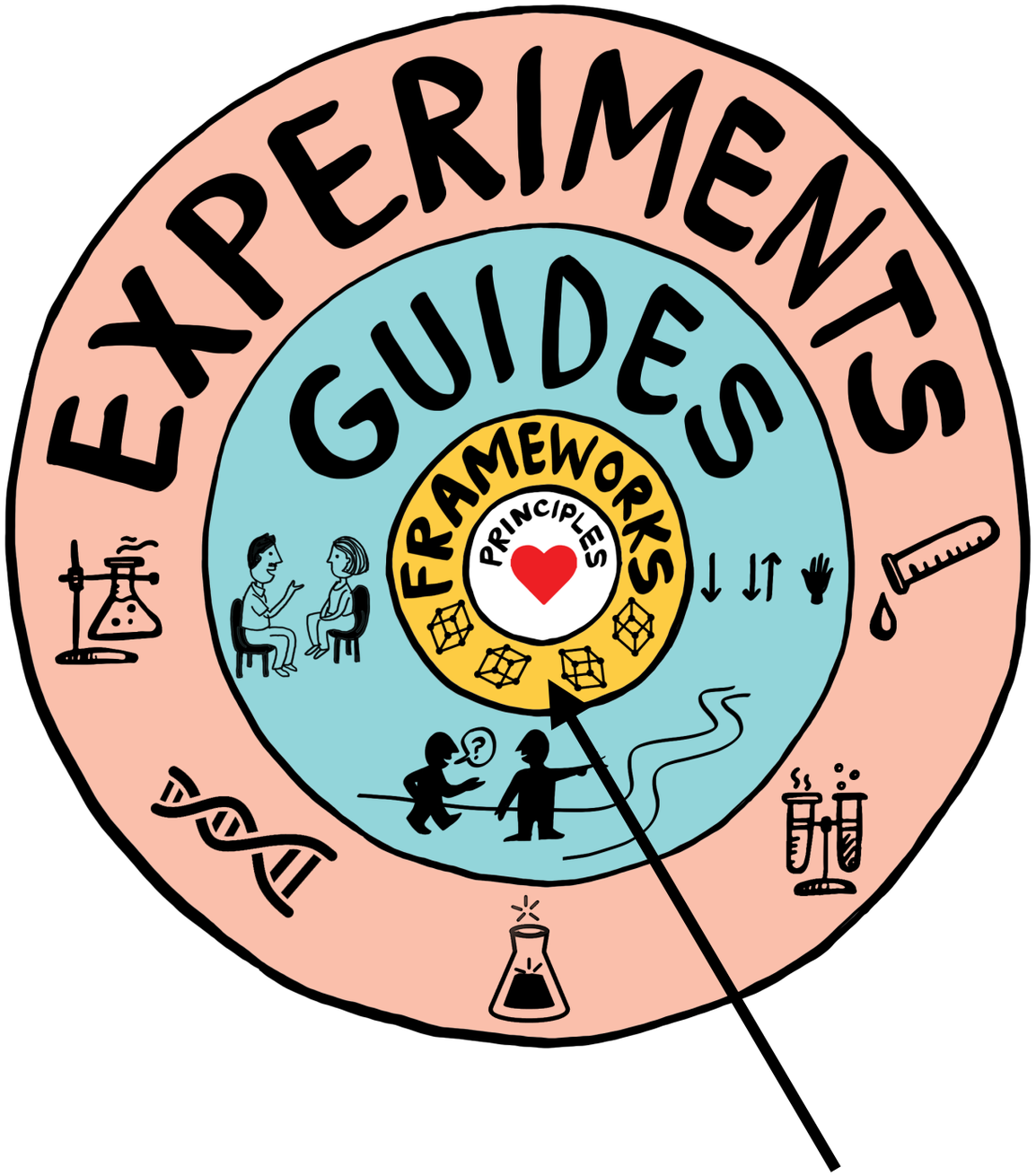
LeSS Sprint

11 Product Backlog Refinement 249
12 Sprint Planning 275
13 Coordination and Integration 285
14 Review & Retrospective 313


HOW TO FOCUS IN THE AGE OF DISTRACTION
TAKE A DIGITAL TECHNOLOGY DETOX
SAY NO TO...
No
MOBILE
EMAIL
FACEBOOK
TV
SPENDING TIME WITH FAMILY & FRIENDS
SAY YES TO...
Yes
CREATING
READING A BOOK
MANAGING YOUR SPACE
SPEND 10 MINS EACH DAY DECLUTTERING
USE AN INTRAY FOR INCOMING PAPER
CLEAR AWAY ALL DISTRACTIONS
IF READING ONLY HAVE THE BOOK IN FRONT OF YOU
DO 1 THING
HOW TO WORK
USE A TIMER
WORK IN FOCUSED BURSTS
50 MINUTES OF WORK 20 MINUTES BREAK
GET AWAY FROM THE COMPUTER
AREA WITH NO INTERNET
USE ONLY A NOTEPAD AND PEN
CREATE RITUALS/HABITS
MORNING
MORNING QUIET
MEDITATE
READ
WRITE MORNING PAGES
SHOWER
HAVE A HEALTHY BREAKFAST
GO FOR A WALK
BEDTIME
CLEAN ALL DISHES/ EMPTY YOUR SINK
READ BOOK
WASH FACE
STRETCH
REFOCUS
WALK
LOOK AT THINGS TO DO LIST
MANAGING EMAIL
DISCONNECTION TIMES
8am–10am
CHECK EMAIL AT SET POINTS IN THE DAY
2–3 TIMES PER DAY
TURN OFF EMAIL ALERTS
HELP FOR ADDICTS
Twitter
email
Facebook
BLOCKER PROGRAMS
FREEDOM
LOCKS YOU OUT OF THE INTERNET
SELF CONTROL
LOCKS YOU OUT OF SPECIFIC SITES
WRITING PROGRAMS
WRITE ROOM
TYPEWRITER
OMMWRITER
TAKE TIME TO REFLECT AND REVIEW
WHAT WORKED?
FIRST THING IN THE MORNING
WHAT ARE THE 3 MOST IMPORTANT THINGS TO DO TODAY?
DON'T CHECK YOUR EMAIL UNTIL YOU GET THE TASK COMPLETE
AT THE END OF THE DAY
1 WHAT DID I DO?
2 WHAT WILL I FOCUS ON TOMORROW?
3 WHAT CAN BE IMPROVED?
© Learning Fundamentals
www.learningfundamentals.com.au

LeSS Rules

LeSS Rules



the LeSS rules
define the 2 frameworks

“we don’t want rules”

so why do they exist?

Scaling Sweet Spot & **Shu**-Ha-Ri

“barely sufficient methodology”

Why “Rules”? **Shu** & Focus on Creating…

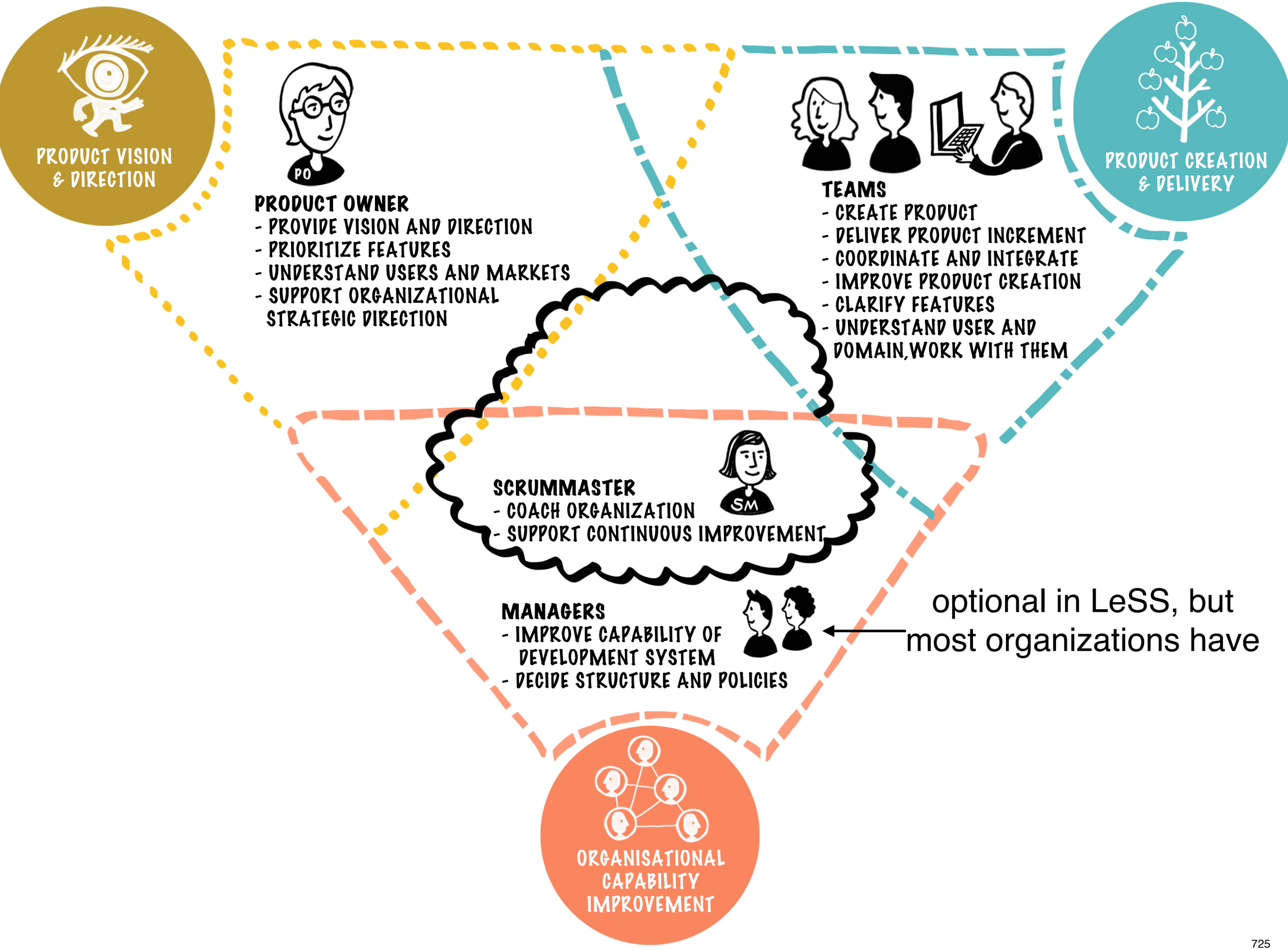
>transparency

>whole-product focus

>global systems Optimisation

>empirical process control

LeSS Roles


PRODUCT VISION & DIRECTION
PO
PRODUCT OWNER
- PROVIDE VISION AND DIRECTION
- PRIORITIZE FEATURES
- UNDERSTAND USERS AND MARKETS
- SUPPORT ORGANIZATIONAL STRATEGIC DIRECTION
PRODUCT CREATION & DELIVERY
TEAMS
- CREATE PRODUCT
- DELIVER PRODUCT INCREMENT
- COORDINATE AND INTEGRATE
- IMPROVE PRODUCT CREATION
- CLARIFY FEATURES
- UNDERSTAND USER AND DOMAIN,WORK WITH THEM
SM
SCRUMMASTER
- COACH ORGANIZATION
- SUPPORT CONTINUOUS IMPROVEMENT
MANAGERS
- IMPROVE CAPABILITY OF DEVELOPMENT SYSTEM
- DECIDE STRUCTURE AND POLICIES
optional in LeSS, but most organizations have
ORGANISATIONAL CAPABILITY IMPROVEMENT

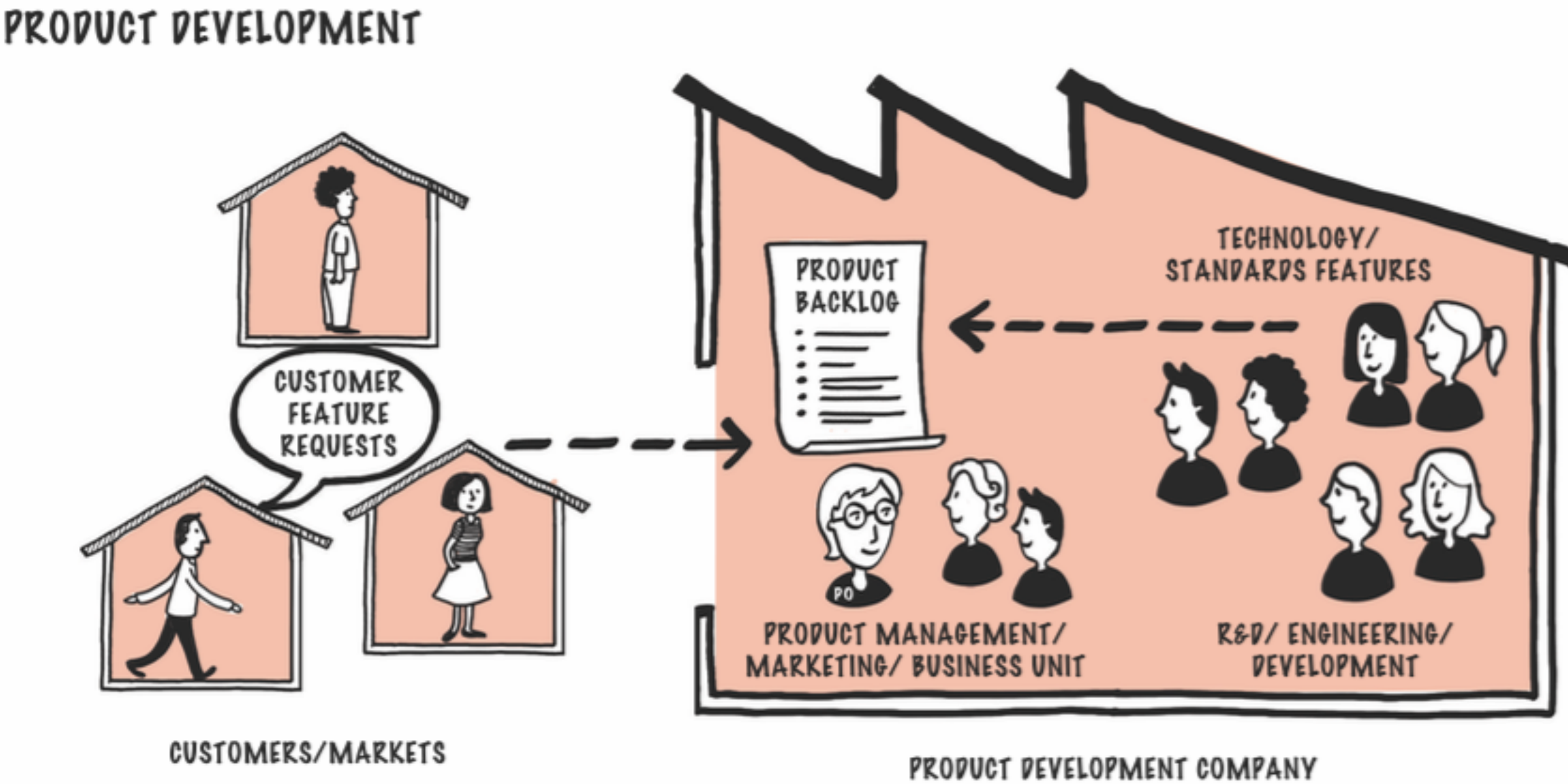
Where is the Product Owner?

