

# SAFe® Scrum Master

Applying the Scrum Master role within a SAFe enterprise



**SAFe® Course:** Attending this course gives learners access to the SAFe® Scrum Master exam and related preparation materials.

**SCALED AGILE®**

© Scaled Agile, Inc.

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

- ▶ Describe Scrum in a SAFe Enterprise
- ▶ Facilitate Scrum events
- ▶ Facilitate effective Iteration execution
- ▶ Support effective Program Increment execution
- ▶ Support relentless improvement
- ▶ Coach Agile Teams for maximum business results
- ▶ Support DevOps implementation



## SAFe Scrum Master topics

1. Introducing Scrum in SAFe
2. Characterizing the role of the Scrum Master
3. Experiencing PI Planning
4. Facilitating Iteration execution
5. Finishing the PI
6. Coaching the Agile Team

## Logistics

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



↓ + ↓

Elephant	North
Penguin	South
Giraffe	East
Dolphin	West
Zebra	(Back to top)
Wren	

## Exercise: Spell it out SAFe Scrum Master with certification

- ▶ Pair with a learner that has the same compass direction: North, South, East, or West
- ▶ In your pair, introduce yourself and your role
- ▶ Choose one letter from the course title. Use it to explain what you hope to learn, and have fun!
  - Example: “I selected ‘C’ for ‘communication’ because I want to know how and when to communicate better.”



# Lesson 1

## Introducing Scrum in SAFe

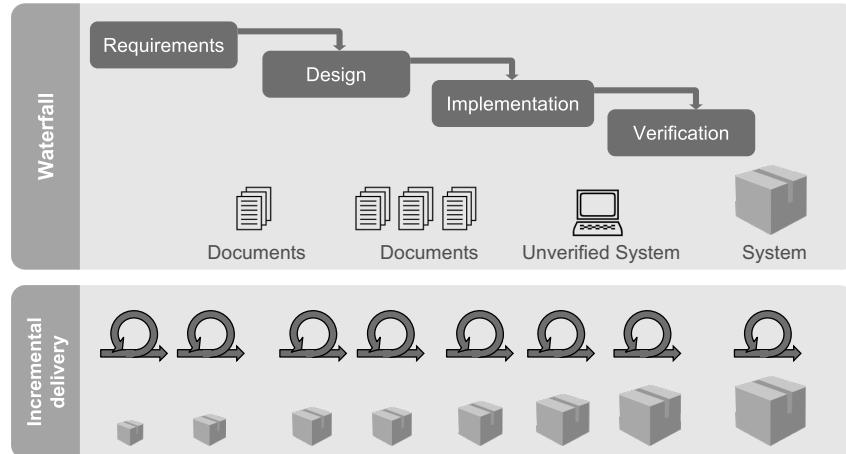
1. Introducing Scrum in SAFe
2. Characterizing the role of the Scrum Master
3. Experiencing PI Planning
4. Facilitating Iteration Execution
5. Finishing the PI
6. Coaching the Agile Team

## Learning objectives

- 1.1 Examine basic Agile development concepts
- 1.2 Explore Scrum basics
- 1.3 Position an Agile Team in a SAFe Enterprise

## 1.1 Examine basic Agile development concepts

## Agile and waterfall development



## Exercise: Fast focus!

Using a blank back side of a page in your workbook to write on (such as one of the Big Picture pages in the front):

1. When the instructor says Go, enter the numbers 1-26 as many times as you can – until the instructor says STOP.
2. When the instructor says Go, enter the letters A-Z as many times as you can – until the instructor says STOP.
3. When the instructor says Go, enter the number/letter pairs (i.e. 1A, 2B, 3C), as many times as you can – until the instructor says STOP.



