

SAFe® Advanced Scrum Master

Advancing Scrum Master Servant
Leadership with SAFe

SAFe® Course – Attending this course gives students access to the SAFe Advanced Scrum Master exam and related preparation materials.

5.1



Welcome to the course!

Make the Most of Your Learning



Access the SAFe Community Platform

Manage your member profile, access videos and training resources, join Communities of Practice, and more.



Prepare Yourself

Access your learning plan featuring your digital workbook, study materials, and certification practice test



Become a Certified SAFe Professional

Get certified to validate your knowledge, expand your professional capabilities, and open the door to new career opportunities.



Access SAFe Content and Tools

Access professional development resources and your trainer enablement plan to teach SAFe instructor-led courses.



Collaborate with Your Team

Choose from hundreds of collaboration templates to easily set up events like PI Planning and work in real time with your team and others—all with SAFe Collaborate.



Showcase SAFe Credentials

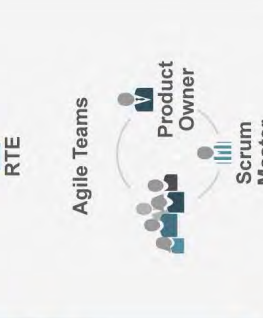
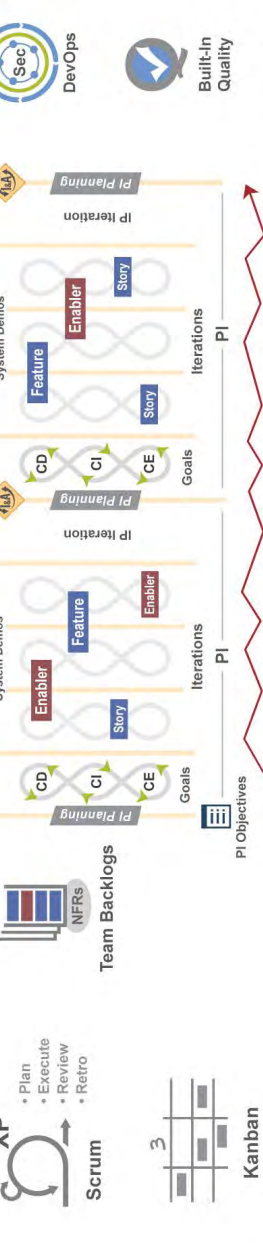
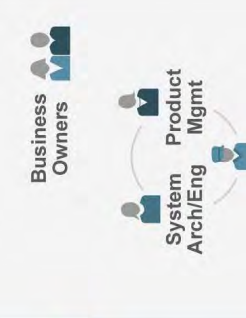
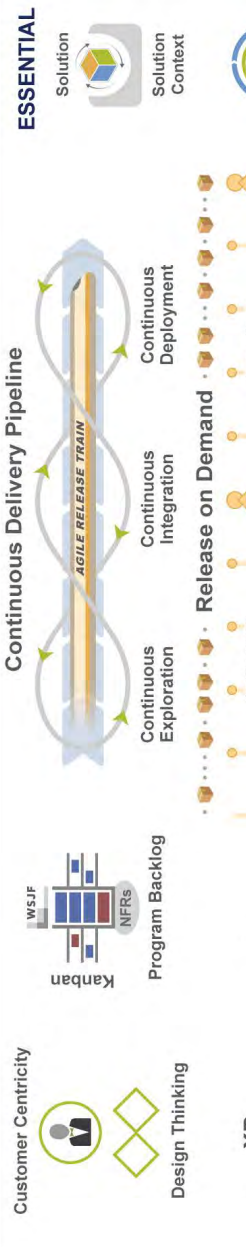
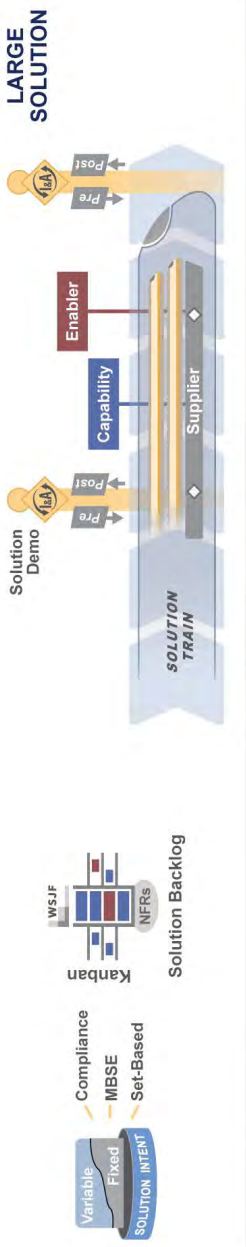
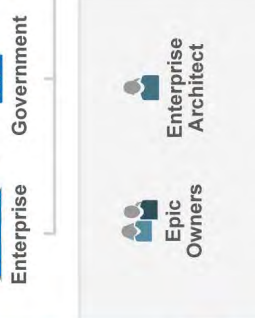
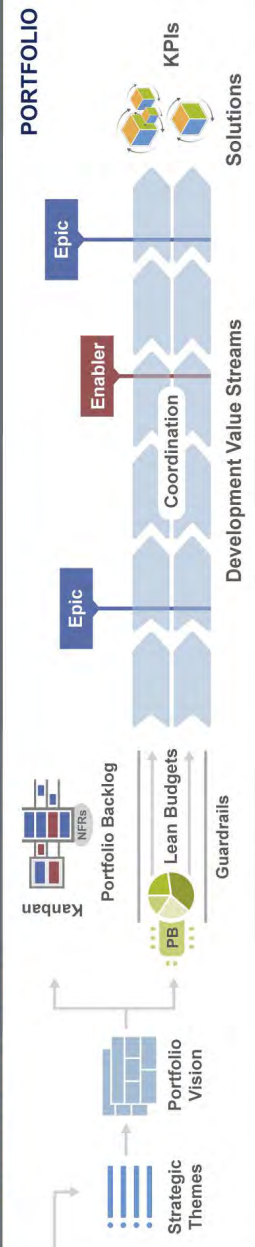
Display your digital badge to promote your SAFe capabilities and proficiencies throughout your career.

community.scaledagile.com

Business Agility

Operational Value Streams

Measure & Grow



Organizational Agility

Lean Portfolio Management

Enterprise Solution Delivery

Agile Product Delivery

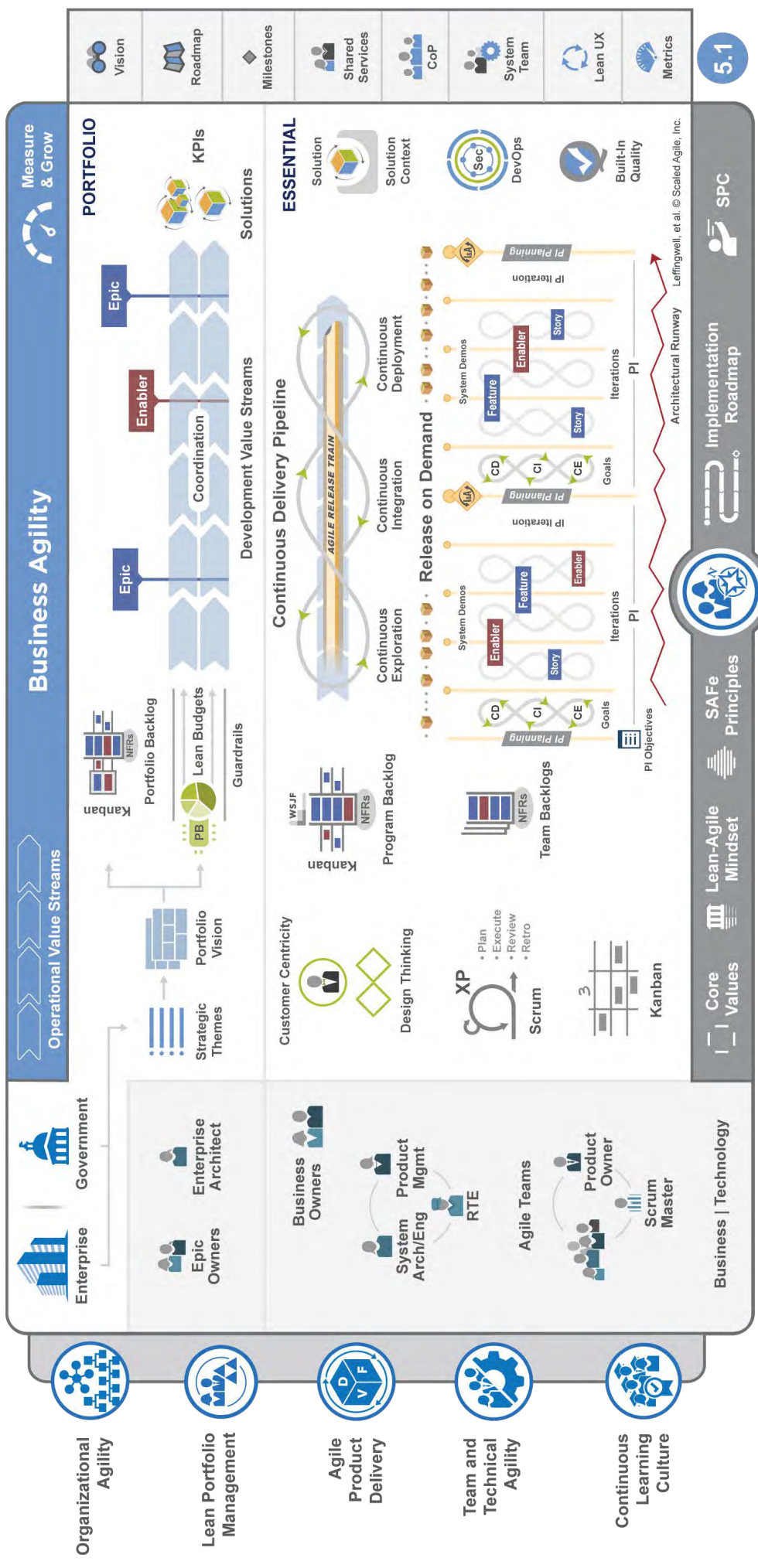
Team and Technical Agility

Continuous Learning Culture

- Vision
- Roadmap
- Milestones
- Shared Services
- CoP
- System Team
- Lean UX
- Metrics

5.1

Lean-Agile Leadership



Organizational Agility

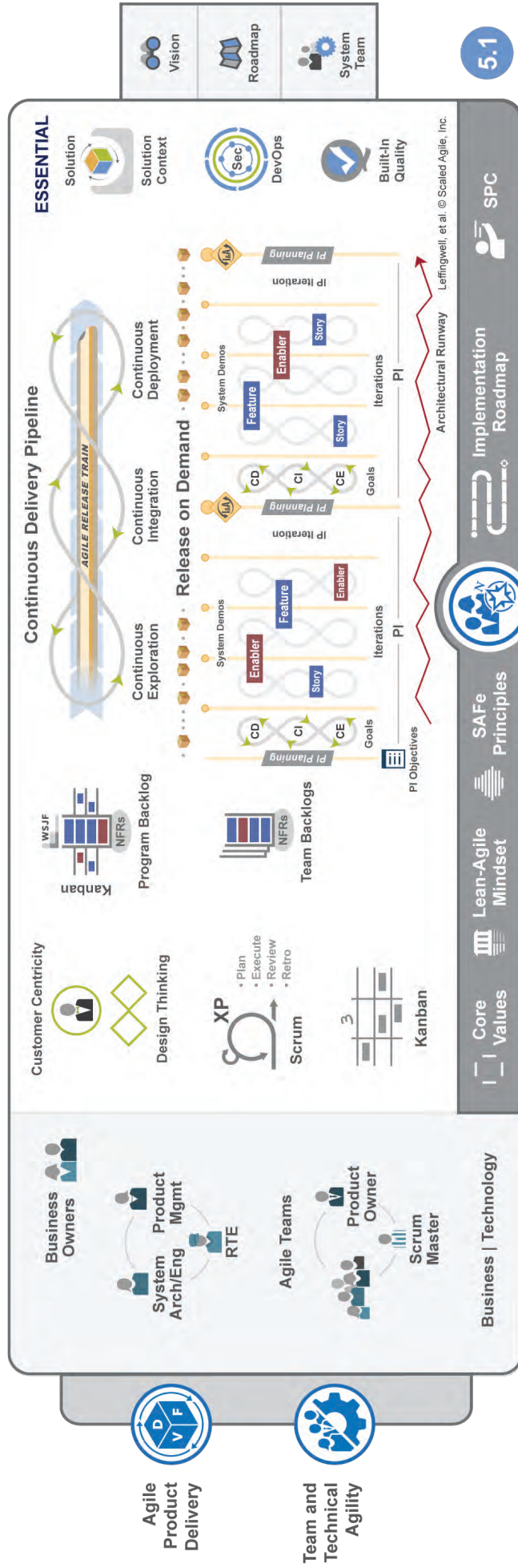
Lean Portfolio Management

Agile Product Delivery

Team and Technical Agility

Continuous Learning Culture

Lean-Agile Leadership

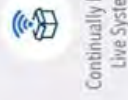


Lean-Agile Leadership



Enterprise Solution Delivery

- Apply Lean system engineering to build really big systems
- Coordinate and align the full supply chain
- Continually evolve live systems



Lean Portfolio Management

- Align strategy, funding, and execution
- Optimize operations across the portfolio
- Lightweight governance empowers decentralized decision-making

Strategy & Investment Funding

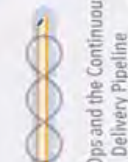
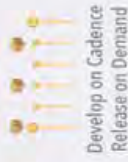


Lean Governance

Agile Portfolio Operations

Agile Product Delivery

- The customer is the center of your product strategy
- Develop on cadence and release on demand
- Continuously explore, integrate, deploy, and innovate



Team And Technical Agility

- High-performing, cross-functional, Agile teams
- Business and technical teams build business solutions
- Quality business solutions delight customers



Agile Teams

Teams of Agile Teams

Built-in Quality

Lean-Agile Leadership

- Inspire others by modeling desired behaviors
- Align mindset, words, and actions to Lean-Agile values and principles
- Actively lead the change and guide others to the new way of working



Leading by Example

Mindset & Principles

Leading Change

Continuous Learning Culture

- Everyone in the organization learns and grows together
- Exploration and creativity are part of the organization's DNA
- Continuously improving solutions, services, and processes is everyone's responsibility



Learning Organization

Innovation Culture

Relentless Improvement



Lean-thinking People and Agile Teams

Lean Business Operations

Strategy Agility

SAFe® Courses and Certifications














Course	Description	Certification
Leading SAFe®	Thriving in the Digital Age with Business Agility	 with SAFe® 5 Agilist Certification
Implementing SAFe®	Achieving Business Agility with the Scaled Agile Framework	 with SAFe® 5 Program Consultant Certification
SAFe® for Government	Applying Lean-Agile Practices in the Public Sector with SAFe®	 with SAFe® 5 Government Practitioner Certification
Lean Portfolio Management	Aligning Strategy with Execution	 with SAFe® 5 Lean Portfolio Manager Certification
SAFe® Product Owner/Product Manager	Delivering Value through Effective Program Increment Execution	 with SAFe® 5 Product Owner/Product Manager Certification
Agile Product Management	Using Design Thinking to Create Valuable Products in the Lean Enterprise	 with SAFe® 5 Agile Product Manager Certification
SAFe® Scrum Master	Applying the Scrum Master Role within a SAFe® Enterprise	 with SAFe® 5 Scrum Master Certification
SAFe® Advanced Scrum Master	Advancing Scrum Master Servant Leadership with SAFe®	 with SAFe® 5 Advanced Scrum Master Certification
SAFe® Release Train Engineer	Facilitating Lean-Agile Program Execution	 with SAFe® 5 Release Train Engineer Certification
SAFe® for Architects	Architecting for Continuous Value Flow with SAFe®	 with SAFe® 5 Architect Certification
SAFe® DevOps	Optimizing Your Value Stream	 with SAFe® 5 DevOps Practitioner Certification
SAFe® for Teams	Establishing Team Agility for Agile Release Trains	 with SAFe® 5 Practitioner Certification
Agile Software Engineering	Enabling Technical Agility for the Lean Enterprise	 with SAFe® 5 Agile Software Engineer Certification

Table of Contents

Privacy Notice.....	Page 11
Course Introduction.....	Page 12
Lesson 1: Exploring the Scrum Master Role in SAFe.....	Page 14
Lesson 2: Applying the SAFe Lean-Agile Principles.....	Page 27
Lesson 3: Exploring Agile and Scrum Anti-Patterns.....	Page 60
Lesson 4: Facilitating Program Execution.....	Page 72
Lesson 5: Improving Flow with Kanban and XP.....	Page 113
Lesson 6: Building High-Performing Teams.....	Page 139
Lesson 7: Improving Program Performance.....	Page 153
Lesson 8: Practicing SAFe.....	Page 167
SAFe Glossary.....	Page 175

Privacy Notice

Your name, company and email address will be shared with Scaled Agile, Inc. for course fulfillment, including testing and certification. Your information will be used in accordance with the Scaled Agile privacy policy available at <https://www.scaledagile.com/privacy-policy/>.

SAFe® Advanced Scrum Master

Advancing Scrum Master Servant
Leadership with SAFe

SAFe® Course - Attending this course gives students access to the SAFe Advanced Scrum Master exam and related preparation materials.

5.1



SAFe® PROVIDED BY **SCALED AGILE®**
© Scaled Agile, Inc.

Logistics

- ▶ Course meeting times
- ▶ Breaks
- ▶ Facilities
- ▶ Technology requirements
- ▶ Working agreements



Discussion: Introductions

Duration



- ▶ **Step 1:** Meeting in pairs, introduce yourself and share:
 - Your strengths as a Scrum Master and your favorite event to facilitate
 - What you would like to learn and practice in this course to improve in your role as a Scrum Master
- ▶ **Step 2:** Introduce each other to the class. Note similar strengths and challenges.



Course outline

- ▶ Lesson 1: Exploring the Scrum Master Role in SAFe
- ▶ Lesson 2: Applying SAFe Principles
- ▶ Lesson 3: Exploring Agile and Scrum Anti-Patterns
- ▶ Lesson 4: Facilitating Program Execution
- ▶ Lesson 5: Improving Flow with Kanban and XP
- ▶ Lesson 6: Building High-Performing Teams
- ▶ Lesson 7: Improving Program Performance
- ▶ Lesson 8: Practicing SAFe

Lesson 1

Exploring the Scrum Master Role in SAFe

SAFe® Course - Attending this course gives students access to the SAFe Advanced Scrum Master exam and related preparation materials.



Lesson Topics

- 1.1** Explore Scrum Master challenges in the Enterprise
- 1.2** Explain the purpose and the basic constructs of SAFe
- 1.3** Establish Scrum Master connections in SAFe



1-6

Learning Objectives

At the end of this lesson, you should be able to:

- ▶ Define the challenges of the Scrum Master in the Enterprises
- ▶ Analyze the basic constructs of SAFe
- ▶ Create connections between the role of the Scrum Master and the SAFe competencies

1.1 Explore Scrum Master challenges in the Enterprise



Discussion: Solving issues with facilitation



- ▶ **Step 1:** Working in your groups, brainstorm team challenges that require facilitation and that originate from **within** and **outside** the team.
- ▶ **Step 2:** Add the challenge to the Scrum Master challenges board.
- ▶ **Step 3:** Share a few of the challenges with the class.
- ▶ **Step 4:** We will address the challenges and how to address them throughout the course.



