

CHANGE REQUIRES INSIGHT

2016-
10-22

Knowit

1

1. Knowit

- 2. Why go Agile – why scale?
- 3. Scaling Agile Frameworks
- 4. LeSS Framework
- 5. SAFe Framework
- 6. Comparisons
- 7. Discussions / Reflections

AGENDA

Daniel Borgentun



Region Manager at Knowit
Agile Enterprise Coach
Systems Architect

Daniel.borgentun@knowit.se

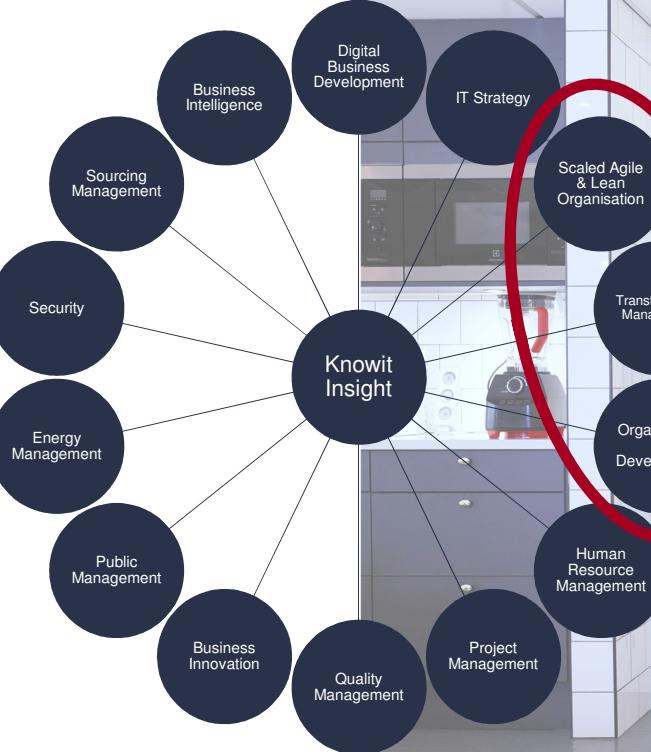
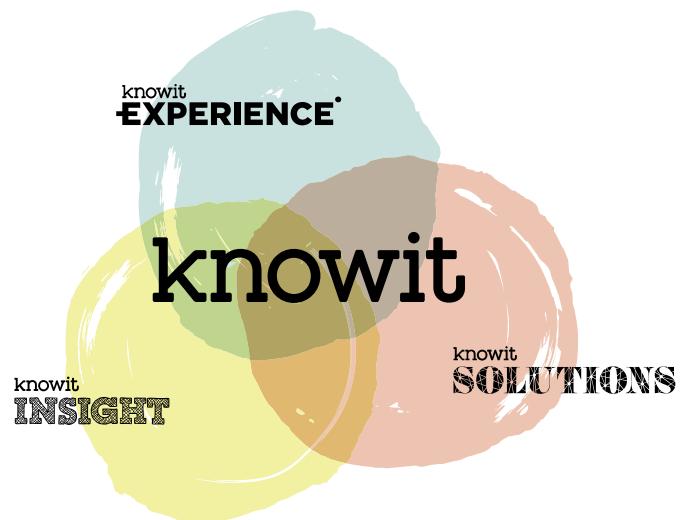
Page 3

WE ARE HERE!



Knowit shortly

- Nordic consultant company
- 2000 experts
 - 1200 specialists developing IT solutions
 - 500 consultants in the largest digital agency in the Nordics
 - 230 management consultants



1. Knowit

knowit INSIGHT

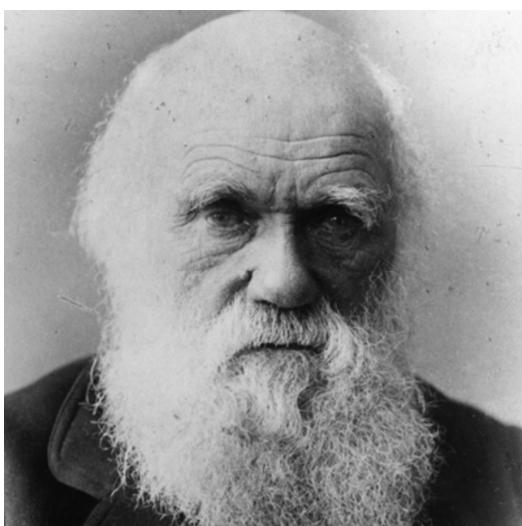
2.Why go Agile – why scale?

- 3. Scaling Agile Frameworks
- 4. LeSS Framework
- 5. SAFe Framework
- 6. Comparisons
- 7. Discussions / Reflections

AGENDA

SURVIVAL OF THE FITTEST

knowit INSIGHT



It is not the strongest of the species that survives, nor the most intelligent that survives. It is the one that is most adaptable to change.

Average Lifetime of S&P 500 Companies

knowit INSIGHT



This company will be going strong one hundred and even five hundred years from now.

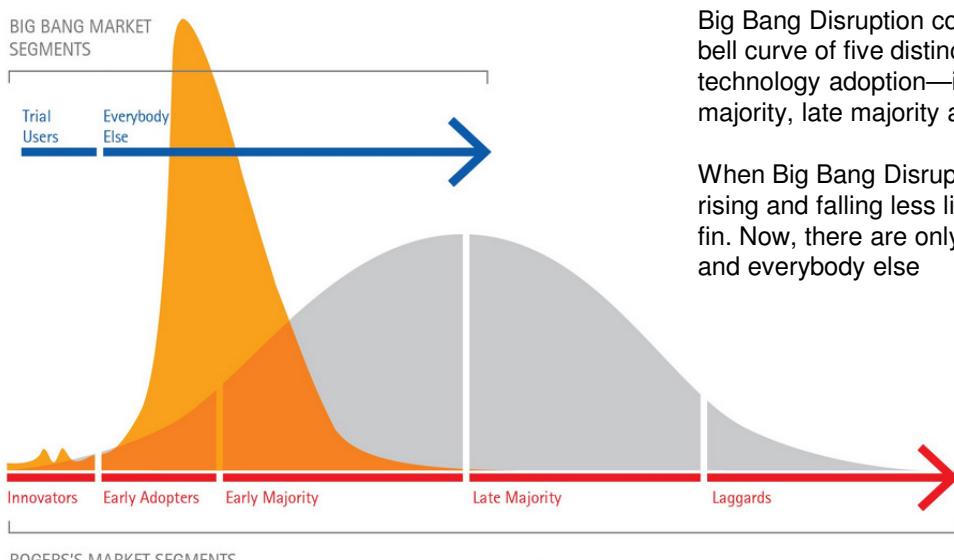
C. JAY PARKINSON, PRESIDENT OF ANACONDA MINES
statement made three years in advance of Anaconda's bankruptcy

Page 9

THE ERA OF BIG BANG DISRUPTION

knowit INSIGHT

Big Bang Market Adoption

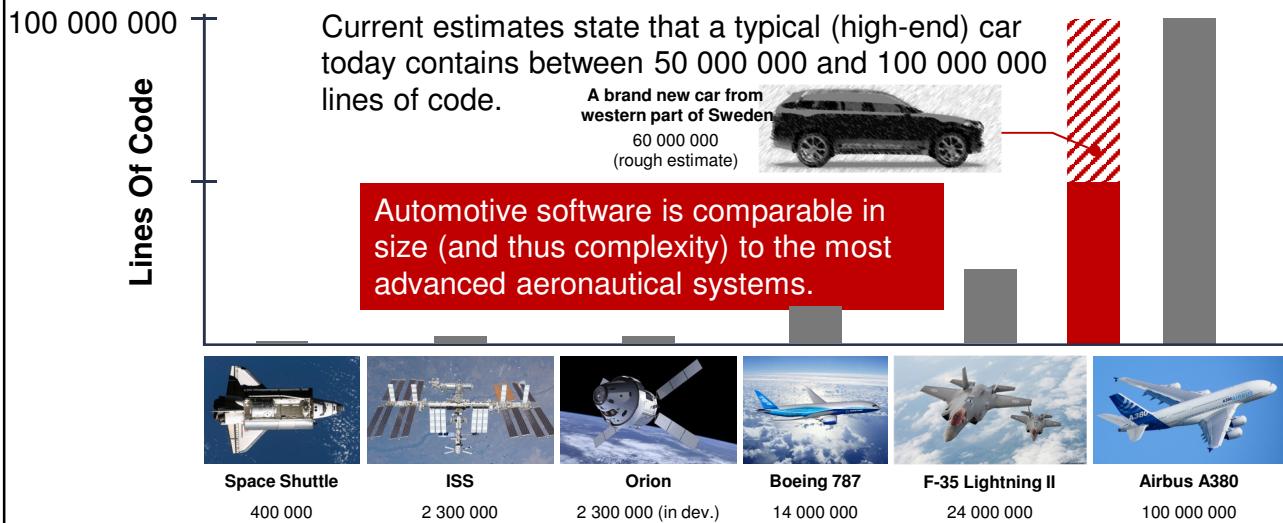


Big Bang Disruption collapses Everett Rogers's classic bell curve of five distinct customer segments for technology adoption—innovators, early adopters, early majority, late majority and laggards.