3 Types of Development

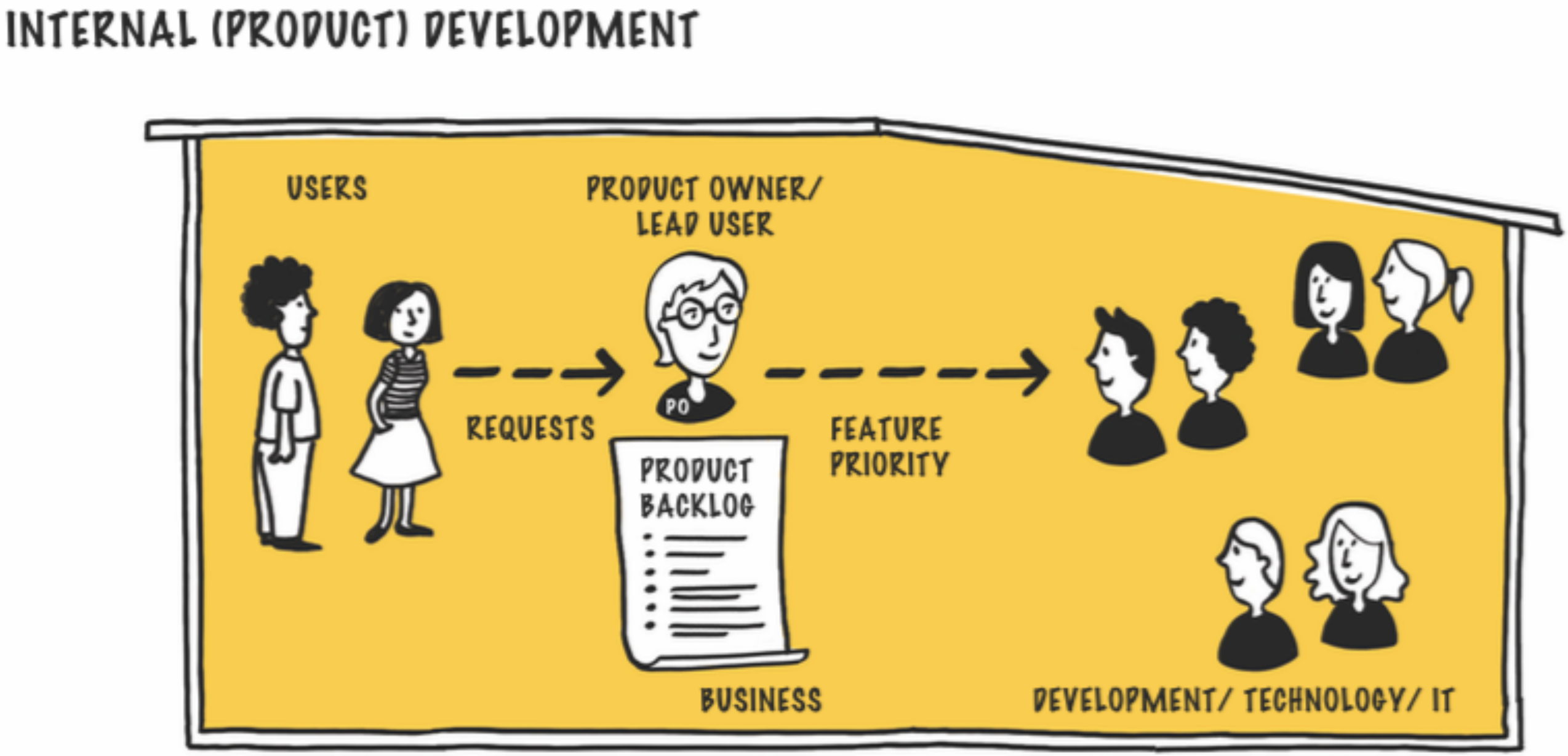
Why Learn 3 Types of Development?

- >where is the Product Owner?
- >major “pattern” groupings?

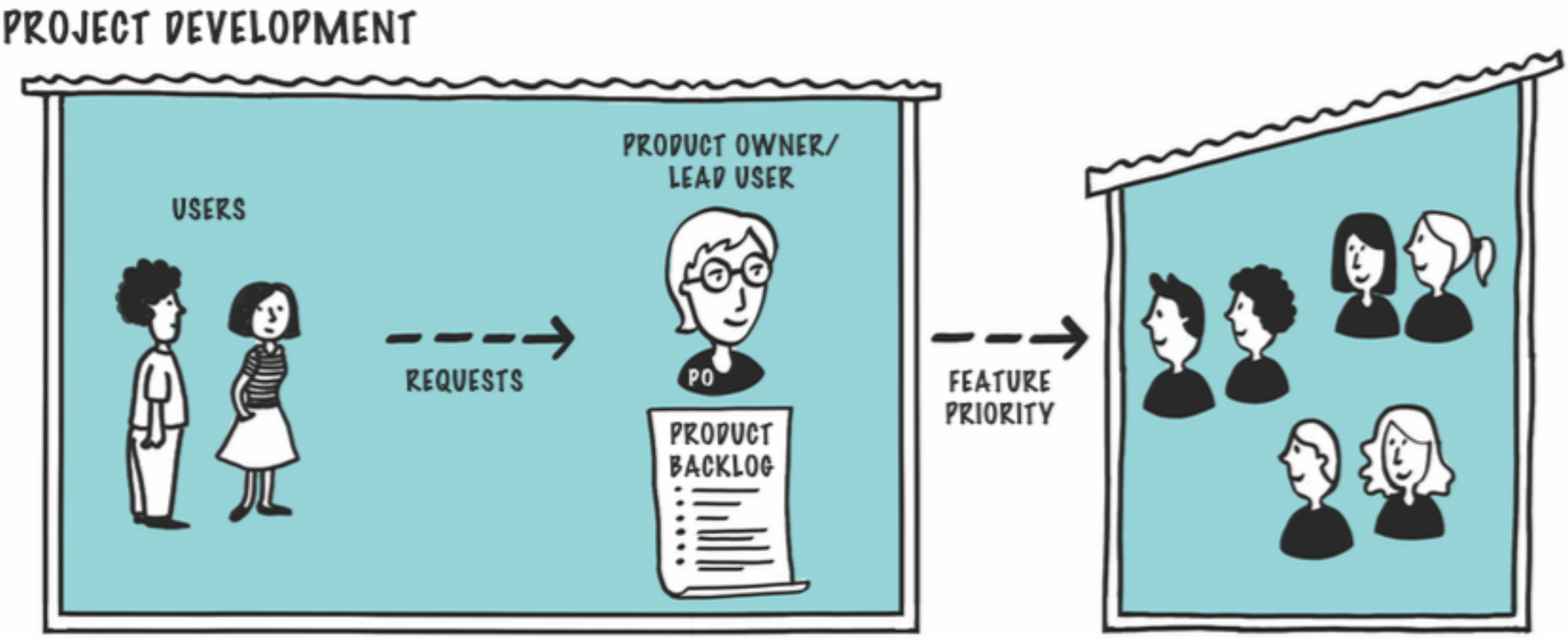
(External) Product Development



Internal (Product) Development

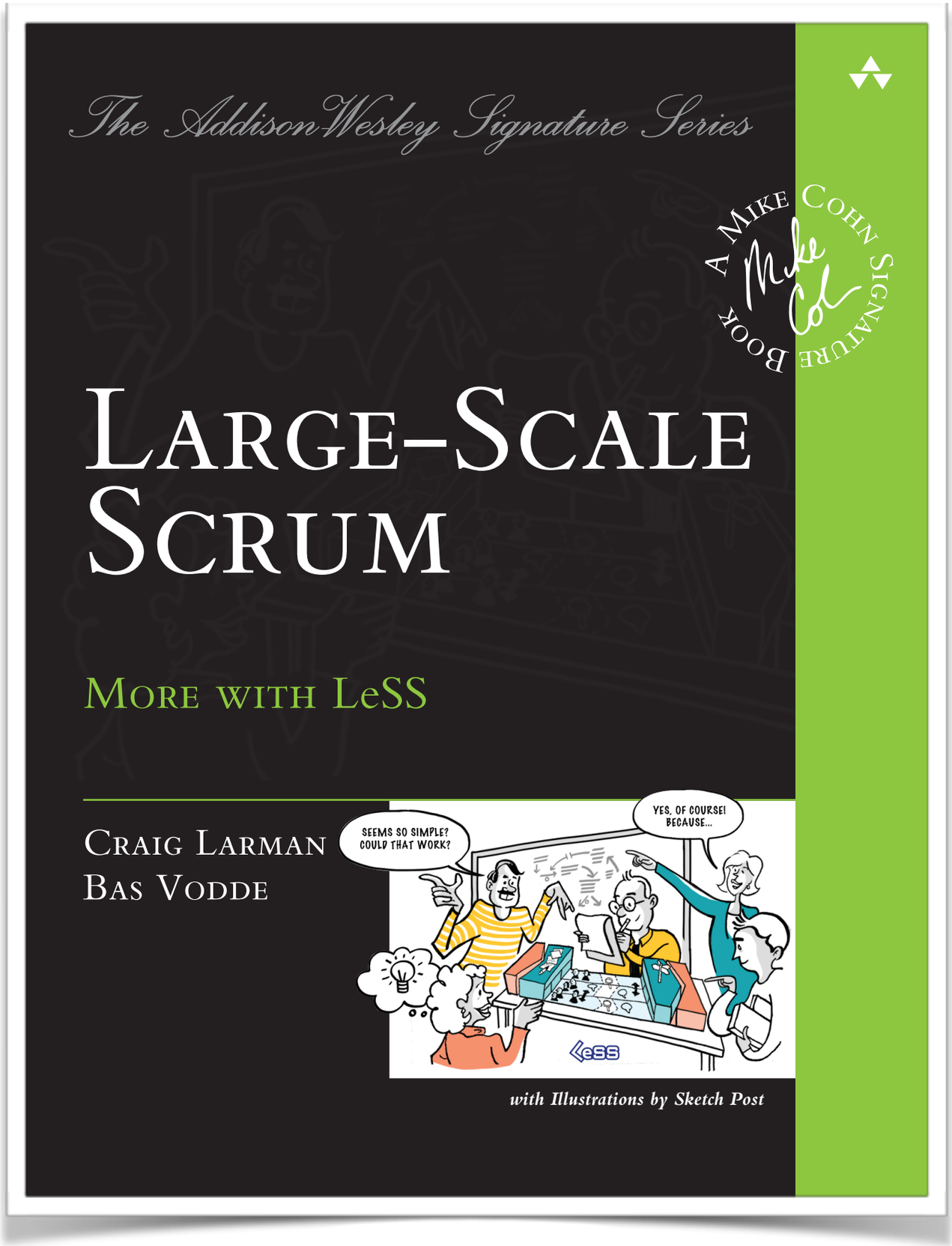


(Outsourced) Project (Product) Development



RECEIVER & USER OF PRODUCT

OUTSOURCER



1 More with LeSS 1
2 LeSS 5

LeSS Structure

3 Adoption 41
4 Organize by Customer Value 67
5 Management 111
6 ScrumMasters 133

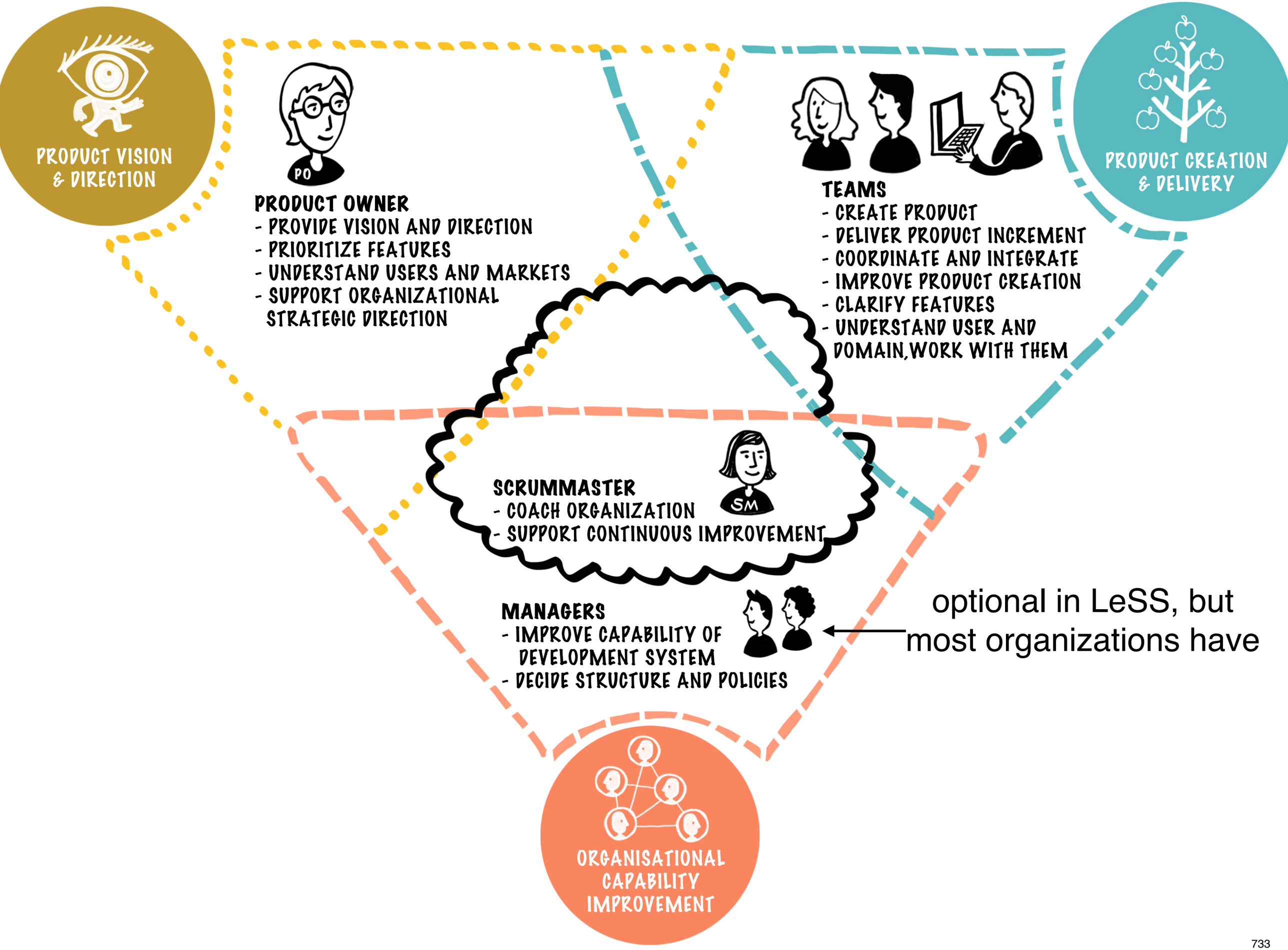
LeSS Product

7 Product 155
8 Product Owner 171
9 Product Backlog 197
10 Definition of Done 231

LeSS Sprint

11 Product Backlog Refinement 249
12 Sprint Planning 275
13 Coordination and Integration 285
14 Review & Retrospective 313

