

# SAFe® for Teams

Empowering teams to deliver value in the SAFe enterprise

**SAFe® Course** Attending this course gives students access to the SAFe Practitioner exam and related preparation materials.



V4.5.0.1

**SCALED AGILE®**

© Scaled Agile, Inc.

## SAFe® for Teams

- 1. Introducing the Scaled Agile Framework
- 2. Building an Agile Team
- 3. Planning the Iteration
- 4. Planning the Iteration
- 5. Executing the PI

# Logistics

- ▶ Class times
- ▶ Breaks
- ▶ Lunch
- ▶ Restrooms
- ▶ Other



To perform the role of a SAFe Practitioner, you should be able to:

- ▶ Apply SAFe to scale Lean and Agile development in your enterprise
- ▶ Know your team and its role on the Agile Release Train
- ▶ Know all other teams on the train, their roles, and the dependencies between the teams
- ▶ Plan iterations
- ▶ Execute iterations and demonstrate value
- ▶ Plan program increments
- ▶ Integrate and work with other teams on the train



# Lesson 1

## Introducing the Scaled Agile Framework

- 1. Introducing the Scaled Agile Framework
- 2. Building an Agile Team
- 3. Planning the Iteration
- 4. Executing the Iteration
- 5. Executing the PI

**SAFe® Course** Attending this course gives students access to the SAFe Practitioner exam and related preparation materials.

## Learning objectives

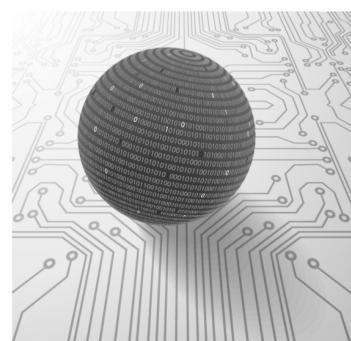
- 1.1 Connect with the Scaled Agile Framework
- 1.2 Explore Lean, the Agile Manifesto, and SAFe Principles
- 1.3 Identify Scrum, Kanban and XP Practices

## 1.1 Connect with the Scaled Agile Framework

### How do we keep pace?

Our development methods must keep pace with an increasingly complex world.

- ▶ We've had Moore's Law for hardware, and now software is eating the world
- ▶ Our development practices haven't kept pace; Agile shows the greatest promise, but was developed for small teams
- ▶ We need a new approach, one that harnesses the power of Agile and Lean, and applies to the needs of the enterprises who build the world's most important software and systems.



We thought we'd be developing like this.



SCALED AGILE® © Scaled Agile, Inc.

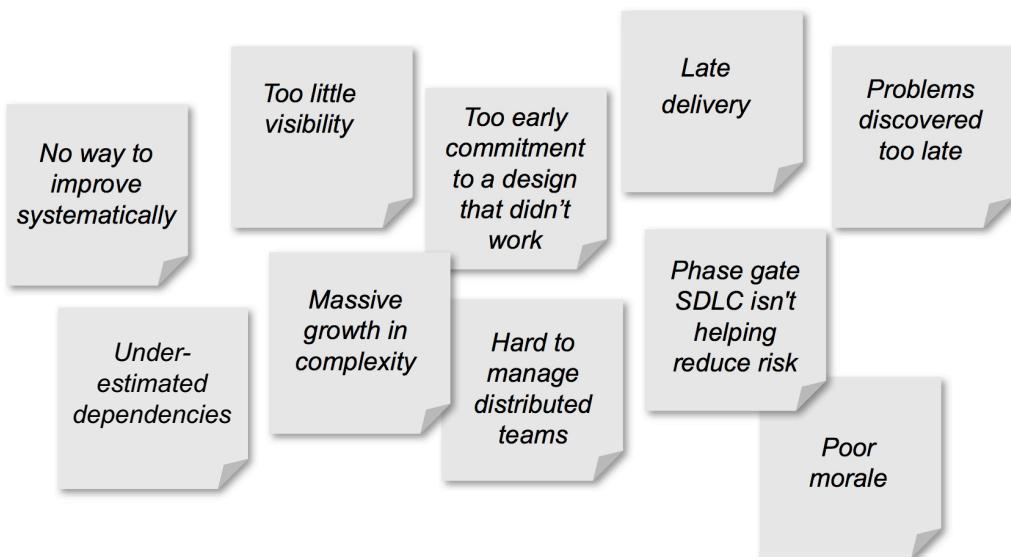
But sometimes it feels like this.



Library of Congress

SCALED AGILE® © Scaled Agile, Inc.

And our retrospectives read like this:



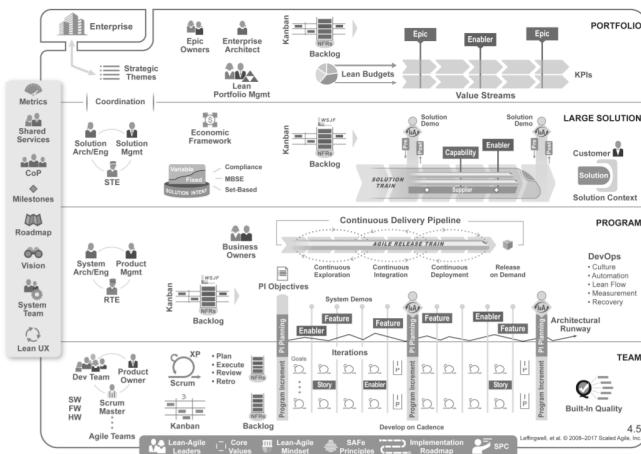
*Knowledge for people building the world's most important systems*

SAFe® is a freely revealed knowledge base of integrated, proven patterns for enterprise Lean-Agile development.



# The Scaled Agile Framework® (SAFe®)

Synchronizes alignment, collaboration, and delivery for large numbers of teams.