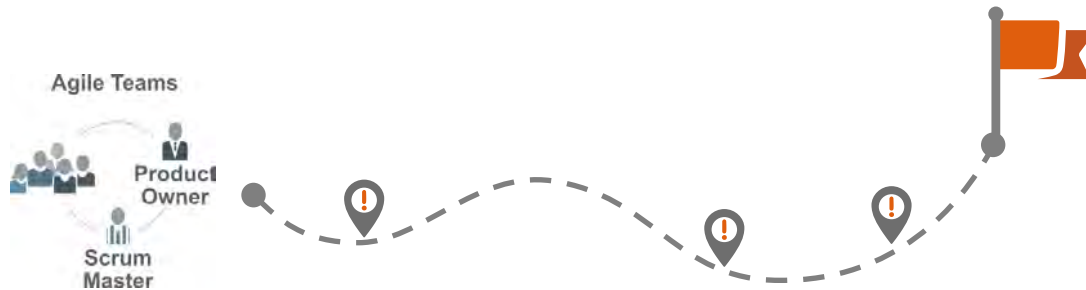
Iteration Event Facilitator's Guides

- ▶ Available on the SAFe Community Platform
- ▶ Provides event purpose, typical structure, and potential issues that can arise
- ▶ Tips for remote facilitation
- ▶ Offers various checklists to streamline preparation and execution

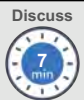


Teams must relentlessly improve

Aside from the impediments to developing and delivering value, Agile Teams in the Enterprise may encounter significant roadblocks to growth.



Discussion: Enable team growth



- ▶ **Step 1:** Discuss the following:
 - What are some impediments that you have experienced with your Agile Team?
 - How do these impediments prevent your team from growing, improving, learning, and becoming stronger?
 - What strategies and techniques have you used to remove impediments?

1.2 Explain the purpose and the basic constructs of SAFe

Why SAFe?

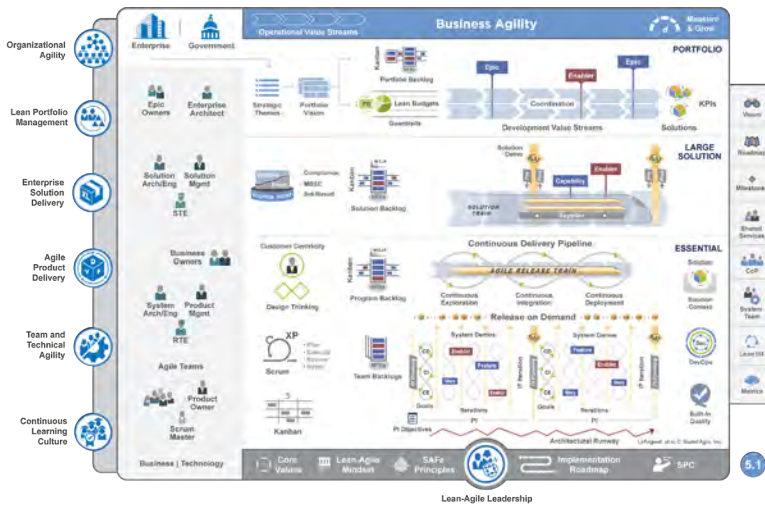
SAFe's business benefits are derived directly from case studies written by SAFe customers.



Typical results from scaledagile.com/customer-stories

The Scaled Agile Framework

SAFe synchronizes alignment, collaboration, and delivery for large numbers of teams.



SCALED AGILE © Scaled Agile, Inc.

Core Values



1-17

The Seven Core Competencies of Business Agility



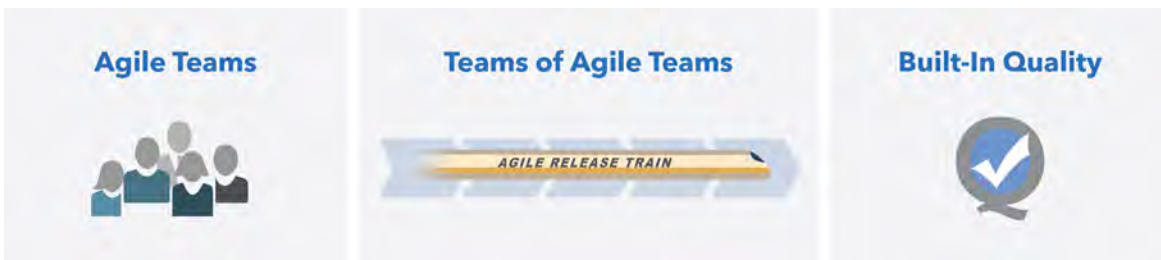
SCALED AGILE © Scaled Agile, Inc.

1-18



Team and Technical Agility

- ▶ High-performing, cross-functional Agile Teams
- ▶ Teams of business and technical teams build Solutions
- ▶ Quality business Solutions delight Customers



SCALED AGILE © Scaled Agile, Inc.

1-19



Agile Product Delivery

- ▶ The Customer is the center of your product strategy
- ▶ Decouple the release of value from the development cadence
- ▶ Continuously explore, integrate, deploy, and release



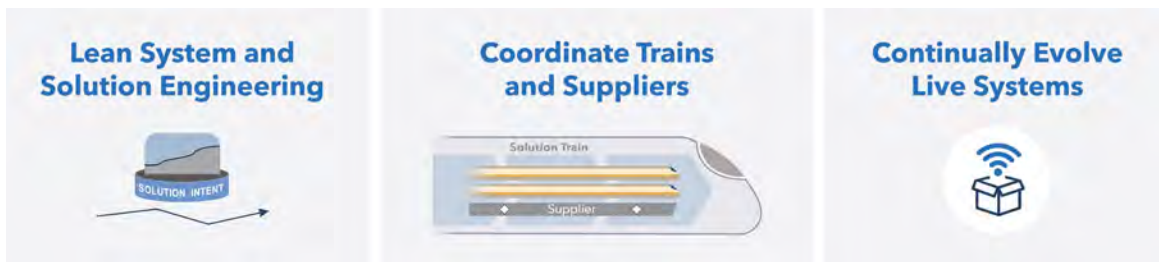
SCALED AGILE © Scaled Agile, Inc.

1-20



Enterprise Solution Delivery

- ▶ Apply Lean system engineering practices to build really big systems
- ▶ Coordinate and align the full supply chain
- ▶ Continue to enhance value after release



Lean Portfolio Management

- ▶ Align strategy, funding, and execution
- ▶ Optimize operations across the portfolio
- ▶ Lightweight governance empowers decentralized decision-making





Organizational Agility

- ▶ Create an enterprise-wide, Lean-Agile mindset
- ▶ Map and continuously improve business processes
- ▶ Respond quickly to opportunities and threats

Lean-thinking People and Agile Teams



Lean Business Operations



Strategy Agility



Continuous Learning Culture

- ▶ Everyone in the organization learns and grows together
- ▶ Exploration and creativity are part of the organization's DNA
- ▶ Continuously improving Solutions, services, and processes is everyone's responsibility

Learning Organization



Innovation Culture



Relentless Improvement





Lean-Agile Leadership

- ▶ Inspire others by modeling desired behaviors
- ▶ Align mindset, words, and actions to Lean-Agile values and principles
- ▶ Actively lead the change and guide others to the new way of working



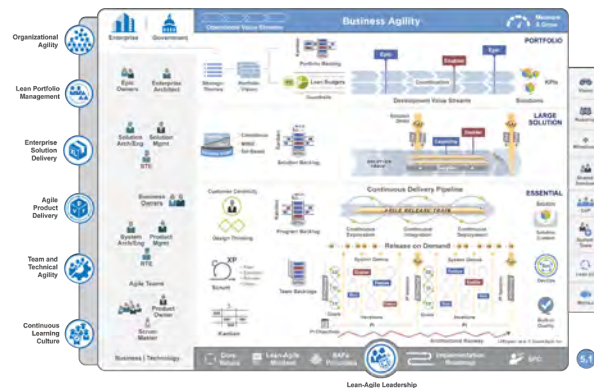
1.3 Establish Scrum Master connections in SAFe



Activity: The Seven Core Competencies and the Scrum Master role



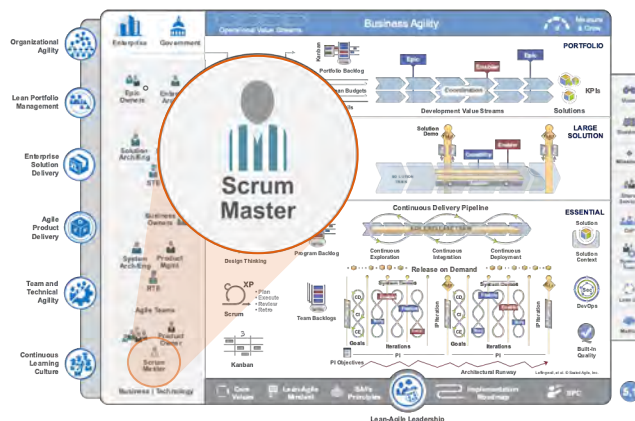
- ▶ **Step 1:** Working in your groups, use the SAFe Big Picture graphic and draw connections from the Scrum Master to other Framework elements, based on:
 - Communication
 - Collaboration
 - Problem-solving
 - Inputs/outputs
 - Other ideas you have
- ▶ **Step 2:** Be ready to present and discuss the identified connections



Lesson review

In this lesson you:

- ▶ Defined the challenges of the Scrum Master in the Enterprises
- ▶ Analyzed the purpose and basic constructs of SAFe
- ▶ Created connections between the role of the Scrum Master SAFe competencies



<https://www.scaledagileframework.com/scrum-master/>

Lesson notes



Enter your notes below:

A large, empty rectangular box intended for students to enter their lesson notes.

Reminder: If using a digital workbook, save your PDF often so you don't lose any of your notes.

Lesson 2

Applying SAFe® Principles

SAFe® Course - Attending this course gives students access to the SAFe Advanced Scrum Master exam and related preparation materials.



Lesson Topics

2.1 Apply the SAFe Principles in the role of a Scrum Master



Learning Objectives

At the end of this lesson, you should be able to:

- ▶ Apply the SAFe Principles in the role of a Scrum Master

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

#10 Organize around value

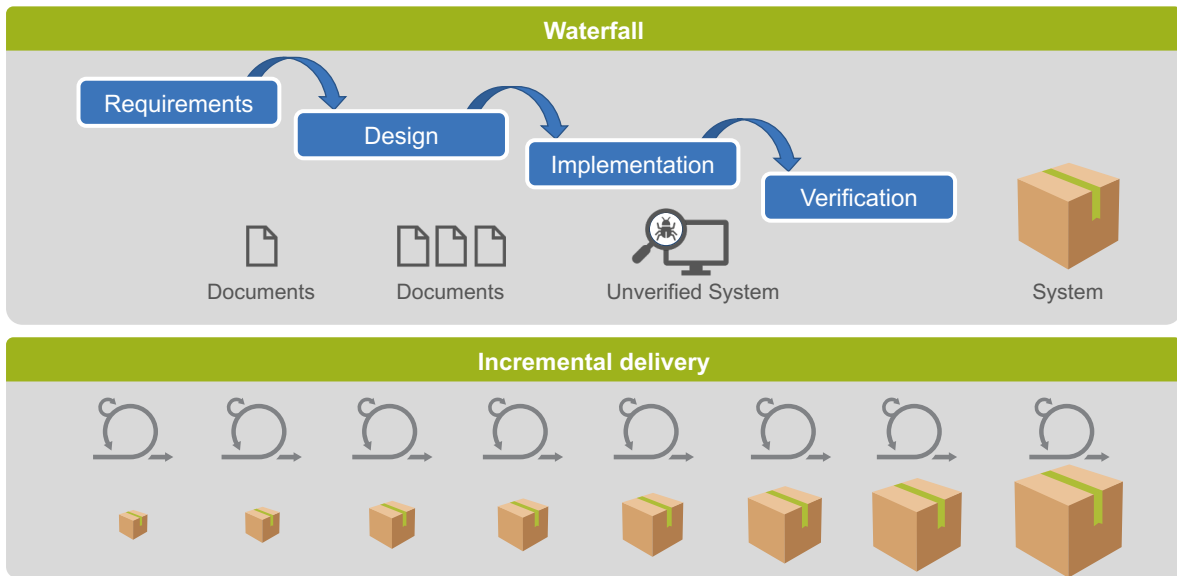
#1 Take an economic view

Apply a comprehensive economic framework

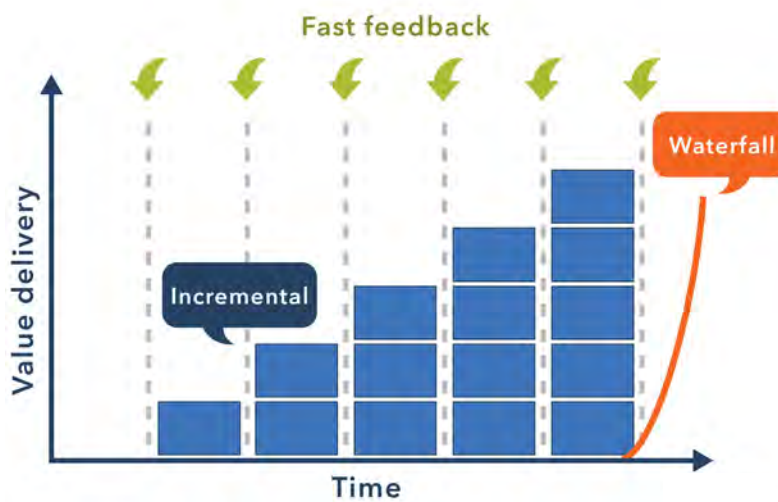
SAFe's economic framework contains four primary elements:

- ▶ Operating within Lean Budgets and Guardrails
- ▶ Understanding Solution economic trade-offs
- ▶ Leveraging Suppliers
- ▶ Sequencing jobs for the maximum benefit

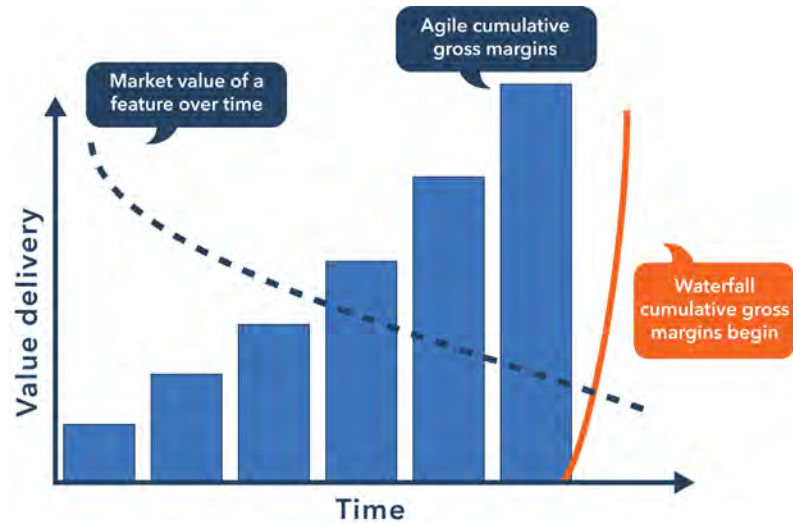
Agile economics: Deliver early and often



Deliver value incrementally



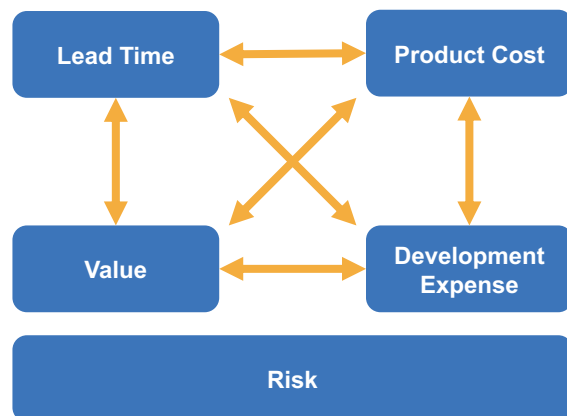
Early delivery has higher value



Solution economic trade-offs

Understanding trade-off parameters

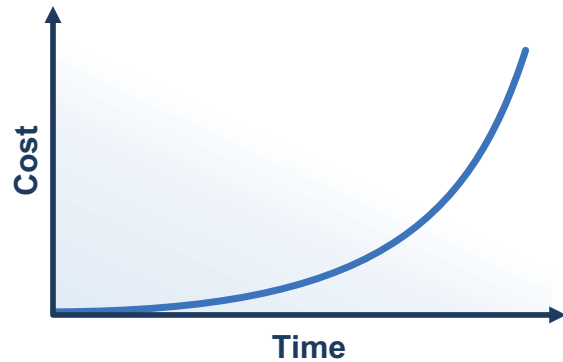
- ▶ Sequence jobs for maximum benefit
- ▶ Do not consider money already spent
- ▶ Make economic choices continuously
- ▶ Empower local decision making
- ▶ If you only quantify one thing, quantify the cost of delay



Examples of high-impact indicators

Here are examples of indicators that may have a surprisingly high economic impact in the long run.

- ▶ Cost of late defect fixing
- ▶ Cost of branching with late merge
- ▶ Cost of delayed performance testing
- ▶ Cost of large batch of cross-team dependencies
- ▶ Economic value of test automation
- ▶ Economic value of Enablers like research spikes, refactors, etc...



The cost of fixing a defect grows nearly exponentially over time.



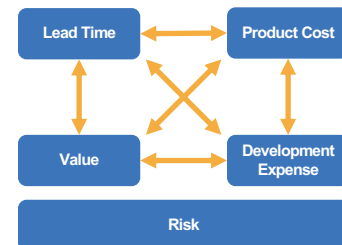
Discussion: Economic decisions



- ▶ **Step 1:** Looking at the economic trade-offs, discuss with your group the economic impact of the following challenges:

- Cost of late defect fixing
- Cost of branching with late merge
- Cost of delayed performance testing
- Cost of large batch of cross-team dependencies
- Economic value of test automation and Enablers, such as research spikes, refactors, etc.