When Big Bang Disruptors take off, they do so quickly, rising and falling less like a curve and more like a shark's fin. Now, there are only two market segments: trial users and everybody else

SOFTWARE INTENSIVE SYSTEMS

knowit INSIGHT

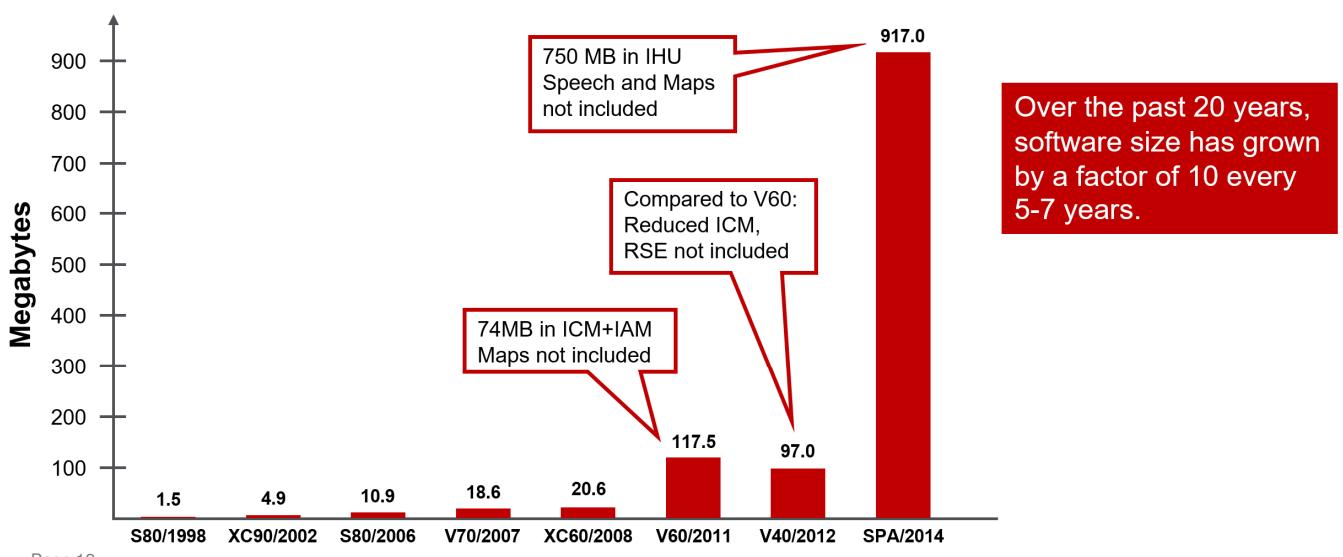


Page 11

Source: Martin Hiller, Volvo

SOFTWARE SIZE EVOLUTION AT A BIG CAR COMPANY

knowit INSIGHT

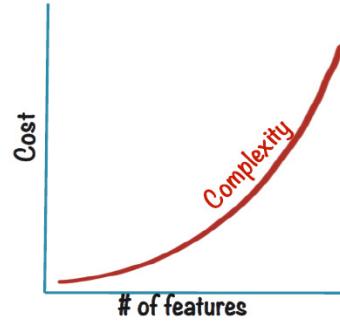
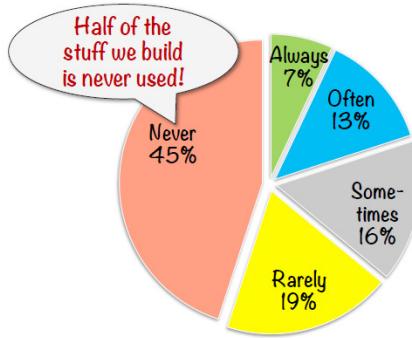


Page 12

Source: Martin Hiller, Volvo

We tend to build the wrong thing

Features and functions used in a typical system



Sources:
Standish group study reported at XP2002 by Jim Johnson, Chairman

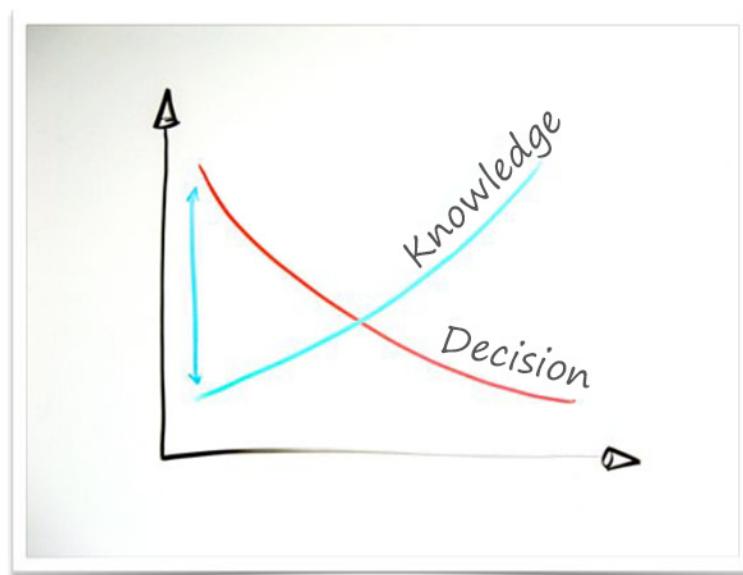
The right-hand graph is courtesy of Mary Poppendieck

Henrik Kniberg

Page 13

THE PROJECT PARADOX

knowit INSIGHT

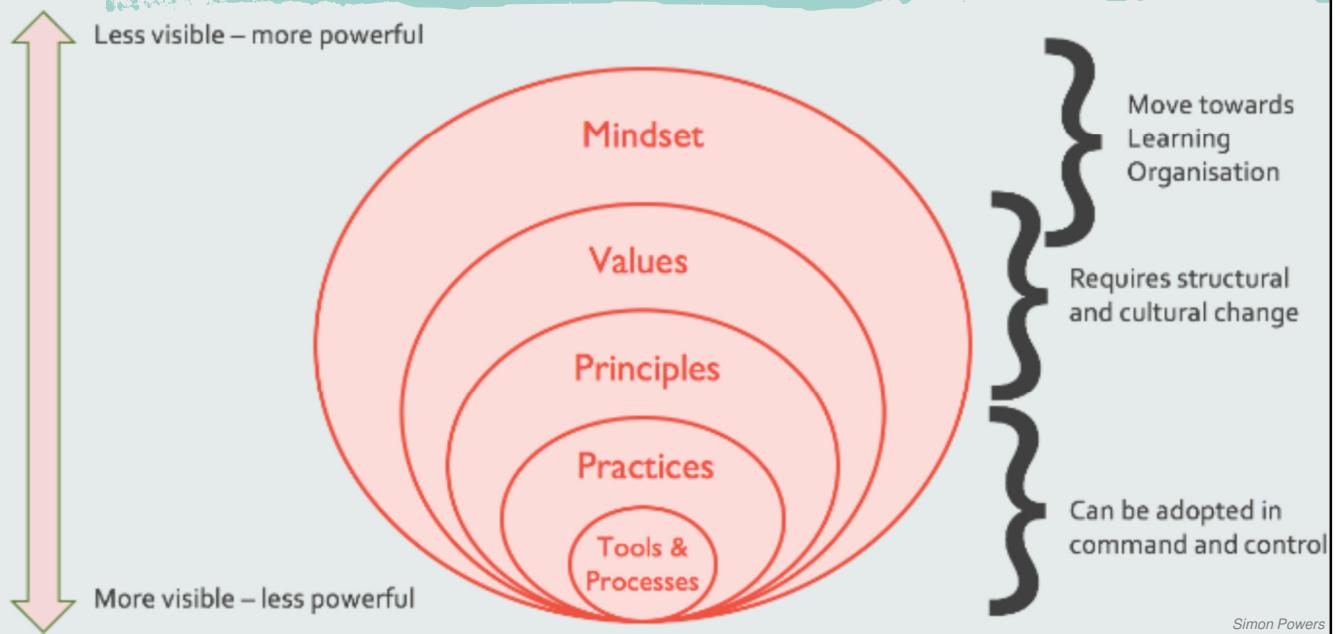


Page 15

Tobias Fors, Citerus AB

WHAT IS AGILE?

knowit INSIGHT



1. Knowit
2. Why go Agile – why scale?

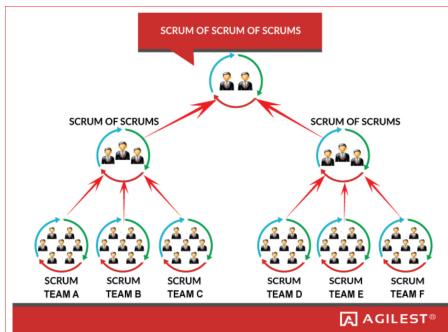
3. Scaling Agile Frameworks

4. LeSS Framework
5. SAFe Framework
6. Comparisons
7. Discussions / Reflections

knowit INSIGHT

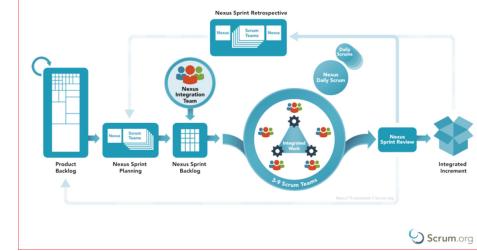
AGENDA

FRAMEWORKS

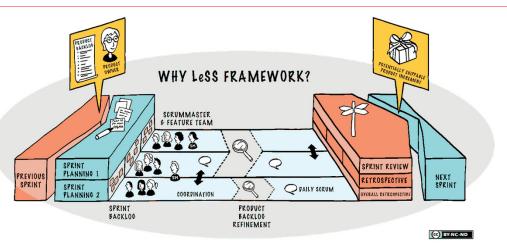
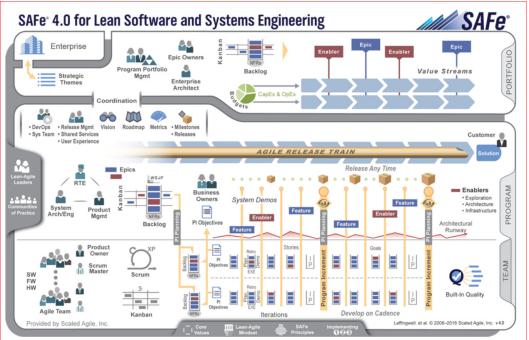


Scrum of Scrums

Nexus Framework



SAFe



LeSS

1. Knowit

knowit INSIGHT

2. Why go Agile – why scale?

3. Scaling Agile Frameworks

4.LeSS Framework

5. SAFe Framework

6. Comparisons

7. Discussions / Reflections

AGENDA

SCALING AGILE WITH LeSS ''



"LeSS IS SCRUM APPLIED TO MANY TEAMS WORKING TOGETHER ON ONE PRODUCT"

LeSS is Scrum

- How can we apply the principles, purpose, elements, and elegance of Scrum in a large-scale context, as simply as possible.

