



1

Craig Larman
craiglarman.com & less.works

Please...
Do not copy or share this material, or re-use for other education.
Exceptions require prior written consent of the author.

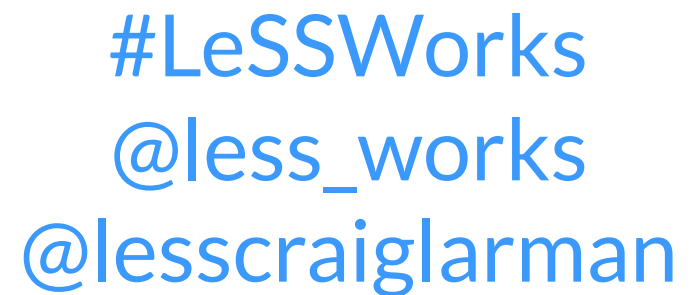
Copyright © 2016 Craig Larman, All rights reserved.
May not be reproduced without written consent of the author.

v.49

less.works

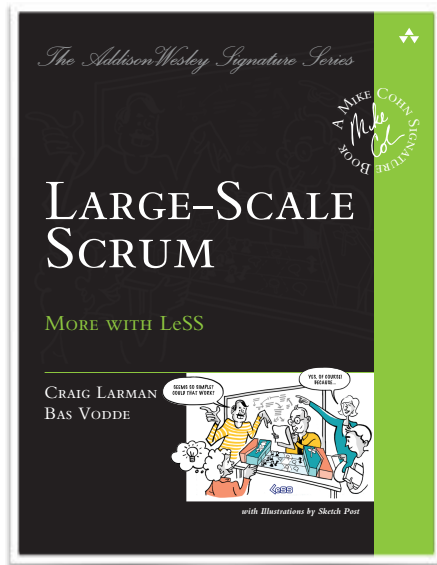


3



3

Book Raffle?



5

Scaling?

6

first, a caution...

7



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?” ...**

8

His answer:
“Find the ten best
people and write
the entire thing
themselves.”

[Horowitz74]

9

after years working in
large
multisite
offshore
development,
our **key advice?** ...

10

large - don't
multisite - don't
offshore - don't

11

but groups still ‘scale’,
for reasons...

compelling
 (“create self-driving car”)

questionable
 (“low-cost sites”)

12

so is LeSS for **scaling**?

13

13

Descaling &
Simplifying

14

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

15

15

is this the right
question? ...

16

16

traditional large groups are complicated — though not because they need to be, but because **their organizational designs create an illusion of “necessary” complexity**

17

This is an Important Question...

“How can we **simplify** the unnecessarily big and complex organizational design, and **be agile** rather than **do agile**?”

18



BIG Idea

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

19

LeSS
More with LeSS

20

Learning Resources

21

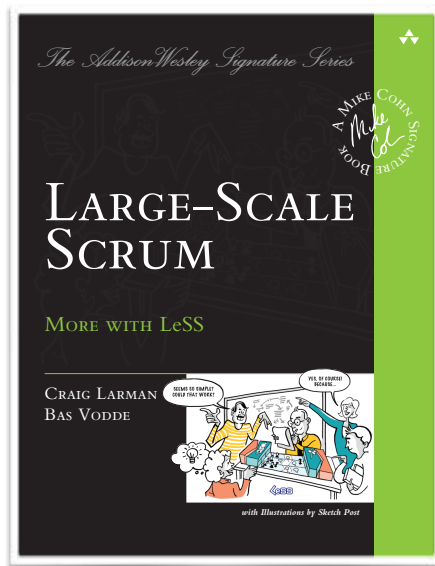
Craig Larman

co-creator of LeSS (with Bas Vodde)

large + multisite + 'offshore'
large-scale embedded systems
large-scale financial systems
large-scale telecom systems

22

3rd LeSS book...



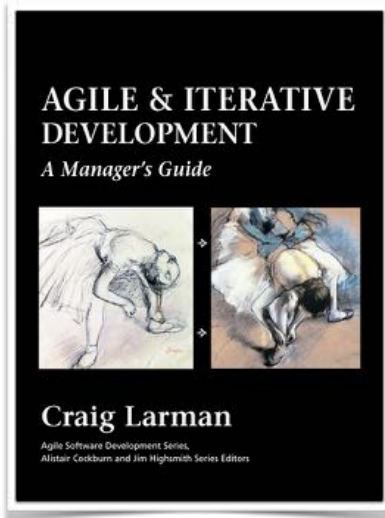
23

First Two LeSS books...