- ▶ **Step 2:** Be prepared to share with the class

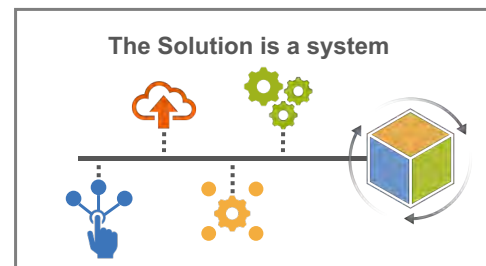


#2 Apply systems thinking

Attributes of systems thinking

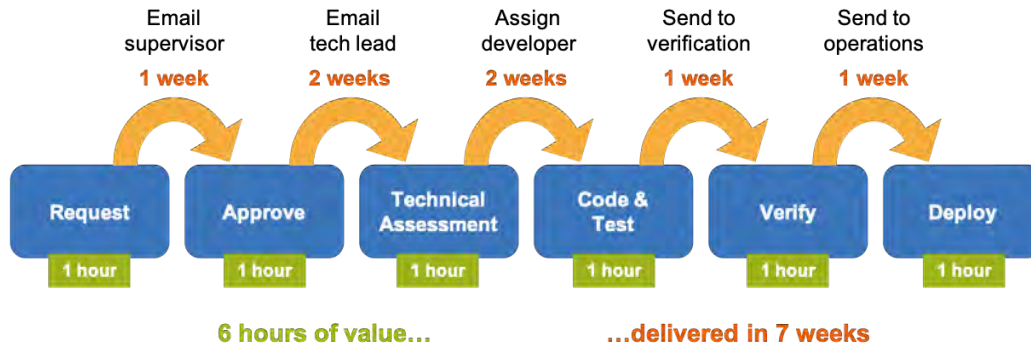
The Solution and the Enterprise are both affected by the following:

- ▶ Optimizing a component does not optimize the system
- ▶ For the system to behave well as a system, a higher-level understanding of behavior and architecture is required
- ▶ The value of a system passes through its interconnections
- ▶ A system can evolve no faster than its slowest integration point



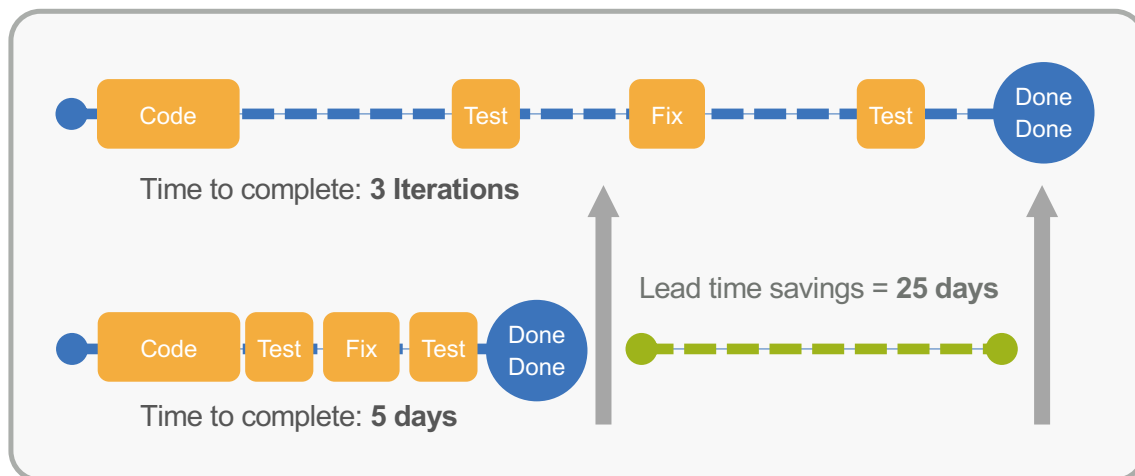
Optimize the full Value Stream

- ▶ Most problems with your process will surface as **delays**
- ▶ Most of the time spent getting to market is a result of these delays
- ▶ Reducing delays is the fastest way to reduce time-to-market



Reduce lead times and improve flow

A tale of two Stories

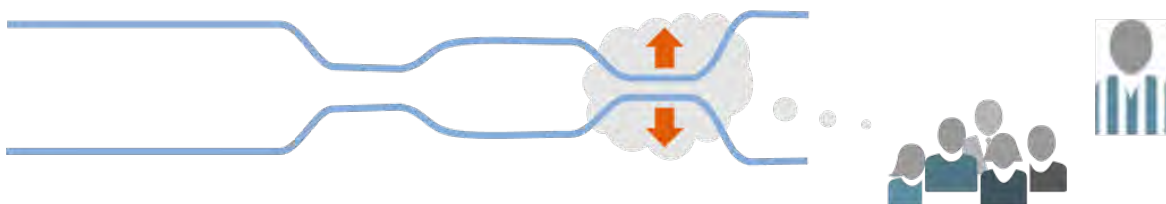


“I say an hour lost at a bottleneck is an hour out of the entire system. I say an hour saved at a non-bottleneck is worthless. Bottlenecks govern both throughput and inventory.”

— Eliyahu M. Goldratt, *The Goal*

Move from bottleneck to bottleneck

- ▶ The Scrum Master helps the team identify and remove bottlenecks
- ▶ Every system has only one or two bottlenecks that significantly constrain performance
- ▶ Once you have identified and removed the current bottleneck, there will be another one, but the system is already operating at a higher level of performance





Discussion: Identifying delays



- ▶ **Step 1:** Working in your groups, select a part of the development process from your own context or from the example below.
- ▶ **Step 2:** Discuss ways you could increase flow by reducing work in process (WIP) and eliminating bottlenecks.
- ▶ **Step 3:** Be prepared to share with the class.



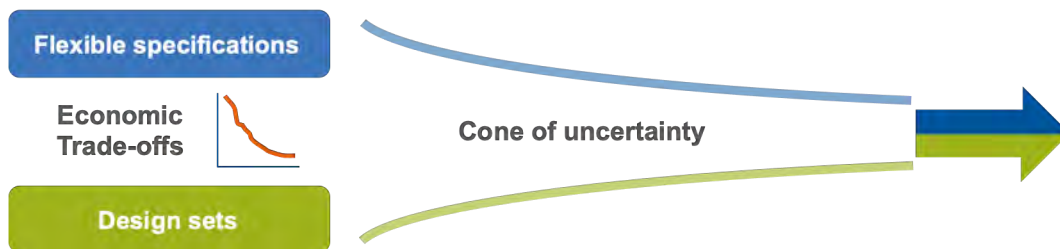
#3 Assume variability; preserve options

Development occurs in an uncertain world

Aggressively evaluate alternatives. Converge specifications and solution set.

—Allen C. Ward

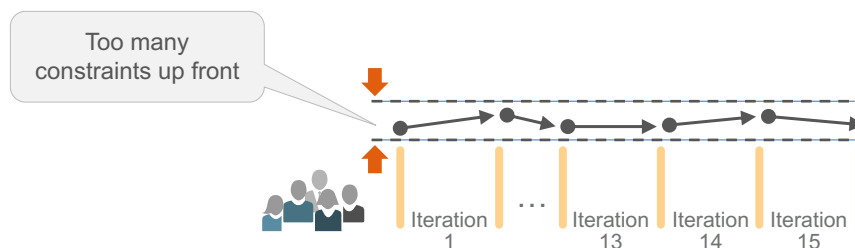
- ▶ You cannot possibly know everything at the start
- ▶ Requirements and designs must be flexible to build an optimal Solution
- ▶ Iterative, incremental development can reduce uncertainty over time



Common problem of many organizations

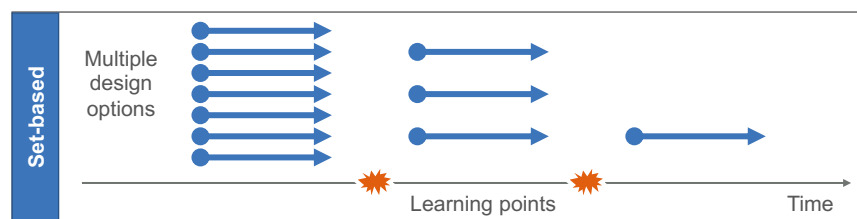
When Agile practices are adopted on top of a traditional, phase-gate mindset, teams end up with a typical problem:

- ▶ They follow an iterative and incremental development model while committing to a specific Solution specification early in the process
- ▶ As a result, the power of Agile is significantly underused and reduced to applying only minor adjustments



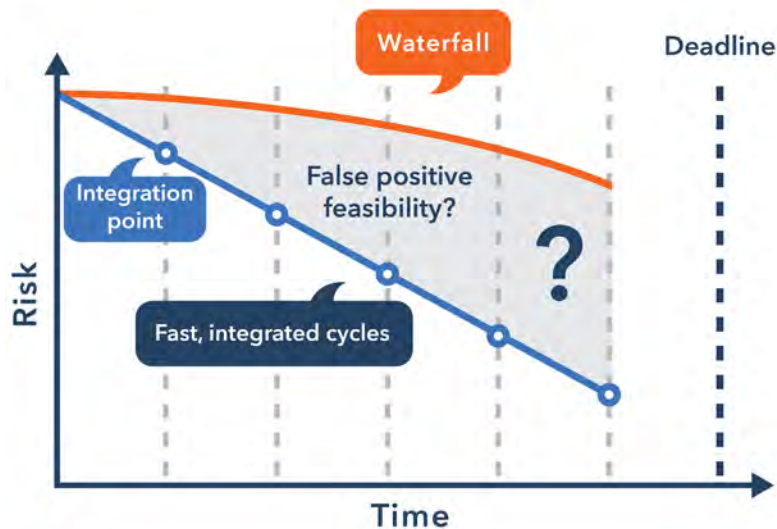
Different thought process is needed

- ▶ 'Just in time' elaboration of requirements and design
- ▶ Not everything should be defined at once
- ▶ Better requirements and design options will emerge over the course of Iterations
- ▶ Up-front thinking is not enough; 'learning by doing' must extend the paradigm
- ▶ Set-based design
- ▶ If you can't be right early on, preserve multiple options until you have more certainty
- ▶ Narrow them down over the course of the learning points and Iterations



#4 Build incrementally with fast, integrated learning cycles

Integration points reduce risk

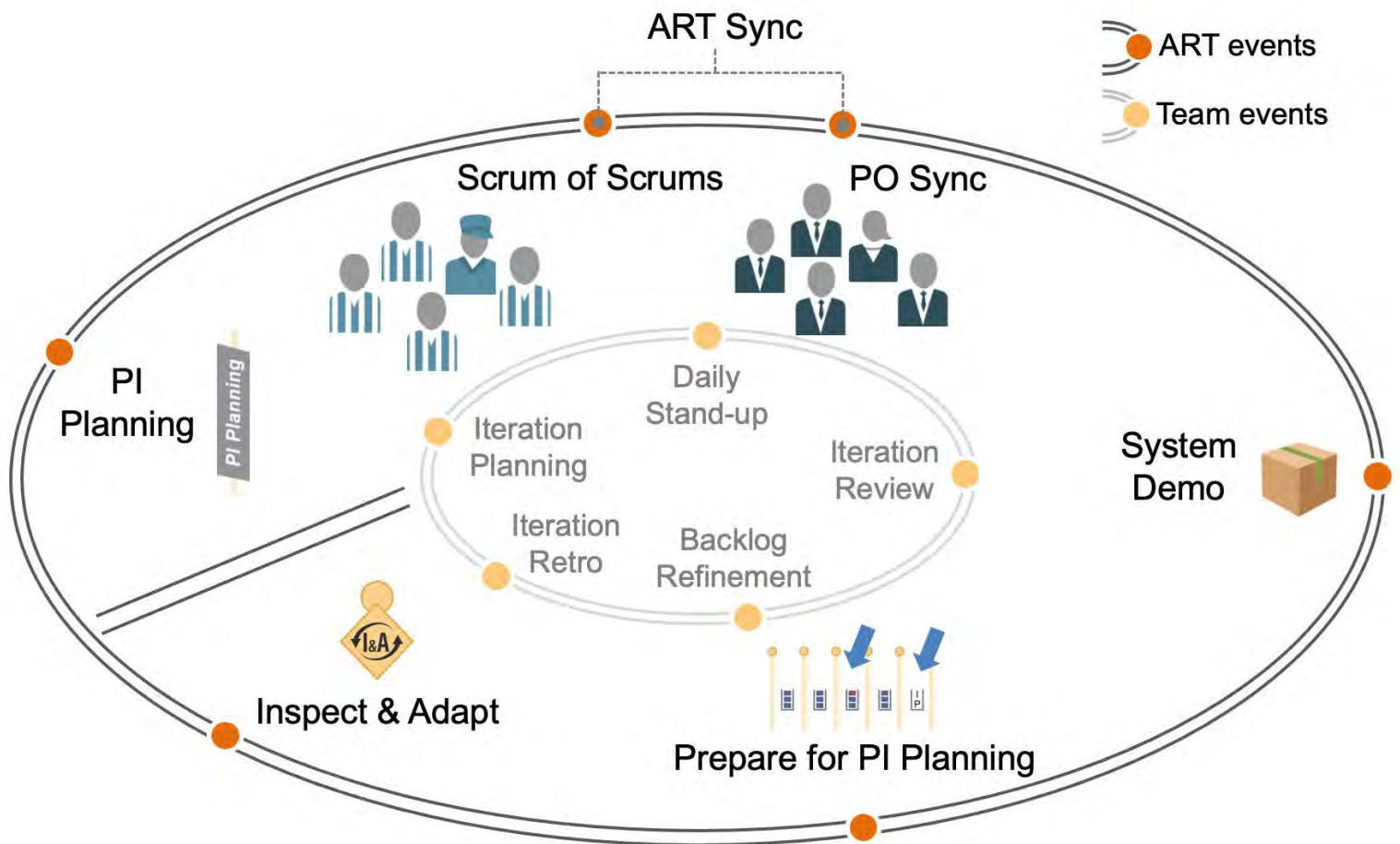


Activity: PDCA learning cycles representations



- ▶ **Step 1:** Individually, review the list of team and ART events in your workbook
- ▶ **Step 2:** Working in your groups, create a representation of the PDCA learning cycle for your team and for the ART events
- ▶ **Step 3:** Discuss where, as a Scrum Master, you anticipate bigger challenges and why
- ▶ **Step 4:** Present your PDCA representations and share some insights with the class

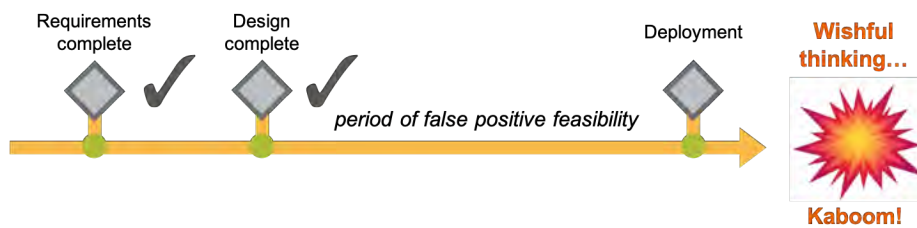
Team and ART Events



#5 Base milestones on objective evaluation of working systems

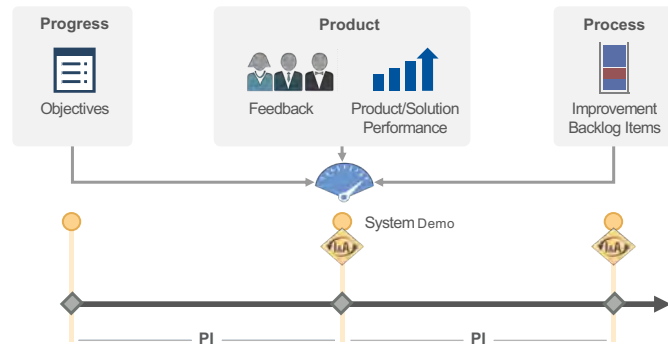
The problem of phase gate milestones

- ▶ They force design decisions too early; this encourages false-positive feasibility
- ▶ They assume a 'point' Solution exists and can be built correctly the first time
- ▶ They create huge batches and long queues, and they centralize requirements and design in program management



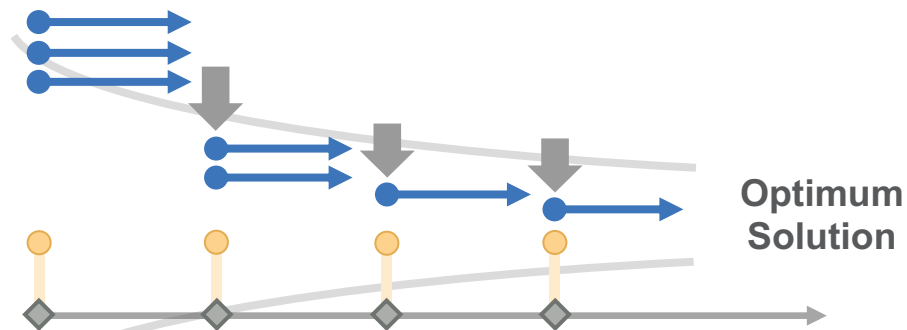
Apply objective Milestones

- ▶ Build the system in increments, each of which is an integration point that demonstrates some evidence of the feasibility of the Solution in process.
- ▶ Identify Milestones based on objective evaluation of working systems.



Iterate to the optimum Solution

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



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



Video: Visualize and limit WIP



Visualize  limit WIP

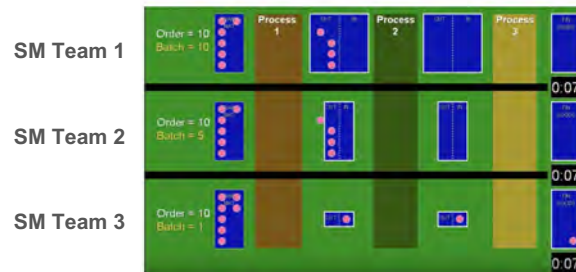
http://bit.ly/visualize_and_limit_WIP



Discussion: WIP and Flow



- ▶ **Step 1:** Working in your group, consider the flow of work represented in the previous video and discuss:
 - How do you think the Scrum Masters for each team feel about the progress? What can the Scrum Master from Team 1 learn from Team 2 and 3?
 - What is a large batch in your context and how can you improve flow?
- ▶ **Step 2:** Be prepared to share with the class.



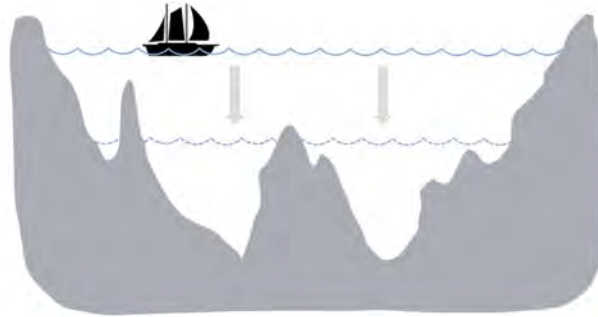
How can we reduce lead times?

- ▶ Reduce size of work
- ▶ Reduce bottlenecks
- ▶ Reduce waiting
- ▶ Increase swarming
- ▶ Improve quality



Identify bottlenecks in work in process (WIP)

Track work at the right level to reveal bottlenecks in the workflow.



At the surface, however, it all looks smooth and promising.

Too much WIP slows down the enterprise

Having too much WIP confuses priorities, causes frequent context switching, and increases overhead.

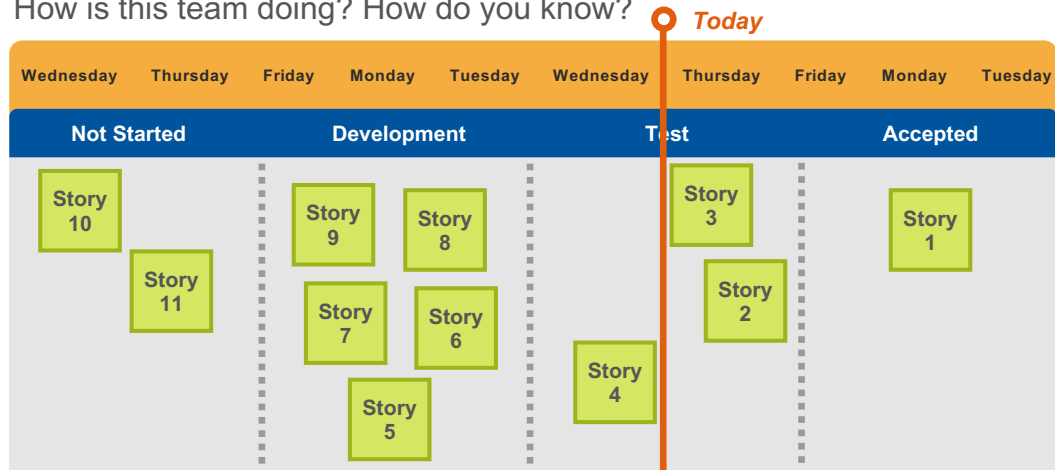
Ensure the teams are aligned to produce value as an ART.



Visualize and limit WIP

One team's big visible information radiator (BVIR)

How is this team doing? How do you know?