... applied to
many teams

- Cross-functional, cross-component, — to create done items and a shippable product.

... working
together

- Towards a common goal to deliver one common shippable product at the end of a common Sprint.

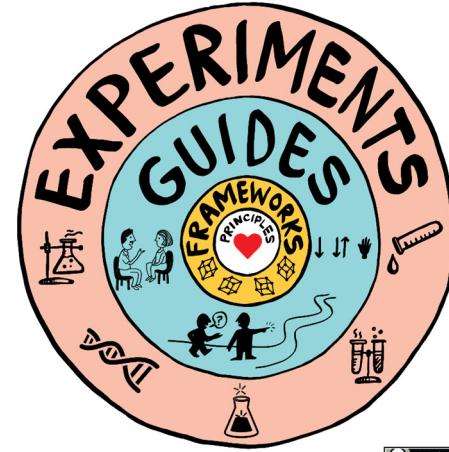
... on one product

- A broad complete end-to-end customer-centric solution that real customers use.

THREE DESIGN PRINCIPLES FOR LeSS

knowit INSIGHT

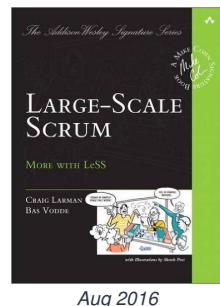
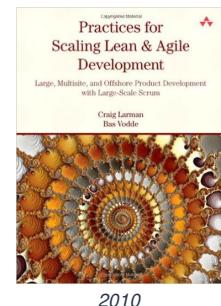
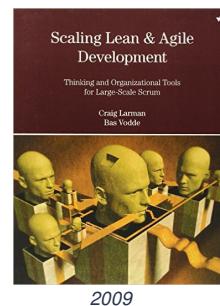
1. Simple and barely sufficient
2. Build up instead of tailor down
3. Descaling over scaling



22

BACKGROUND

knowit INSIGHT



Bas Vodde and Craig Larman

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



[Less.works](http://less.works)

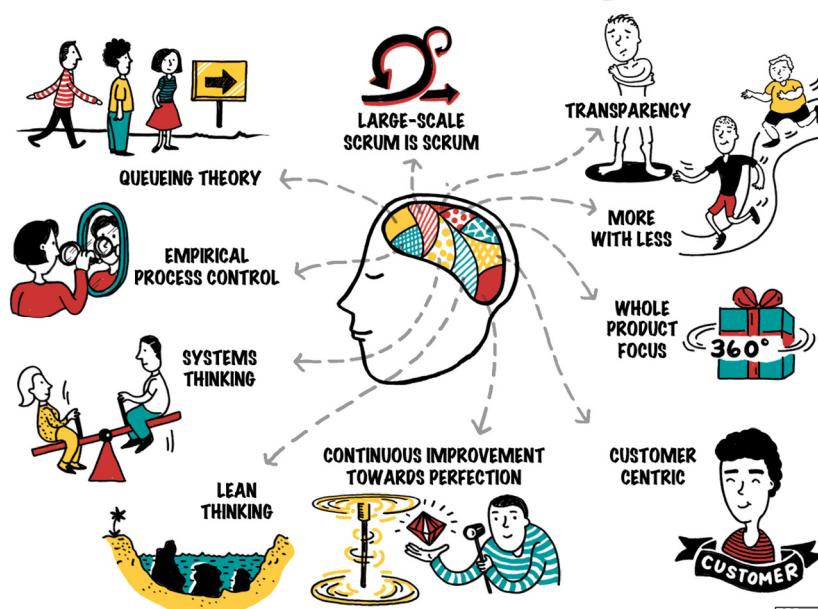
23

LeSS PRiNCiPLES



24

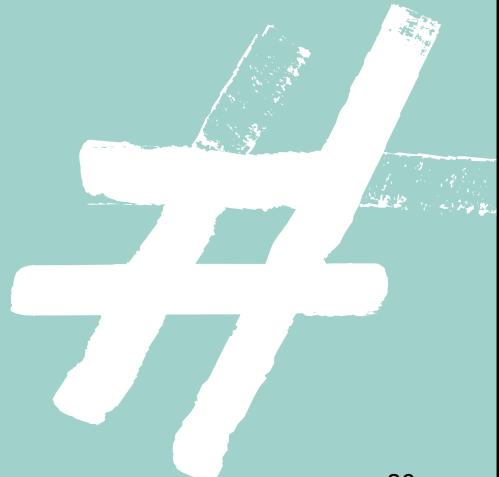
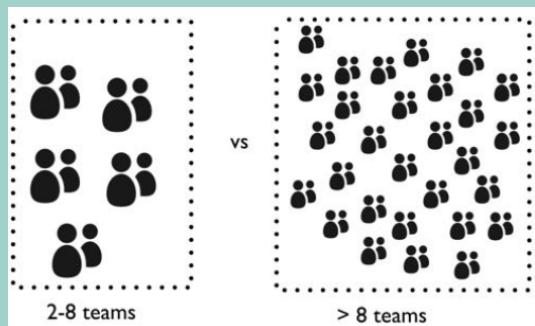
LeSS PRiNCiPLES



25

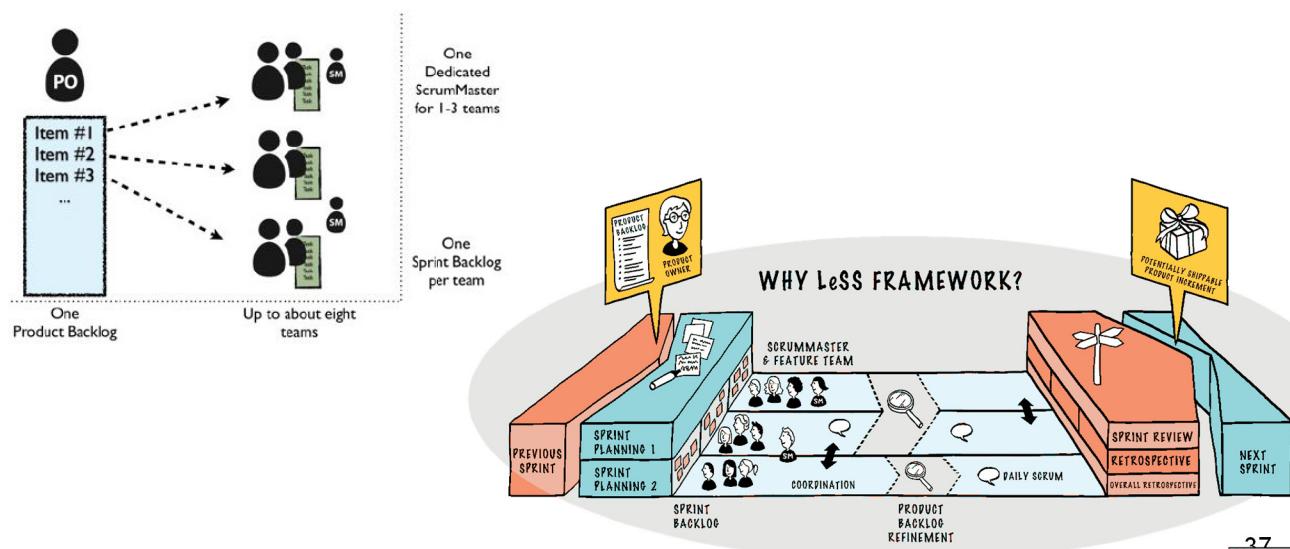
TWO FRAMEWORKS

- LeSS and LeSS Huge



36

LeSS

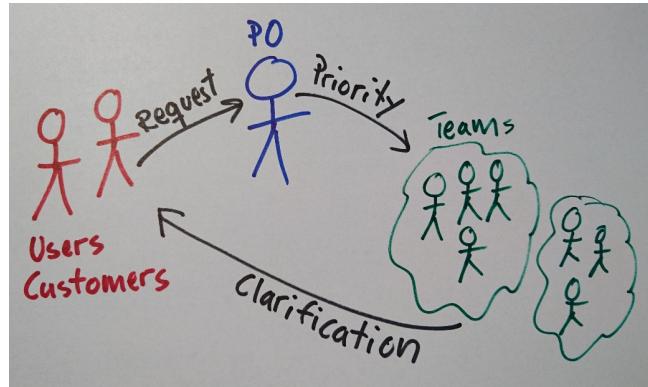


37

PRODUCT OWNER

knowit INSIGHT

- The Product Owner need to proactively ensure the old structures are replaced, and act as a connector of developers and users.
- Prioritization over Clarification