Product Owner in LeSS


PRODUCT VISION & DIRECTION
PO
PRODUCT OWNER
- PROVIDE VISION AND DIRECTION
- PRIORITIZE FEATURES
- UNDERSTAND USERS AND MARKETS
- SUPPORT ORGANIZATIONAL STRATEGIC DIRECTION
PRODUCT CREATION & DELIVERY
TEAMS
- CREATE PRODUCT
- DELIVER PRODUCT INCREMENT
- COORDINATE AND INTEGRATE
- IMPROVE PRODUCT CREATION
- CLARIFY FEATURES
- UNDERSTAND USER AND DOMAIN,WORK WITH THEM
SM
SCRUMMASTER
- COACH ORGANIZATION
- SUPPORT CONTINUOUS IMPROVEMENT
MANAGERS
- IMPROVE CAPABILITY OF DEVELOPMENT SYSTEM
- DECIDE STRUCTURE AND POLICIES
optional in LeSS, but most organizations have
ORGANISATIONAL CAPABILITY IMPROVEMENT

LeSS Rule(s)

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

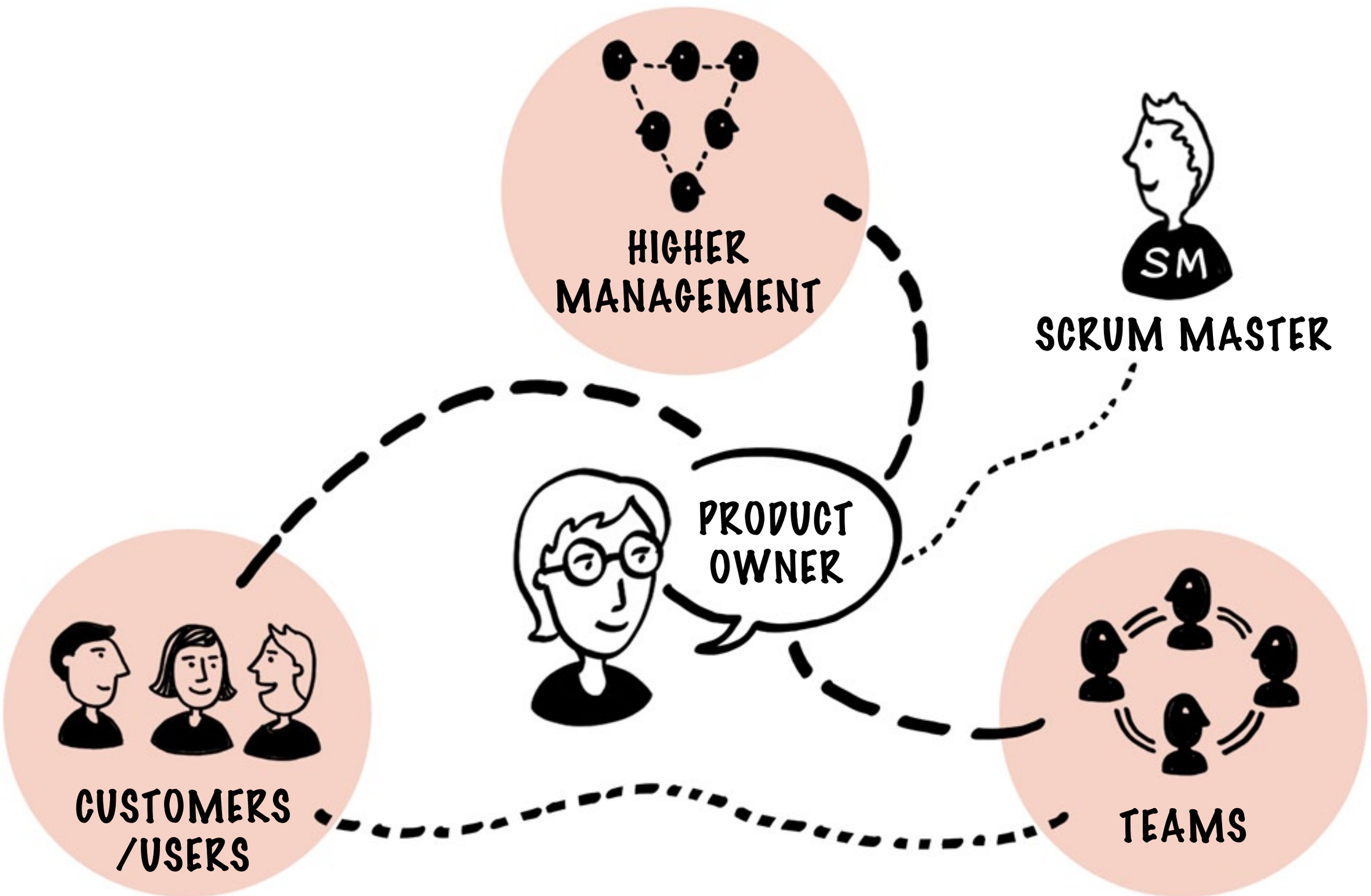
The Product Owner shouldn't work alone on Product Backlog refinement; it is mostly done by the multiple Teams working directly with customers, users, and other stakeholders.

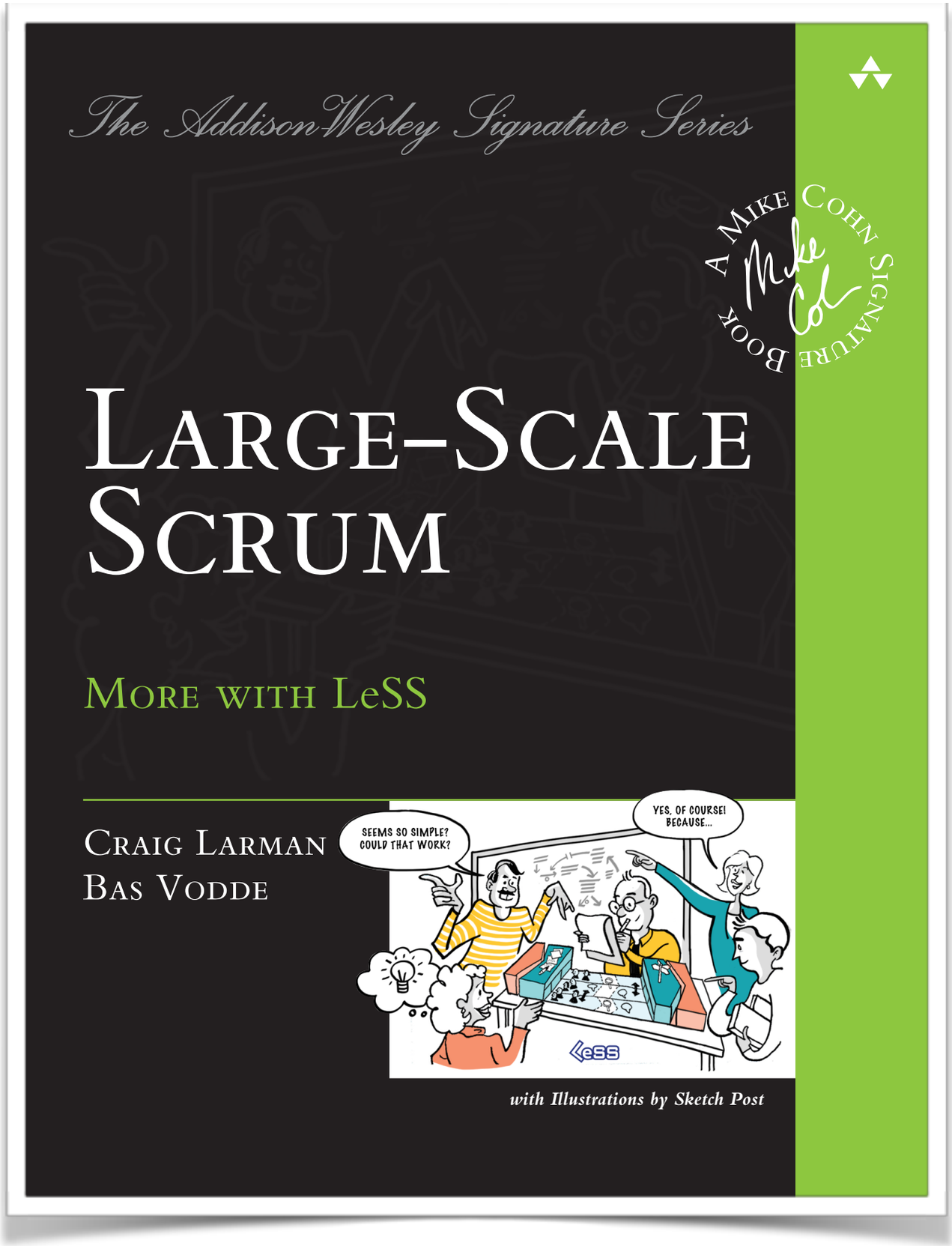
All prioritization (ordering) goes through the Product Owner, but clarification is as much as possible directly between the Teams and customer, users, and other stakeholders.

Guides in LeSS: Product Owner

- Who Should be Product Owner?
- Who are those Users/ Customers?
- Prioritization over Clarification
- Don't Do It
- Helpers
- Five Relationships
- Customer Collaborations over...
- Ship At Least Every Sprint
- Don't Be Nice
- Let Go
- Don't Let Undone Work be Your Undoing

5 relationships





1 More with LeSS 1
2 LeSS 5

LeSS Structure

3 Adoption 41
4 Organize by Customer Value 67
5 Management 111
6 ScrumMasters 133

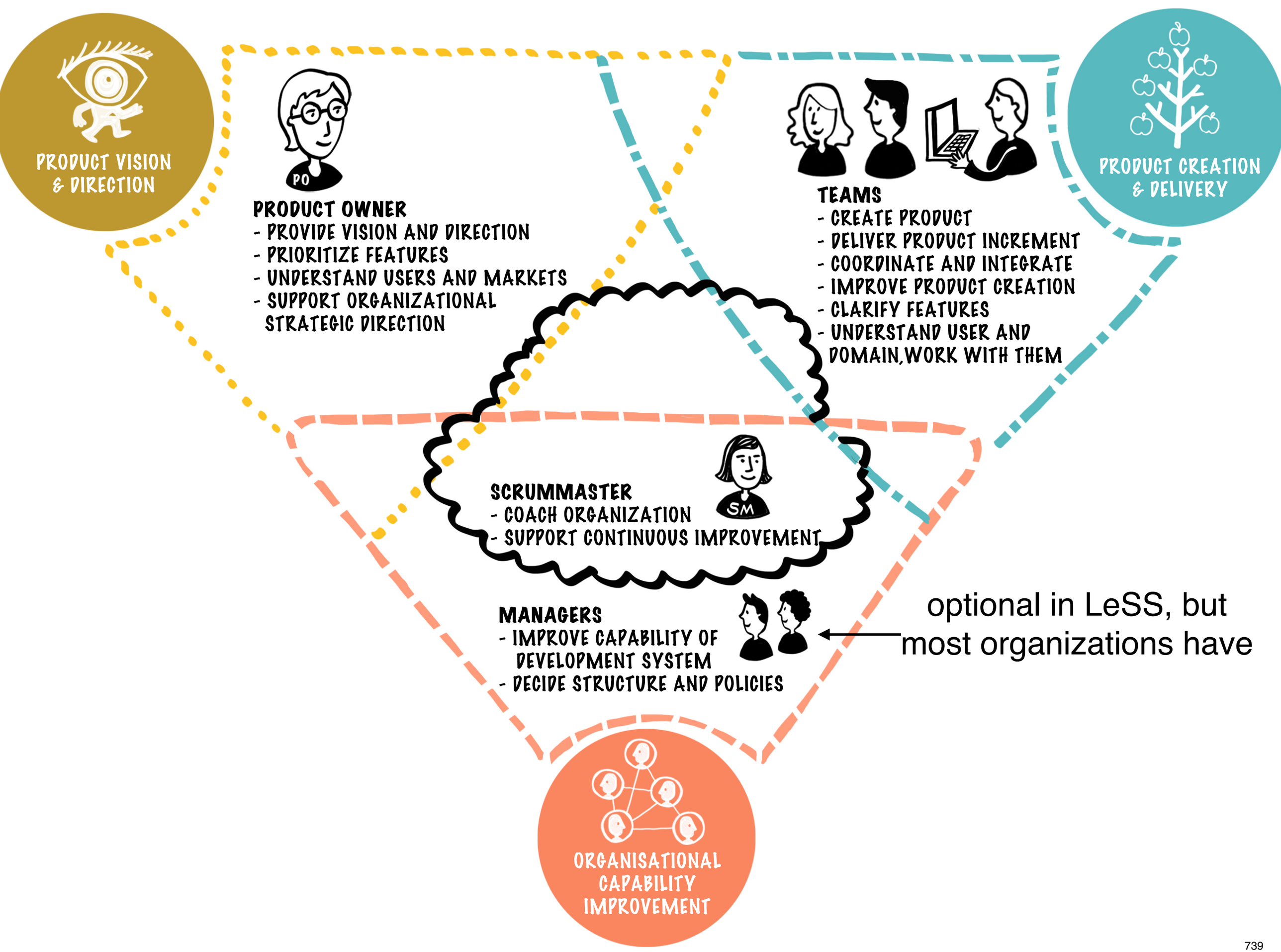
LeSS Product

7 Product 155
8 Product Owner 171
9 Product Backlog 197
10 Definition of Done 231

LeSS Sprint

11 Product Backlog Refinement 249
12 Sprint Planning 275
13 Coordination and Integration 285
14 Review & Retrospective 313

Scrum Master in LeSS


PRODUCT VISION & DIRECTION
PO
PRODUCT OWNER
- PROVIDE VISION AND DIRECTION
- PRIORITIZE FEATURES
- UNDERSTAND USERS AND MARKETS
- SUPPORT ORGANIZATIONAL STRATEGIC DIRECTION
PRODUCT CREATION & DELIVERY
TEAMS
- CREATE PRODUCT
- DELIVER PRODUCT INCREMENT
- COORDINATE AND INTEGRATE
- IMPROVE PRODUCT CREATION
- CLARIFY FEATURES
- UNDERSTAND USER AND DOMAIN,WORK WITH THEM
SM
SCRUMMASTER
- COACH ORGANIZATION
- SUPPORT CONTINUOUS IMPROVEMENT
MANAGERS
- IMPROVE CAPABILITY OF DEVELOPMENT SYSTEM
- DECIDE STRUCTURE AND POLICIES
optional in LeSS, but most organizations have
ORGANISATIONAL CAPABILITY IMPROVEMENT

LeSS Rule(s)

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 doesn't only focus on a team but on the overall organizational system.