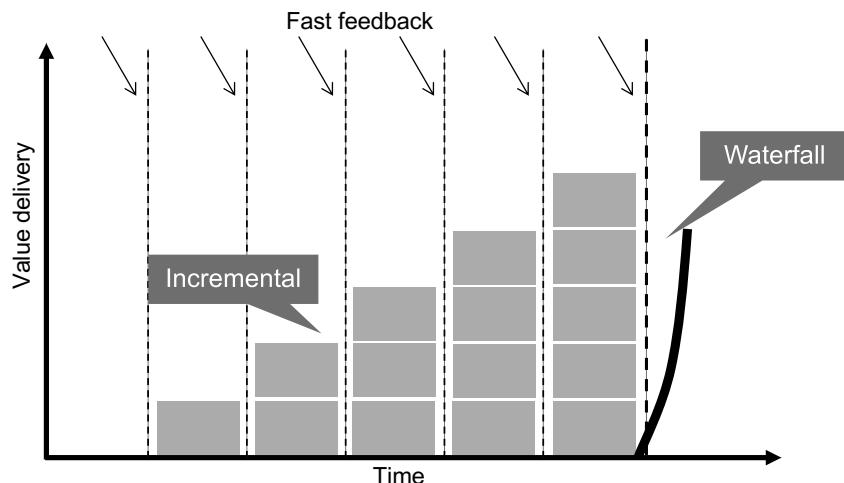
## Exercise: Fast focus! Debrief

Discuss at your tables:

- ▶ How many numbers did you write down? How many letters?  
How many number/letter pairs?
- ▶ How many active projects are you currently juggling? (Studies have shown that for each additional activity you take on, you lose as much as 20% of your total effectiveness.)
- ▶ How much of your day is actually spent adding value versus running from meeting to meeting?
- ▶ Context switching is VERY disruptive to our productivity.  
The more focused we are, the more productive we are.



## Deliver value incrementally



## Exercise: Manifesto for Agile Software Development

Fill in the following value statements, using the phrases in the box to the right:

\_\_\_\_\_ over processes and tools

Working software over \_\_\_\_\_

Customer collaboration over \_\_\_\_\_

\_\_\_\_\_ over following a plan

### PHRASES:

- contract negotiations
- comprehensive documentation
- individuals and interactions
- responding to change



## 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.

## Agile Manifesto principles

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)

1.15

SCALED AGILE® © Scaled Agile, Inc.

## Agile Manifesto principles (Cont.)

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)

1.16

SCALED AGILE® © Scaled Agile, Inc.

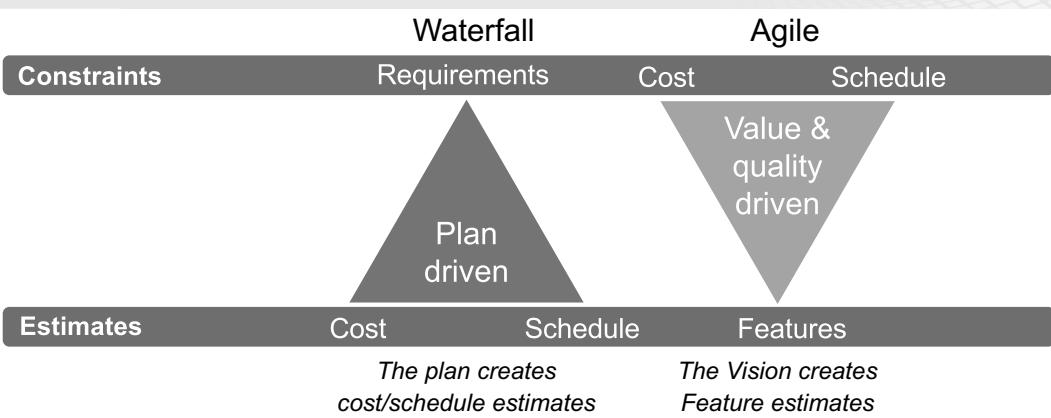
## Exercise: We've got values

Discuss as a team:

- ▶ Do the Agile values and principles align with your company culture?
  - ▶ Any contradictions?
  - ▶ Which value or principle stands out to you?
  - ▶ Discuss the biggest areas where Agile challenges traditional development



## 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

## Agile frameworks

### Agile Development

A general term defined by **values** and **principles**

#### Agile frameworks

SAFe  
Scrum

Crystal  
Kanban

eXtreme Programming (XP)  
Feature-Driven Development

Timeboxing  
User Stories  
Daily Stand-Ups

**Practices**  
Frequent Demos  
Acceptance Test-Driven Development

Information Radiators  
Retrospectives  
Continuous Integration

## Agile at scale gets business results



See <http://www.ScaledAgileFramework.com/case-studies>

## 1.2 Explore Scrum basics

### The roots of Scrum

*The ‘relay race’ approach to product development ... may conflict with the goals of maximum speed and flexibility.*

*Instead, a holistic or ‘rugby’ approach—where a team tries to go the distance as a unit, passing the ball back and forth—may better serve today’s competitive requirements.*

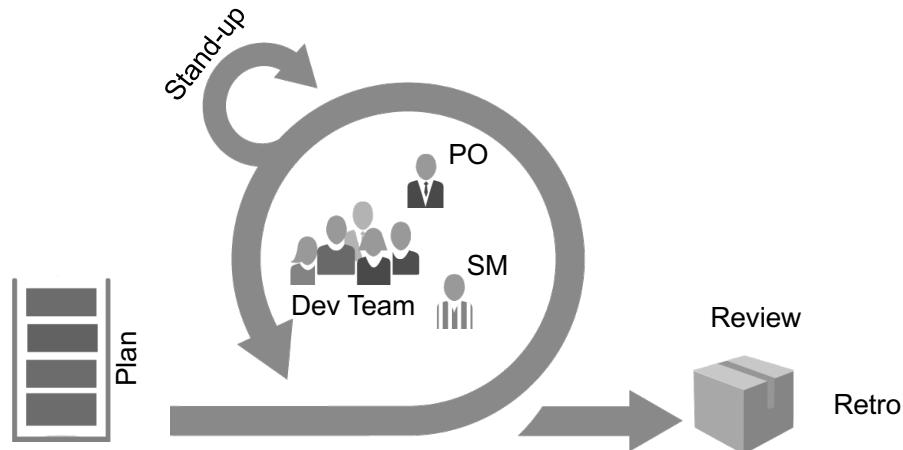
—Hirotaka Takeuchi and Ikujiro Nonaka,  
“The New New Product Development Game,”  
*Harvard Business Review*, January 1986



All Blacks vs Wallabies  
<https://youtu.be/KWnwI0-aeq0>  
1:31

## Scrum in one slide

Scrum is built on 3 pillars: transparency, inspection, and adaptation.



SCALED AGILE® © Scaled Agile, Inc.

1.23

## Differences from Scrum Guide

Scrum Guide	SAFe
Sprint Planning	Iteration Planning
Sprint Review	Iteration Review
Sprint Retrospective	Iteration Retrospective
Sprint Goals	Iteration Goals
Sprint Backlog	Iteration Backlog
Daily Scrum	Daily Stand-Up (DSU)
Increment	Team Increment
The Scrum Team	Agile Team

SCALED AGILE® © Scaled Agile, Inc.

1.24

## Scrum values

Make these things transparent: the process, the workflow, progress

**Courage**



**Focus**



**Openness**



**Commitment**



**Respect**



## Exercise: Scrum values create transparency

**Courage**



**Focus**



**Openness**



**Commitment**



**Respect**

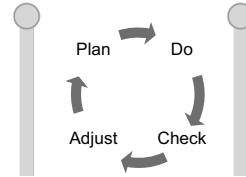


PREPARE  
5 min

SHARE  
2 min

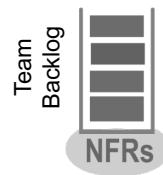
## Iteration (sprint) basics

- ▶ Iteration length is one to four weeks
  - SAFe advises two weeks (small batch size)
  - Each Iteration is the same length, running back to back
- ▶ Goal is to deliver working software/hardware at the end of each Iteration
- ▶ Avoid adding scope once the Iteration has begun
- ▶ **Important: Team composition does not change during an Iteration, otherwise the velocity is invalid**