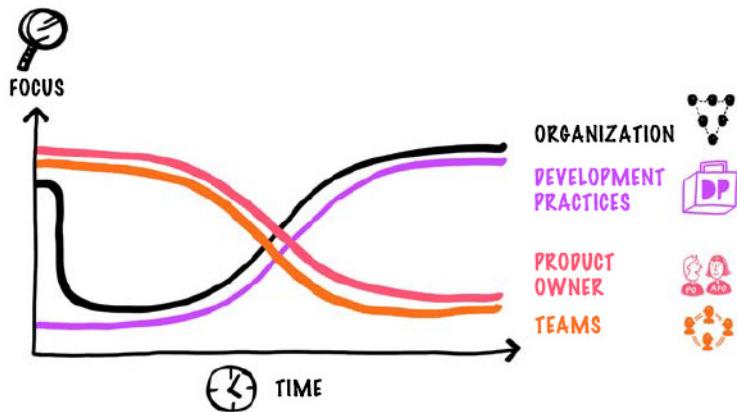


38

SCRUM MASTER

knowit INSIGHT

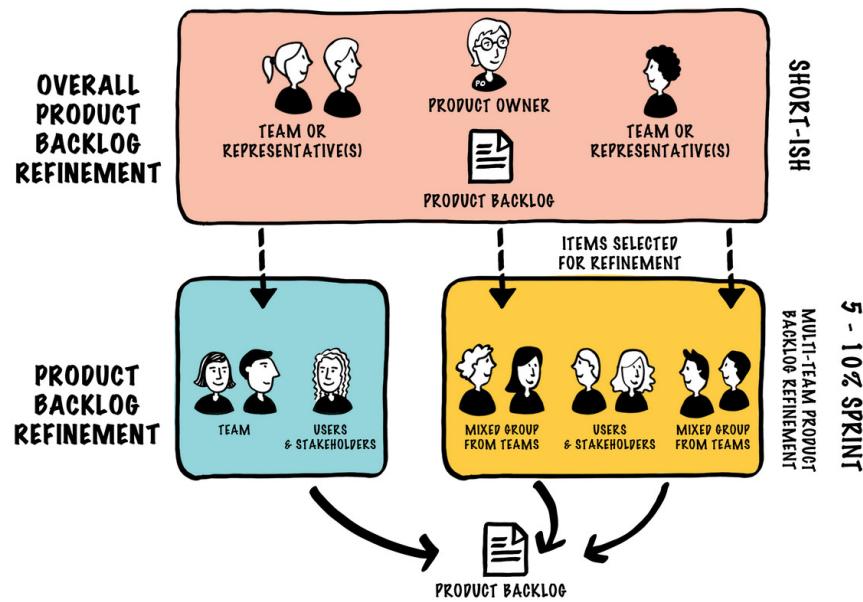
- A Scrum Master is not part-time team member, team representative or “team lead”
- Role often misunderstood by organizations new to agile



39

PRODUCT BACKLOG REFINEMENT

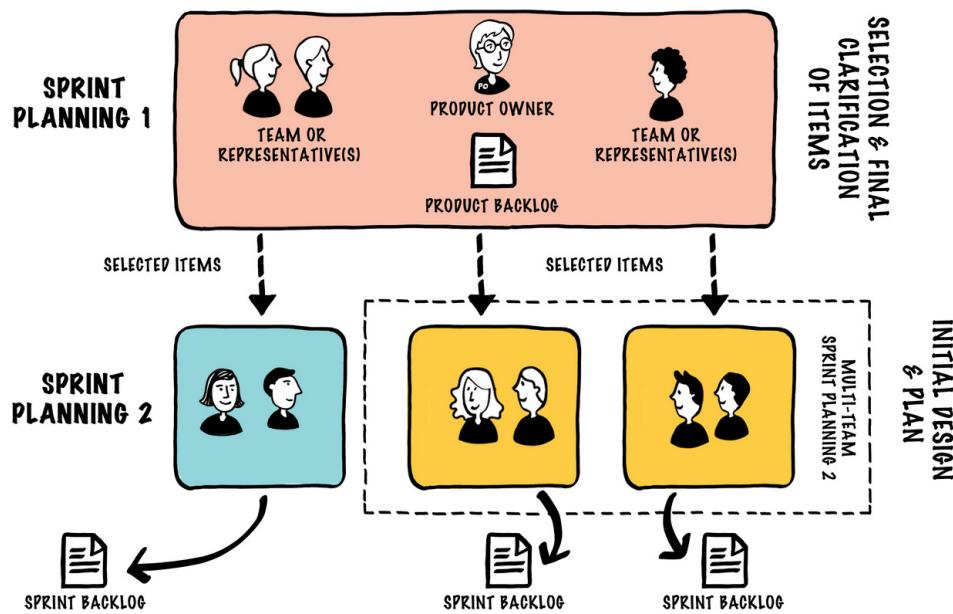
knowit INSIGHT



40

SPRINT PLANNING

knowit INSIGHT

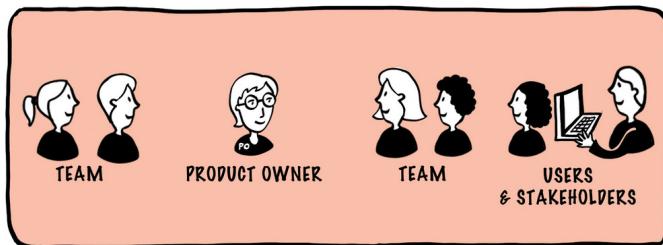


41

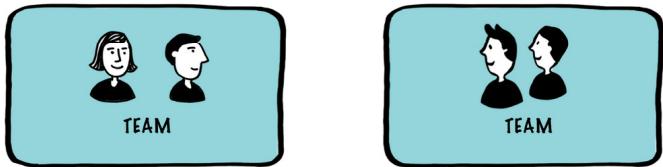
SPRINT REVIEW AND RETROSPECTIVE

knowit INSIGHT

SPRINT REVIEW



TEAM RETROSPECTIVE



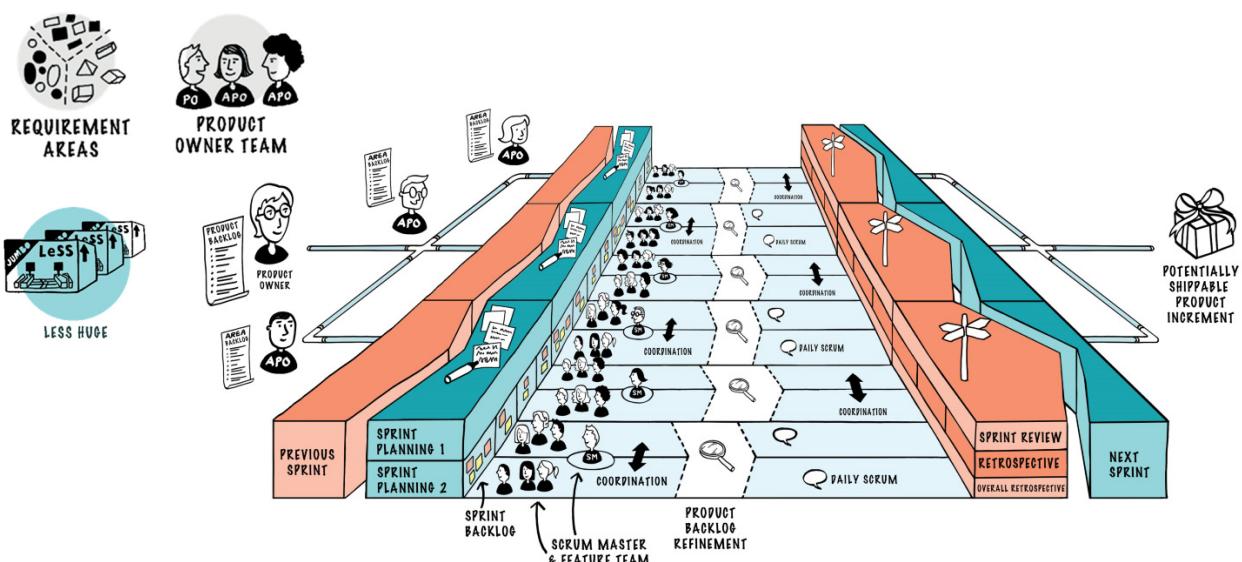
OVERALL RETROSPECTIVE



42

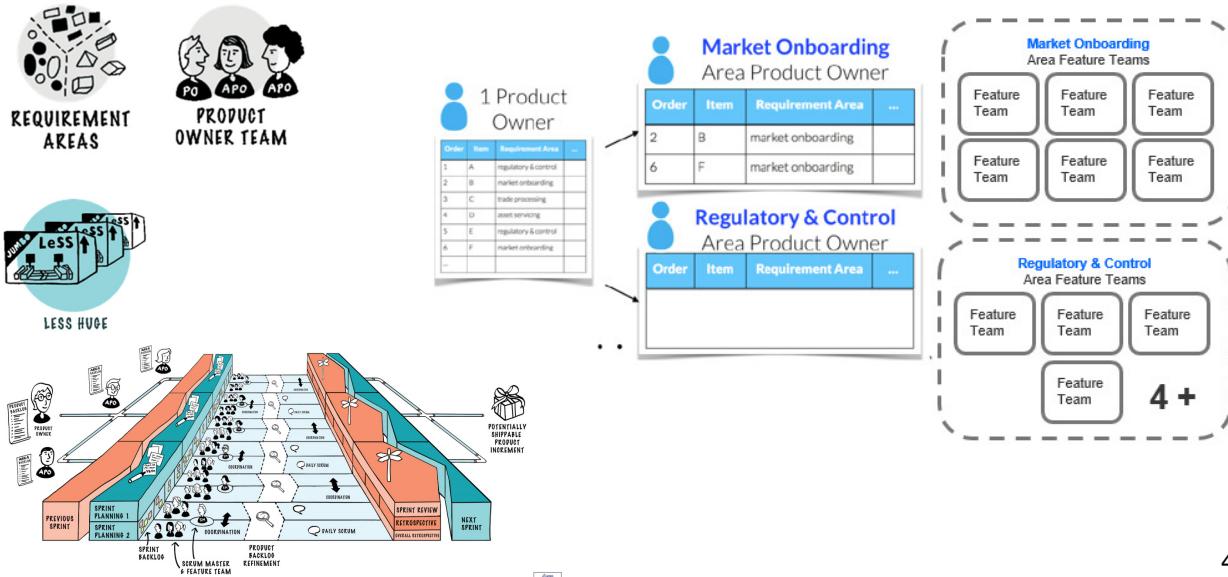
LeSS HUGE – STACKS OF LeSS

knowit INSIGHT



LeSS HUGE – REQUIREMENT AREAS

knowit INSIGHT



44

MANAGERS

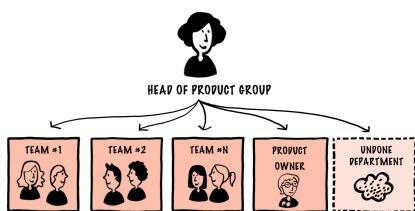


46

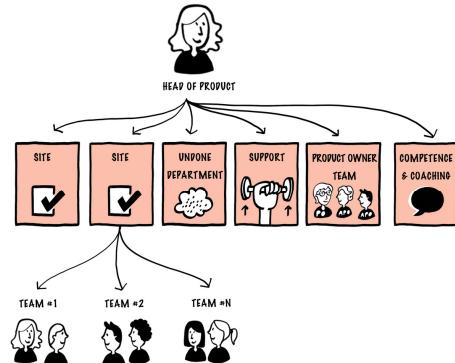
ORGANIZATIONAL STRUCTURE

knowit INSIGHT

LeSS



LeSS Huge



- Surprisingly simple structure
- LeSS Huge may add
 - Support, such as CM & CI
 - Undone Departments, such as Architecture, QA & Test
 - Competence & Coaching