Activity: WIP improvement opportunities

Discuss



- ▶ **Step 1:** Referring to the example, discuss the effect of a three-story WIP constraint on the Development and Test stages.
- ▶ **Step 2:** Consider this scenario: You're a developer. You just finished Story 6. What would you do if:
 - There is no WIP constraint
 - The three-Story WIP constraint is in place
- ▶ **Step 3:** Which scenario has the highest throughput?



The importance of small batches

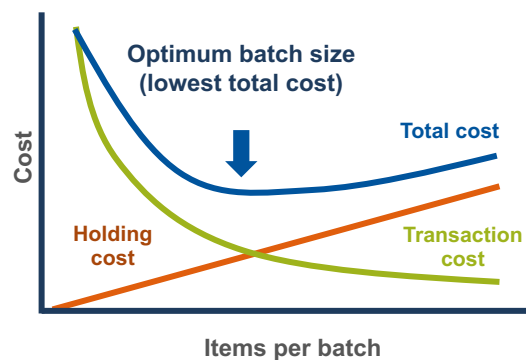
Small batches go through the system faster with lower variability.

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

Finding optimum batch size

Optimum batch size is an example of a U-curve optimization.

- ▶ Total costs are the sum of holding costs and transaction costs
- ▶ Higher transaction costs make optimal batch size bigger
- ▶ Higher holding costs make batch size smaller

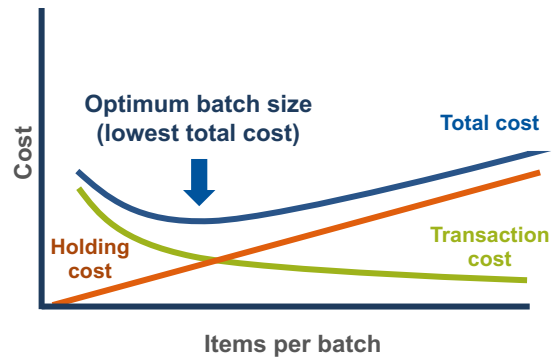


Principles of Product Development Flow, Don Reinertsen

Reducing optimum batch size

Reducing transaction costs reduces total costs and lowers optimum batch size.

- ▶ Reducing batch size:
 - Increases predictability
 - Accelerates feedback
 - Reduces rework
 - Lowers cost
- ▶ Batch size reduction probably saves **twice** what you would think



Principles of Product Development Flow, Don Reinertsen

Manage queue lengths

Email from a client service organization:

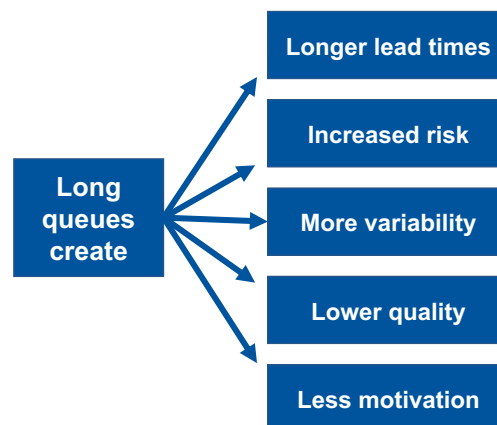
Thank you for contacting us.



We are experiencing increased volumes and apologize in advance for the delay.

Our goal is to contact you within...

Long queues: All bad



Principles of Product Development Flow, Don Reinertsen



Video: Formula 1 Pit Stops: 1950 and Today



<http://bit.ly/Formu1aPitStops>

Reduce queue lengths

- ▶ Understand Little's Law
 - Faster processing time decreases wait
 - Shorter queue lengths decrease wait
- ▶ Control wait times by controlling queue lengths:
 - WIP limits, small batches, defer commitments

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

Average wait time = average queue length divided by average processing rate

Example – Given an average processing speed of 10 Features per quarter and a committed set of 30 Features, a new Feature will experience an approximate wait time of:

$$\frac{30 \text{ items}}{10 \text{ items/Quarter}} = 3 \text{ Quarters}$$

#7 Apply cadence, synchronize with cross-domain planning

Cadence and synchronization

Cadence

- ▶ Converts unpredictable events into predictable occurrences and lowers cost
- ▶ Makes waiting times for new work predictable
- ▶ Supports regular planning and cross-functional coordination
- ▶ Limits batch sizes to a single interval
- ▶ Controls injection of new work
- ▶ Provides scheduled integration points

Note: Delivering on cadence requires scope or capacity margin

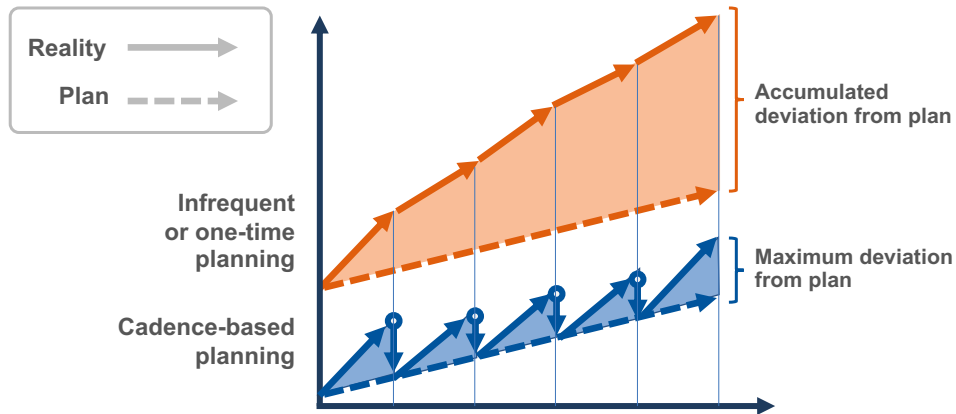
Synchronization

- ▶ Causes multiple events to happen simultaneously
- ▶ Facilitates cross-functional trade-offs
- ▶ Provides routine dependency management
- ▶ Supports full stem integration and assessment
- ▶ Provides multiple feedback perspectives

Note: To work effectively, design cycles must be synchronized

Control variability with planning cadence

Cadence-based planning limits variability to a single interval.



Synchronize with cross-domain planning

Future product development tasks can't be predetermined. Distribute planning and control to those who can understand and react to the end results.

—Michael Kennedy, *Product Development for the Lean Enterprise*

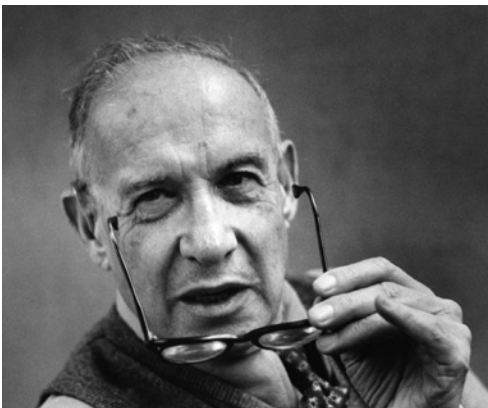
- ▶ Everyone plans together at the same time
- ▶ Management sets the mission with minimum constraints
- ▶ Requirements and design emerge
- ▶ Important decisions are accelerated
- ▶ Teams take responsibility for their own plans



#8 Unlock the intrinsic motivation of knowledge workers

On managing knowledge workers

Workers are knowledge workers if they know more about the work they perform than their bosses.
—Peter Drucker



Used with permission from The Drucker Institute at Claremont Graduate University

- ▶ Workers themselves are most qualified to make decisions about how to perform their work.
- ▶ The workers must be heard and respected for management to lead effectively.
- ▶ Knowledge workers must manage themselves. They need autonomy.
- ▶ Continuing innovation must be part of the work, the tasks, and the responsibilities of knowledge workers.

#9 Decentralize decision-making

Decentralize decision-making

Define the economic logic behind a decision; empower others to make the changes.

Centralize	Decentralize everything else
<ul style="list-style-type: none">▶ Infrequent – Not made very often and usually not urgent (Example: Internationalization strategy)▶ Long-lasting – Once made, highly unlikely to change (Example: Common technology platform)▶ Significant economies of scale – Provide large and broad economic benefit (Example: Compensation strategy)	<ul style="list-style-type: none">▶ Frequent – Routine, everyday decisions (Example: Team and Program Backlog)▶ Time critical – High cost of delay (Example: Point release to Customer)▶ Requires local information – Specific and local technology or Customer context is required (Example: Feature criteria)



Activity: Decentralize decision-making

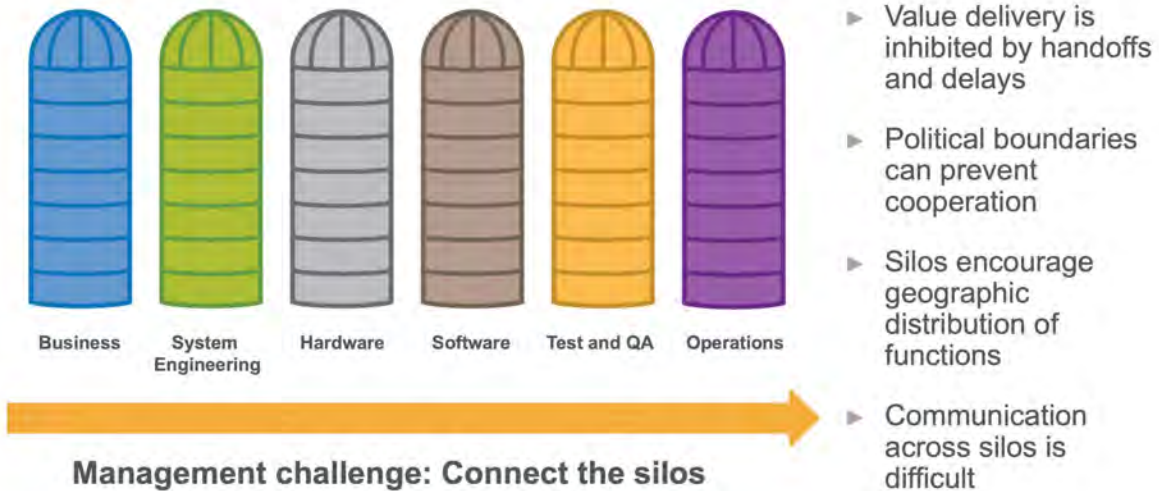


- ▶ **Step 1:** Consider three significant decisions you are currently facing. Capture those decisions.
- ▶ **Step 2:** Rate each decision based on the frequency, time criticality, and economies of scale, assigning a value of 0, 1, or 2.
- ▶ **Step 3:** Add the total values: 0 – 3 centralize and 4 – 6 decentralize.

Decision	Frequent? Y=2 N=0	Time-critical? Y=2 N=0	Economies of scale? Y=0 N=2	Total

#10 Organize around value

Value doesn't follow silos

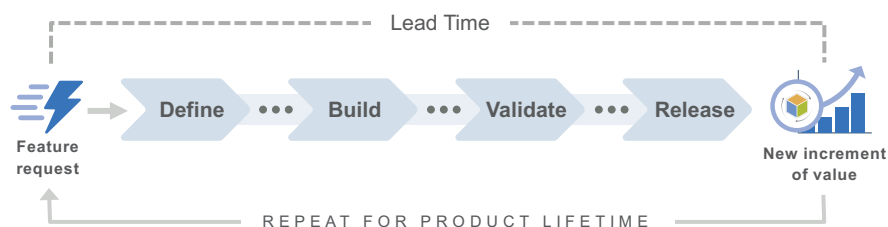


Instead, organize around Development Value Streams

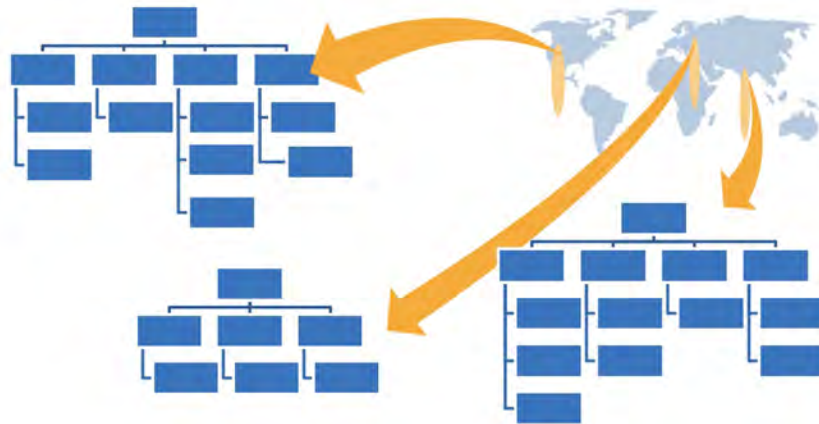
The aim of development is in fact the creation of profitable operational value streams.

—Allen C. Ward

- ▶ Includes activities from recognizing an opportunity through release and validation
- ▶ Contains the steps, the flow of information and material, and the people who develop the Solutions used by the operational Value Streams

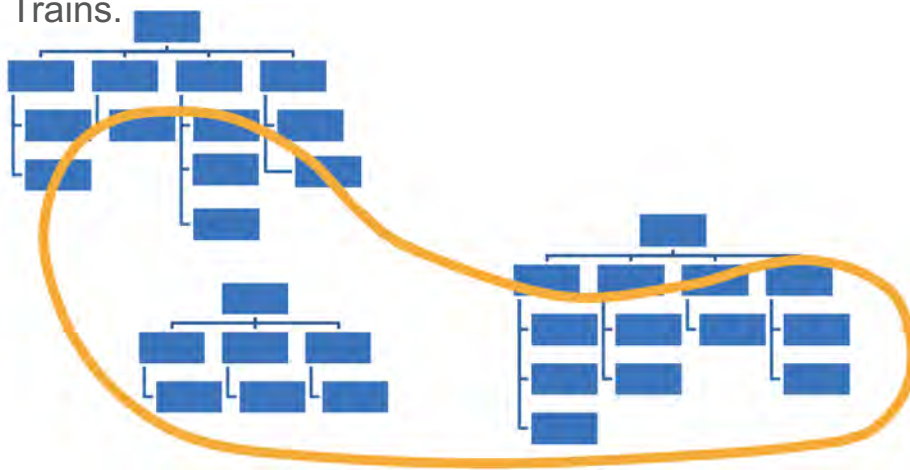


Value at scale is distributed



Value flows across organizational boundaries

Identify the Value Streams within which to build one or more Agile Release Trains.





Activity: Building and presenting SAFe Principles poster



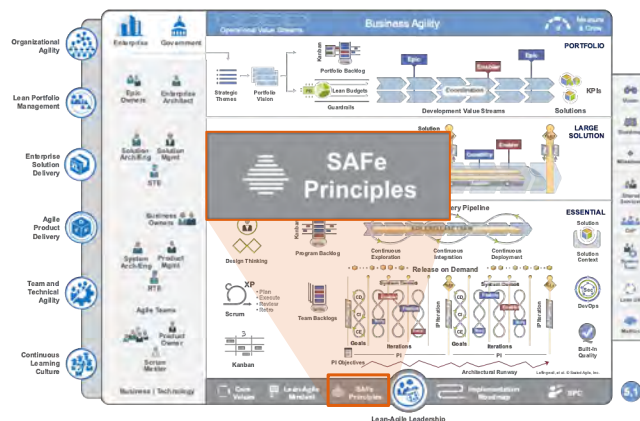
- ▶ **Step 1:** In your group, pick one SAFe Principle
- ▶ **Step 2:** Document the SAFe Principle with the following requirements:
 - Title: Your team's chosen SAFe Principle (e.g., Take an economic view)
 - How does a Scrum Master apply the principle?
 - Capture strategies, actions, and events that apply to the principle in the format of your choice
- ▶ **Step 3:** Present your applications to the class



Lesson review

In this lesson you:

- ▶ Applied the SAFe Principles in the role of a Scrum Master



<https://www.scaledagileframework.com/safe-lean-agile-principles/>

Lesson notes



Enter your notes below:

Reminder: If using a digital workbook, save your PDF often so you don't lose any of your notes.

Lesson 3

Exploring Agile and Scrum Anti-Patterns

SAFe® Course - Attending this course gives students access to the SAFe Advanced Scrum Master exam and related preparation materials.



Lesson Topics

- 3.1** Explore anti-patterns associated with the Product Owner role
- 3.2** Explain how Stories and tasks may lead to anti-patterns
- 3.3** Identify context-specific anti-patterns in your environment



Learning Objectives

At the end of this lesson, you should be able to:

- ▶ Identify anti-patterns associated with the Product Owner role
- ▶ Analyze how Stories and tasks may lead to anti-patterns
- ▶ Evaluate context-specific anti-patterns in your environment

3.1 Explore anti-patterns associated with the Product Owner role

Recognizing anti-patterns

As an Agile coach, the Scrum Master must learn to recognize anti-patterns in the process. An anti-pattern can be ...

Structural or behavioral		Internal or external	
Structural example	Team has more than one Product Owner	Internal example	Developers don't work collaboratively on Stories
Behavioral example	Partially completed Stories are being carried over from Iteration to Iteration	External example	Lack of coordination with other teams leads to excessive WIP

Many anti-patterns can be traced to the PO role

Underperforming in the Product Owner role can lead to dysfunction on the team.

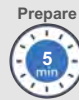
Key responsibilities of the Product Owner:

- ▶ Facilitate Team Backlog refinement
- ▶ Prepare for and participate in Iteration Planning
- ▶ Elaborate Stories and Enablers just-in-time
- ▶ Address team questions; be the voice of the customer
- ▶ Accept Stories
- ▶ Participate in the Iteration Review and Retrospective
- ▶ Coordinate with other Product Owners to manage dependencies





Discussion: Anti-patterns that involve the Product Owner



- ▶ **Step 1:** Working in your groups, brainstorm anti-patterns that arise from the interaction between the Product Owner and the rest of the team
- ▶ **Step 2:** Discuss how you as a Scrum Master will address some of these anti-patterns
- ▶ **Step 3:** Share with the class



3.2 Explain how Stories and tasks may lead to anti-patterns

Big Stories are a frequent source of anti-patterns

A team that can't iterate isn't able to inspect and adapt.