## Core Values

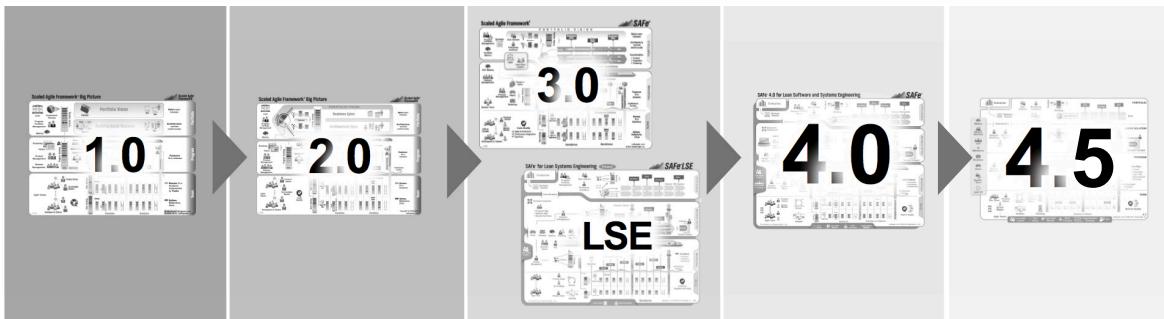
1. Built-In Quality
2. Program execution
3. Alignment
4. Transparency

## Roots, past, present and future

Field experience at enterprise scale

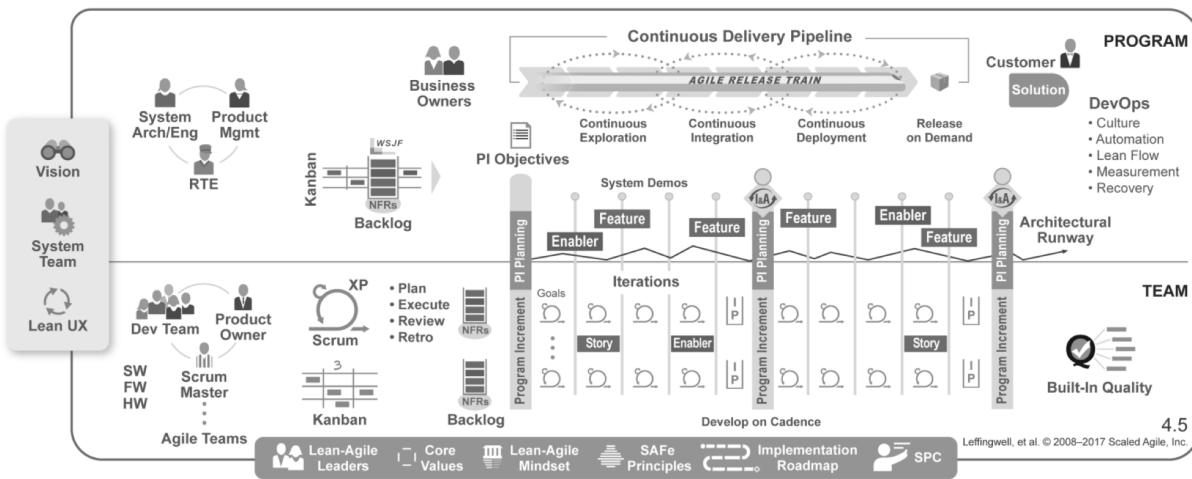
2011

Now...



Agile development | Lean product development | Systems thinking

## Essential SAFe provides the basis for success



## Nothing beats an Agile Team

- ▶ Empowered, self-organizing, self-managing, cross-functional team
- ▶ Delivers valuable, tested, working system every two weeks
- ▶ Uses a team framework which combines the best of Scrum project management, XP-inspired technical practices and Kanban for flow