50

SUMMARY - More with LeSS

knowit INSIGHT



59

Do more

- System optimization
- Empirical process control
- “Barely sufficient methodology”
- Descaling with simplicity & freedom
- Teams own & evolve their processes
- Value
- Experimenting & learning & improving
- Theory Y
- Build up from “why”

Do less

- Local optimization
- Defined & prescriptive processes
- Big methodology
- Scaling with complexity & control
- Defined processes pushed on to teams
- Waste, roles, artifacts, processes (*delete, don't add*)
- Conforming to “best practices”
- Theory X
- Tailoring down

60

1. Knowit
2. Why go Agile – why scale?
3. Scaling Agile Frameworks
4. LeSS Framework

5.SAFe Framework

6. Comparisons
7. Discussions / Reflections

AGENDA

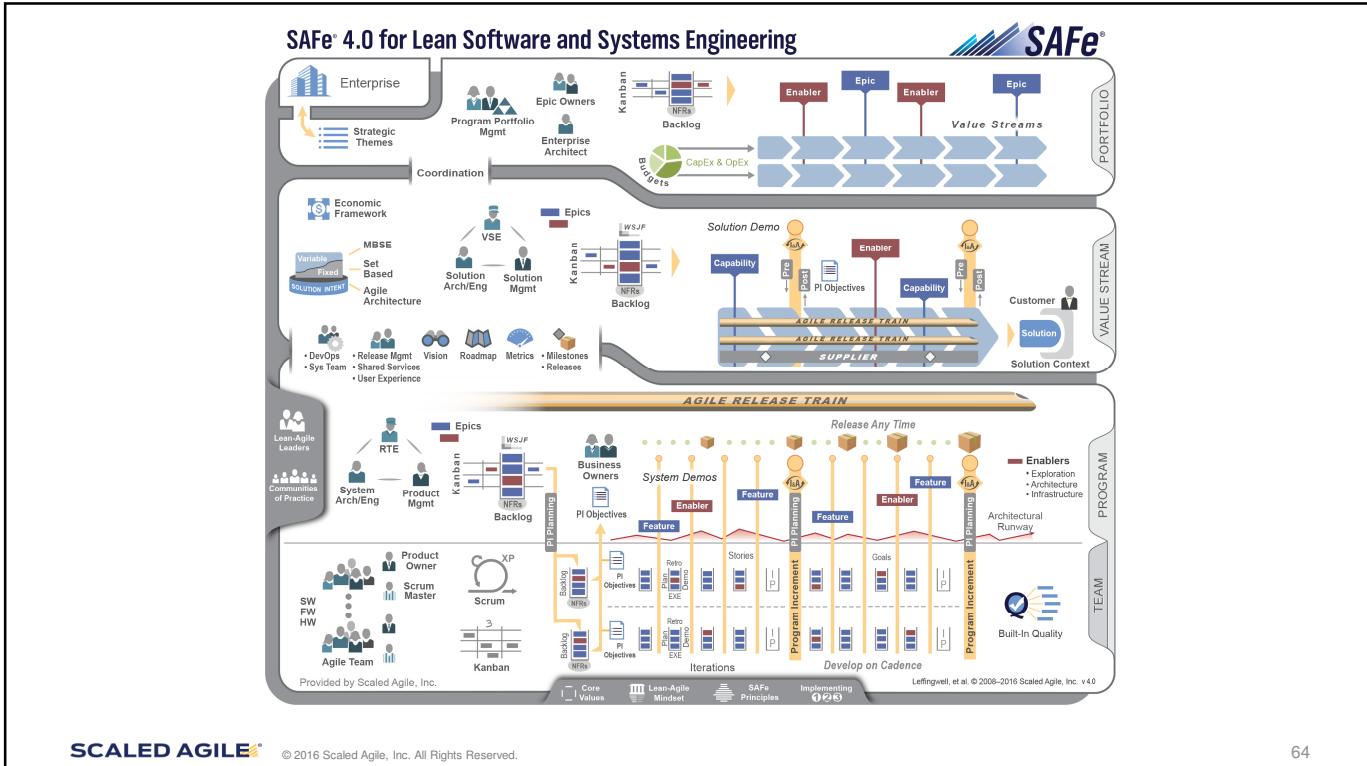


SCALED AGILE®

© 2016 Scaled Agile, Inc. All Rights Reserved.