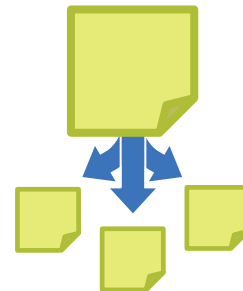
- ▶ Big Stories do not support team Iteration
- ▶ Smaller Stories allow for faster, more reliable implementation
- ▶ Splitting bigger Stories into smaller ones is an essential skill



When Stories are too big to fit into an Iteration, they are split

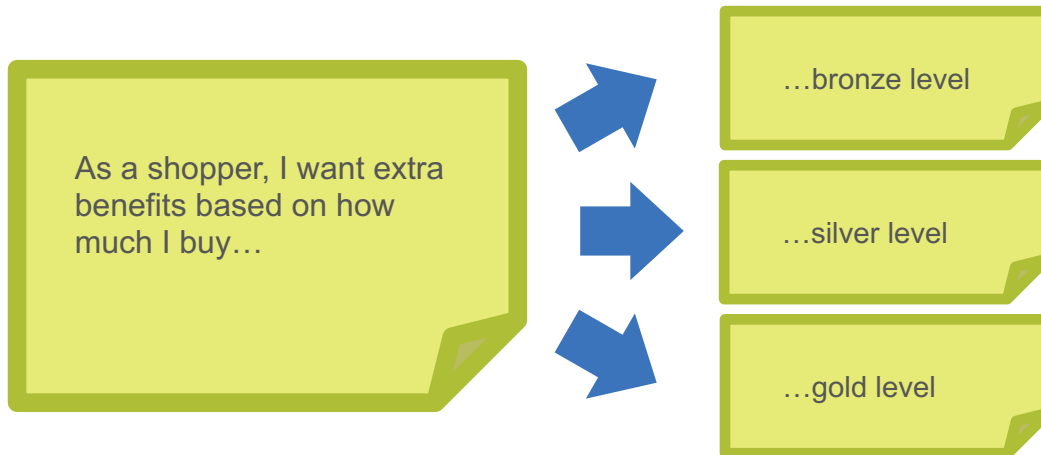
Stories are split using the same techniques as decomposing Features

- | | |
|----------------------------|--------------------------|
| 1 Workflow steps | 6 Data entry methods |
| 2 Business rule variations | 7 Defer system qualities |
| 3 Major effort | 8 Operations |
| 4 Simple/complex | 9 Use case scenarios |
| 5 Variations in data | 10 Break out a spike |



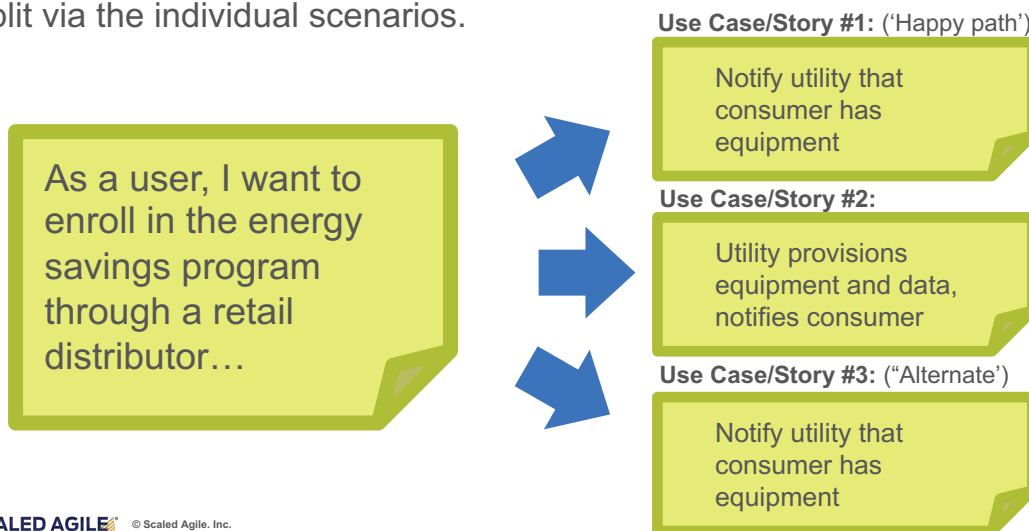
Split by business rule variations

Business rule variations often provide a straightforward splitting scheme.



Split by use case scenarios

If use cases are used to represent complex interactions, the Story can be split via the individual scenarios.



Split by simple/complex

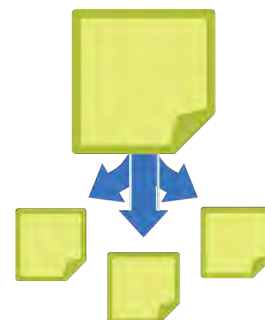
Simplify. What's the simplest version that can possibly work?



Activity: Splitting Stories



- ▶ **Step 1:** Working in your groups, select a big Story from your actual backlog that is too big to complete in an Iteration
- ▶ **Step 2:** Considering the patterns we discussed, split the Story into two or three smaller Stories that could be completed in an Iteration
- ▶ **Step 3:** Share with the class





Discussion: A Story that can't be split?



- ▶ **Step 1:** Consider the following scenario: Your team is insisting that big Stories in the backlog cannot be split into smaller ones.
- ▶ **Step 2:** In your group, discuss the following:
 - How might you coach the team to reconsider?
 - What specific steps would you take to shift their mindset?
 - What event is appropriate to have this discussion with the team?
- ▶ **Step 3:** Roleplay how you will facilitate the discussion with the team.
- ▶ **Step 4:** Be prepared to share with the class.

Story Writing and Splitting resources

The following resources are available on the SAFe Community Platform:

- ▶ Writing and Splitting Stories Guide
 - Also available in the Iteration Execution Toolkit
- ▶ Video Playlist: Stories



3.3 Identify context-specific anti-patterns in your environment

PO and backlog, planning, and commitment anti-patterns

PO and Backlog

Product Owner and team do Iteration Planning without preparation

There is more than one PO per team

PO is not sufficiently involved during Iteration execution

Planning

Planning is based on tasks, not on Stories and acceptance criteria

Commitment

Team does not commit to clear Iteration goals

Execution, Demos, and Retrospective anti-patterns

Execution

Developers don't work collaboratively on Stories

Waterfalling Iterations:
Team integrates and tests
Stories only at Iteration
end

Done isn't 'done': Debt is
carried forward Iteration to
Iteration

Demos

Story reported but not
demonstrated (non-UI
Stories, spikes, refactors,
etc.)

Retrospectives

"Idea-fest" instead of
focus on near-term
incremental
improvements



Discussion: Anti-patterns



- ▶ **Step 1:** In your group, discuss the following:
 - Which of the anti-patterns resonate most strongly with you?
 - What other anti-patterns have you experienced?
 - Could these anti-patterns be fully resolved by the Scrum Master?
- ▶ **Step 2:** Choose two or three of these anti-patterns. For each anti-pattern, build a list of action items and strategies that will allow the Scrum Master to resolve the anti-pattern.
- ▶ **Step 3:** Share your master list with the class.

SAFe resources to help avoid and overcome anti-patterns

Use the following resources, available on the SAFe Community Platform:

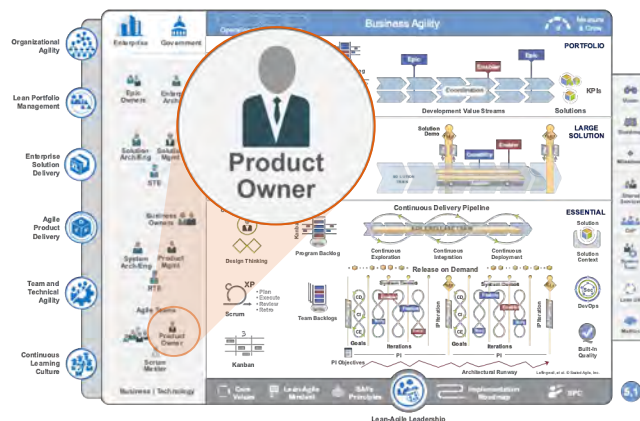
- ▶ Iteration Execution Toolkit
- ▶ Iteration Retrospective Facilitator Checklist (also included in the Iteration Execution Toolkit)
- ▶ Video: How to Run an Effective Iteration Retrospective Meeting
- ▶ SAFe Collaborate Retrospective templates:
 - Start – Stop – Continue & I Wish
 - Actions for Retrospectives



Lesson review

In this lesson, you:

- ▶ Identified anti-patterns associated with the Product Owner role
- ▶ Analyzed how Stories and tasks may lead to anti-patterns
- ▶ Evaluated context-specific anti-patterns in your environment



<https://www.scaledagileframework.com/product-owner/>

Lesson notes



Enter your notes below:

Reminder: If using a digital workbook, save your PDF often so you don't lose any of your notes.

Lesson 4

Facilitating Program Execution

SAFe® Course - Attending this course gives students access to the SAFe Advanced Scrum Master exam and related preparation materials.



Lesson Topics

- ▶ **4.1** Synchronize development with the Agile Release Train
- ▶ **4.2** Organize teams around the flow of value
- ▶ **4.3** Plan the Program Increment



Lesson Topics

- ▶ 4.4 Execute the Program Increment
- ▶ 4.5 Enable teams to release value on demand
- ▶ 4.6 Prepare for the next PI Planning session



Learning objectives

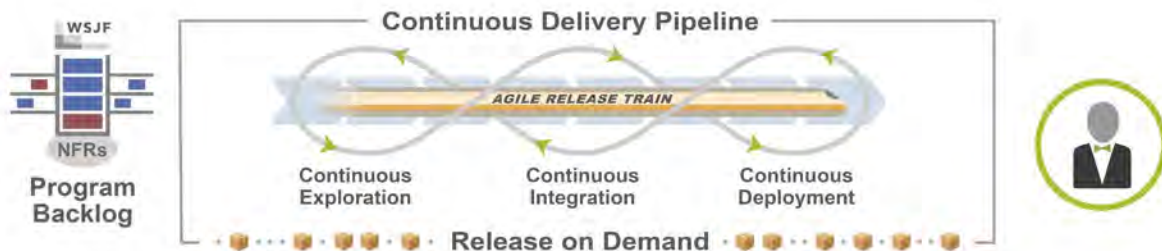
At the end of this lesson, you should be able to:

- ▶ Execute development with the Agile Release Train
- ▶ Establish teams around the flow of value
- ▶ Organize the Program Increment
- ▶ Execute the Program Increment
- ▶ Manage teams to release value on demand
- ▶ Prepare for the next PI Planning session

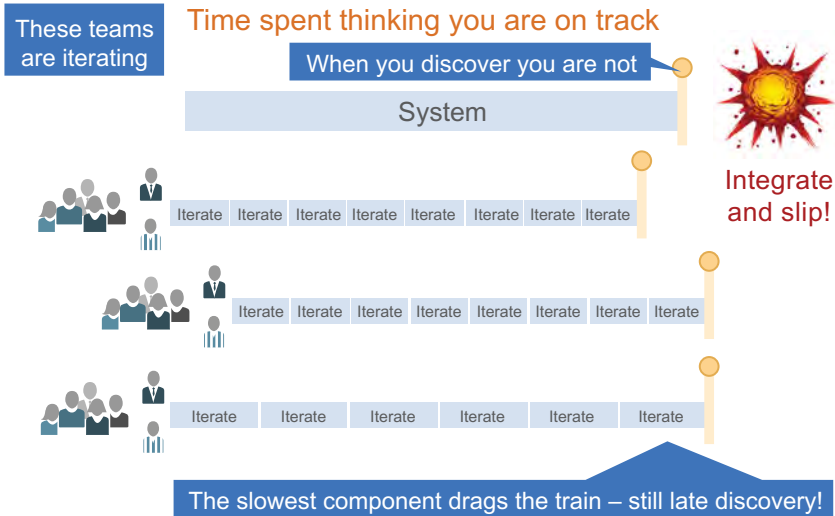
4.1 Synchronize development with the Agile Release Train

Agile Release Trains (ARTs) deliver Solutions

An ART is a long-lived, self-organizing team of Agile Teams.



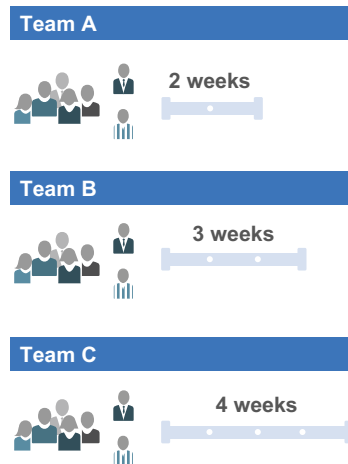
Cadence without synchronization is not enough



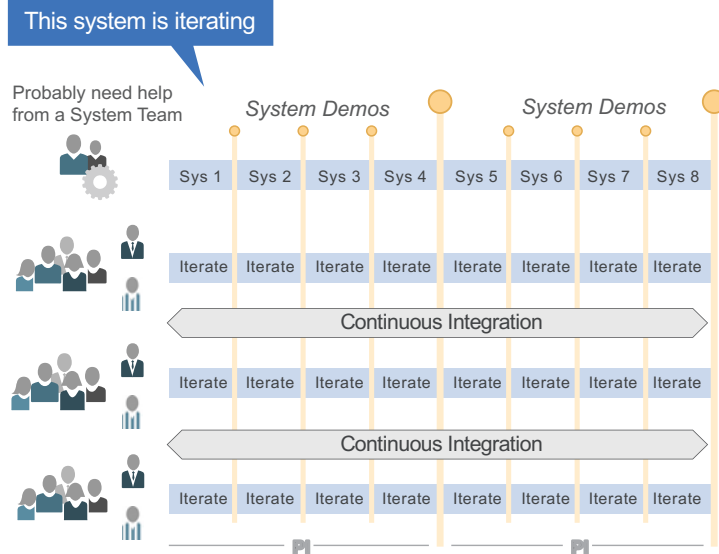
Activity: Different cadences



- ▶ **Step 1:** Consider a scenario in which three teams are working on different Iteration cadences:
 - Team A: 2-week Iterations
 - Team B: 3-week Iterations
 - Team C: 4-week Iterations
- ▶ **Step 2:** In your groups, discuss the following:
 - If the teams start at the same time, when is the first point in time they can align on the Iteration outcomes?
- ▶ **Step 3:** Share with the class.

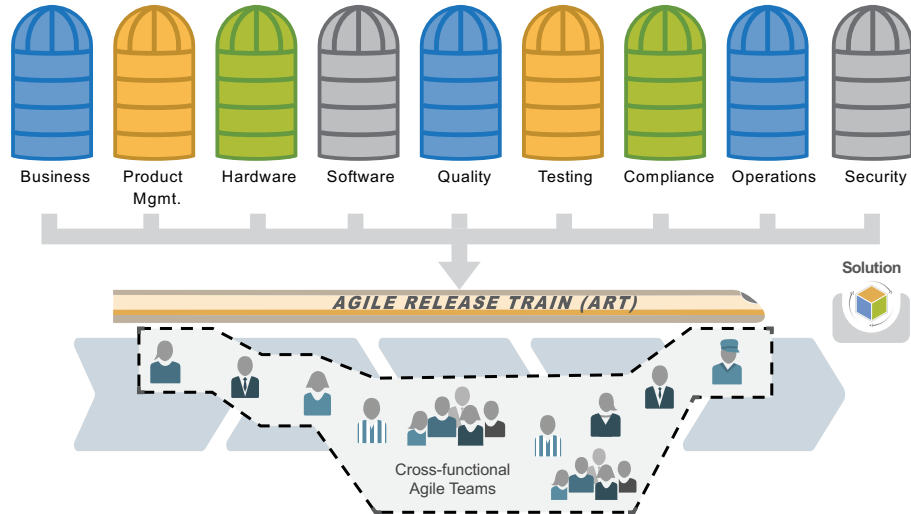


Synchronize to assure delivery

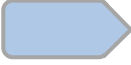





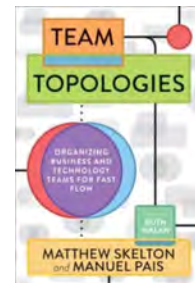
4.2 Organize teams around the flow of value

Build cross-functional Agile Release Trains



Teams on the ART are organized for flow

-  **Stream-aligned team** – organized around the flow of work and has the ability to deliver value directly to the Customer or end user.
-  **Complicated subsystem team** – organized around specific subsystems that require deep specialty skills and expertise.
-  **Platform team** – organized around the development and support of platforms that provide services to other teams.
-  **Enabling team** – organized to assist other teams with specialized capabilities and help them become proficient in new technologies.

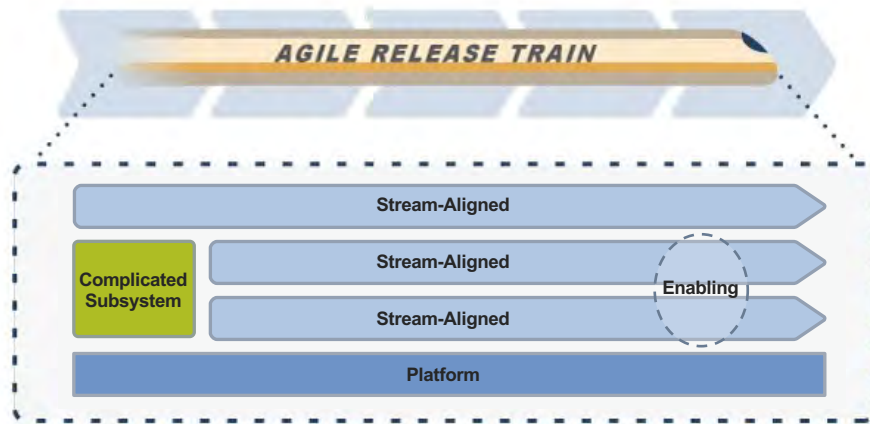


More information in the Advanced Topic Article:

<https://www.scaledagileframework.com/organizing-agile-teams-and-arts-team-topologies-at-scale/>

ARTs are organized to deliver value continuously

Consider the necessary interactions between the teams and organize to maximize flow.



Roles on the Agile Release Train



Release Train Engineer (RTE) 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.



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



Business Owners are key stakeholders on the Agile Release Train.



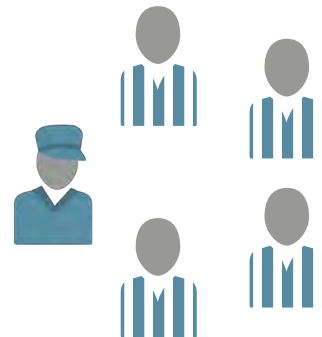
Responsibilities of the RTE

- ▶ Manage and optimize flow of value through the ART
- ▶ Facilitate PI Planning readiness and the event itself
- ▶ Aggregate and communicate PI Objectives
- ▶ Assist with execution and Feature completion tracking
- ▶ Assist with economic decision-making through Feature estimation and roll-up to Value Stream and portfolio
- ▶ Escalate and track impediments
- ▶ Foster collaboration between teams and system-level stakeholders; manage risks and dependencies
- ▶ Drive relentless improvement via Inspect and Adapt



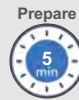
Operating as a community

- ▶ RTEs often the best fit to assist Scrum Masters in removing systemic impediments
- ▶ RTEs and Scrum Masters see problems with the train/team structure firsthand
- ▶ Together RTEs and Scrum Masters are able to take a systems view of the Agile Release Train
- ▶ Operating as a community is important
 - Regularly meet to discuss problems
 - Exchange experiences





Discussion: Scrum Master responsibilities



- ▶ **Step 1:** In your group, discuss the following:
 - What are your responsibilities as a member of the Scrum Master and RTE community, that go beyond the day-to-day facilitation of your Agile Team?
 - What are some examples and actions you have taken to promote the Scrum Master and RTE community to innovate the ART?
 - What challenges might you face in performing this aspect of your role?
- ▶ **Step 2:** Share with the class.

Product Management owns the Program Backlog

Assumptions about requirements need to be validated.

- ▶ Primary responsibilities of Product Management:
 - Understand customer needs; validate Solutions
 - Work with System Architect/Engineering to understand the value of Enablers
 - Develop and communicate Vision and Roadmap



Product Management owns the Program Backlog (cont.)

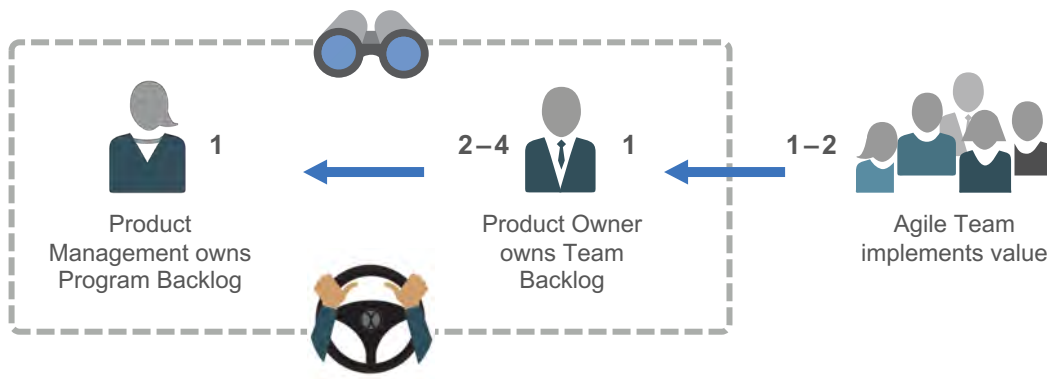
Additional responsibilities and activities of Product Management:

- ▶ Manage and prioritize the flow of work to the program
- ▶ Prepare for and participate in PI Planning
- ▶ Define releases and program increments
- ▶ Participate in demos and Inspect and Adapt
- ▶ Build an effective Product Management/Product Owner team



The PO/PM team steers the ART

At scale, a single person cannot handle product and market strategy while also being dedicated to an Agile Team.