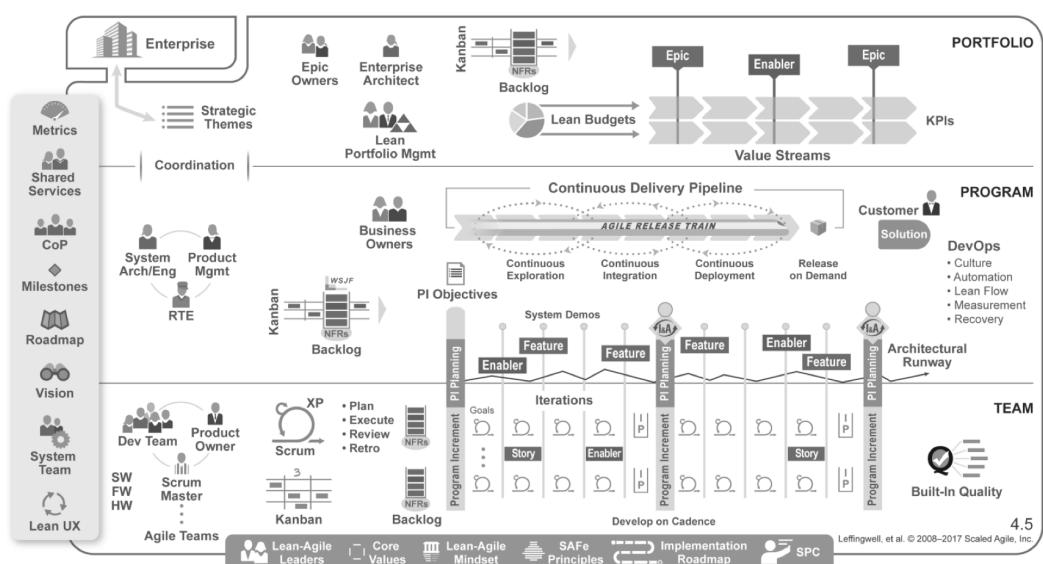


## Except a team of Agile Teams

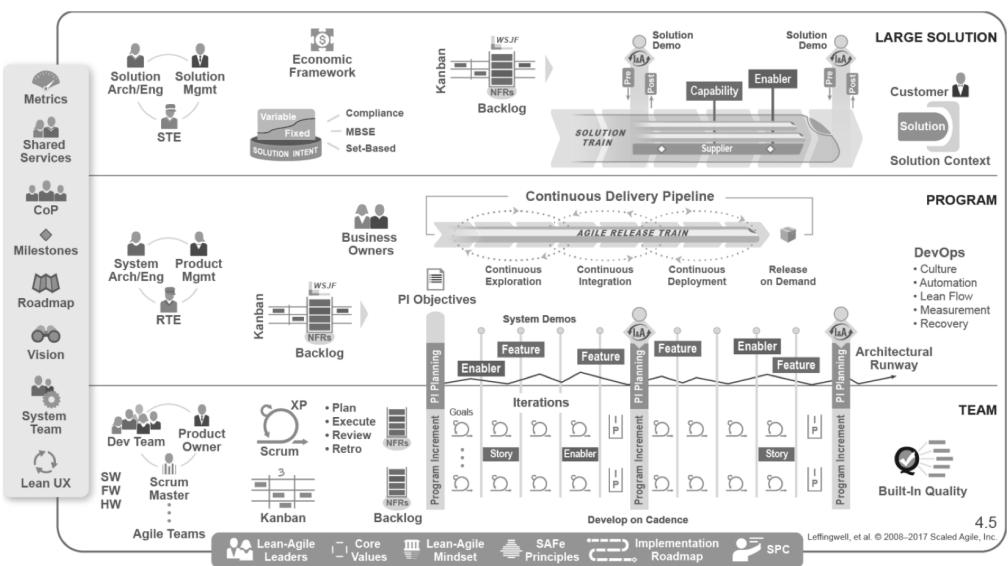
- ▶ Self-organizing, self-managing, team of agile teams
- ▶ Delivers working, tested full system increments every two weeks
- ▶ Operates with Vision, architecture and UX guidance
- ▶ Common iteration lengths and estimating
- ▶ Face-to-face planning for collaboration, alignment, and adaptation



## Portfolio SAFe adds Lean Portfolio governance



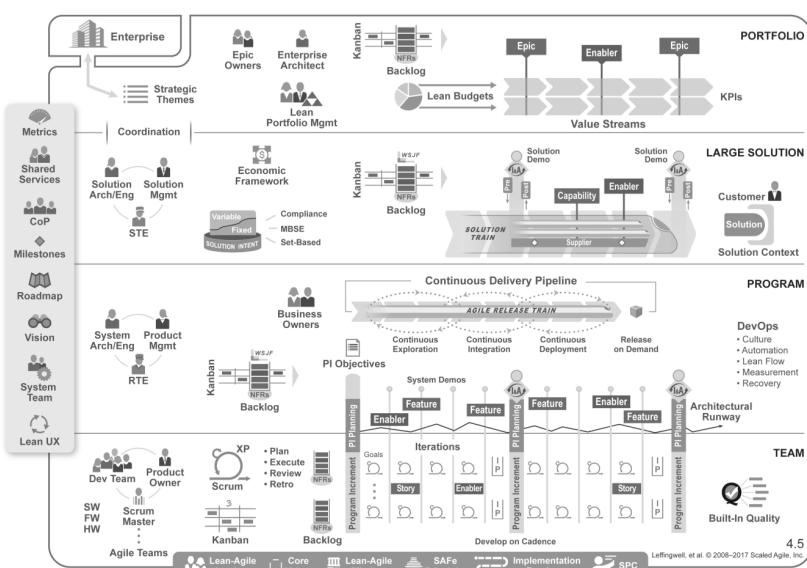
## Large Solution SAFe is coordinated by a Solution Train



SCALED AGILE® © Scaled Agile, Inc.

1.19

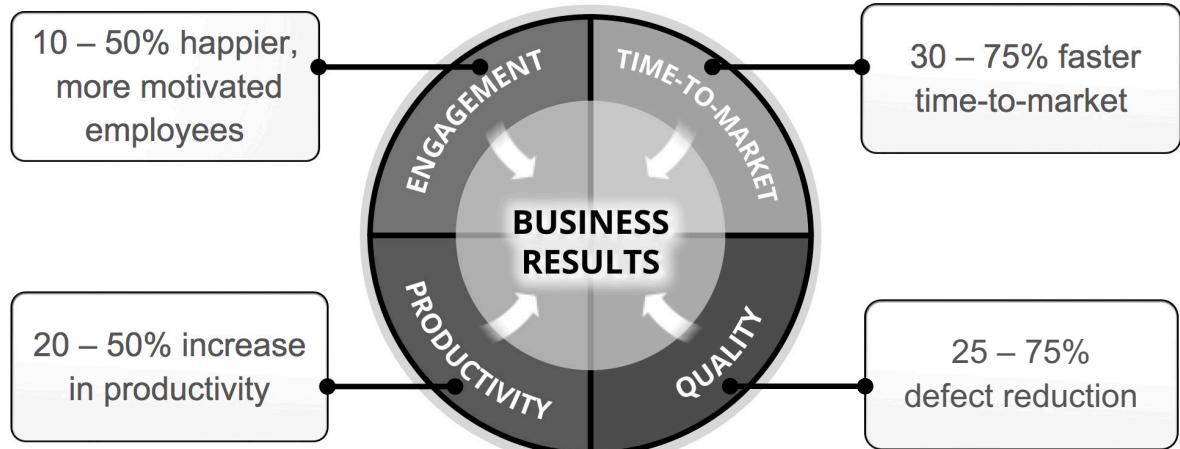
## Some enterprises require Full SAFe



SCALED AGILE® © Scaled Agile, Inc.