A Scrum Master is a dedicated full-time role. One ScrumMaster can serve 1-3 teams.

team: standing

> What may happen if the big group is moving to LeSS and…

> "a Scrum Master is only allowed to serve 1 Team"

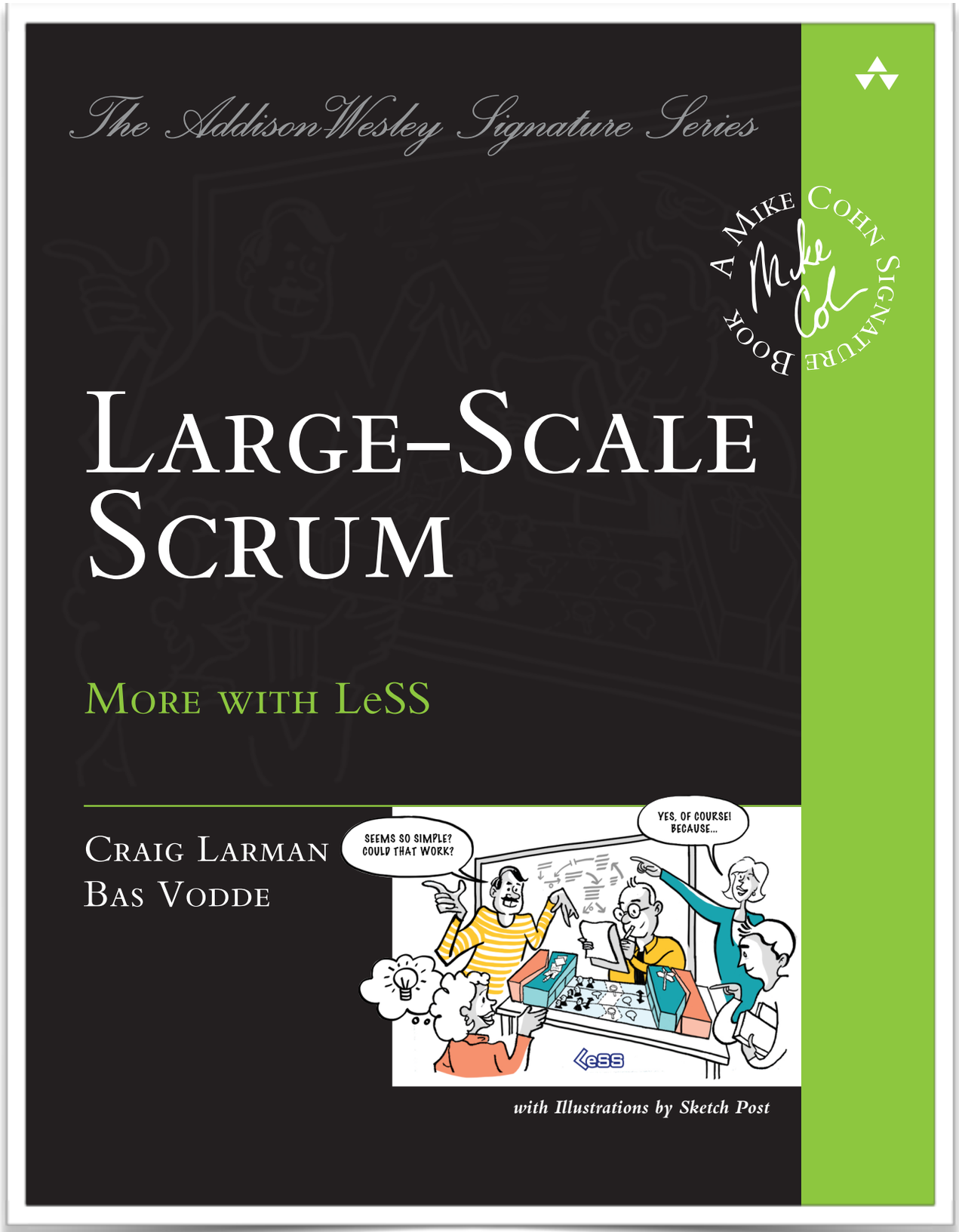
> or, "a Scrum Master can serve 6 Teams"

Scrum Master is
another name for…

~~project manager~~
~~agile project manager~~
~~team manager~~
~~tracker/status provider~~
~~Jira updater~~
~~note-taker~~

Guides in LeSS: Scrum Masters

- Scrum Master Focus
- Five Scrum Master Tools
- Large-Group Facilitation
- Promote Learning & Multiple Skills
- Community Work
- Scrum Master Surviving Guide
- Scrum Master Reading List
- Especially Pay Attention To...
- Avoid Requirement Area Silos



1 More with LeSS 1
2 LeSS 5

LeSS Structure

3 Adoption 41
4 Organize by Customer Value 67
5 Management 111
6 ScrumMasters 133

LeSS Product

7 Product 155
8 Product Owner 171
9 Product Backlog 197
10 Definition of Done 231

LeSS Sprint

11 Product Backlog Refinement 249
12 Sprint Planning 275
13 Coordination and Integration 285
14 Review & Retrospective 313

Managers in LeSS

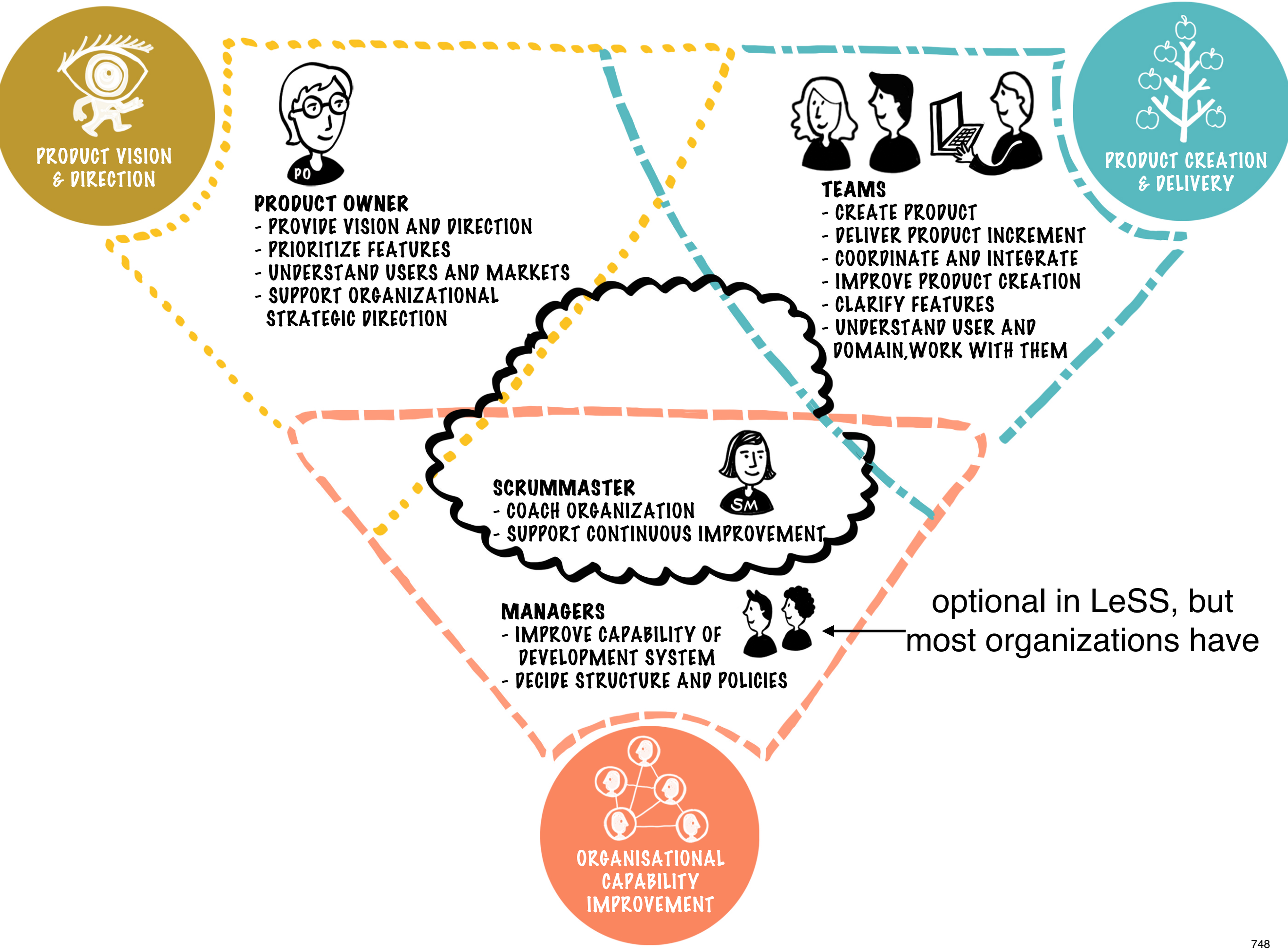
teams

> traditional activities/ responsibilities of managers (program, project, functional, component, resource, team, …)

coach: discuss

Manager Responsibilities

- the "other" column tasks
  - corporate admin tasks, etc.


PRODUCT VISION & DIRECTION
PO
PRODUCT OWNER
- PROVIDE VISION AND DIRECTION
- PRIORITIZE FEATURES
- UNDERSTAND USERS AND MARKETS
- SUPPORT ORGANIZATIONAL STRATEGIC DIRECTION
PRODUCT CREATION & DELIVERY
TEAMS
- CREATE PRODUCT
- DELIVER PRODUCT INCREMENT
- COORDINATE AND INTEGRATE
- IMPROVE PRODUCT CREATION
- CLARIFY FEATURES
- UNDERSTAND USER AND DOMAIN,WORK WITH THEM
SM
SCRUMMASTER
- COACH ORGANIZATION
- SUPPORT CONTINUOUS IMPROVEMENT
MANAGERS
- IMPROVE CAPABILITY OF DEVELOPMENT SYSTEM
- DECIDE STRUCTURE AND POLICIES
optional in LeSS, but most organizations have
ORGANISATIONAL CAPABILITY IMPROVEMENT

Manager Responsibilities

- corporate admin tasks, etc.
- **customer-value-delivery capability of org system**

do you recall? ...

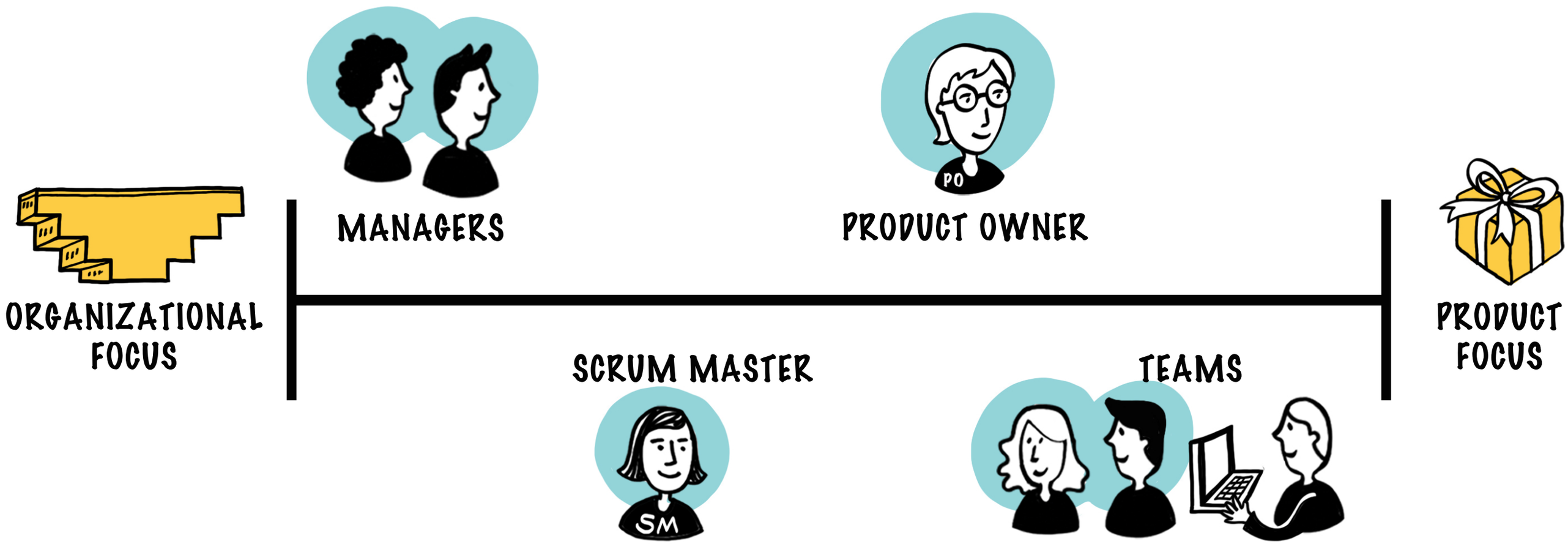
imperfect
product definition

imperfect
feature teams

Manager Responsibilities

- corporate admin tasks, etc.
- customer-value-delivery capability of org system
- **expanding product definition**
- **expanding feature teams**

Focus of Different Roles



LeSS Rule(s)

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**.