V4.0.0

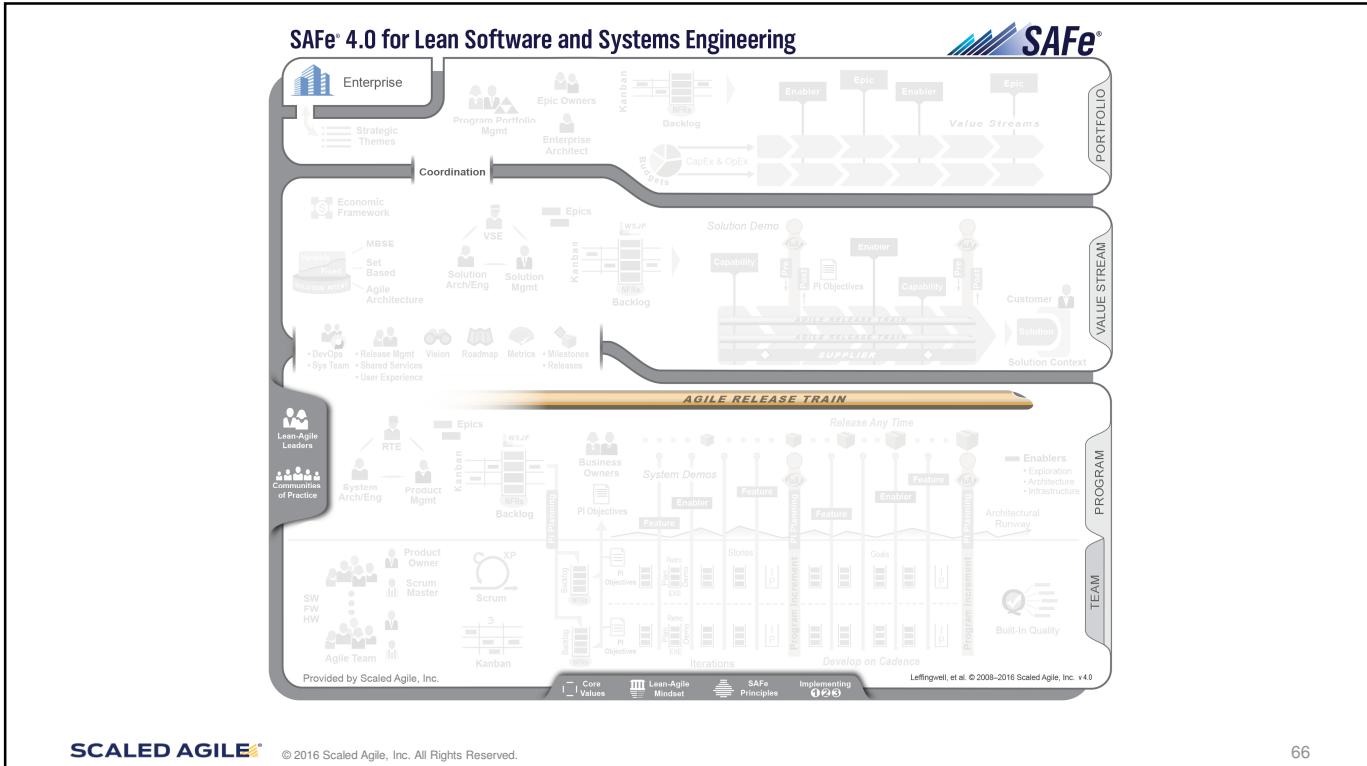
Proven



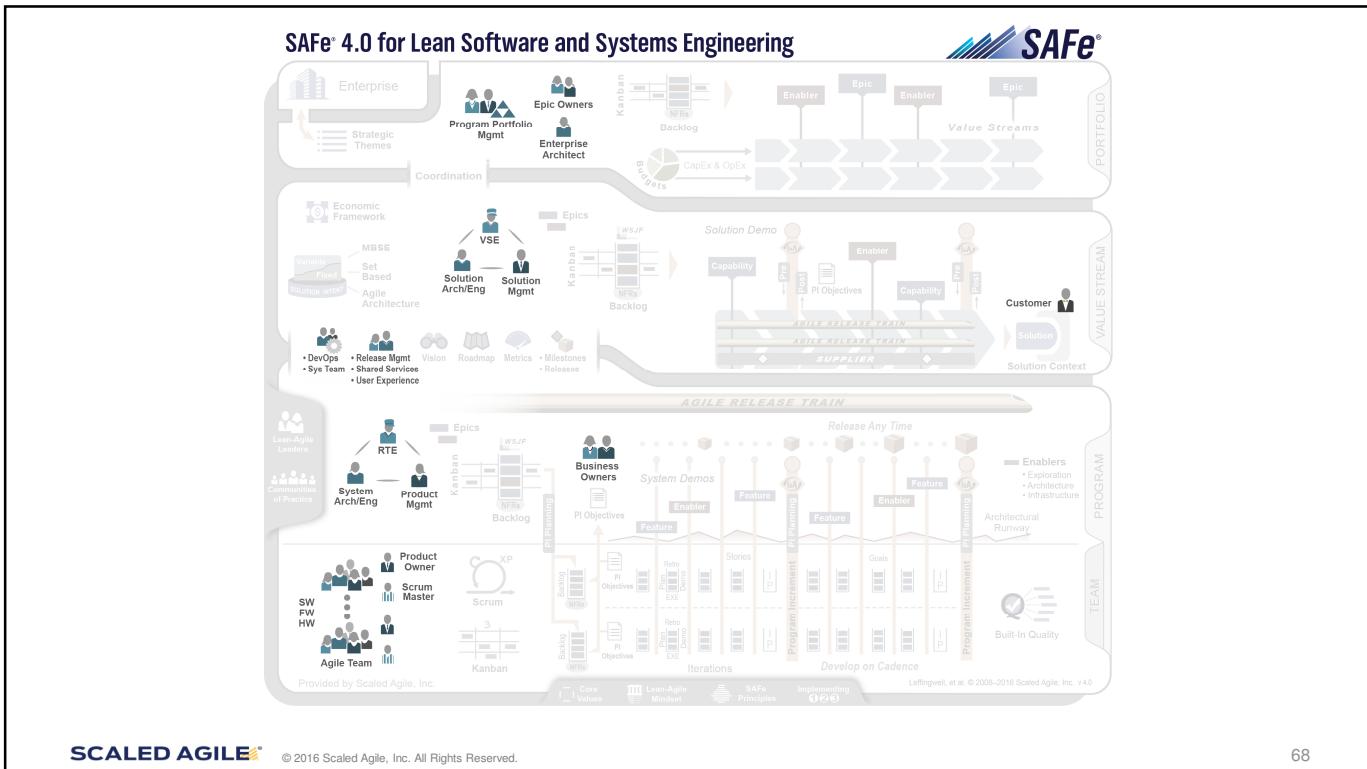
64

The Levels

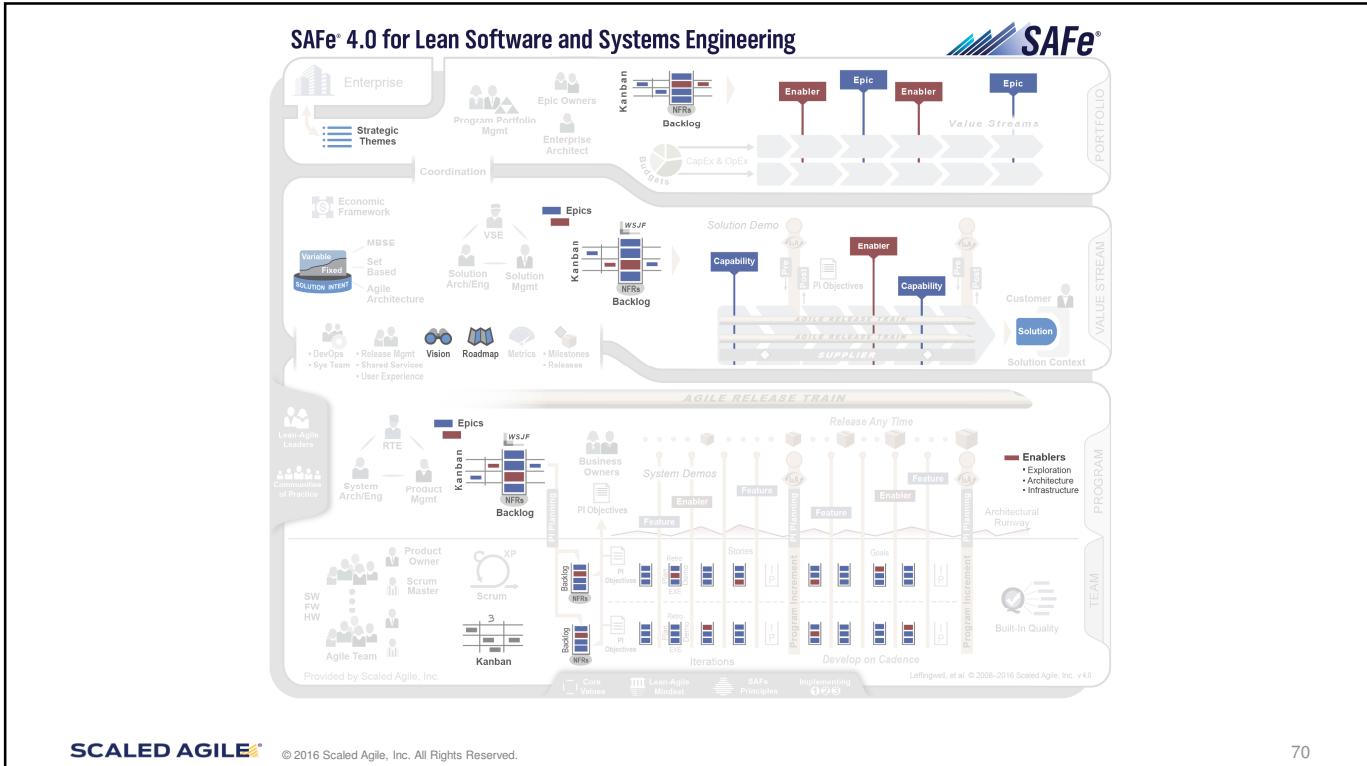
65



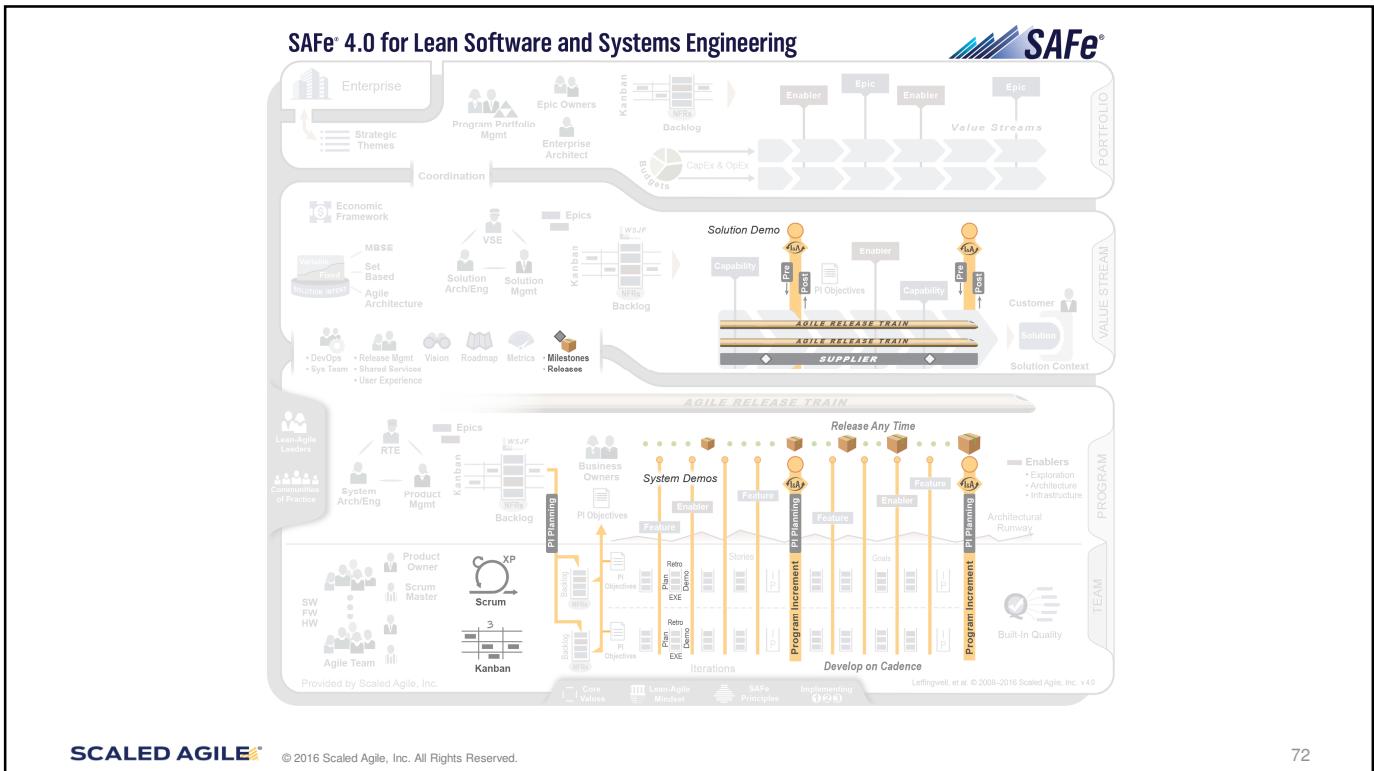
The People