1.20

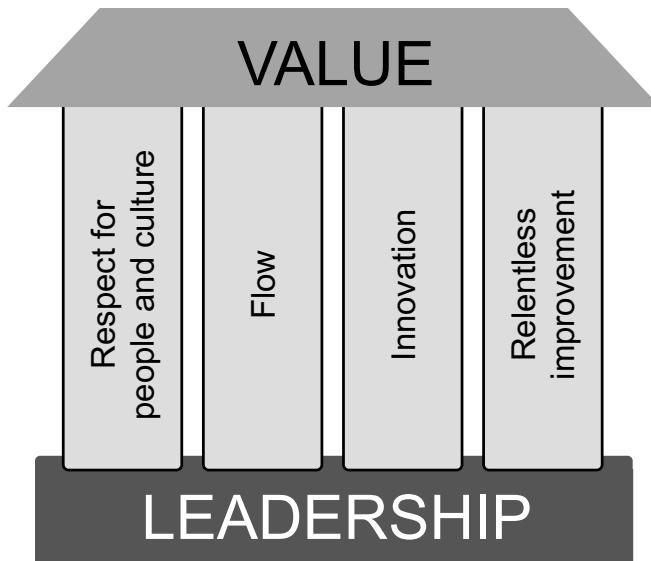
## Get business results



ScaledAgileFramework.com/case-studies

## 1.2 Explore Lean, the Agile Manifesto, and SAFe Principles

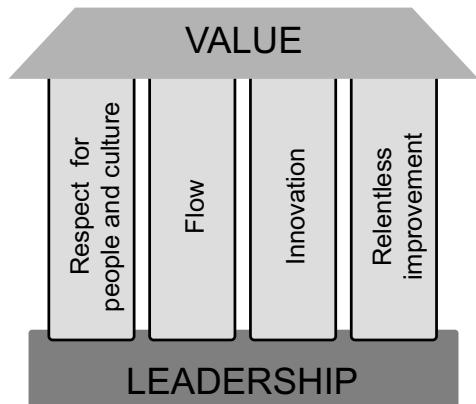
## SAFe House of Lean



SCALED AGILE® © Scaled Agile, Inc.

1.23

## Purpose



Achieve the sustainably shortest lead time with:

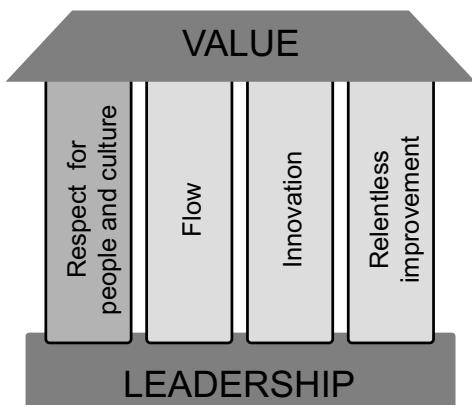
- ▶ Best quality and value to people and society
- ▶ High morale, safety and customer delight

*There is only one boss. The customer.  
And he can fire everybody in the company.  
—Sam Walton*

SCALED AGILE® © Scaled Agile, Inc.

1.24

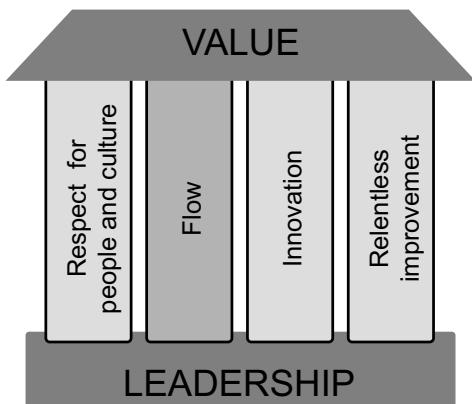
## Respect for people and culture



*Culture eats strategy for breakfast.*  
—Peter Drucker

- ▶ People do all the work
- ▶ Your customer is whoever consumes your work
  - ▶ Don't overload them
  - ▶ Don't make them wait
  - ▶ Don't force them to do wasteful work
  - ▶ Don't impose wishful thinking
- ▶ Build long-term partnerships based on trust
- ▶ Cultural change comes last, not first
- ▶ To change the culture, you have to change the organization

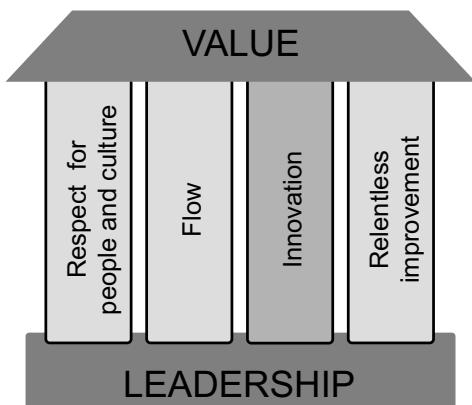
## Flow



*Operating a product development process near full utilization is an economic disaster.*  
—Don Reinertsen

- ▶ Optimize continuous and sustainable throughput of value
- ▶ Avoid start-stop-start project delays
- ▶ Build quality in; flow depends on it
- ▶ Understand, exploit and manage variability
- ▶ Integrate frequently
- ▶ Informed decision-making via fast feedback

## Innovation



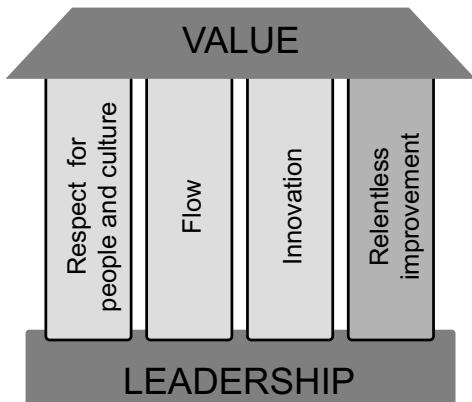
*Innovation comes from the producer.*

—W. Edwards Deming

- ▶ Producers innovate; customers validate
- ▶ Get out of the office (Gemba\*)
- ▶ *No useful improvement was ever invented at a desk*  
— Taiichi Ohno
- ▶ Provide time and space for creativity
- ▶ Apply innovation accounting
- ▶ Pivot without mercy or guilt

\* Gemba: The “real place” where the work is actually done.

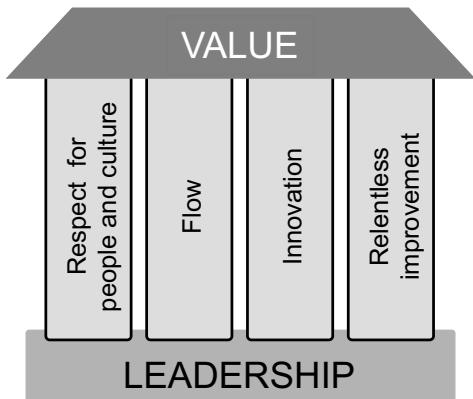
## Relentless improvement



*Those who adapt the fastest, win.*

- ▶ A constant sense of danger
- ▶ Optimize the whole
- ▶ Consider facts carefully, then act quickly
- ▶ Apply lean tools to identify and address root causes
- ▶ Reflect at key milestones; identify and address shortcomings

## Leadership



*People are already doing their best; the problems are with the system. Only management can change the system.*

—W. Edwards Deming

SCALED AGILE® © Scaled Agile, Inc.