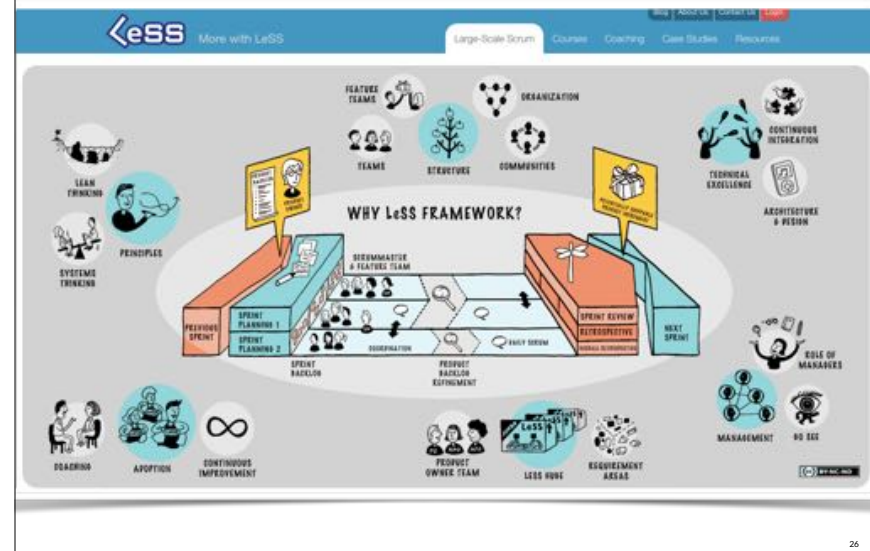
24

Early Agile Book



25

less.works



26

Local Optimization

27

in traditional large-scale
organizational design, the
overarching & repeating theme

local optimization

28

examples of
local optimization...

29



30

focus on delivering dishes?



31

System
Optimization

32



using the analogies, how to
advise system optimization?



33

33

Systems Optimization

“watch the **ball**,
not the players”

“deliver the **dish**,
not the onions”

34

34

the **One True**
system optimizing goal?

35

35

low E2E CT
Highest Value
Highest Agility/Adaptiveness
Busy-ness
Personal Secrecy
Ideation
Manager Prestige & Cash Transfer

...

36

36



BIG Idea

leadership needs to
agree on the system
optimizing goal

37

37

the **LeSS** System Goals

- > company-level system optimization for
 - > **deliver highest customer value first**
 - > **agility driven by learning**
("turn on a dime, for a dime")

38

38

Why LeSS?

39

Organize by
Customer Value:
Feature Teams

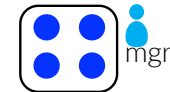
40

Descaling with LeSS

replace
**local optimizations of
single-specialist groups**
with a *majority* of
feature teams

41

analysts and/or
UX/UI designers



DBAs



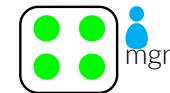
architects



component-1
programmers



component-2
programmers



test/QA
group



a likely traditional
large-scale
organizational
structure before
adopting Scrum

42

analysts and/or
UX/UI designers



DBAs



architects



component-1
programmers



component-2
programmers



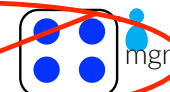
test/QA
group



a cross-functional
team in Scrum
spans **all** functions

43

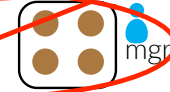
analysts and/or
UX/UI designers



DBAs



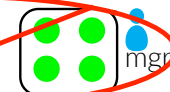
architects



component-1
programmers



component-2
programmers



test/QA
group



single-specialist
groups are dissolved

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

44

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

— Upton Sinclair

45

Guide: Job Safety, but not Role Safety

Job safety
& **salary safety**
but not
role safety

46

Larman's Laws of
Organizational
Behavior

47

why so much?...
Lean-but
Scrum-but
Kanban-but
DevOps-but
AnyChangeIdea-but

48

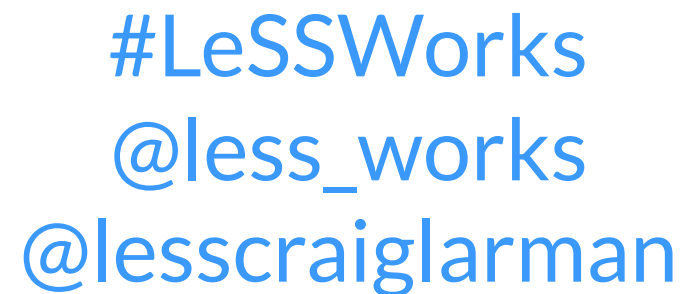
1. Organizations are implicitly optimized to avoid changing the status quo middle- and first-level manager and “specialist” positions & power structures.
2. As a corollary to (1), any change initiative will be reduced to overloading or redefining the new terminology to mean basically the same as status 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)

4

.

50

3



52