## The Team Backlog organizes the team's work

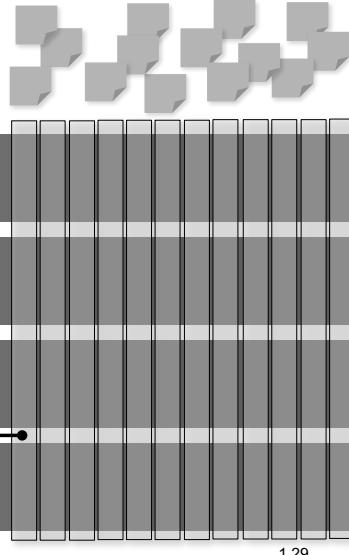
- ▶ It is truly **all** things. If a thing is in there, it might get done. If it isn't there, there is no chance that it will be done.
- ▶ It represents opportunities, not commitments—a list of what we want to do
- ▶ Stories and Enablers may be estimated (preferable), but estimates do not imply committed delivery
- ▶ It has a single owner: the team's Product Owner
- ▶ Created by the Agile Team, prioritized by the Product Owner
- ▶ Stories and Enablers for the next Iteration are more detailed than for later Iterations.



## Team Increment

A Team Increment is a thin vertical slice of functionality.

- ▶ Backlog items must be defined as vertical slices of functionality; otherwise, you'll just have ingredients
- ▶ Each slice can be demonstrated and consumed
- ▶ Each slice represents a single piece of end-to-end functionality



## Scrum events

Event	Timebox	Value
Backlog Refinement	~1 hr	Prepare requirements for Iteration Planning
Iteration Planning	2 – 4 hrs	Team commits to a set of goals to be delivered in the Iteration
Daily Stand-Up	<= 15 mins	Team members sync regarding the progress of the Iteration Goals
Iteration Review	~1 hr	Deliverables reviewed with stakeholders providing feedback
Iteration Retrospective	1 – 1.5 hrs	Team reviews and improves its process before the next Iteration

## Exercise: Scrum events increase transparency

- ▶ Return to your previous groups
- ▶ For each Scrum Event, the team should answer the following questions:
  - Who facilitates it?
  - Why do we have it?
  - How often does it occur?

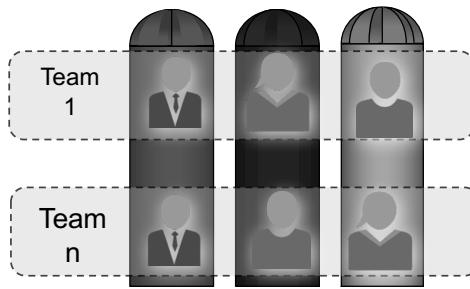
Backlog refinement	Who: Why: How:
Iteration Planning	Who: Why: How:
Daily Stand-Up	Who: Why: How:
Iteration Review	Who: Why: How:
Retrospective	Who: Why: How:

PREPARE      SHARE

4 min      3 min

## Build cross-functional Agile Teams

- ▶ Cross-functional, self-organizing entities that can define, build, and test a Feature or component
- ▶ Optimized for communication and delivery of value
- ▶ Deliver value every two weeks
- ▶ Three roles: Scrum Master, Product Owner, and Development Team



## Scrum Master

- ▶ Coaches team improvement using values, principles, and best practices
- ▶ Facilitates Scrum team events
- ▶ Protects the development team
- ▶ Helps to remove impediments
- ▶ Is a servant leader