1.29

- ▶ Lead the change
- ▶ Know the way; emphasize life-long learning
- ▶ Develop people
- ▶ Inspire and align with mission; minimize constraints
- ▶ Decentralize decision-making
- ▶ Unlock the intrinsic motivation of knowledge workers

## The Agile Manifesto

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.

 agilemanifesto.org

SCALED AGILE® © Scaled Agile, Inc.

1.30

# Agile Manifesto

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference for the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.



[agilemanifesto.org/principles.html](http://agilemanifesto.org/principles.html)

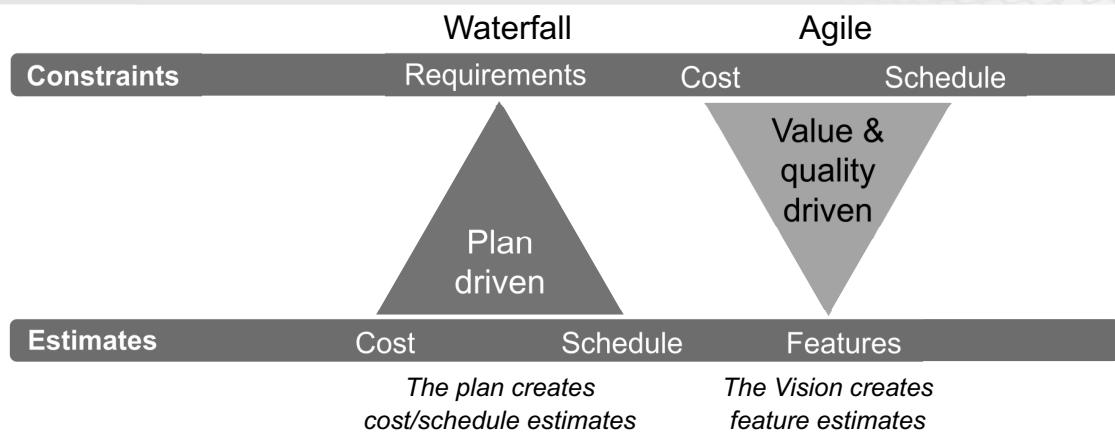
# Agile Manifesto

7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity—the art of maximizing the amount of work not done—is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.



[agilemanifesto.org/principles.html](http://agilemanifesto.org/principles.html)

## Agile turns development upside-down



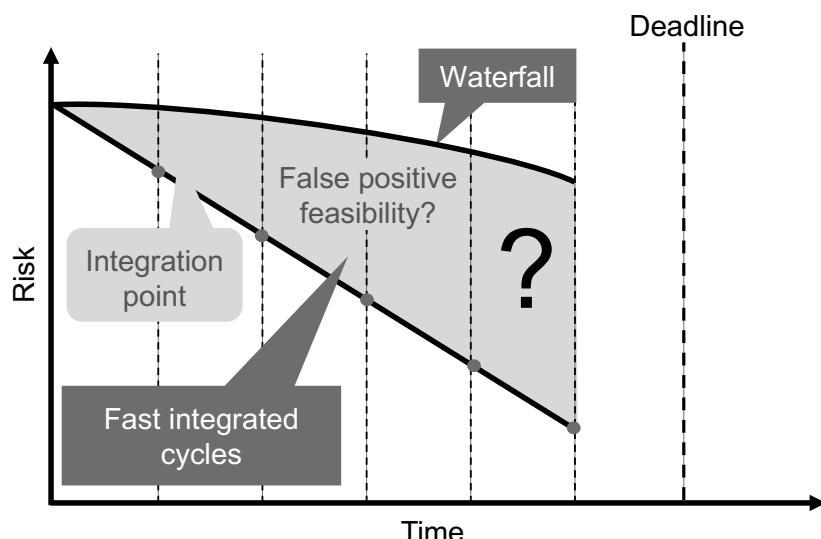
- ▶ Agile Teams show that *dates* matter and they *meet* their commitments
- ▶ Business Owners understand how *priorities* matter
- ▶ Fix *quality*, not scope

## SAFe Lean-Agile Principles

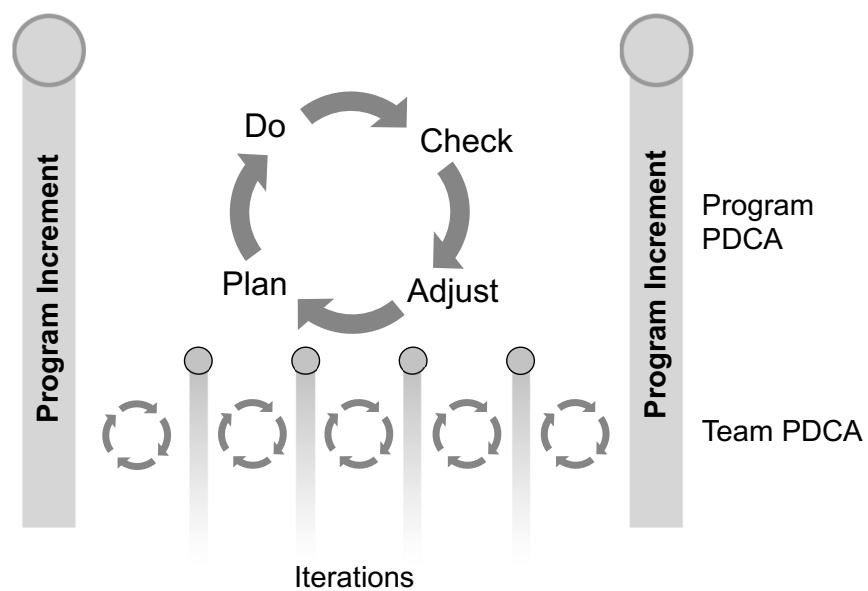
- #1-Take an economic view
- #2-Apply systems thinking
- #3-Assume variability; preserve options
- #4-Build incrementally with fast, integrated learning cycles
- #5-Base milestones on objective evaluation of working systems
- #6-Visualize and limit WIP, reduce batch sizes, and manage queue lengths
- #7-Apply cadence, synchronize with cross-domain planning
- #8-Unlock the intrinsic motivation of knowledge workers
- #9-Decentralize decision-making