LeSS Rule(s)

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

Guide: Go See at *Gemba*

senior managers

manage by means

coach & group

> meaning of “senior managers manage by means” (vs … by results)?

teams

> make 2 lists:

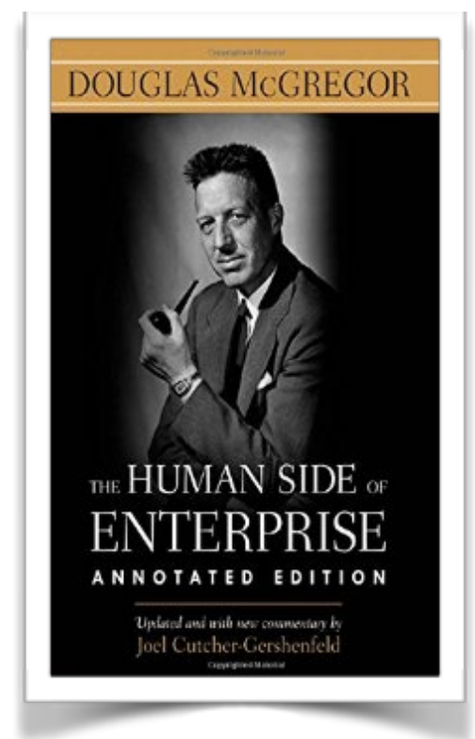
> Theory-of-mind **X** assumptions & behaviors

> Theory-of-mind **Y** assumptions & behaviors

coach: discuss

Guide: Theory (of mind) Y Management

Agile Principle 5: …Give them the environment and support they need, and trust them to get the job done.



class

> scenario:

  > some manager says, “let’s measure each team’s velocity”

  > the result is, “TeamRed has a higher velocity than TeamBlue”

> what dysfunctions arise?

Guide: LeSS Metrics with Less Targets

- **who**
  - metrics created by the **Teams** themselves, or **Product Owner**
- **purpose**
  - to learn & improve

LeSS Metrics: Don't…

> …let anyone other than team members or the Product Owner create metrics

> …measure for comparing teams or people

Harvard Business Review

TALENT MANAGEMENT

Why More and More Companies Are Ditching Performance Ratings

by David Rock and Beth Jones

SEPTEMBER 08, 2015

ECONOMY

Annual Job Review Is 'Total Baloney,' Expert Says

July 8, 2010 · 10:50 AM ET

Heard on Morning Edition

NPR STAFF

Employee performance reviews should be eliminated, according to UCLA business professor Samuel Culbert. "First, they're dishonest and fraudulent. And second, they're just plain bad management," he says

The problem with the practice, Culbert tells NPR's Renee Montagne, is that periodic reviews create circumstances that help neither the employee nor

Deloitte Joins Adobe And Accenture In Dumping Performance Reviews

On Leadership

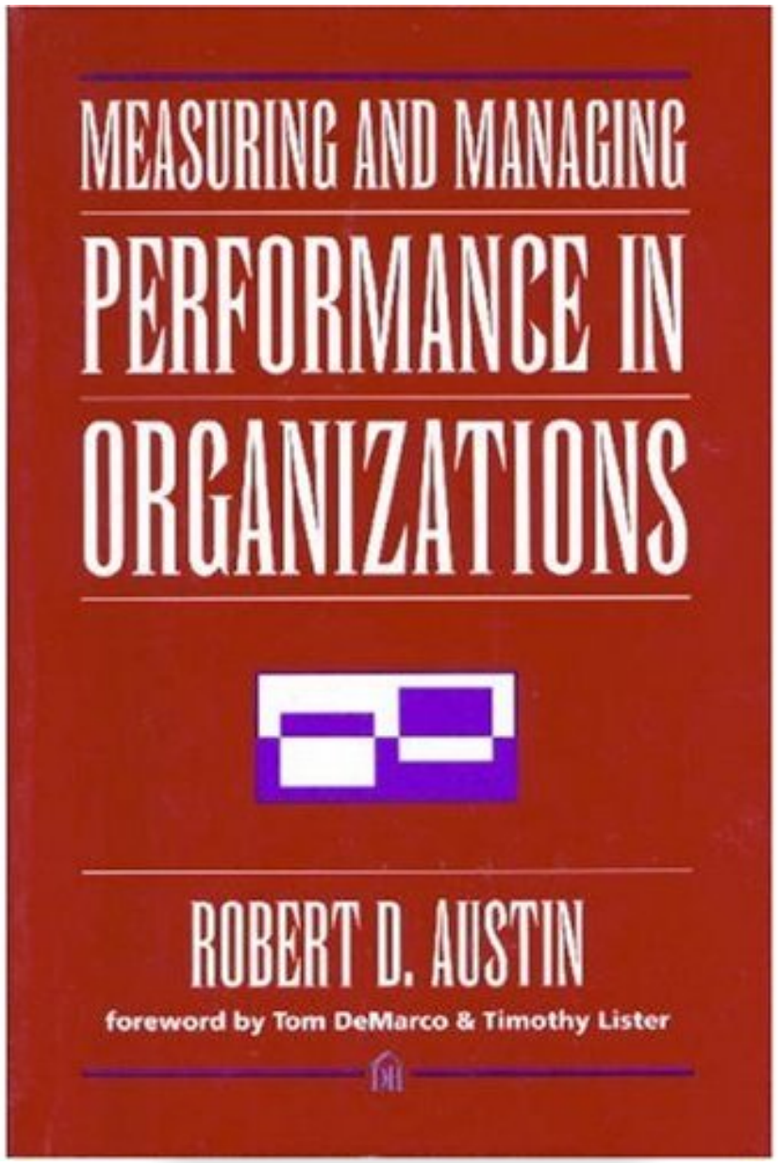
In big move, Accenture will get rid of annual performance reviews and rankings

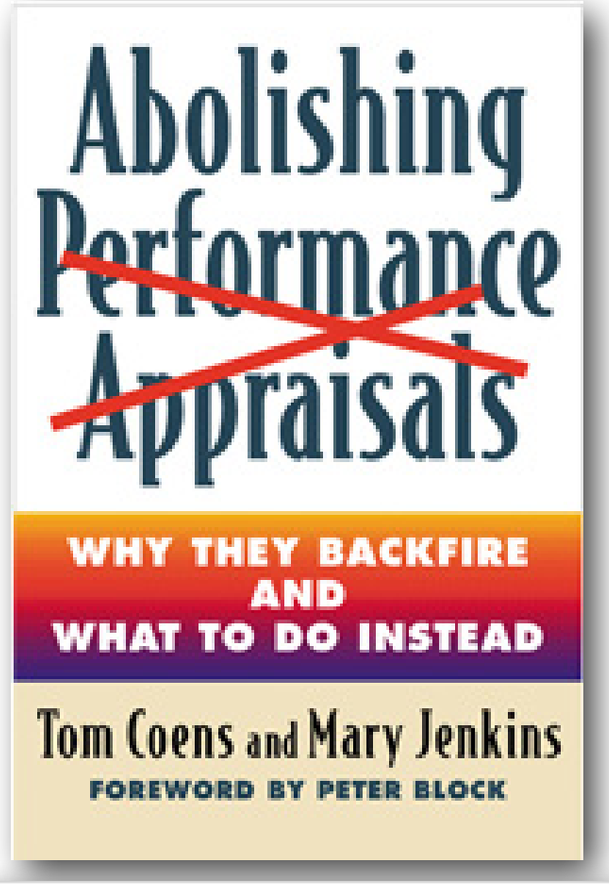
JUN 1, 2016 @ 01:17 PM 9,727 VIEWS

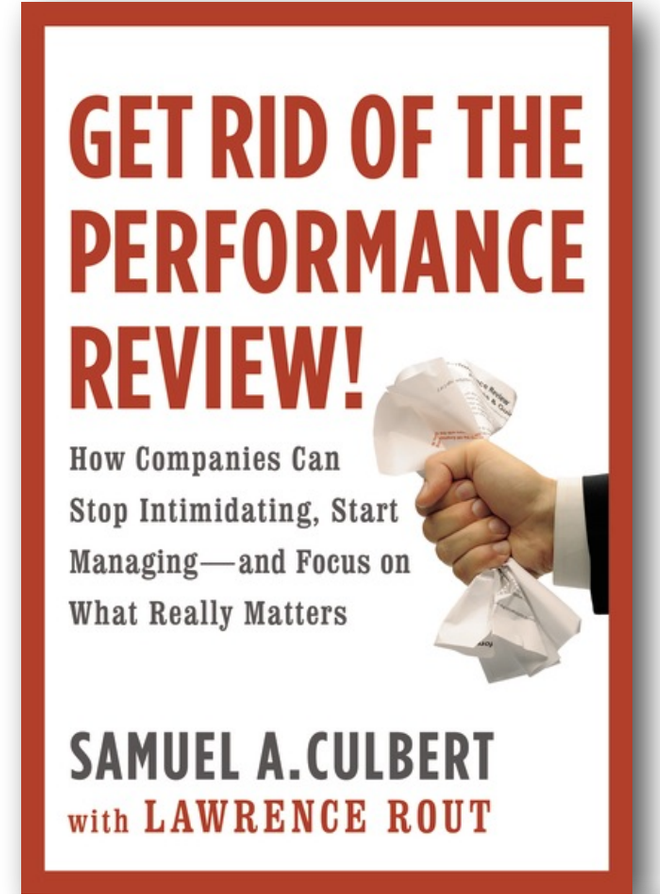
The Little Black Book of Billionaire Secrets

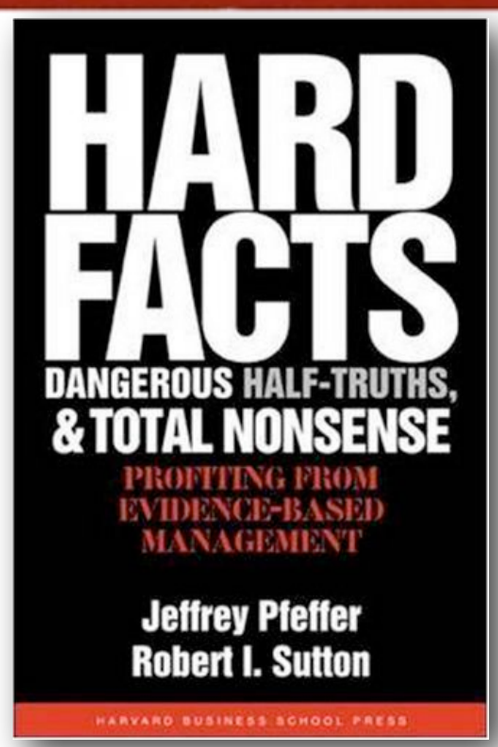
How Adobe Scrapped Its Performance Review System And Why It Worked

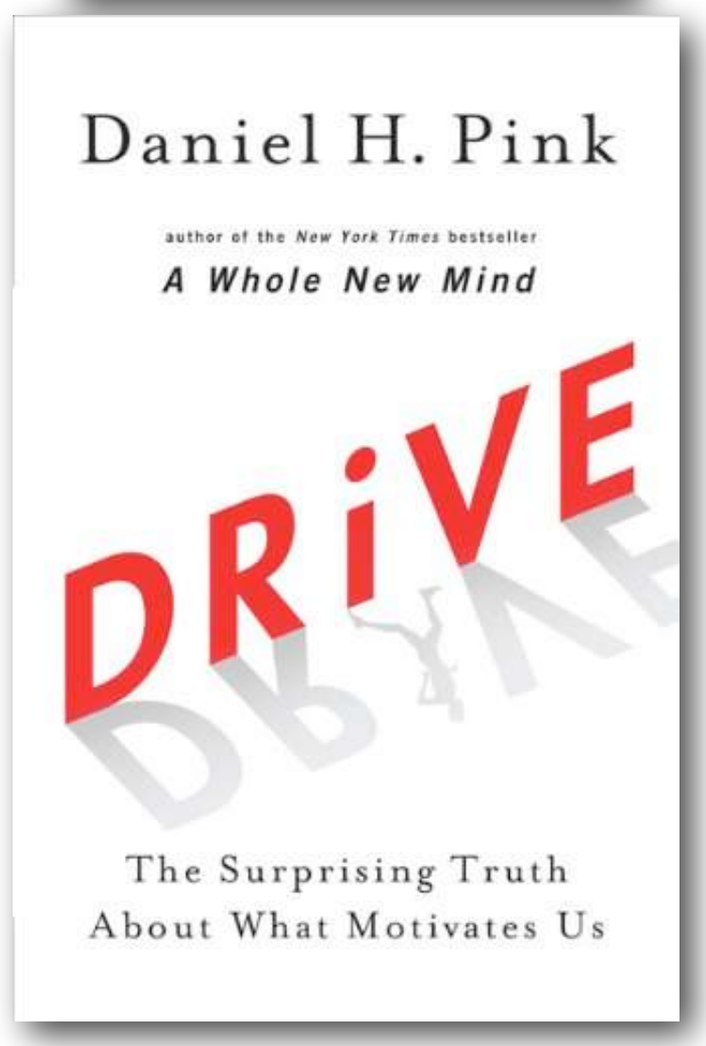
Why GE had to kill its annual performance reviews after more than three decades