## Product Owner

- ▶ The single voice of the customer and stakeholders in the team
- ▶ Owns and manages the Team Backlog
- ▶ Defines and accepts requirements
- ▶ Makes the hard calls on scope and content



## The Development Team

- ▶ Typically 3 to 9 people (excluding Scrum Master and Product Owner)
- ▶ Everyone who is needed to define, build, and test
- ▶ Team members are **only** on this team
- ▶ Self-organizing and accountable
- ▶ Collaborative
- ▶ Cross-functional
- ▶ Empowered



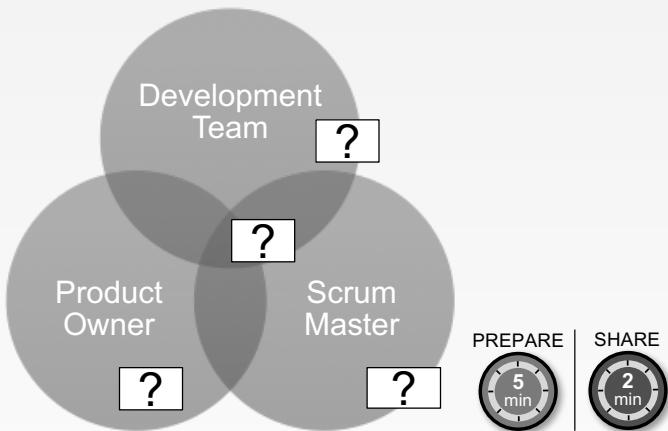
## Development team responsibilities

- ▶ Listen and talk to people
- ▶ Seek and accept help
- ▶ Embrace change
- ▶ Work on items in an order set by the Product Owner
- ▶ Be proactive and self-motivated
- ▶ Be honest (with yourself and others)
- ▶ Be passionate about what you do
- ▶ Embrace ‘all sink or all swim’
- ▶ Everyone involved has skin in the game (Devs, QA, APM, PO, Business, Mgmt, etc.—everyone)



## Exercise: Match it up

- ▶ Create the Venn diagram on a flip chart or white board
- ▶ Make sticky notes for the responsibilities in the previous 4 slides
- ▶ Place them either in a role or at an intersection
- ▶ Prepare to discuss your decisions



## 1.3 Position an Agile Team in a SAFe Enterprise

*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.

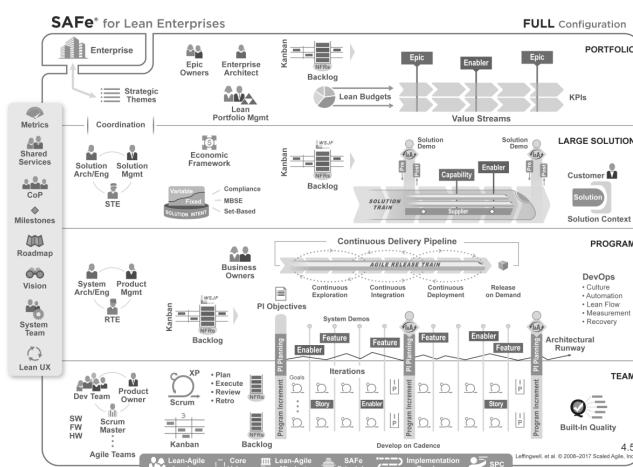
 scaledagileframework.com

SCALED AGILE® © Scaled Agile, Inc.

1.39

## 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

SCALED AGILE® © Scaled Agile, Inc.