#4

## Build incrementally with fast, integrated learning cycles



## Use Iterations and Program Increments to learn fast

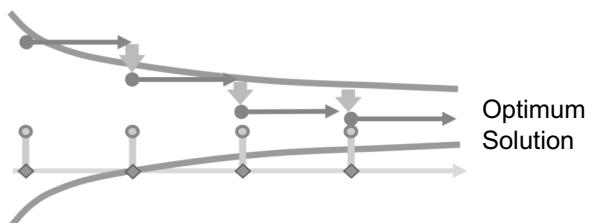


## #5 Base milestones on objective evaluation of working systems



- ▶ Phase gates force too-early design decisions, encourage false-positive feasibility
- ▶ Assume a “point” Solution exists and can be built right the first time

Objective Milestones facilitate learning and allow for continuous, cost-effective adjustments toward an optimum Solution



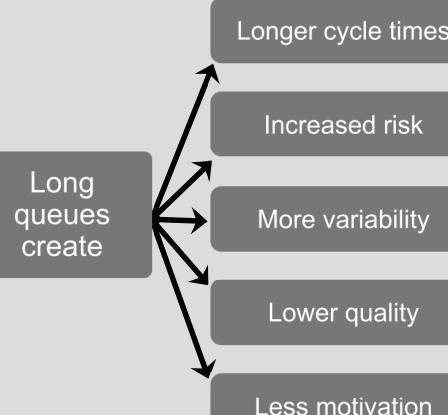
## #6 Visualize and limit WIP, reduce batch size, and manage queue lengths

- ▶ Understand Little’s Law
- ▶ Faster processing time decreases wait
- ▶ Control wait times by controlling queue lengths

$$W_q = \frac{L_q}{\lambda}$$

Average wait time = average queue length / average processing rate

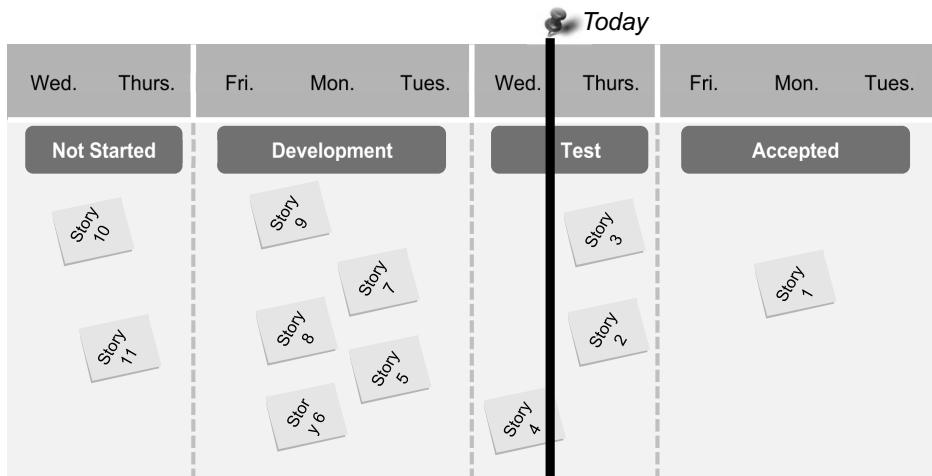
### Long queues: All bad



*Principles of Product Development Flow*, Don Reinertsen

## Visualize and limit work in progress

### One team's Big Visible Information Radiator (B VIR)



How is this team doing? How do you know that?

## Exercise: Large batch push

- In your team, choose a four-person group who will process the 10 coins on the table. One additional person is the group timekeeper. Other members are individual timekeepers.
- Each of the four people flips all coins one at a time, recording his own results (heads or tails)
- Then each person passes all coins at the same time to the next person
- The timekeeper records time from the start of the first flip to the completion of the last flip for the group. Each individual timekeeper records time for a single individual.



## Exercise: Small batch pull

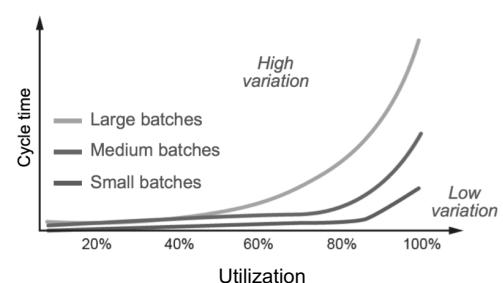
- ▶ Similar four-person process
- ▶ Each of the four people flips each coin one at a time and records the result
- ▶ People pull coins from the previous person as soon as he is done recording them, and process them immediately
- ▶ The timekeeper records the time from the start of the first flip to the completion of the last flip for the group. Each individual timekeeper records time for a single individual.



## Reduce batch size

Small batches go through the system faster, with lower variability.