Activity: Facilitating PO/PM collaboration



- ▶ **Step 1:** As a group, discuss the following:
 - Is your Product Owner effectively collaborating on the priorities and backlog with Product Management?
 - Is the Product Owner sufficiently empowered to represent the voice of the customer?
 - How could you help facilitate PO/PM collaboration to align for backlog prioritization?
- ▶ **Step 2:** Share with the class.

4.3 Plan the Program Increment



Video: Agile Development Story: The Power of PI Planning



http://bit.ly/Agile_development

What is PI Planning?

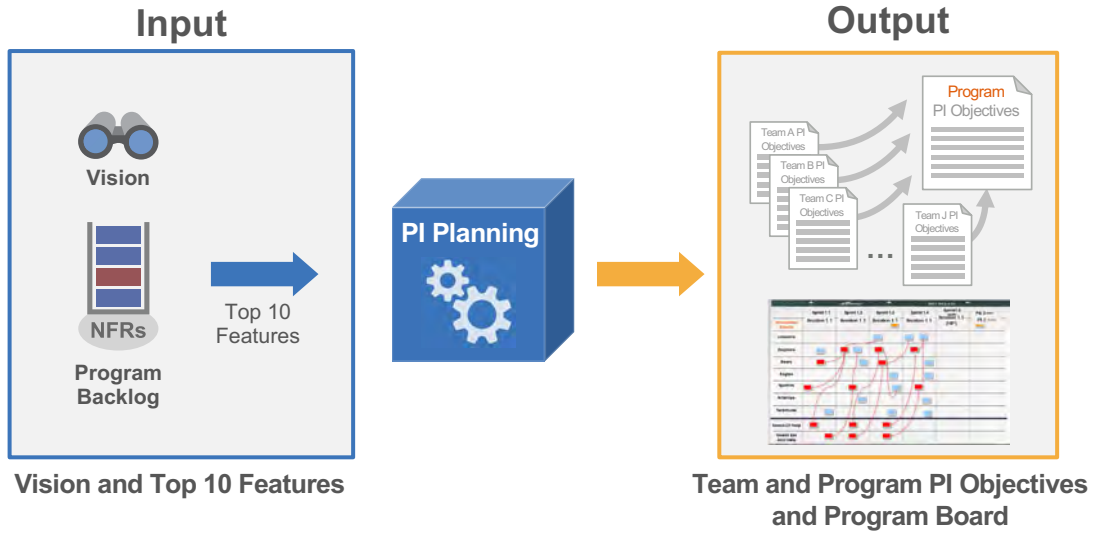
Program Increment (PI) Planning is a cadence-based event that serves as the heartbeat of the Agile Release Train (ART), aligning all teams on the ART to a shared mission and Vision.

- ▶ Two days every 8 – 12 weeks (10 weeks is typical)
- ▶ Everyone plans together
- ▶ Product Management owns Feature priorities
- ▶ Development teams own Story planning and high-level estimates
- ▶ Architect/Engineering and UX work as intermediaries for governance, interfaces, and dependencies

“There is no magic in SAFe...except maybe for PI Planning”

—Dean Leffingwell

The PI Planning process



PI Planning: Day 1 agenda

Business context	8:00 – 9:00	<ul style="list-style-type: none"> State of the business
Product/Solution Vision	9:00 – 10:30	<ul style="list-style-type: none"> Vision and prioritized Features
Architecture Vision and development practices	10:30 – 11:30	<ul style="list-style-type: none"> Architecture, common frameworks, etc. Agile tooling, engineering practices, etc.
Planning context and lunch	11:30 – 1:00	<ul style="list-style-type: none"> Facilitator explains the planning process
Team breakouts	1:00 – 4:00	<ul style="list-style-type: none"> Teams develop draft plans and identify risks and impediments Architects and Product Managers circulate
Draft plan review	4:00 – 5:00	<ul style="list-style-type: none"> Teams present draft plans, risks, and impediments
Management review and problem solving	5:00 – 6:00	<ul style="list-style-type: none"> Adjustments made based on challenges, risks, and impediments

Business context and Product/Solution Vision

PI Planning begins with executive leadership:

- ▶ Sharing the state of the business and upcoming objectives
- ▶ Communicating the key portfolio priorities
- ▶ Analyzing the organization's strengths, weaknesses, opportunities, and threats (SWOT)
- ▶ Introducing Product Management for the Vision and the high-priority Features



Video Playlist: Introduction to PI Planning



<http://bit.ly/PIPlanningPlaylist>

Architecture, User Experience (UX), and development practices

Architecture, UX, and development practices are high priorities in PI Planning, not afterthoughts!

- ▶ A System Architect presents the Vision for architecture, new architecture Epics, and common frameworks
- ▶ Development management may provide updates on Agile tooling and improvements in engineering practices
- ▶ UX professionals provide guidance around usability issues

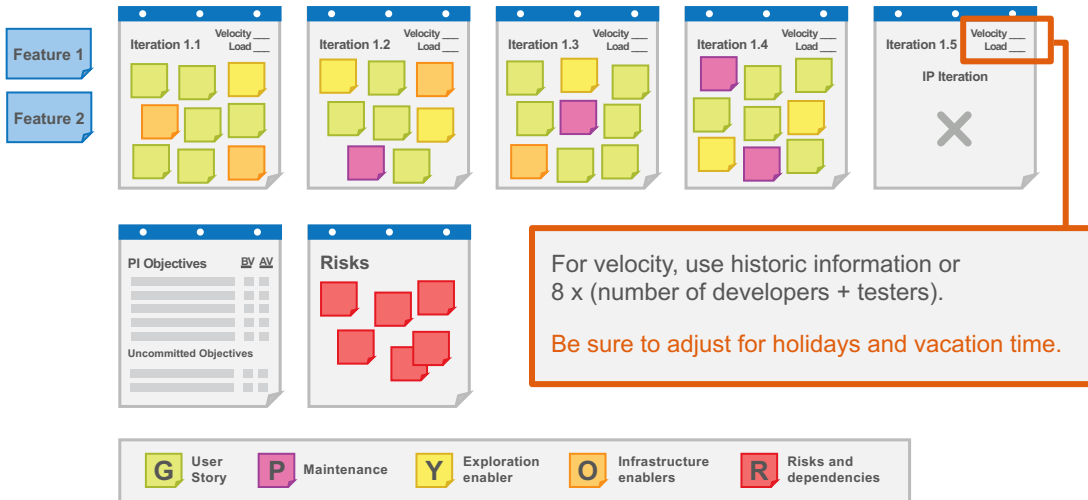


Team breakout #1

- ▶ In breakouts, each team breaks down its Features into User Stories. Stories are estimated and placed into Iterations.
- ▶ There is a lot of back and forth between the teams, mostly about understanding and minimizing dependencies.



Team plan



Activity: Calculating capacity

Duration



- ▶ **Step 1:** Consider the following scenario:
 - There are nine teams on the train with velocities of 32, 48, 61, 30, 65, 18, 25, 62, 38 in the Iteration. The train operates on a five-Iteration PI cadence (with the last Iteration reserved for Innovation and Planning).
- ▶ **Step 2:** Discuss as a group, what is the ART's velocity?
 - **Note:** Each number above is expressed in Story points relevant to that team only, not comparable with other teams' numbers.

Starting fast with normalized Story points

Normalized estimation technique:



Example: Assuming a seven-person team composed of three developers, two testers, one Product Owner, and one Scrum Master, with no vacations.

Exclude Scrum Master and Product Owner from the calculation.

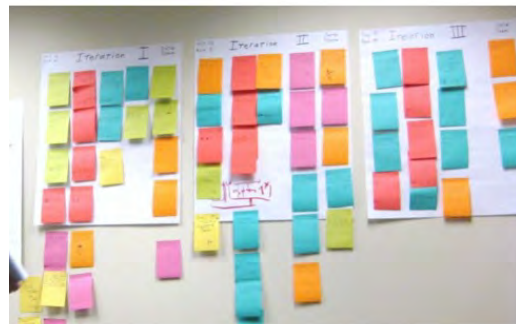
**Estimated Capacity =
5 * 8 pts = 40 pts/iteration**

- ▶ For every full-time developer and tester on the team, give the team eight points (adjust for part-timers).
- ▶ Subtract one point for every team member vacation day and holiday.
- ▶ Find a small Story that would take about a half-day to develop and a half-day to test and validate. Call it a 1.
- ▶ Estimate every other Story relative to that 1-point story.
- ▶ Never look back (don't worry about recalibrating).

Color-coding Stories

We color-code the backlog items to give visibility into the investments.

We can visually see that some teams may have significant backlog items dedicated to things like maintenance.



G User Story

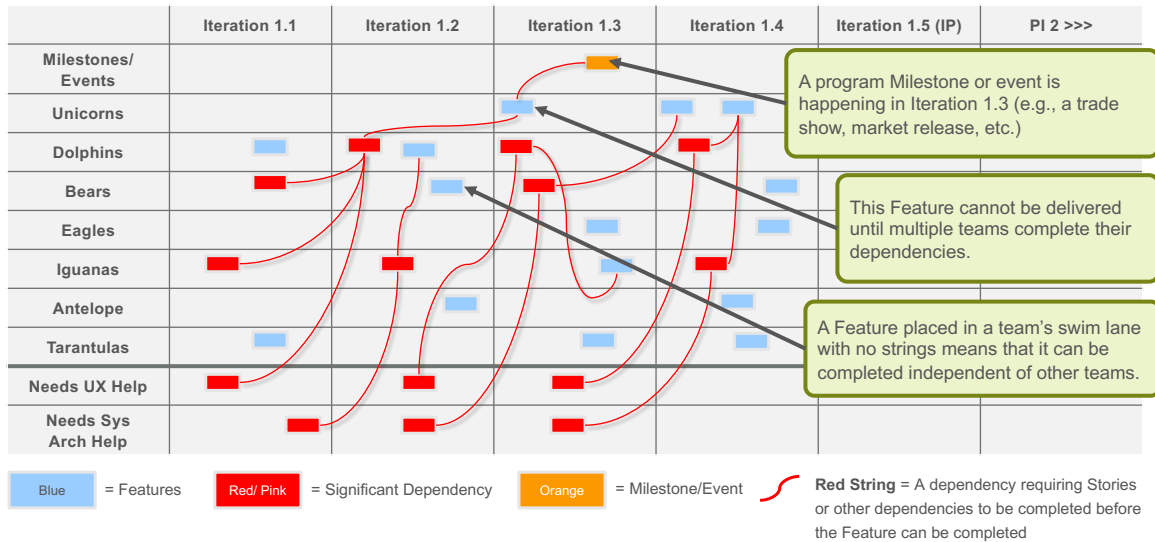
P Maintenance

Y Exploration Enabler

O Infrastructure Enablers

R Risks and dependencies

Program board: Feature delivery, dependencies, and Milestones



Discussion: Identifying problems

Prepare



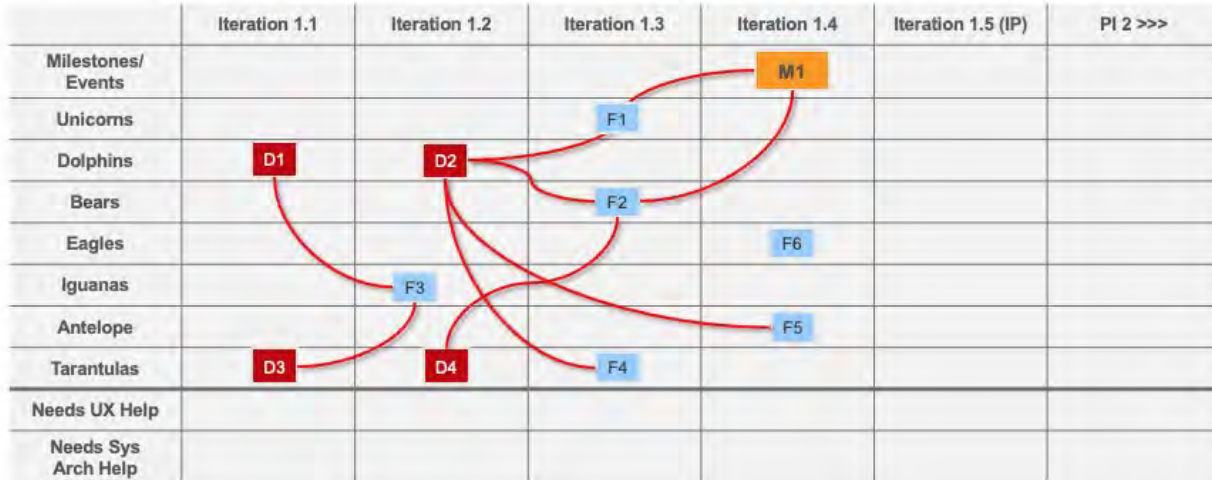
Share



- ▶ **Step 1:** As a group, review the examples of program boards in your workbook.
- ▶ **Step 2:** Discuss the following:
 - What problems can you identify?
 - What can you do during PI planning to mitigate the issues? What can you do after PI planning?
- ▶ **Step 3:** Share with the class.

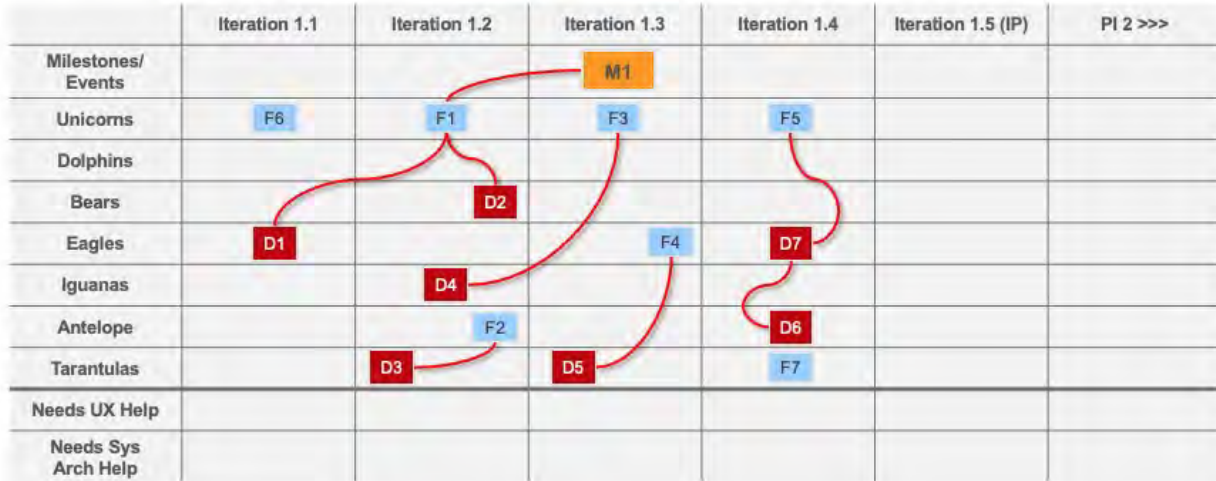
Program Board 1

Example program board #1



Program Board 2

Example program board #2



SCALED AGILE  © Scaled Agile, Inc.

Create Alignment with PI Objectives

- ▶ Objectives are business summaries of what each team intends to deliver in the upcoming PI.
- ▶ They often directly relate to intended Features in the backlog.
- ▶ Other examples:
 - Aggregation of a set of Features
 - A Milestone like a trade show
 - An Enabler Feature supporting the implementation
 - A major refactoring

Objectives for PI 1	BV	ABV
1. Show routing calculations between the 5 most frequent destinations	_____	_____
2. Navigate autonomously from distribution center the most frequent destination	_____	_____
3. Parallel park for a delivery	_____	_____
4. Return to distribution center after delivery	_____	_____
5. Include traffic data in route planning	_____	_____
6. Recall a delivery that is already in progress	_____	_____
Uncommitted Objectives		
7. Spike: Reduce GPS signal loss by 25%	_____	_____
8. Demonstrate real-time rerouting to avoid delays (e.g., accident, construction)	_____	_____

Maintain predictability with uncommitted objectives

Uncommitted objectives help improve the predictability of delivering business value.

- ▶ They are planned and aren't extra things teams do 'just in case you have time'
- ▶ They are not included in the commitment, thereby making the commitment more reliable
- ▶ If a team has low confidence in meeting a PI Objective, it should be moved to uncommitted
- ▶ If an objective has many unknowns, consider moving it to uncommitted and put in early spikes
- ▶ Uncommitted objectives count when calculating load

Objectives for PI 1

Uncommitted Objectives
7. Spike: Reduce GPS signal loss by 25%
8. Demonstrate real-time rerouting to avoid delays (e.g., accident, construction)

Scrum of Scrums (SoS)

The hourly Scrum of Scrums checkpoint helps keep teams on track and supports early identification of risk.

Hourly SoS planning checkpoint:

- ▶ Keeps teams on track with hourly planning Milestones
- ▶ Helps drive out risks, impediments, and dependencies



Activity: Getting back on track with planning

Prepare



Share



- ▶ **Step 1:** In your groups, consider the following scenario:
 - *You are at the second SoS event. The planning radiator shows that your team is quite behind. Some Stories are estimated, but none of the Iterations are completely planned and the team is way too far from formulating the PI Objectives. This happened because the team got into too much detail with the first bunch of Stories they considered.*
- ▶ **Step 2:** The RTE made a clear suggestion that you need to use any tools at your disposal as well as any people in the planning room, but the team must provide a draft plan at the end of the breakout. What would you do next?
- ▶ **Step 3:** Share with the class.

Draft plan review

Plans are peer-reviewed by all teams.



Draft plan review agenda:

1. Capacity and load
2. Draft PI Objectives
3. Program risks and impediments
4. Q&A



Reprinted by permission of TradeStation Technologies

Management review and problem-solving

Management meets to adjust scope and objectives based on Day 1 planning. Some common questions are:

- ▶ Where do we need to adjust Vision? Scope? Resources?
- ▶ Where are the bottlenecks?
- ▶ What Features must be de-scoped?
- ▶ What decisions must we make between now and tomorrow to address these issues?



Activities during Day 2

Day 1		Day 2	
Business context	8:00–9:00	Planning adjustments	8:00–9:00
Product/Solution Vision	9:00–10:30	Team breakouts	9:00–11:00
Architecture Vision and development practices	10:30–11:30	Final plan review and lunch	11:00 –1:00
Planning context and lunch	11:30–1:00	Program risks	1:00–2:00
Team breakouts	1:00–4:00	PI confidence vote	2:00–2:15
Draft plan review	4:00–5:00	Plan rework if necessary	2:15–???
Management review and problem solving	5:00–6:00	Planning retrospective and moving forward	After commitment

4-43

Make planning adjustments

Based on the previous day's management review and problem-solving meeting, adjustments are discussed.

Possible changes:

- ▶ Business priorities
- ▶ Adjustment to plan
- ▶ Changes to scope
- ▶ Movement of resources



Team breakout #2

Based on new knowledge (and a good night's sleep), teams work to create their final plans.