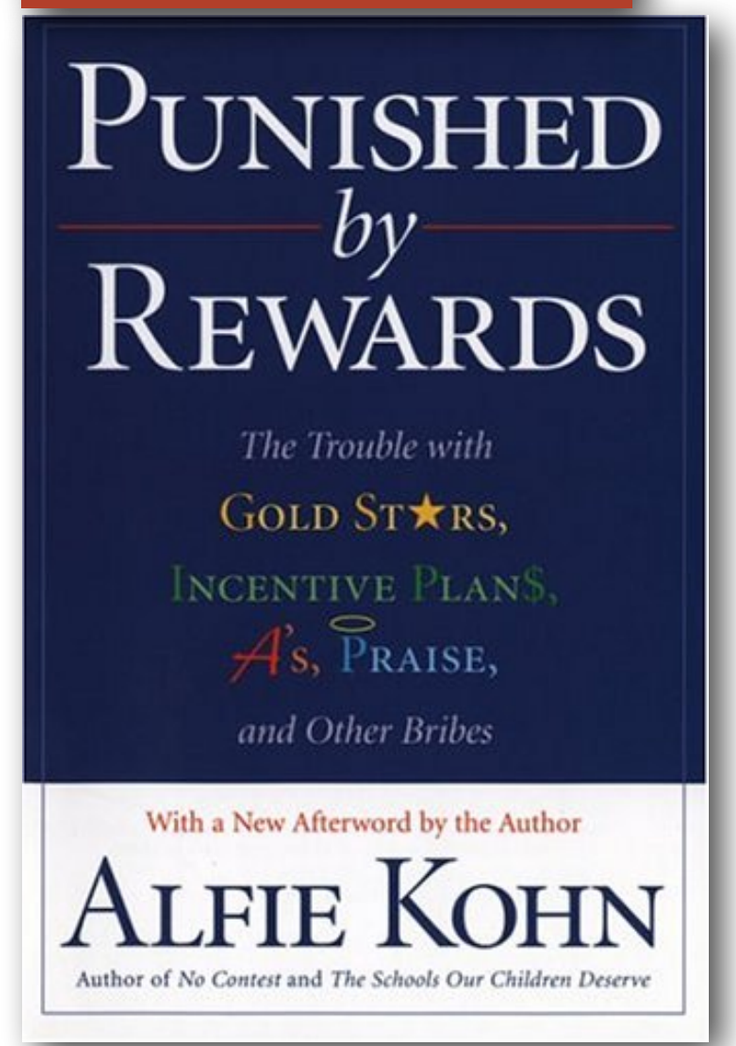

MEASURING AND MANAGING
PERFORMANCE IN
ORGANIZATIONS
ROBERT D. AUSTIN
foreword by Tom DeMarco & Timothy Lister


Abolishing
Performance
Appraisals
WHY THEY BACKFIRE
AND
WHAT TO DO INSTEAD
Tom Coens and Mary Jenkins
FOREWORD BY PETER BLOCK

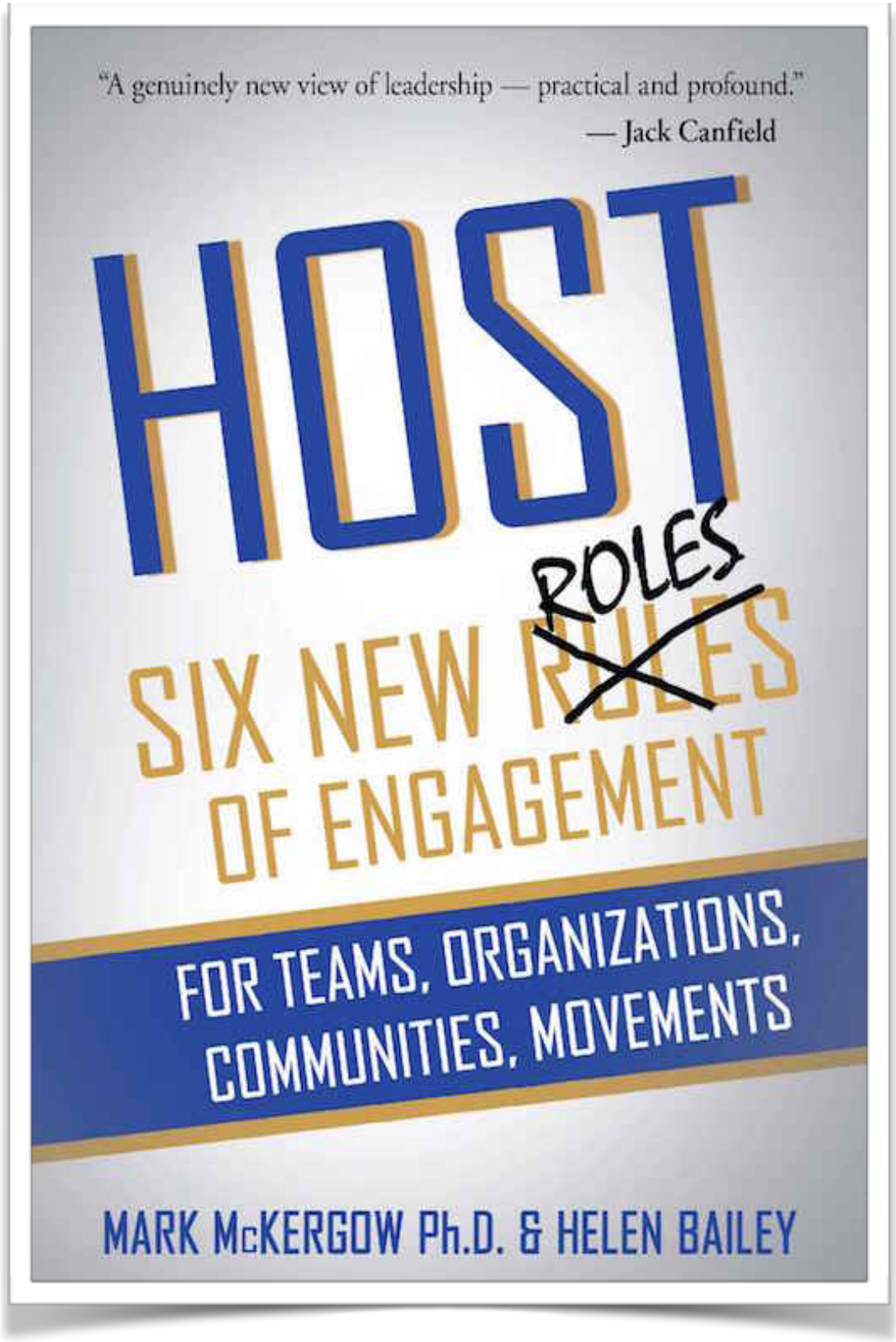

GET RID OF THE
PERFORMANCE
REVIEW!
How Companies Can
Stop Intimidating, Start
Managing—and Focus on
What Really Matters
SAMUEL A. CULBERT
with LAWRENCE ROUT


HARD
FACTS
DANGEROUS HALF-TRUTHS,
& TOTAL NONSENSE
PROFITING FROM
EVIDENCE-BASED
MANAGEMENT
Jeffrey Pfeffer
Robert I. Sutton
HARVARD BUSINESS SCHOOL PRESS


Daniel H. Pink
author of the New York Times bestseller
A Whole New Mind
DRiVE
The Surprising Truth
About What Motivates Us

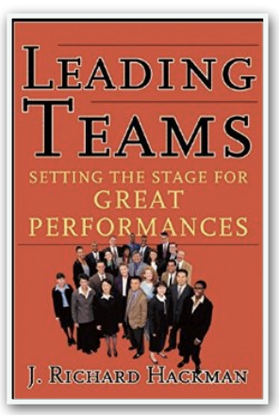

PUNISHED
by
REWARDS
The Trouble with
GOLD STARS,
INCENTIVE PLANS,
A's, PRAISE,
and Other Bribes
With a New Afterword by the Author
ALFIE KOHN
Author of No Contest and The Schools Our Children Deserve

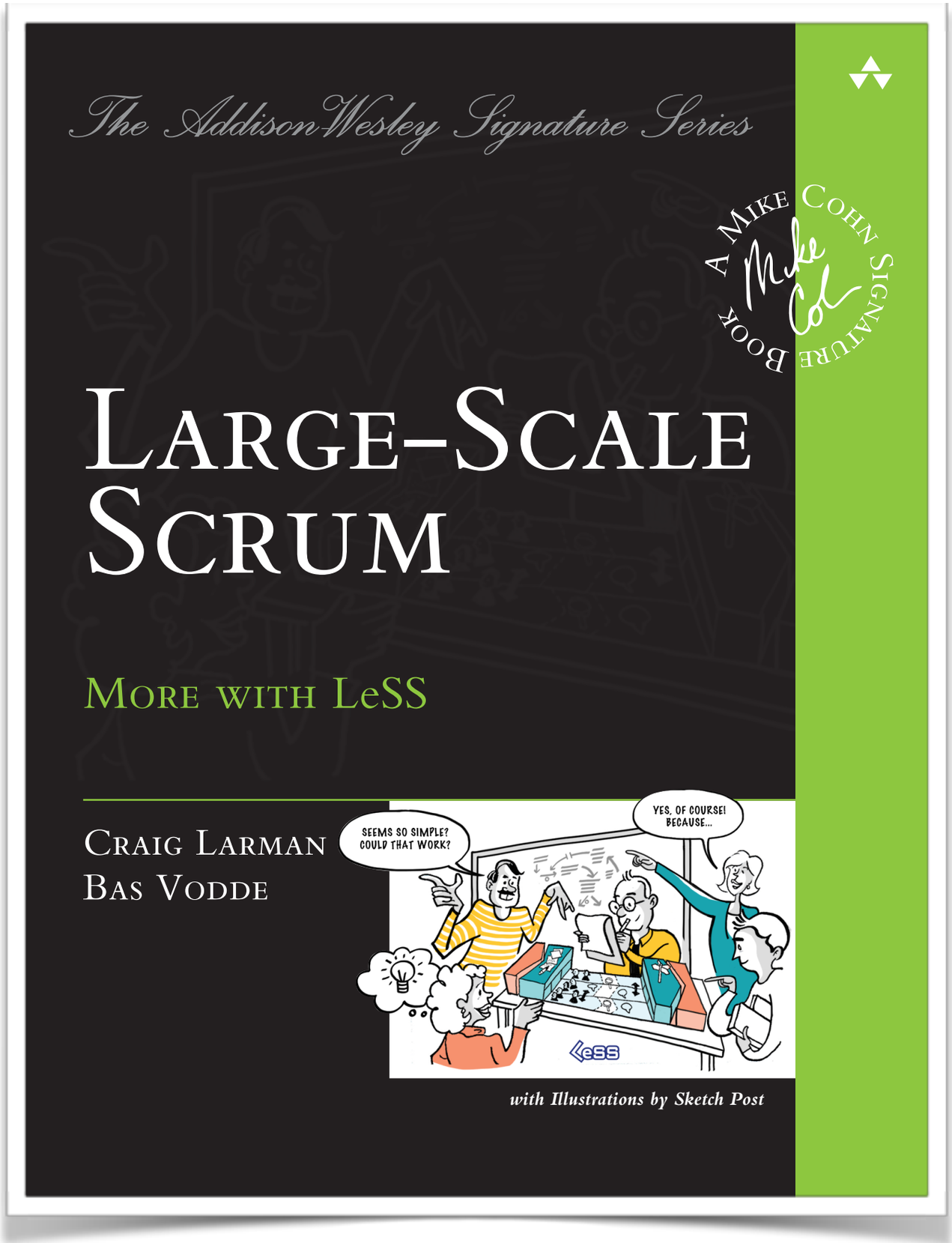
Metaphor: **Host** (… manager)



Self-Managing & Authority Matrix

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





1 More with LeSS 1
2 LeSS 5

LeSS Structure

3 Adoption 41
4 Organize by Customer Value 67
5 Management 111
6 ScrumMasters 133

LeSS Product

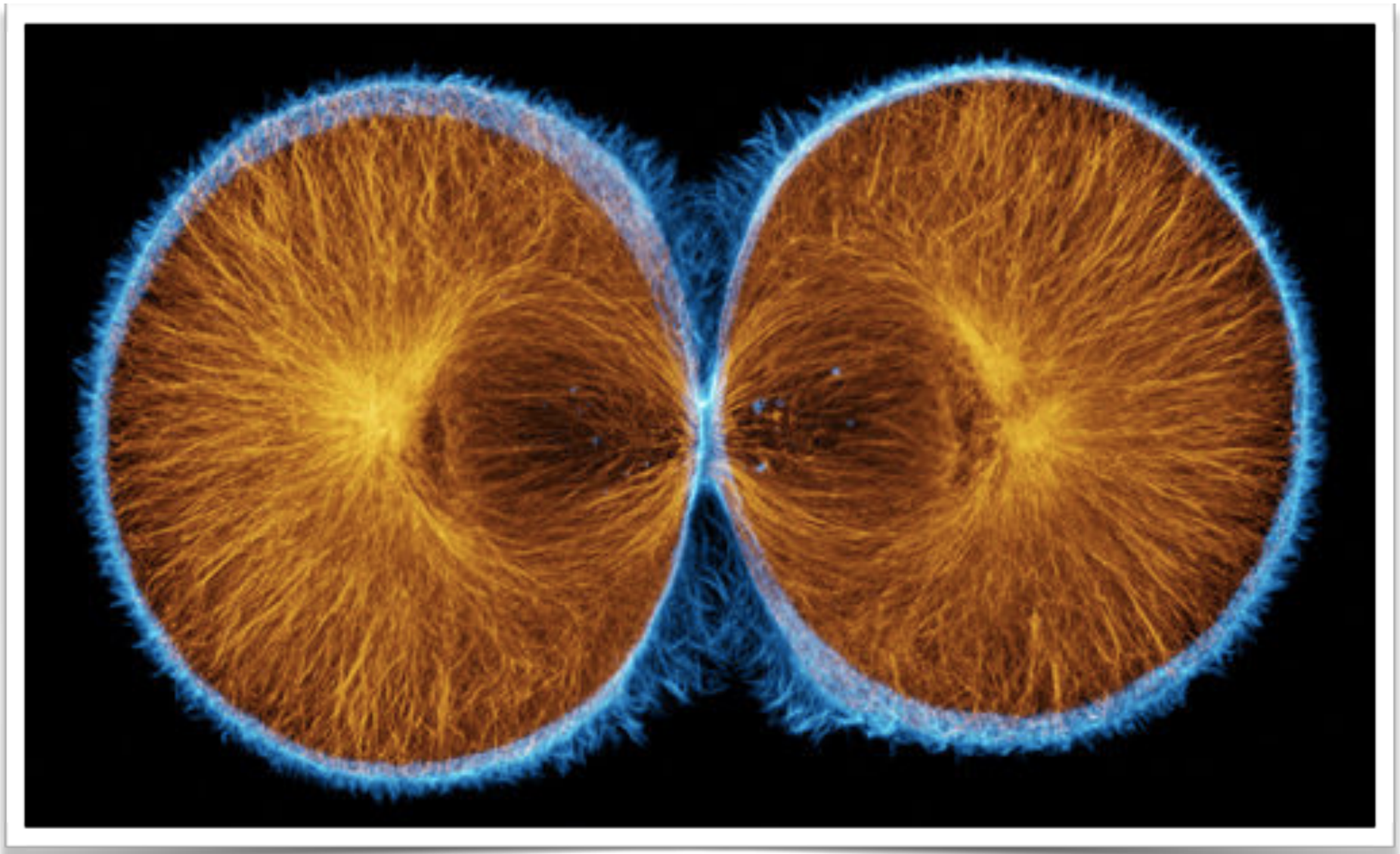
7 Product 155
8 Product Owner 171
9 Product Backlog 197
10 Definition of Done 231

LeSS Sprint

11 Product Backlog Refinement 249
12 Sprint Planning 275
13 Coordination and Integration 285
14 Review & Retrospective 313

Product Backlog & Tools

Guide: Dealing with Parents



Dealing with Parents: **Ancestor** Attribute

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

Guide: Handling Special Items

- defects
- improvements
- innovation or unusual study

Guide: Tools for Large Product Backlogs

- tools aren't agile; agility (adaptiveness) is an organisational quality & behaviour
- what Product Backlog tool at scale?
- nothing more complicated than a **spreadsheet** and **wiki**
- why? …

Why not “Agile Tools”? (1)

- Focus is on tools rather than the deep systemic problems…
  - … and that diverts or avoids focusing on what’s important: changing behaviour and the system. These tools don’t solve the real problems.
- These tools contain and promote reporting features, reinforcing traditional management-reporting and control behaviours.
- They convey a facade of improvement or agile adoption, when nothing meaningful has changed; “agile” tools have nothing to do with being agile.