- ▶ Large batch sizes increase variability
- ▶ High utilization increases variability
- ▶ Severe project slippage is the most likely result
- ▶ Most important batch is the transport (handoff) batch
- ▶ Proximity (co-location) enables small batch size
- ▶ Good infrastructure enables small batches

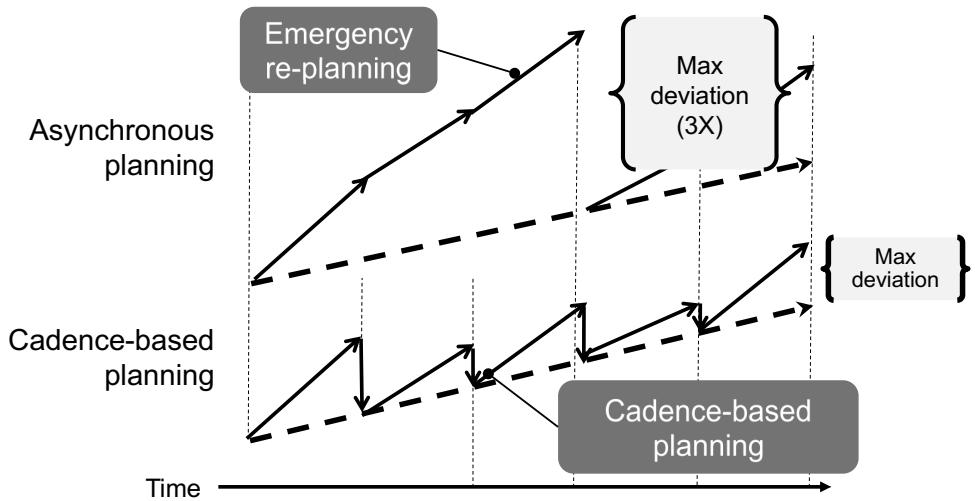


*Implementing Lean Software Development,*  
Mary Poppendieck

*Principles of Product Development Flow,*  
Don Reinertsen

## #7 Control variability with planning cadence

Cadence-based planning limits variability to a single interval.



SCALDED AGILE® © Scaled Agile, Inc.

1.43

## 1.3 Identify Scrum, Kanban, and XP Practices

SCALDED AGILE® © Scaled Agile, Inc.

1.44

## From traditional development to Agile

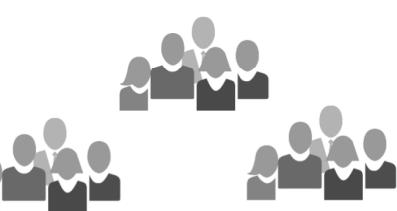
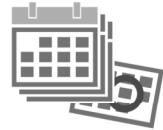
Instead of a large group



Working on all the requirements



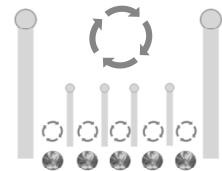
Integrating and delivering value toward the end of development



Have small teams working together as a program

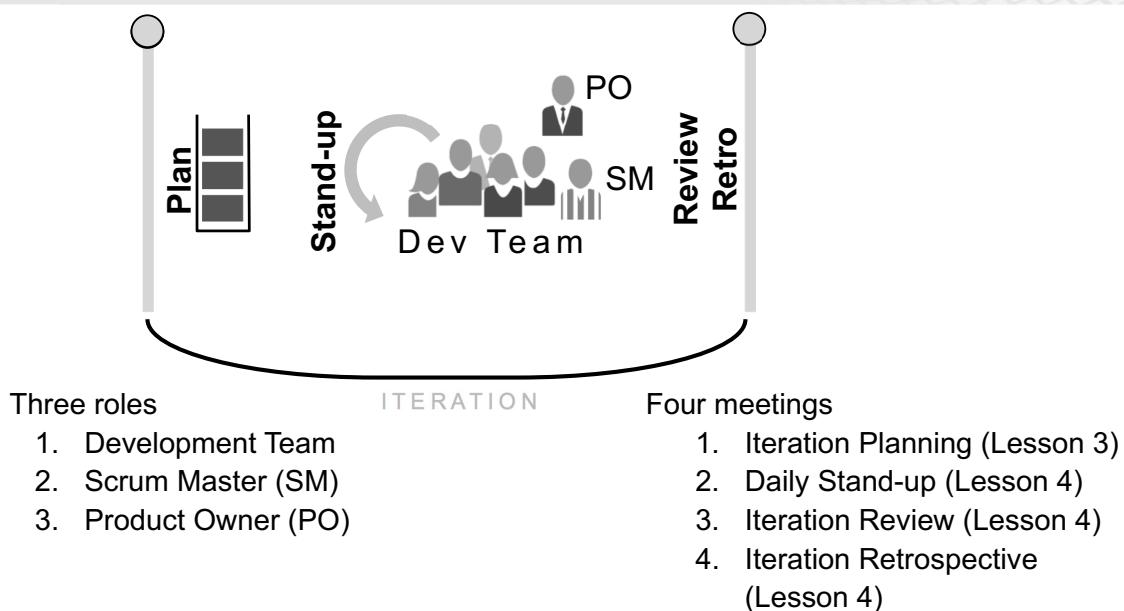


Working on small batches of requirements



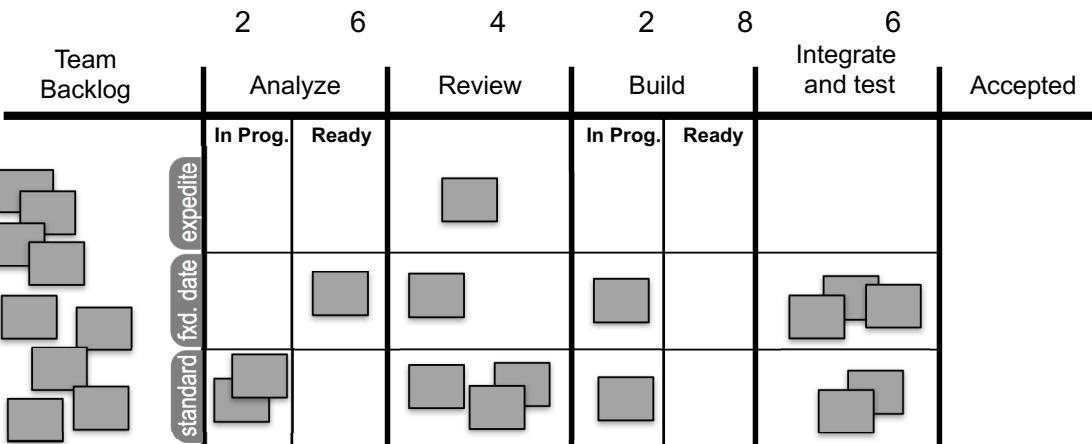
Delivering value in short timeboxes with frequent integration and improvement cycles

## Agile for Teams: Scrum



## Agile for Teams: Kanban

Visualize work flow. Limit work in process. Improve flow.