- ▶ In the second team breakout, Business Owners circulate and assign business value to PI Objectives from low (1) to high (10)
- ▶ Teams finalize the Program Increment plan
- ▶ Teams also consolidate program risks, impediments, and dependencies
- ▶ Uncommitted objectives provide the capacity and guard band needed to increase cadence-based delivery reliability

Objectives for PI 1	BV	ABV
1. Show routing calculations between the 5 most frequent destinations	10	
2. Navigate autonomously from distribution center the most frequent destination	8	
3. Parallel park for a delivery	7	
4. Return to distribution center after delivery	10	
5. Include traffic data in route planning	7	
6. Recall a delivery that is already in progress	7	
Uncommitted Objectives		
7. Spike: Reduce GPS signal loss by 25%	2	
8. Demonstrate real-time rerouting to avoid delays (e.g., accident, construction)	5	



Activity: "We just don't see much business value in it..."



- ▶ **Step 1:** In your groups, consider the following scenario:
 - *Your team is at a breakout session on Day 2. Business Owners ranked a PI Objective of "Building batch processing mechanism for indexing" as 2 and requested that you move it to stretch objectives. This function provides a critical architectural enablement to the entire program in this PI. The team is clearly disappointed and concerned that an important technical item is ranked so low. "We just don't see much business value in it," said the VP of Product.*
- ▶ **Step 2:** Considering the role of the Scrum Master, discuss:
 - How would you solve the problem?
 - What tools or techniques would you use in order to come to a solution?
- ▶ **Step 3:** Be prepared to share with the class.

Objectives for PI 1	BV	ABV
1. Show routing calculations between the 5 most frequent destinations	10	
2. Navigate autonomously from distribution center the most frequent destination	8	
3. Parallel park for a delivery	7	
4. Return to distribution center after delivery	10	
5. Include traffic data in route planning	7	
6. Recall a delivery that is already in progress	7	
Uncommitted Objectives		
7. Spike: Reduce GPS signal loss by 25%	2	
8. Demonstrate real-time rerouting to avoid delays (e.g., accident, construction)	5	

Final plan review

Teams and Business Owners peer-review all final plans

Final plan review agenda

1. Changes to capacity and load
2. Final PI Objectives with business value
3. Program risks and impediments
4. Q&A session



Building the final plan

- ▶ Final plans are collected at the front of the room
- ▶ Final plans are reviewed by all teams
- ▶ Business Owners are asked whether they accept the plan
- ▶ If so, the team's plan and program risk sheet are brought to the front of the room
- ▶ If not, the plans stay in place and the team continues planning after the review



Addressing program risks

After all plans have been presented, remaining program risks and impediments are discussed and categorized.

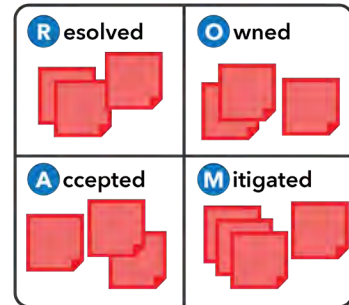
ROAMing risks:

Resolved - Has been addressed. No longer a concern.

Owned - Someone has taken responsibility.

Accepted - Nothing more can be done. If risk occurs, release may be compromised.

Mitigated - Team has plan to adjust as necessary.



Confidence vote: Team and program

After dependencies are resolved and risks are addressed, a confidence vote is taken by the team and program.

A commitment with two parts:

1. Teams agree to do everything in their power to meet the agreed-to objectives
2. In the event that fact patterns dictate that it is simply not achievable, teams agree to escalate immediately so that corrective action can be taken



No confidence



Little confidence



Good confidence



High confidence



Very high confidence

Plan rework if necessary

What happens if there is low confidence? Rework!



No confidence



Little confidence

The PI Planning timebox:

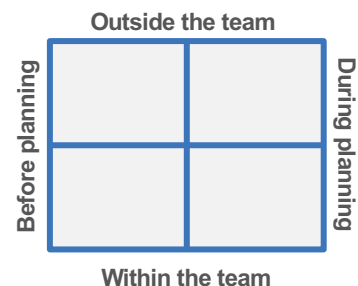
- ▶ Just as the Iteration Planning event is timeboxed, so is the PI Planning event.
- ▶ Leaving the two-day planning event without a committed plan is not an option. Teams stay to rework their plans and 'ROAM' their risks and impediments.



Activity: Being proactive about the confidence vote

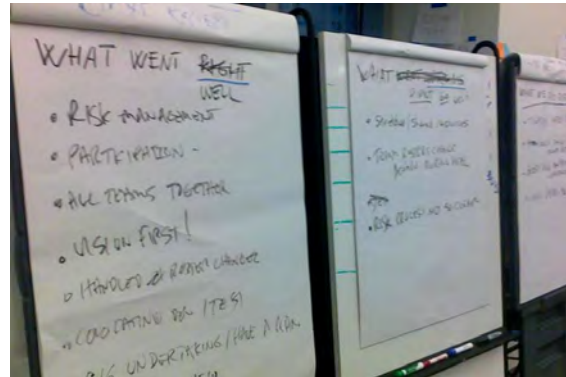
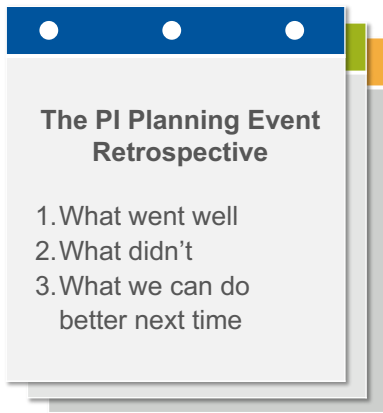


- ▶ **Step 1:** As a group, explore the key factors that impact the team's confidence vote
- ▶ **Step 2:** Create a list of action items that you, as a Scrum Master, would consider to proactively enable a high confidence level on your team
- ▶ **Step 3:** Present the list of actions to the class
 - Hint: Split the sheet into four quadrants and explore action items in each quadrant.



Run a planning event retrospective

The PI Planning event will evolve over time. Ending with a retrospective will help continuously improve it.



Moving forward

The moving forward portion describes what happens after PI Planning ends.

- ▶ Capture objectives and Stories in Agile project management tooling
- ▶ Aggregate Team PI Objectives to Program PI Objectives
- ▶ Set Scrum of Scrum cadence, release management team cadence, System Demo cadence, etc.



Moving forward, (continued)

Additional actions to describe what to do after the PI Planning ends:

- ▶ Refine Program Backlog and prepare for next PI Planning events
- ▶ Summarize changes to engineering practices
- ▶ Clean up PI Planning space (physical and/or digital)



PI Planning resources for Scrum Masters

You'll find the following resources on the SAFe Community Platform:

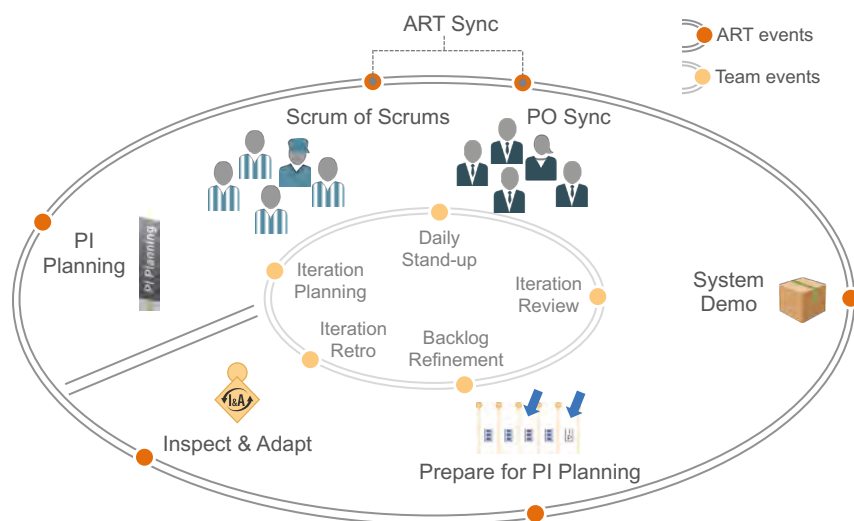
- ▶ PI Planning Toolkit
- ▶ Capacity Allocation spreadsheet
 - Also included in the PI Planning Toolkit
- ▶ Video Playlist: Introduction to PI Planning
- ▶ SAFe Collaborate Template: Being proactive about the confidence vote



4.4 Execute the Program Increment

ART events

ART events create a closed-loop system to keep the train on the tracks.



ART Sync is used to coordinate progress

ARTs coordinate dependencies through sync events.



Scrum of Scrums

- ▶ Visibility into progress and impediments
- ▶ Facilitated by RTE
- ▶ Participants: Scrum Masters, other select team members, SMEs, if necessary
- ▶ Weekly or more frequently, 30–60 minutes
- ▶ Timeboxed and followed by a meet-after

ART Sync

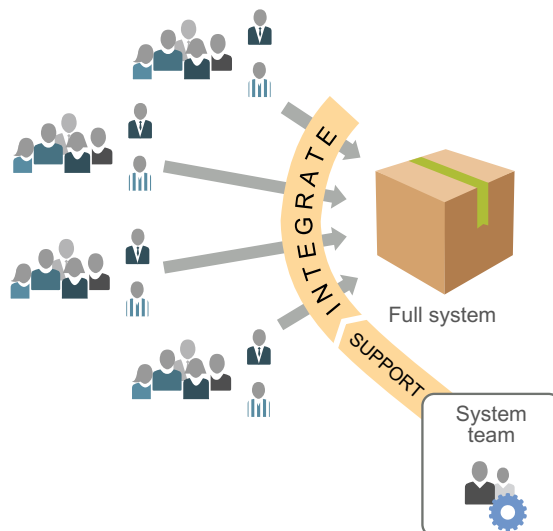


PO Sync

- ▶ Visibility into progress, scope, and priority adjustments
- ▶ Facilitated by RTE or PM
- ▶ Participants: PMs, POs, other stakeholders, and SMEs, as necessary
- ▶ Weekly or more frequently, 30–60 minutes
- ▶ Timeboxed and followed by a meet-after

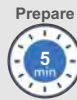
Demo the full system increment every two weeks

- ▶ Features are functionally complete or 'toggled' so as not to disrupt demonstrable functionality
- ▶ New Features work together and with existing functionality
- ▶ Follows the teams' demo (may lag by as much as one iteration, maximum)
- ▶ Demo from a staging environment, resembling production as much as possible





Discussion: "You let us down..."

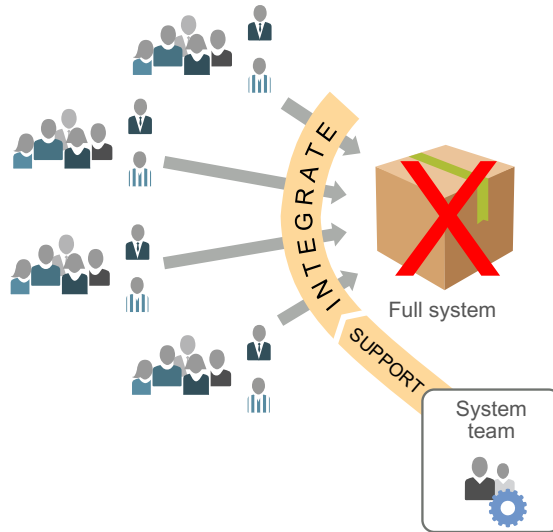


► **Step 1:** Consider the following scenario:

- In SoS, another Scrum Master mentions that at the end of the Iteration, the System Demo did not happen. All teams that had new product functionality merged their changes; your team was the last to merge, and the process didn't go well. You have nothing to show at the demo and other Scrum Masters are looking at you and judging you.

► **Step 2:** Discuss as a group:

- From a Scrum Master perspective, what would you do?



Innovation and Planning (IP) Iteration

Provide sufficient capacity margin to enable cadence. —Donald G. Reinertsen

Facilitate reliability, Program Increment readiness, planning, and innovation

- **Innovation:** Opportunity for innovation, hackathons, and infrastructure improvements
- **Planning:** Provides for cadence-based planning
- Estimating **guard band** for cadence-based delivery

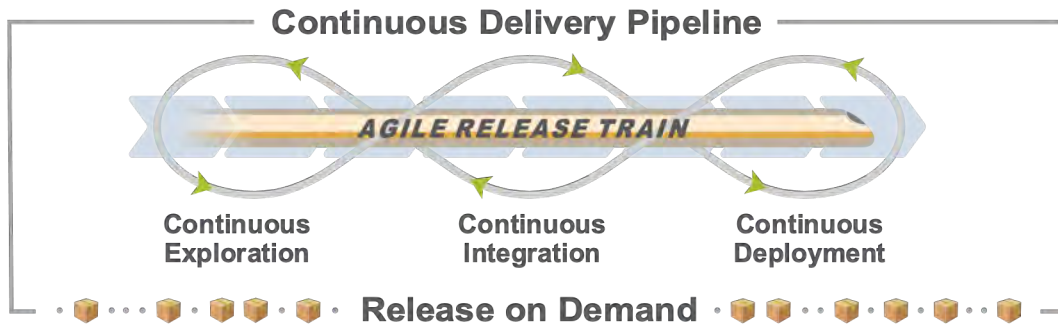


Example IP Iteration calendar

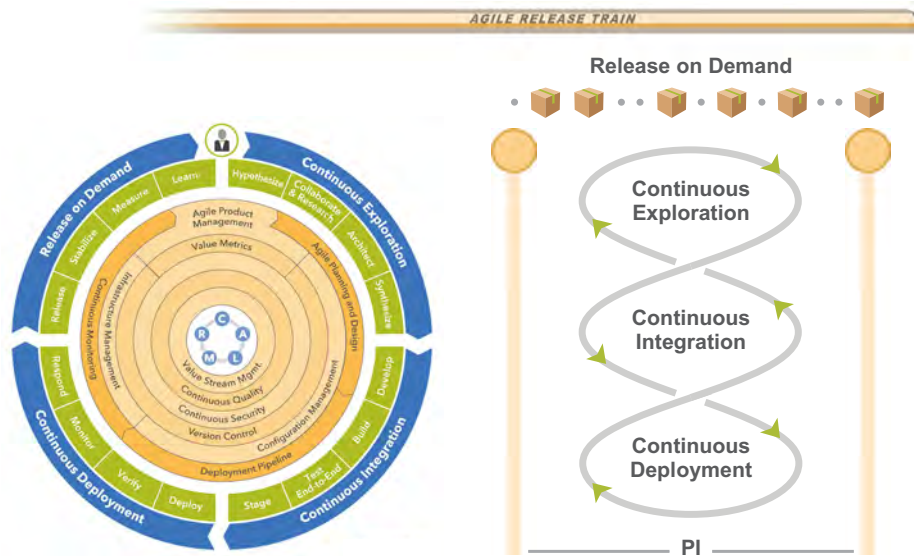
Monday	Tuesday	Wednesday	Thursday	Friday
1	2	3	4	5
Buffer for leftover work				
Final verification and validation, and documentation (if releasing)				
Innovation				
PI planning readiness				
8	9	10	11	12
Innovation continues	Continuing education	PI Planning		Optional time for distributed planning
PI Planning readiness	Inspect and adapt workshop	Business context	Planning adjustments	
		Product/solution vision	Team breakouts	
		Architecture vision and development practices	Final plan review and lunch	
		Planning requirements and lunch	Program risks	
		Team breakouts	PI confidence vote	
		Draft plan review	Plan rework if necessary	
		Management review and problem-solving	Planning retrospective and moving forward	

4.5 Enable teams to release value on demand

ARTs release value on demand



DevOps enables Continuous Delivery





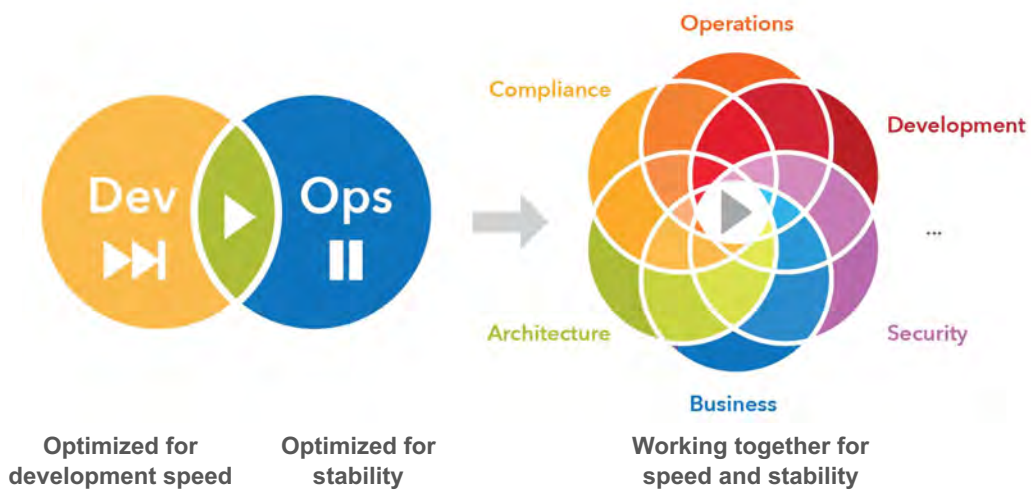
Video: The Continuous Delivery Pipeline

Duration



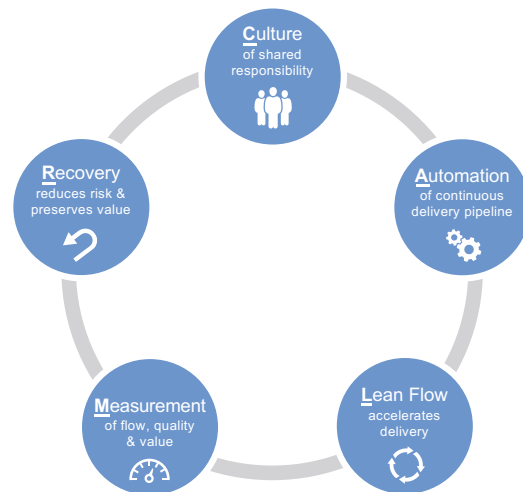
<https://bit.ly/SAFeCDP>

Maximize speed *and* stability



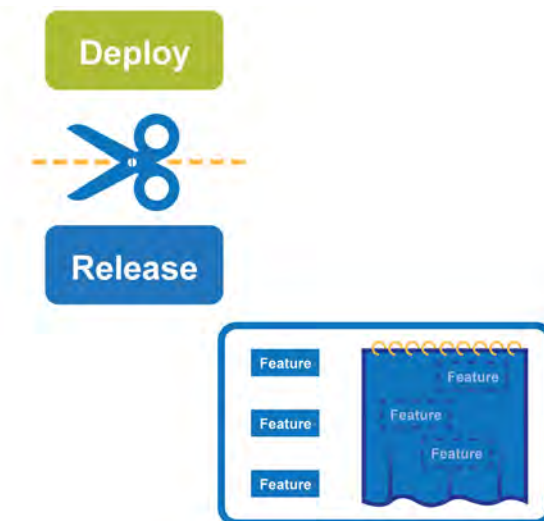
A CALMR approach to DevOps

C	Culture	Establish a culture of shared responsibility for development, deployment, and operations.
A	Automation	Automate the Continuous Delivery Pipeline.
L	Lean flow	Keep batch sizes small, limit WIP, and provide extreme visibility.
M	Measurement	Measure the flow through the pipeline. Implement full-stack telemetry.
R	Recovery	Architect and enable low-risk releases. Establish fast recovery, fast reversion, and fast fix-forward.