1.40

# 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

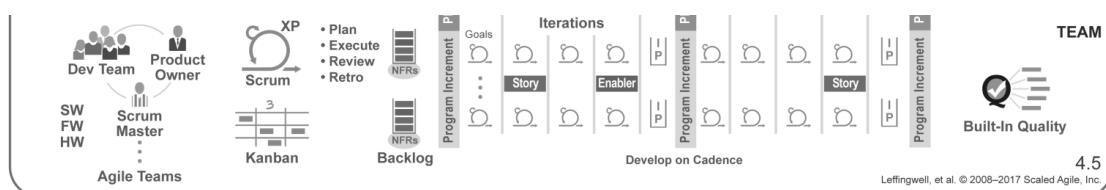
#7-Apply cadence, synchronize with cross-domain planning

#8-Unlock the intrinsic motivation of knowledge workers

#9-Decentralize decision-making

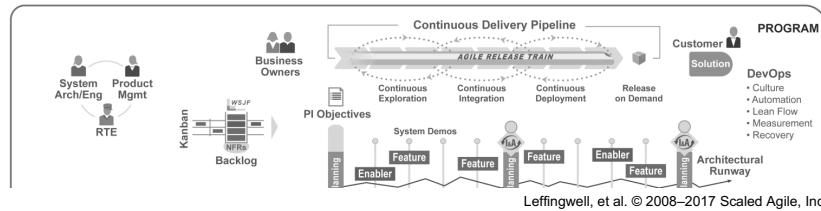
## Nothing beats an Agile Team

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



## 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
- ▶ Value delivery via Features and benefits

## The Agile Release Train

- ▶ A virtual organization of 5 – 12 teams (50 – 125+ individuals) that plans, commits, and executes together
- ▶ Program Increment (PI) is a fixed timebox; default is 10 weeks
- ▶ Synchronized Iterations and PIs
- ▶ Aligned to a common mission via a single Program Backlog
- ▶ Operates under architectural and UX guidance
- ▶ Frequently produces valuable and evaluable System Level Solutions



## ART roles



Release Train Engineer acts as the Chief Scrum Master for the train.



Product Management owns, defines, and prioritizes the Program Backlog.



System Architect/Engineering provides architectural guidance and technical enablement to the teams on the train.



The System Team provides processes and tools to integrate and evaluate assets early and often.



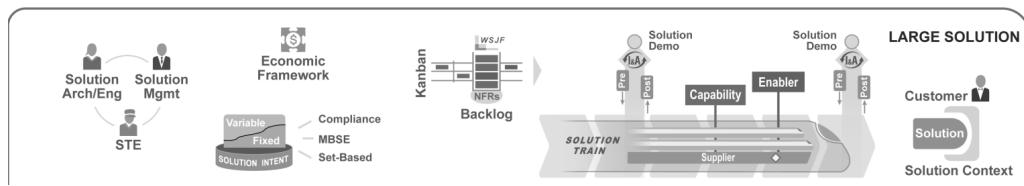
Business Owners are the key stakeholders on the Agile Release Train.

## Program events

Event	Time box	Value
PI Planning	2 days	Teams commit to a set of objectives to be delivered in the PI
ART Sync	1 hour	Train teams to sync regarding the progress of the PI
System Demo	2 hours	Deliverables reviewed with stakeholders providing feedback
Inspect and Adapt event	½ day	The train reviews and improves its process before the next PI

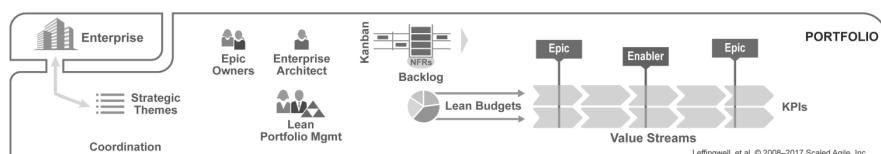
## Large solutions require coordination

- ▶ Coordinates development of large Solutions
- ▶ Synchronizes multiple ARTs and suppliers
- ▶ Manages Solution Intent
- ▶ Integrates suppliers as partners
- ▶ Value delivery via Capabilities



Leffingwell, et al. © 2008–2017 Scaled Agile, Inc.