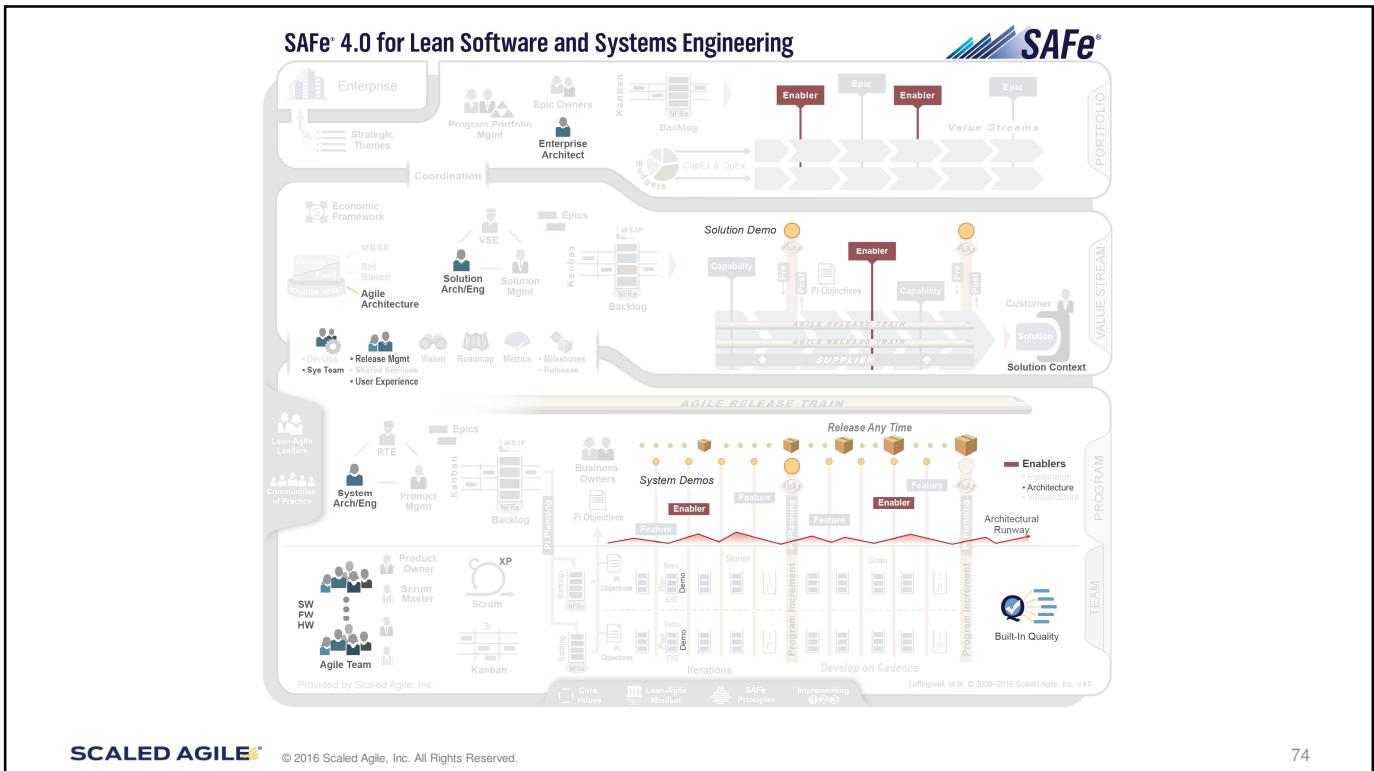
The Backlogs



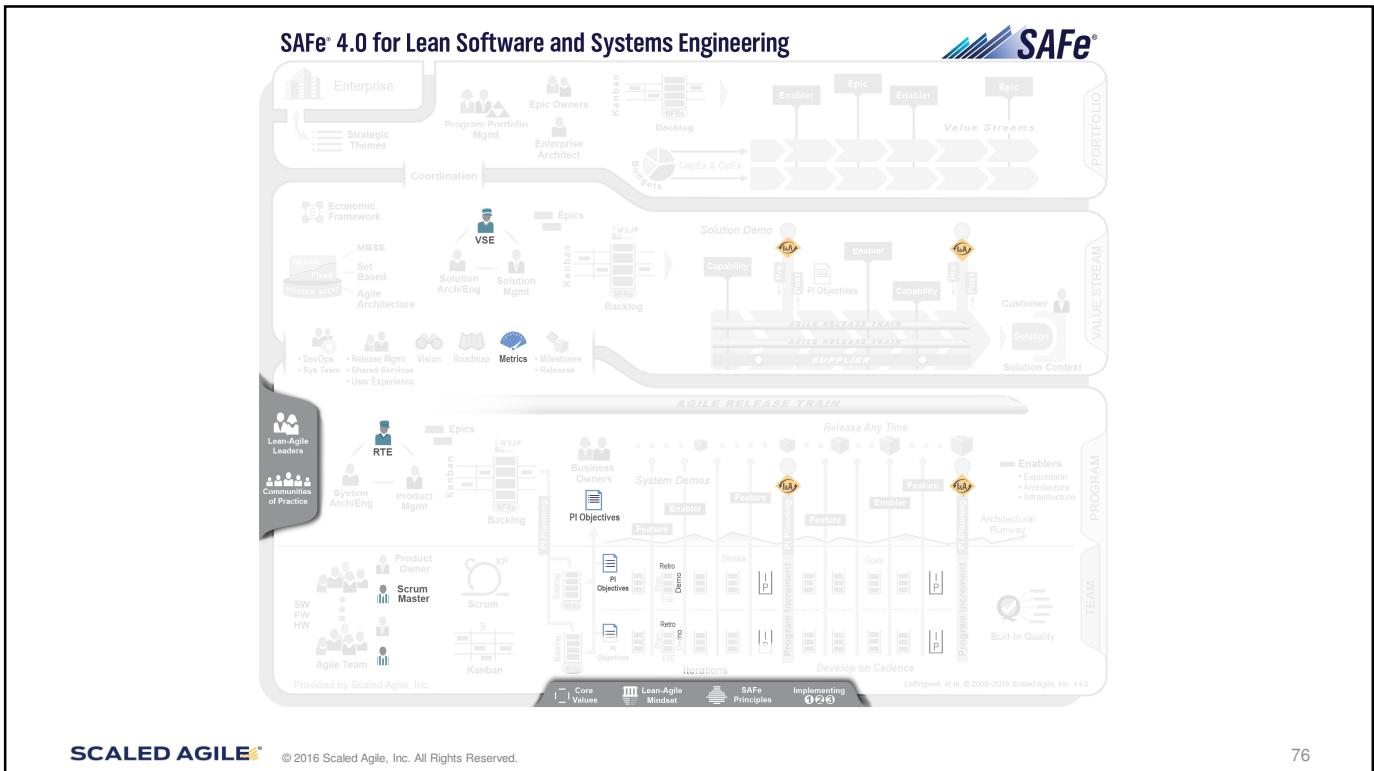
The Cadence



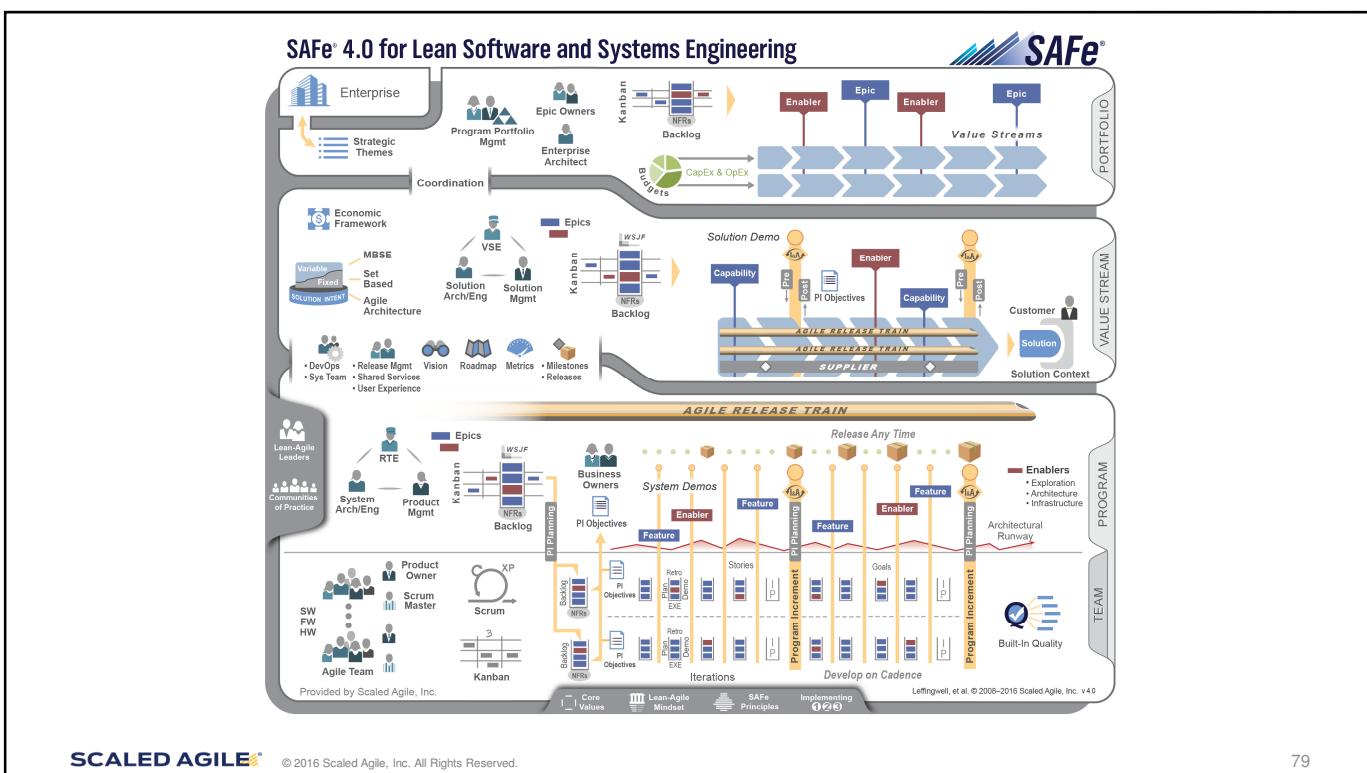
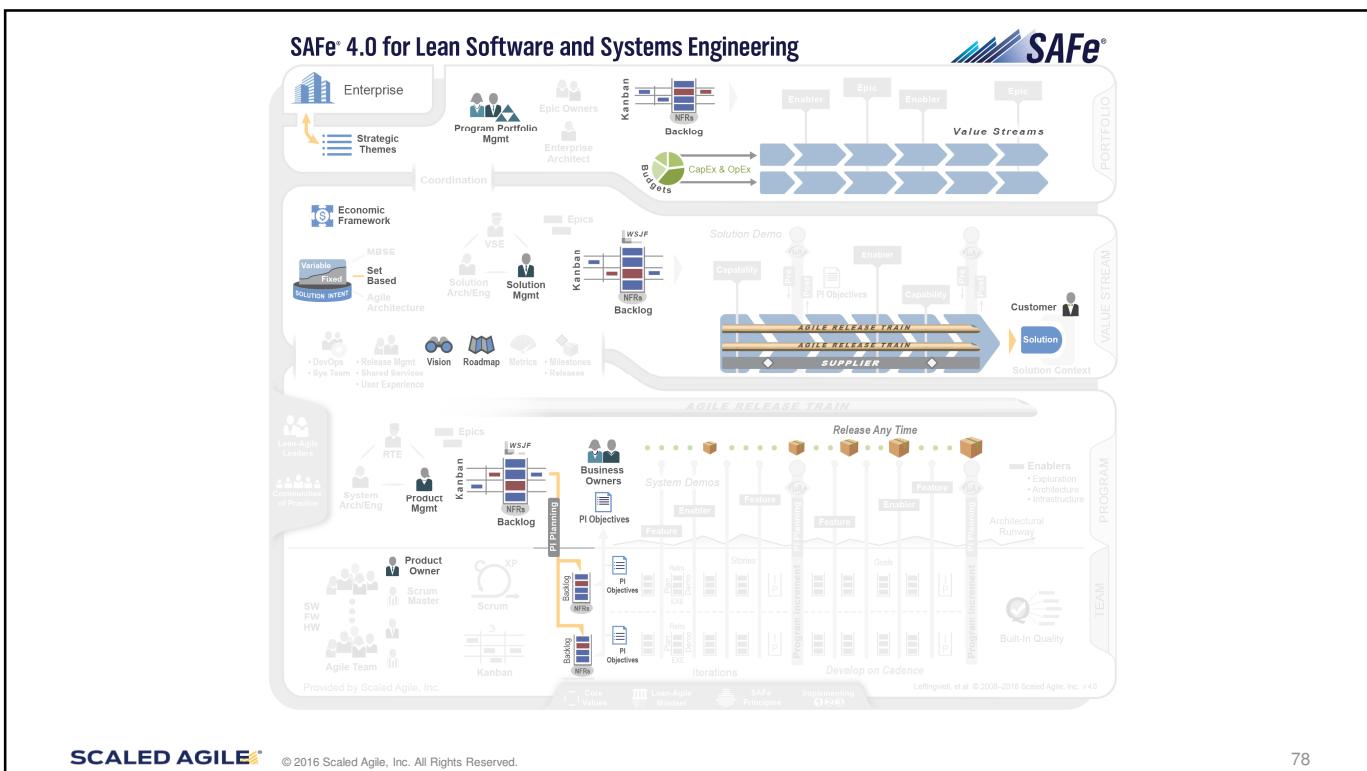
Quality