Why not “Agile Tools”? (2)

- They often impose inflexible terminology and workflows to the teams, taking away process ownership and restricting improvement.
- The Product Backlog is often hidden for most people as access requires an expensive account.
- These tools enable complexifying rather than simplifying.

don’t use same tool for Product Backlog and Sprint Backlogs

Guide: Area Backlog

Item	Requirement Area	...
B	market on-boarding	
F	market on-boarding	

Item	Requirement Area	...
B	market on-boarding	
C	trade processing	
D	asset servicing	
F	market on-boarding	
...		

Area Backlogs via **Tool Views**

Item	Requirement Area	...
B	market on-boarding	
F	market on-boarding	

Item	Requirement Area	...
B	market on-boarding	
C	trade processing	
D	asset servicing	
F	market on-boarding	
...		

a VIEW for one requirement area

it is NOT a separate artifact

Area Backlogs via **Separate Artifacts**

Item	Requirement Area	...
B	market on-boarding	
F	market on-boarding	

Item	Requirement Area	...
B	market on-boarding	
C	trade processing	
D	asset servicing	
F	market on-boarding	
...		

an ARTIFACT for one requirement area

it IS a separate artifact

Area Backlogs as **views**

vs

Area Backlogs as **separate artifacts**

splitting items within a Requirement Area…

Splitting Case 1: Minor Discrepancy

Overall PB

Item	Area
B	market onboarding
C	trade processing
D	market onboarding

Market Onboarding Area Backlog

Item	Ancestor
B-1	B
D	
B-2	B

Splitting Case 2: Major Discrepancy

Item	Area
B	market onboarding
C	trade processing
D	market onboarding
E	market onboarding
F	market onboarding
...	

Item	Ancestor
B-1	B
B-2	B
D	
E	
F	
B-3	B
B-4	B

Splitting Case 2: **Un-splitting** (merging)

Overall PB

Item	Area
BX (generalization of B1, B2)	market onboarding
C	trade processing
D	market onboarding
E	market onboarding
F	market onboarding
BY (generalization of B3, B4)	market onboarding

Market Onboarding Area Backlog

Item	Ancestor
BX-1 (old B-1)	**BX**
BX-2 (old B-2)	**BX**
D	
E	
F	
BY-1 (old B-3)	**BY**
BY 2 (old B4)	**BY**

Guide: Three Levels Max

Overall PB

Item	Ancestor	Area
XA	X	trade processing
XB	X	trade processing
...		

2 “levels”

now, 3 levels

the Ancestor column also links the the Area & Overall PB

Trade Processing Area Backlog

Item	Ancestor
XA-1	XA
XA-2	XA
...	

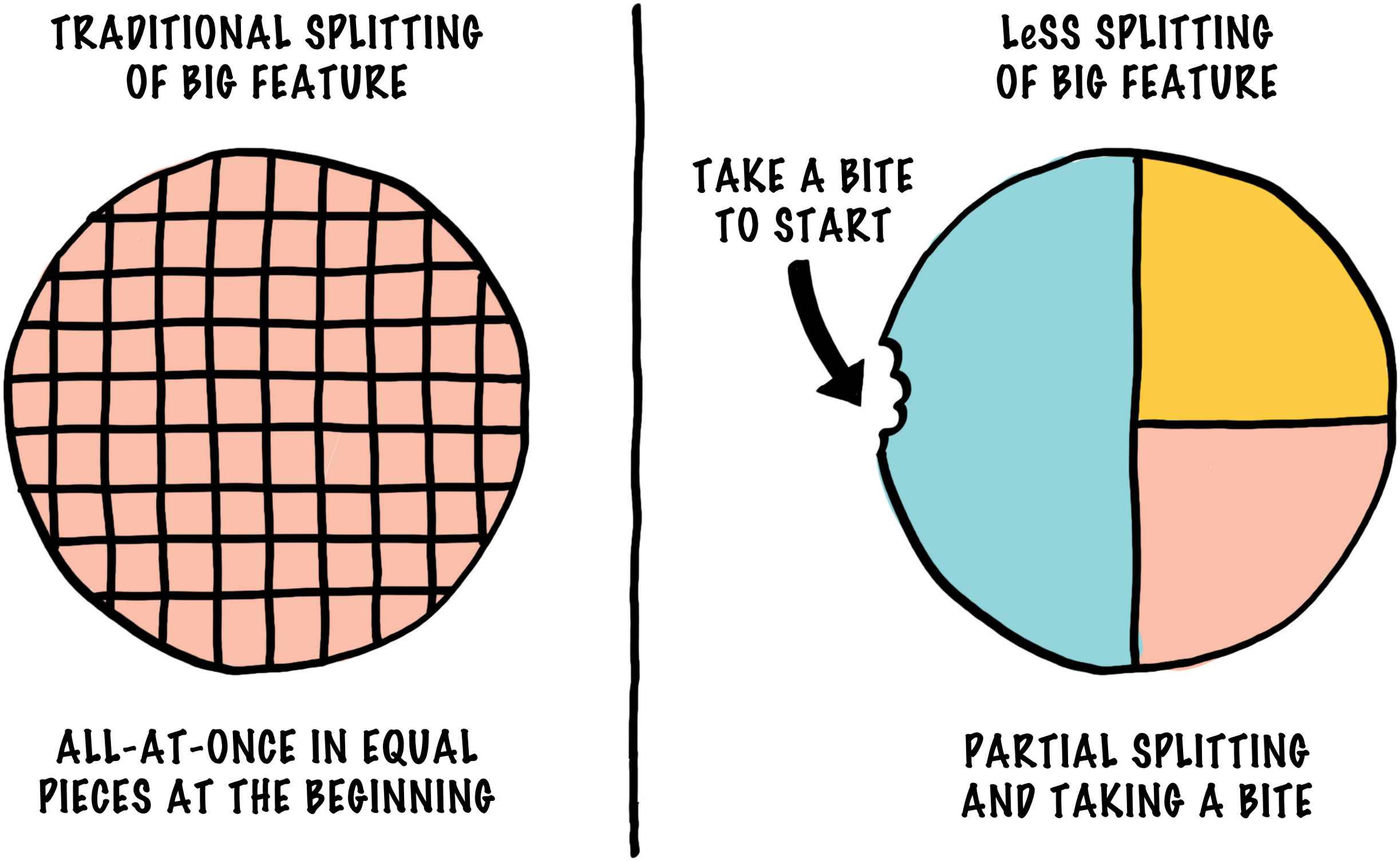
prioritization of large backlog?

(see PBR module)

Guide: Don't "Manage Dependencies between Products" but Minimize Constraints

- Do "their part"
- Pair-work "their part"
- Simplify or split item-A so that the other group's change is small
- Split item-A into (1) item with a stub, and (2) fully integrate item
- Split item-A into (1) item using an alternative interface, and (2) item using the final interface
- Explain the constraint
- Bypass the constraint
- Achieve the outcome a different way

Guide: Take a Bite



Closing

Agenda

- ~~Introduction~~
- ~~Principles~~
- ~~Overview~~
- ~~Organised by Customer Value~~

- ~~Product~~
- ~~Definition of Done~~
- ~~Product Owner and Product Backlog~~

- ~~Sprint Planning, Review Retrospective~~
- ~~Integration and Coordination~~
- ~~Management and Orientation~~
- ~~Adoption~~
- ~~Scrum Master~~

Agile Gurukul all rights reserved.

one key reminder? ...

BIG Idea

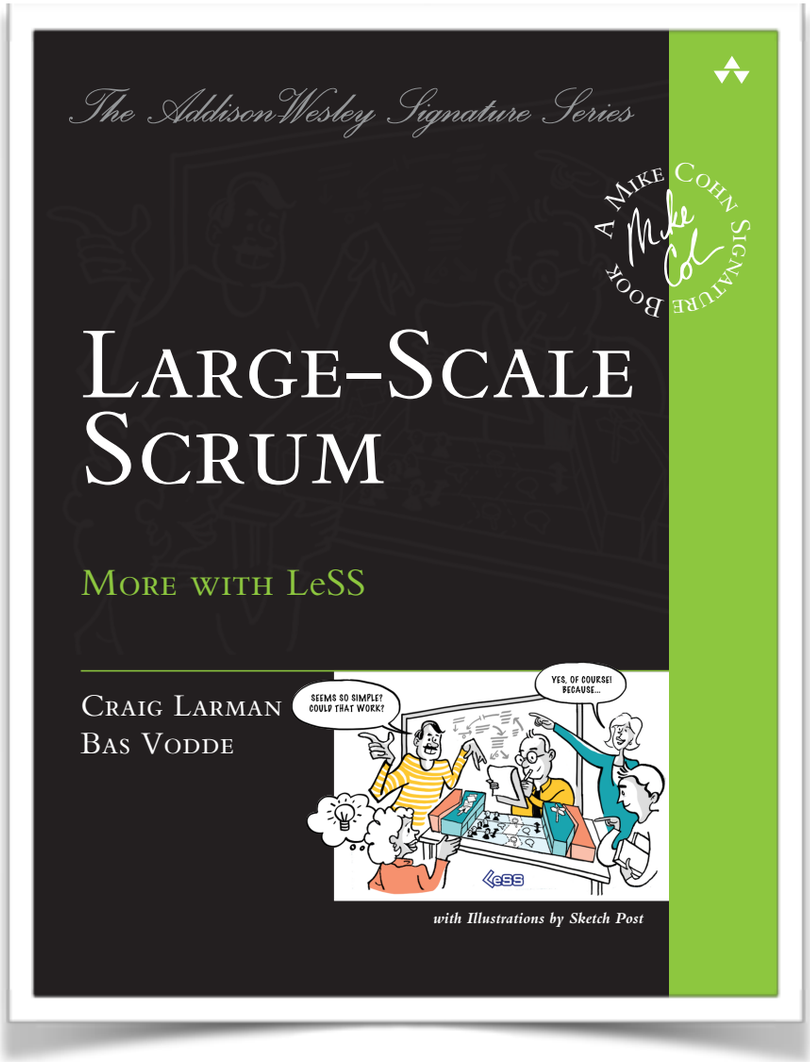
management's remit is to
see the whole system
optimise the whole system
(not locally optimise)