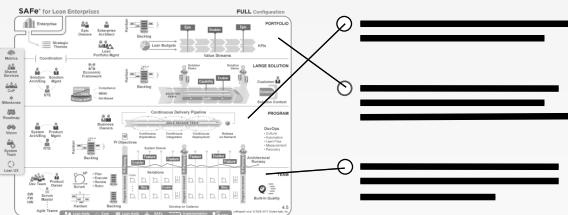
## Value Streams are part of Portfolios



- ▶ Organized around the flow of value
- ▶ Lean Budgets empowers decision makers
- ▶ Kanban system provides Portfolio visibility and WIP limits
- ▶ Enterprise architecture guides larger technology decisions
- ▶ KPIs support governance and improvement
- ▶ Value delivery via Epics

## Exercise: Think fast, you're right!

Draw lines to show in which level of the Framework the activity described belongs. The same activity can tie to more than one level.



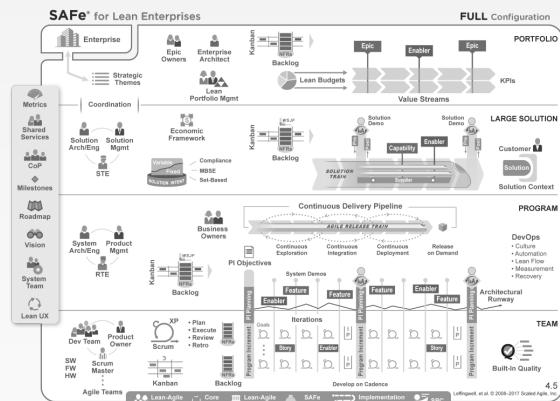
**PREPARE** **SHARE**

SCALED AGILE® © Scaled Agile, Inc.

1.49

## Exercise: Think fast, you're right! Round 1

- Synchronizing multiple ARTs and suppliers
- Face-to-face planning for collaboration, alignment, and adaptation
- Visualizing and limiting WIP of Portfolio Epics
- Iteration retrospective



SCALED AGILE® © Scaled Agile, Inc.

1.50

## Exercise: Think fast, you're right! Round 2

The diagram illustrates the SAFe for Lean Enterprises framework, organized into four main levels:

- Enterprise:** Strategic Themes, Epic Owners, Enterprise Architect, Lean Mgmt, Portfolio Mgmt.
- FULL Configuration:** Portfolio, Value Streams, Epics, Enabler, Epics, KPIs.
- LARGE SOLUTION:** Solution ArchEng, Solution Mgmt, Solution Context, Customer Solution.
- PROGRAM:** Continuous Delivery Pipeline, RELEASE TRAIN, PI Objectives, Continuous Exploration, Continuous Integration, Continuous Deployment, Release on Demand.
- TEAM:** DevOps Culture, Architecture, User Flow, Metrics, Recovery, Built-in Quality.
- SAFÉ for Lean Enterprises:** Shared Services, CoP, Milestones, Roadmap, Vision, System Team, Lean UX, Dev Team, Product Owner, RTE, Business Owners, PI Objectives, Iterations, Story, Feature, Enable, System Demos, Feature, Feature, Feature, Enable, Features, Architectural Runway, Built-in Quality.

Legend at the bottom includes: SAFe for Lean Enterprises, Scaled Agile, Inc., Leffingwell et al. © 2008-2017, License: MIT, 4.5.

- Delivering valuable, tested, working increment every two weeks
- Releasing value to the customer
- Managing Solution Intent
- Guiding technology decisions which impact multiple value streams

1.51

## Lesson summary

In this lesson, you:

- ▶ Reviewed Agile development and Agile frameworks
- ▶ Explored the basics of Scrum and its roles, responsibilities, and events
- ▶ Explored how Agile Teams operate as part of a SAFe Enterprise

Suggested Scaled Agile Framework reading:  
 "Lean-Agile Mindset," "Scrum," "Scrum Master,"  
 "Product Owner," and "Agile Team" articles