Relentless Improvement



Value Delivery



1. Knowit
2. Why go Agile – why scale?
3. Scaling Agile Frameworks
4. LeSS Framework
5. SAFe Framework
6. Comparisons

knowit INSIGHT

7.Discussions / Reflections

AGENDA

6.2 Prioritize the Program Backlog

PROGRAM
6.99

- Enablers
 - Exploration
 - Architecture
 - Infrastructure

Architectural Runway

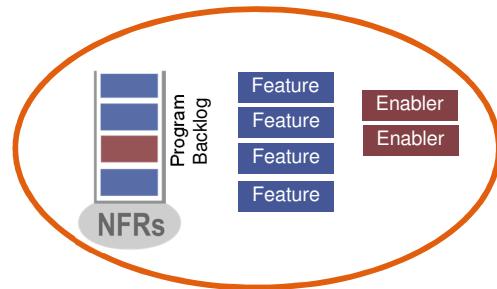


Prioritize Features for optimal ROI

In a flow system, **job sequencing** is the key to economic outcomes.

To prioritize based on lean economics,
we need to know two things:

1. What is the Cost of Delay (CoD) in delivering value?
2. What is the cost to implement the valuable thing?

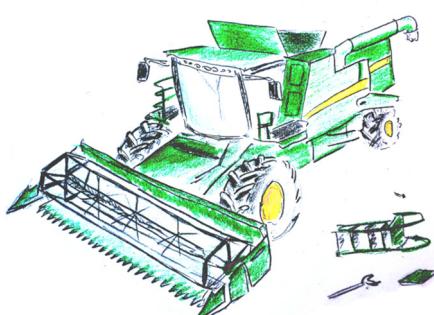


If you only quantify one thing, quantify the Cost of Delay.

—Donald G. Reinertsen, *Principles of Product Development Flow E3*

Example with equal CoD: which job first?

A \$\$, 1 day



B \$\$, 3 days

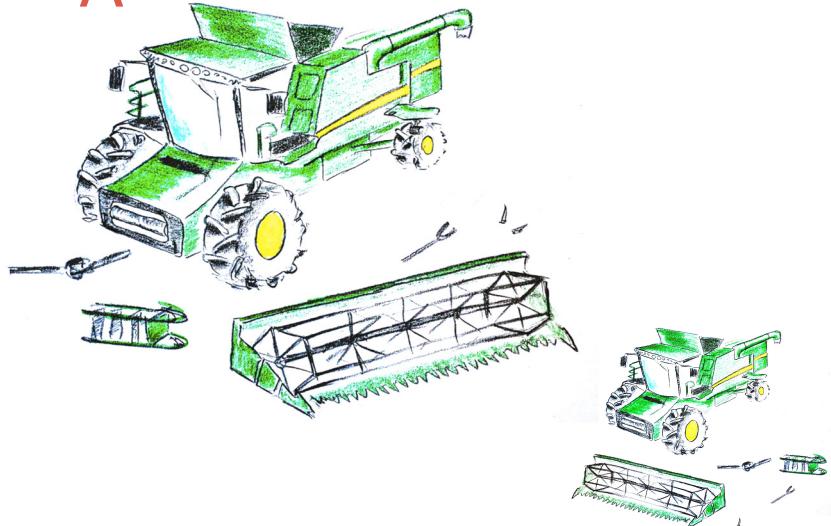


C \$\$, 10 days



Example with equal Duration: which job first?

A \$\$\$, 3 days



B \$\$, 3 days



C \$, 3 days

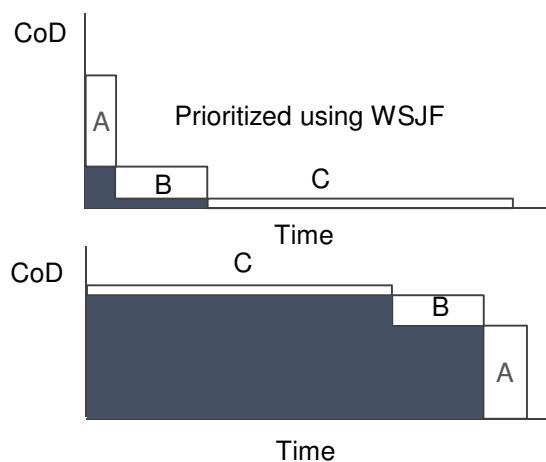


SCALED AGILE® © 2016 Scaled Agile, Inc. All Rights Reserved.

6.102

General case: any CoD and Duration

In the general case, give preference to jobs with **shorter Duration** and **higher CoD**, using *Weighted Shortest Job First* (WSJF):



$$\text{WSJF} = \frac{\text{CoD}}{\text{Duration}}$$

Feature	Duration	CoD	WSJF
A	1	10	10
B	3	3	1
C	10	1	0.1

— Dark area: total Cost of Delay

Adapted from *The Principles of Product Development Flow*, Donald G. Reinertsen

6.103

SCALED AGILE® © 2016 Scaled Agile, Inc. All Rights Reserved.

Components of Cost of Delay

User and business value

Relative value to the customer or business

- ▶ They prefer this over that
- ▶ Revenue impact?
- ▶ Potential penalty or other negative impact?

Time criticality

How User/Business Value decays over time

- ▶ Is there a fixed deadline?
- ▶ Will they wait for us or move to another solution?
- ▶ What is the current effect on customer satisfaction?

Risk Reduction & Opportunity Enablement (RR & OE)

What else does this do for our business

- ▶ Reduce the risk of this or future delivery?
- ▶ Is there value in the information we will receive?
- ▶ Enable new business opportunities?

Calculate WSJF with relative estimating

- ▶ In order to calculate WSJF, teams need to estimate Cost of Delay and duration
- ▶ For duration, use job size as a quick proxy for duration
- ▶ Relative estimating is a quick technique to estimate job size and relative value
- ▶ WSJF stakeholders: Business Owners, Product Managers, Product Owners, System Architects

$$\text{WSJF} = \frac{\text{CoD}}{\text{Job size}} = \frac{\text{User-business value} + \text{Time criticality} + \text{RR | OE value}}{\text{Job size}}$$

WSJF prioritization matrix

The job with the highest WSJF provides the greatest economic benefit.

$$\text{WSJF} = \frac{\text{CoD}}{\text{Job size}} = \frac{\text{User-business value} + \text{Time criticality} + \text{RR | OE value}}{\text{Job size}}$$

Feature	User-business value	Time criticality	RR OE value	CoD	Job size	WSJF

Scale for each parameter: 1, 2, 3, 5, 8, 13, 20

Note: Do one *column* at a time, start by picking the smallest item and giving it a “1.”

There must be at least one “1” in each column!