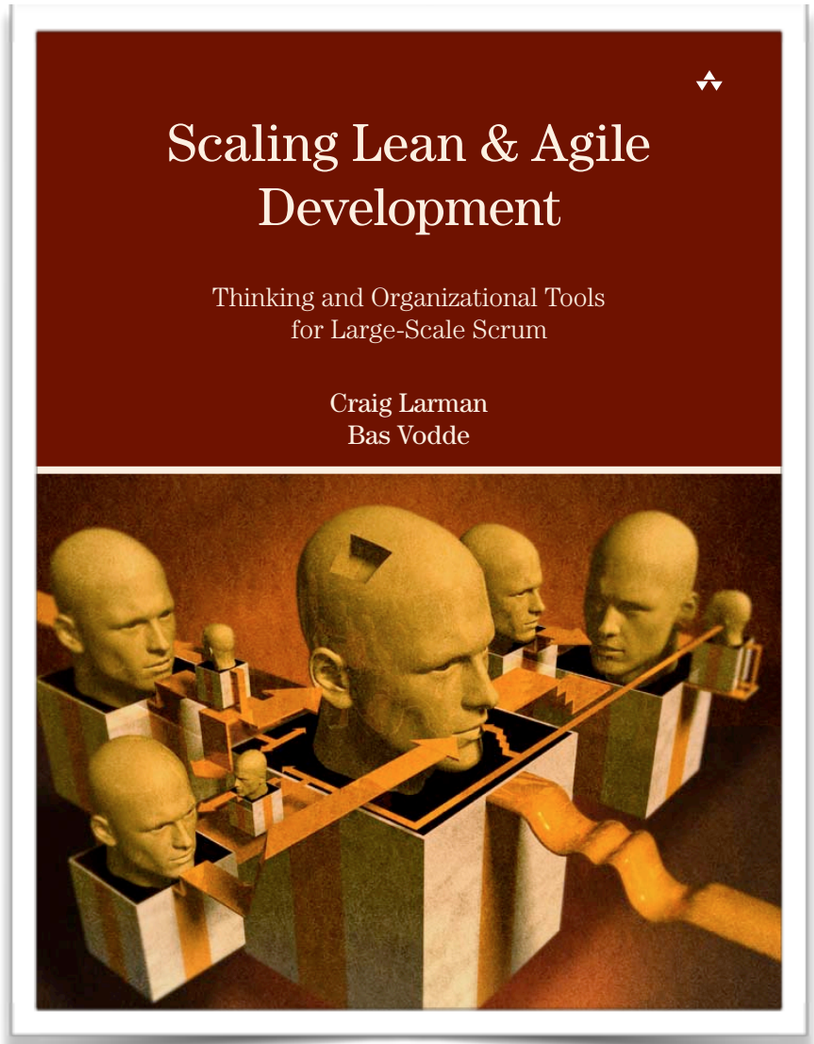
BIG Idea

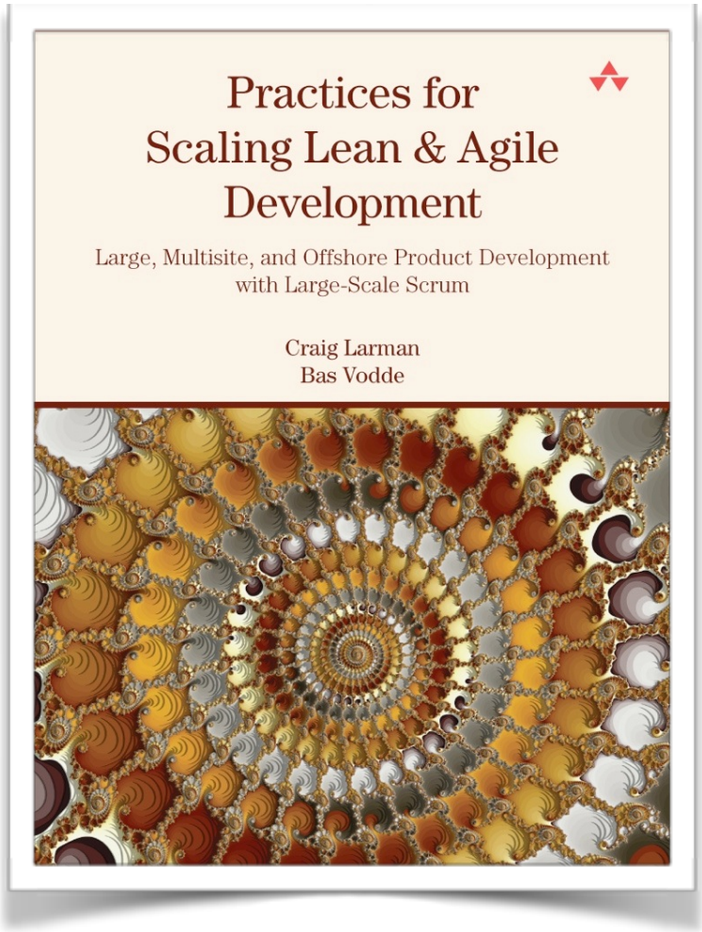
so… failures in…
“delivering quality”,
“successful change”, etc
are failures of the system,
and thus
“failures in management”

wrapping up…

Read these thoroughly before adoption… many people in the group







after you can see
local optimisation,
you can never unsee it

this can be frustrating

sorry! ;)

by the end, you will understand the following:

you won’t successfully be able to return to workplace and “give a summary” of your insights; it is futile & won’t be understood

Certified LeSS Practitioner

>i will register you at less.works

>you can change your email address at any time

Your Account @ less.works

–books 1 & 3

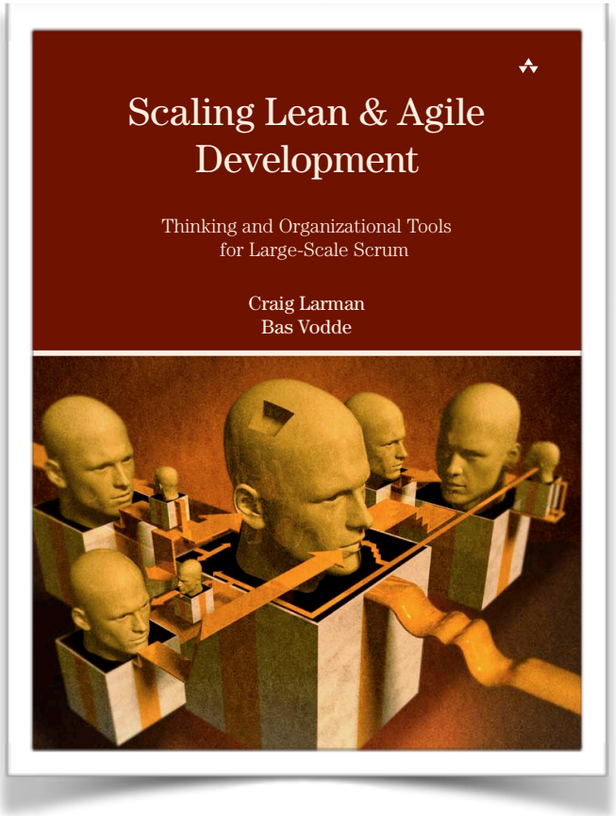
–flipchart/wall photos

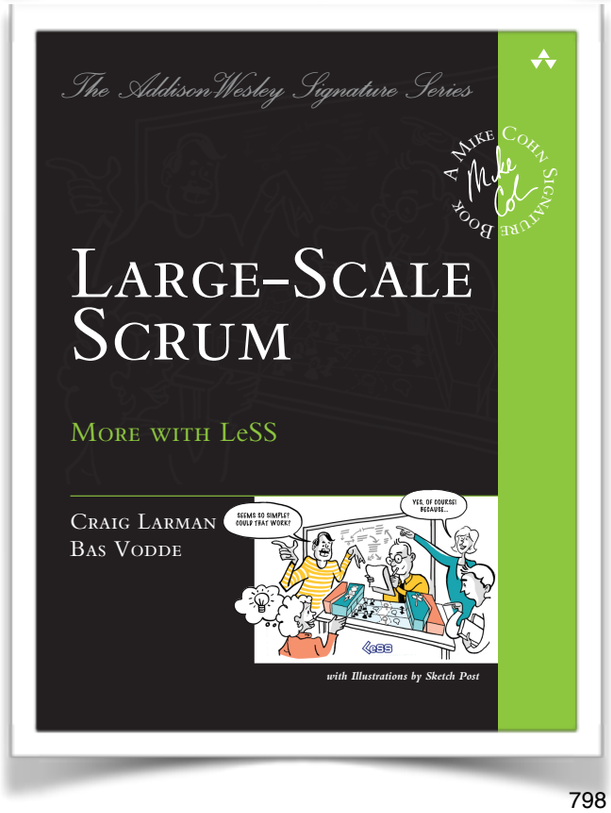
–course notes pdf

–contacts

–certificate

–class photo





Connections

LeSS Site: http://less.works

LeSS Twitter: #LeSSWorks

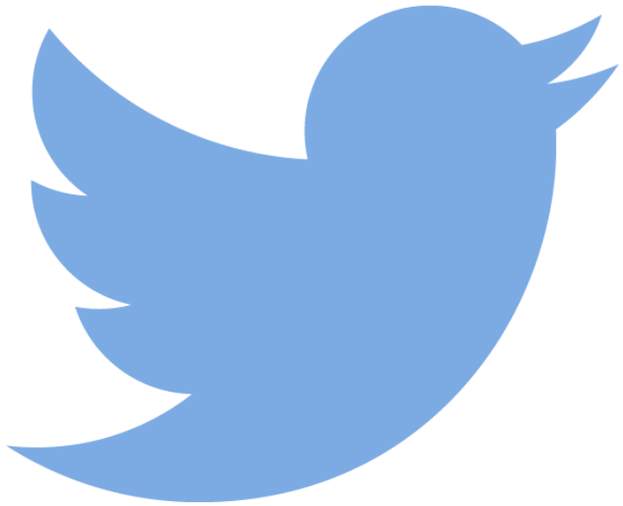
LinkedIn Group: LeSS - Large-Scale Scrum

LinkedIn Group: Certified LeSS Practitioner

Slack: http://less-works.slack.com/

LeSS Discussion Group:
http://groups.google.com/forum/#!forum/largescalescrum

LeSS on Facebook
https://www.facebook.com/less.works



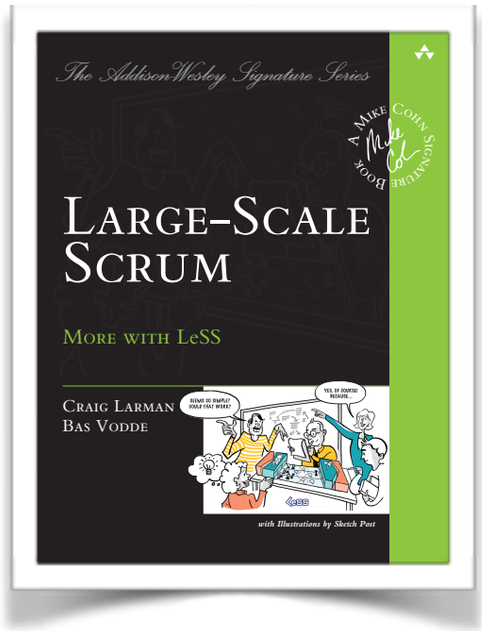
#LeSSWorks

share!

blog!

tweet!

spread the word!



amazon

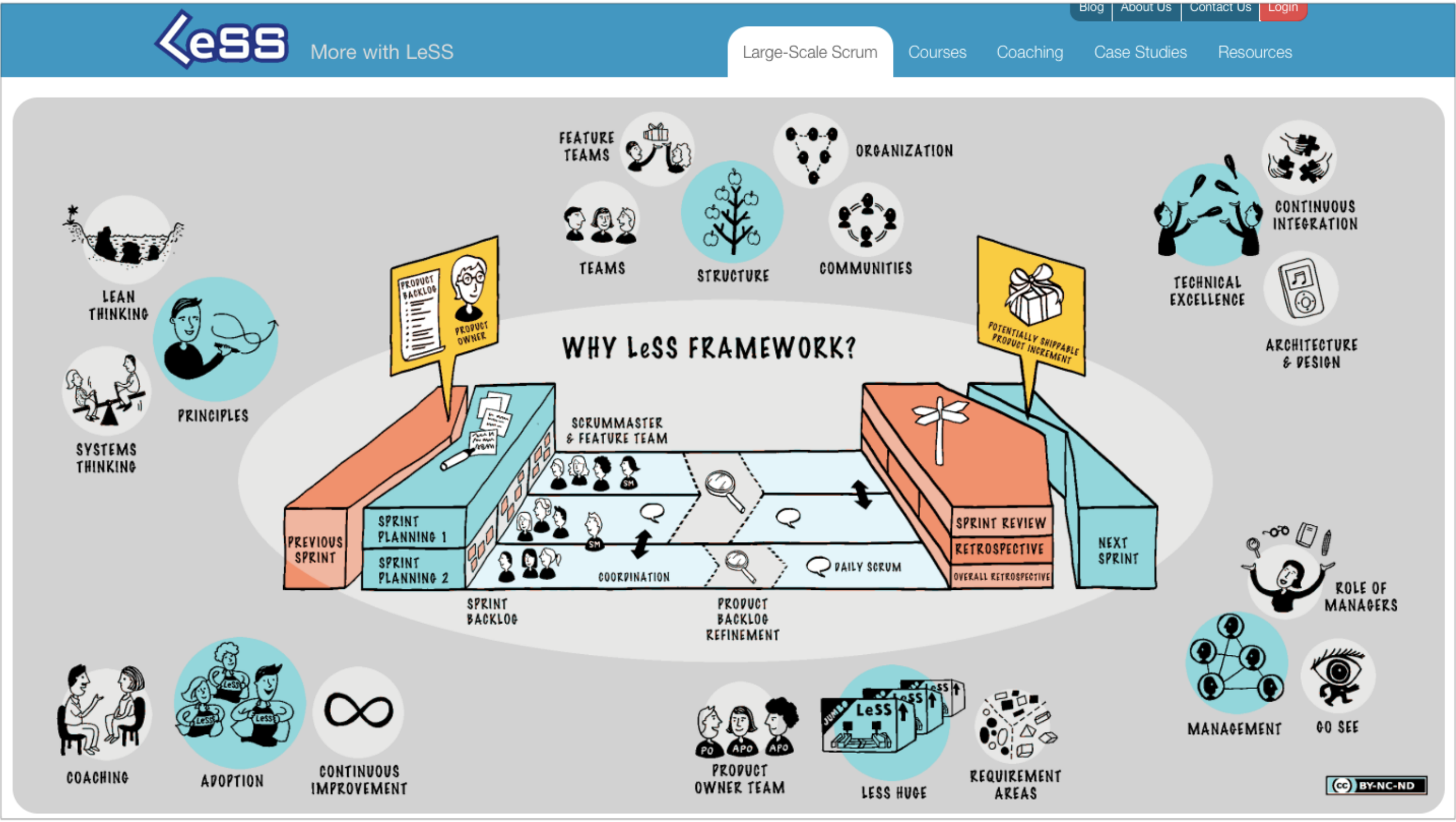
Amazon reviews of new book are appreciated ;)

team: round-robin: standing

> “how do i feel?…”

> please sit when team is done

less.works


LeSS
More with LeSS
Blog
About Us
Contact Us
Login
Large-Scale Scrum
Courses
Coaching
Case Studies
Resources
FEATURE TEAMS
ORGANIZATION
TEAMS
STRUCTURE
COMMUNITIES
CONTINUOUS INTEGRATION
TECHNICAL EXCELLENCE
ARCHITECTURE & DESIGN
LEAN THINKING
PRINCIPLES
SYSTEMS THINKING
WHY LeSS FRAMEWORK?
PRODUCT BACKLOG
PRODUCT OWNER
POTENTIALLY SHIPPABLE PRODUCT INCREMENT
SCRUMMASTER & FEATURE TEAM
PREVIOUS SPRINT
SPRINT PLANNING 1
SPRINT PLANNING 2
SPRINT BACKLOG
COORDINATION
PRODUCT BACKLOG REFINEMENT
DAILY SCRUM
SPRINT REVIEW
RETROSPECTIVE
OVERALL RETROSPECTIVE
NEXT SPRINT
ROLE OF MANAGERS
MANAGEMENT
GO SEE
COACHING
ADOPTION
CONTINUOUS IMPROVEMENT
PO
APO
APO
PRODUCT OWNER TEAM
LESS HUGE
REQUIREMENT AREAS
BY-NC-ND

last words

group: photo