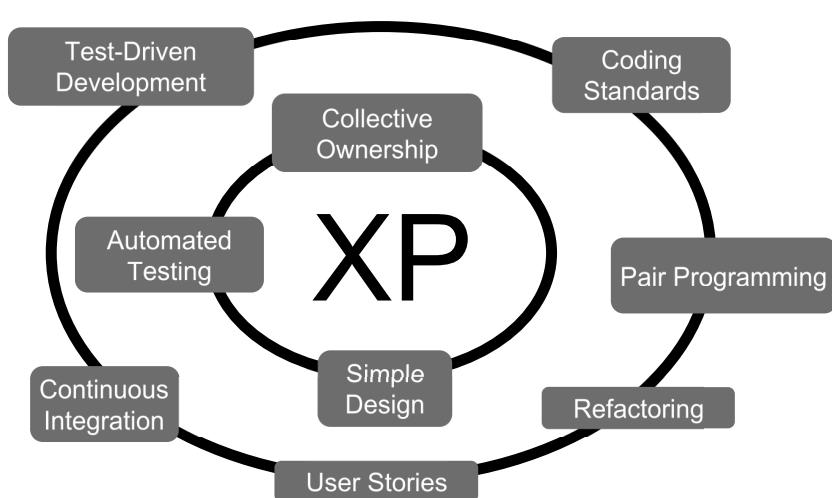
## Agile for Teams: XP

XP practices drive endemic code quality to unprecedented levels.

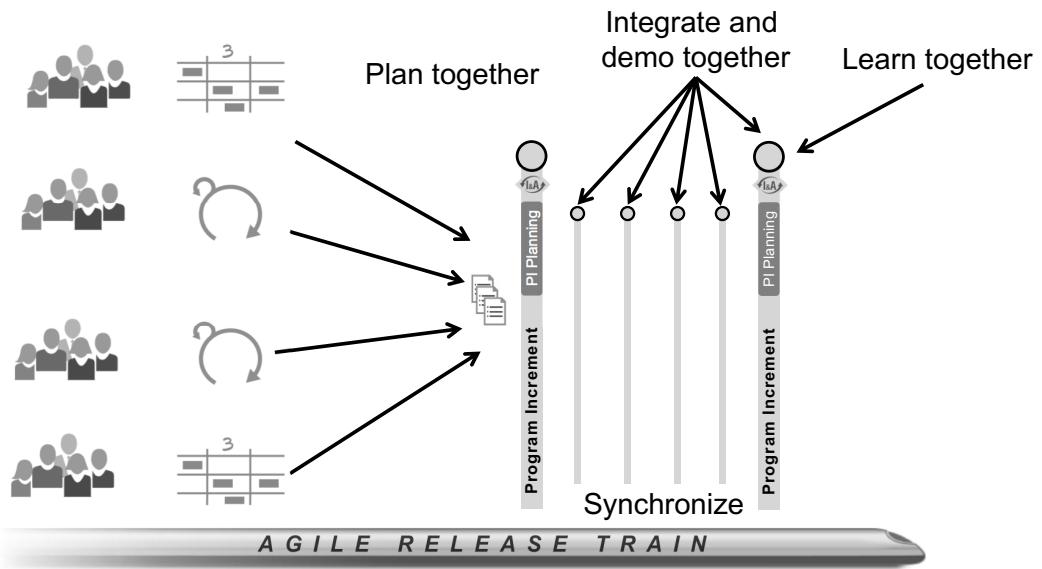
*Most high-performance teams use Scrum and XP together.*

*It is hard to get a Scrum with extreme velocity without XP engineering practices.*

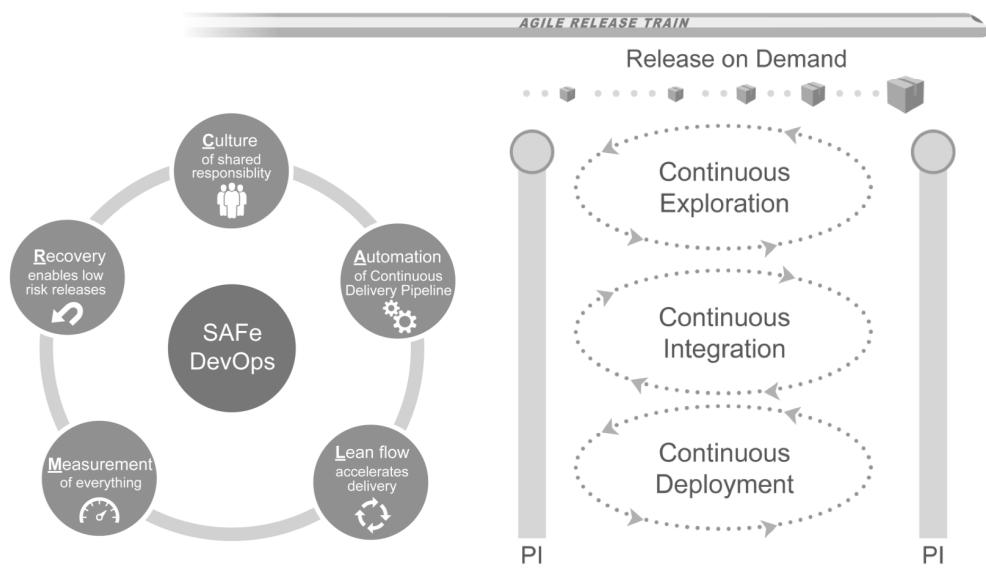
—Jeff Sutherland,  
co-creator of Scrum



## Teams in SAFe are part of an Agile Release Train



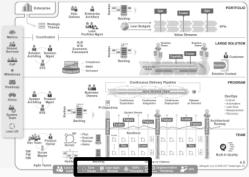
## The ART and Teams continuously deliver value



# Lesson summary

In this lesson, you:

- ▶ Connected with SAFe
- ▶ Explored Lean, the Agile Manifesto, and SAFe Principles
- ▶ Identified Scrum, Kanban, and XP practices



*Suggested Scaled Agile Framework reading:  
“Core Values,” “Lean-Agile Mindset, and  
“SAFe Principles” articles*