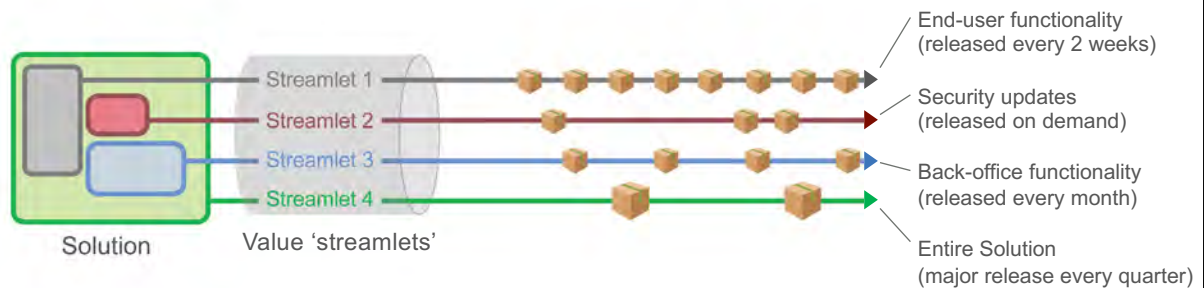
Separate deploy from release

- ▶ Separate deploy to production from release
- ▶ Hide all new functionality under feature toggles
- ▶ Enables testing background and foreground processes in the actual production environment before exposing new functionality to users
- ▶ Timing of the release becomes a business decision



Decouple release elements from the total Solution

- ▶ Different parts of the Solution require different release strategies
- ▶ Architect the Solution to enable the various strategies and to shift them over time based on business demand



Activity: Improving flow

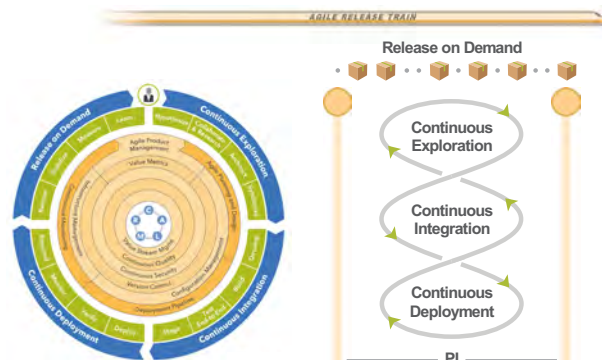
Prepare



Share



- ▶ **Step 1:** Find a partner and think of three ideas that you can implement as a Scrum Master to improve flow through the Continuous Delivery Pipeline and DevOps
- ▶ **Step 2:** Find a new partner and share the three ideas you've had with each other
- ▶ **Step 3:** Prepare to share ideas with the class.



4.6 Prepare for the next PI Planning event

Key stakeholders prepare briefings

In preparation for PI Planning, leadership creates a series of briefings to set context.

- ▶ Executive briefing: State of the business and upcoming objectives
- ▶ Product Vision briefing(s): Vision and top 10 Features
- ▶ Architectural Vision briefing: Vision for architecture, new architectural Epics, common frameworks, and more
- ▶ Development context: Changes to standard practices, new tools and techniques, and more



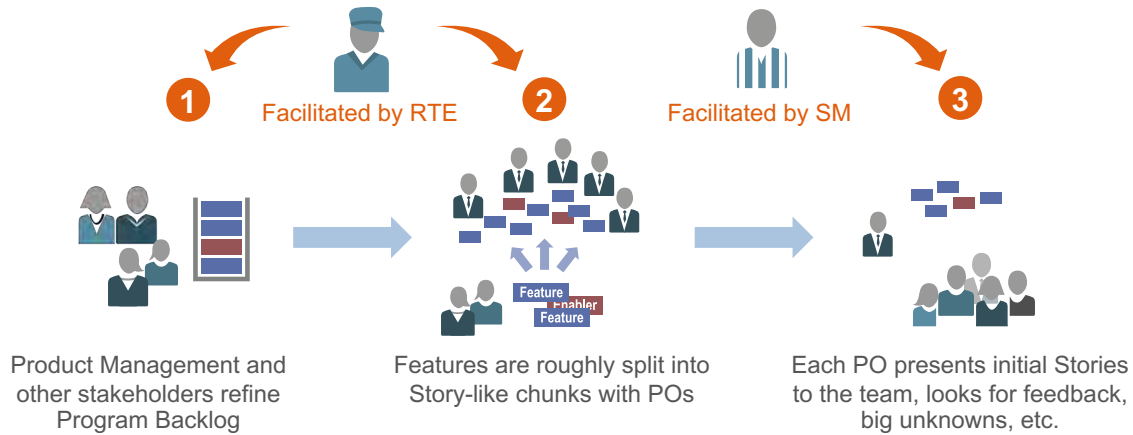
Feature

Enabler

NFRs

New PI content should not be a surprise

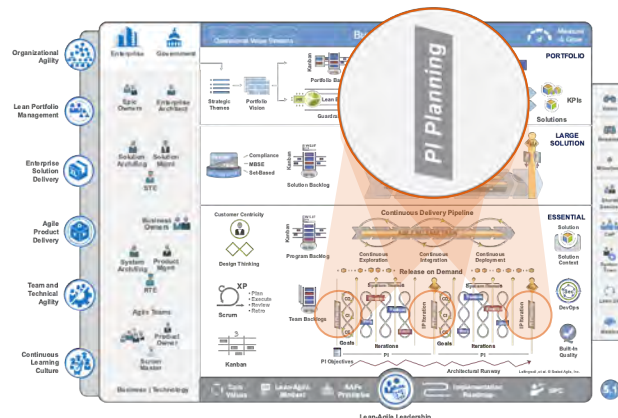
Upfront presentation of content to the teams solves a lot of problems later during PI Planning.



Lesson review

In this lesson you:

- ▶ Executed development with the Agile Release Train
- ▶ Established teams around the flow of value
- ▶ Organized the Program Increment
- ▶ Executed the Program Increment
- ▶ Managed teams to release value on demand
- ▶ Prepared for the next PI Planning



<https://www.scaledagileframework.com/pi-planning/>

Lesson notes



Enter your notes below:

Reminder: If using a digital workbook, save your PDF often so you don't lose any of your notes.

Lesson 5

Improving Flow with Kanban and XP

SAFe® Course - Attending this course gives students access to the SAFe Advanced Scrum Master exam and related preparation materials.



Lesson Topics

- 5.1** Build your Kanban board
- 5.2** Measure and optimize flow
- 5.3** Build quality in
- 5.4** Foster engineering craftsmanship
- 5.5** Facilitate collaboration with Architects, System Team, and Operations



5-2

Learning Objectives

At the end of this lesson, you should be able to:

- ▶ Build a Kanban board for your team
- ▶ Implement metrics to measure and optimize the flow of work on your team
- ▶ Apply activities for the team to build in quality
- ▶ Enable teams for engineering craftsmanship
- ▶ Establish collaboration with Architects, System Team, and Operations

5.1 Build your Kanban board



Video Playlist: Kanban for Teams

Duration



<http://bit.ly/KanbanPlaylist>

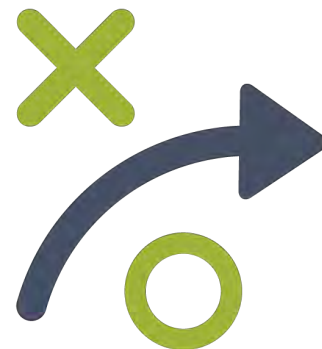
SCALED AGILE © Scaled Agile, Inc.

5-5

Kanban description

Primary aspects for applying Kanban in development:

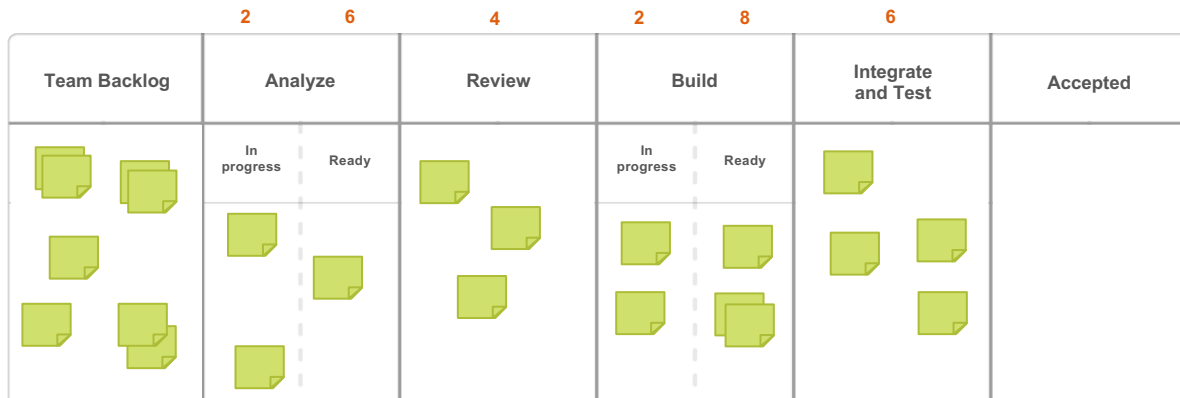
- ▶ The progress of items is tracked by visualizing all work
- ▶ Teams agree on specific WIP limits for each state and change them when necessary to improve flow
- ▶ Policies are adopted to specify the management of work
- ▶ Flow is measured
- ▶ Classes of service are used to prioritize work based on the Cost of Delay (CoD)



SCALED AGILE © Scaled Agile, Inc.

5-6

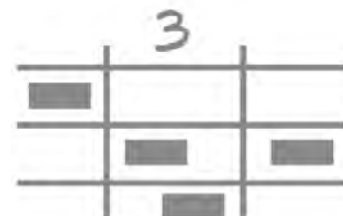
One team's initial Kanban board



Average WIP and duration are measured from the point work is pulled from the backlog until it is accepted.

Applicability

- ▶ Kanban perfectly extends Scrum by providing granular pull mechanisms that drive more effective Iteration execution
- ▶ Kanban connects capacity-based planning in Scrum with a throughput-based approach
- ▶ It helps improve Iteration outcomes
- ▶ It allows better visibility into the progress of work based on the team-specific workflow

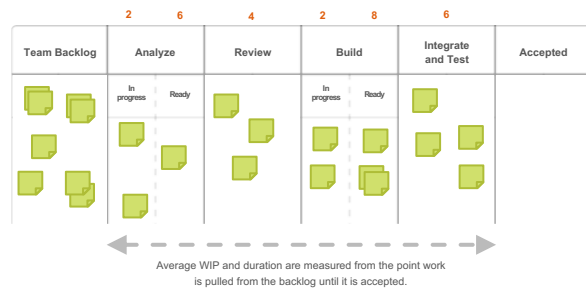




Activity: Build your own Kanban board



- ▶ **Step 1:** Working in your groups, pick one context from any of the team members
- ▶ **Step 2:** Build a team Kanban board using the example previously discussed
- ▶ **Step 3:** Present your board to the class



5.2 Measure and optimize flow



Video Playlist: Kanban - Measuring and Improving

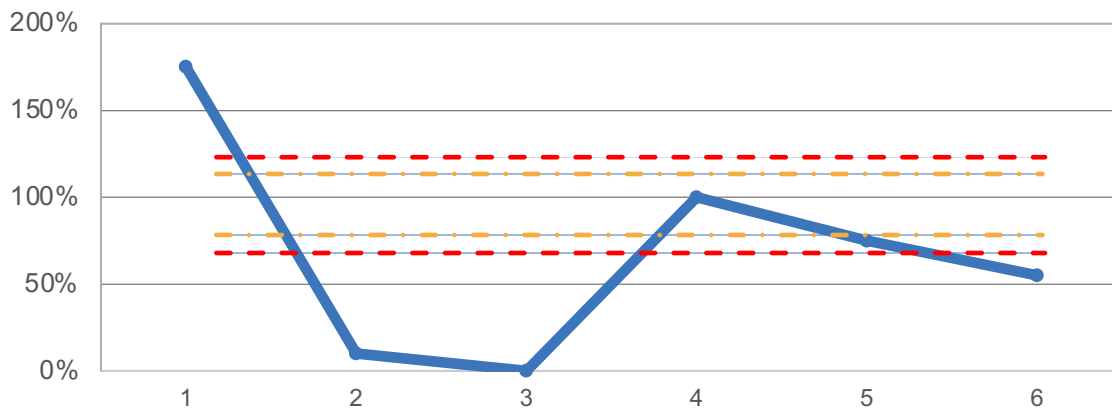
Duration



<http://bit.ly/KanbanPlaylist>

Points accepted vs. points planned

Iteration productivity
Points accepted vs points planned

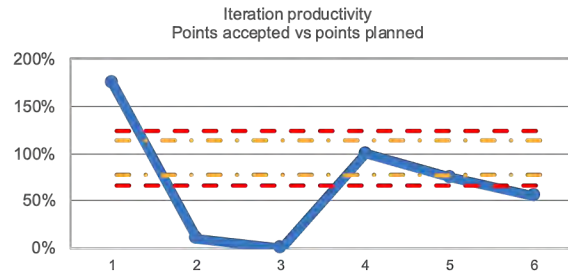




Discussion: Iteration KPIs (key performance indicators)



- ▶ **Step 1:** Working in your groups, discuss what the example chart illustrates about the team's flow and predictability
- ▶ **Step 2:** Share with the class how you, as a Scrum Master, can improve flow and predictability



Iteration performance Metrics

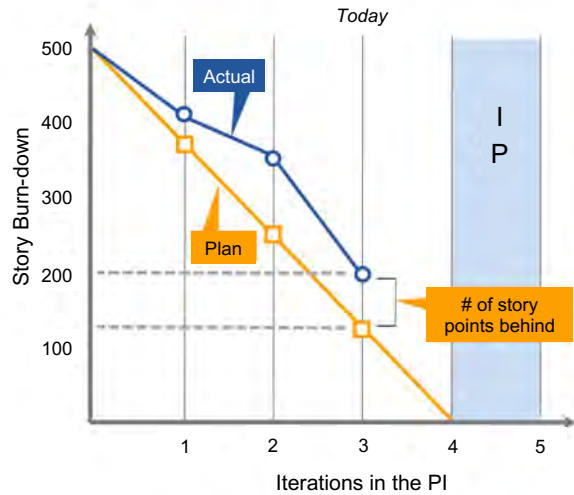
- ▶ **Functionality Metrics** measure things like velocity and throughput
- ▶ **Quality Metrics** measure the ability to build quality into the process

Functionality	Iteration 1	Iteration 2	Iteration 3
# Stories (loaded at beginning of Iteration)	0	0	0
# accepted Stories (defined, built, tested, and accepted)	0	0	0
% accepted	0%	0%	0%
# not accepted (not achieved within the Iteration)	0	0	0
# pushed to next Iteration (rescheduled in next Iteration)	0	0	0
# not accepted: deferred to later date	0	0	0
# not accepted: deleted from backlog	0	0	0
# added (during Iteration; should typically be 0)	0	0	0
Quality			
% Story Complete with test available/test automated	0%	0%	0%
Defect count at start of Iteration	0	0	0
Defect count at end of Iteration	0	0	0
# new test cases	0	0	0
# new test cases automated	0	0	0
# new manual test cases	0	0	0
Total automated tests	0	0	0
Total manual tests	0	0	0
% tests automated	0%	0%	0%
Unit test coverage percentage	0	0	0

Program execution Metrics: PI burn-down chart

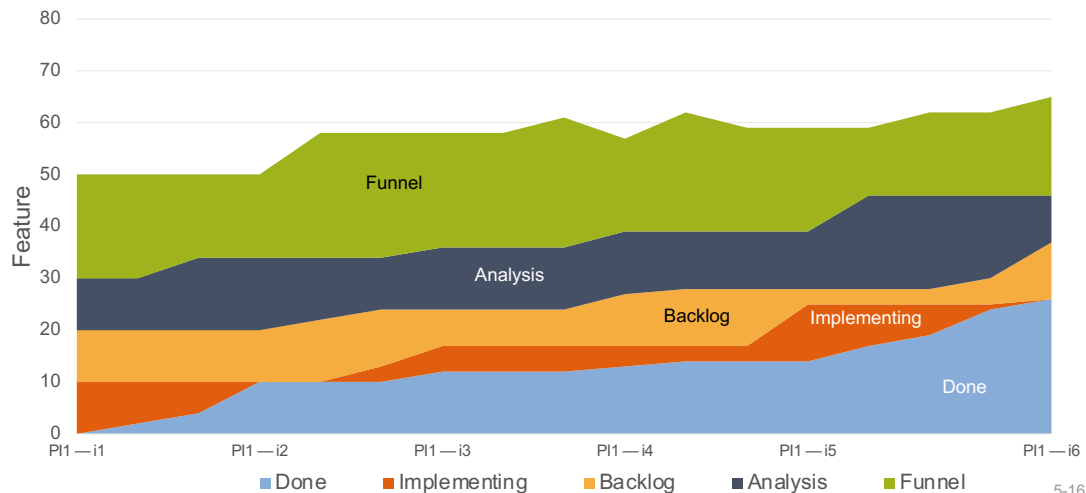
The PI burn-down chart shows the progress being made toward the Program Increment timebox.

- ▶ The horizontal axis of the PI burn-down chart shows the Iterations within the PI
- ▶ The vertical axis shows the aggregated amount of work (Story points) remaining at the start of each Iteration for the ART
- ▶ Iteration boundaries provide the most meaning



Program execution Metrics: Cumulative flow diagram (CFD)

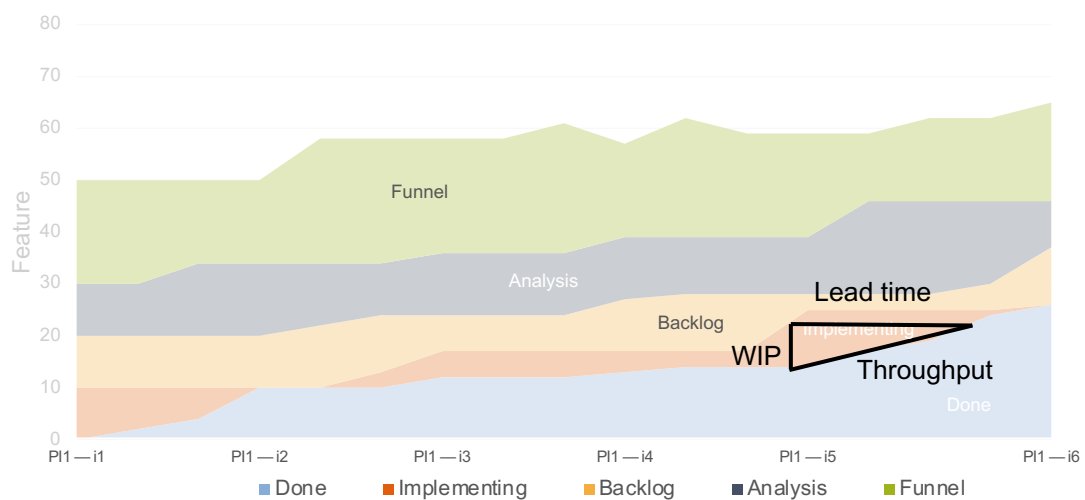
The CFD is made up of a series of lines or areas representing the amount of work in the various Kanban states.



Typical program measures in a CFD

- ▶ **Lead time** - The time a backlog item spends in the system after it has been pulled from the backlog and before it is accepted
- ▶ **WIP in the system** - The number of backlog items currently in process (all items between funnel and done)
- ▶ **Throughput** - The number of items that can be finished per unit of time

Reading cumulative flow diagrams



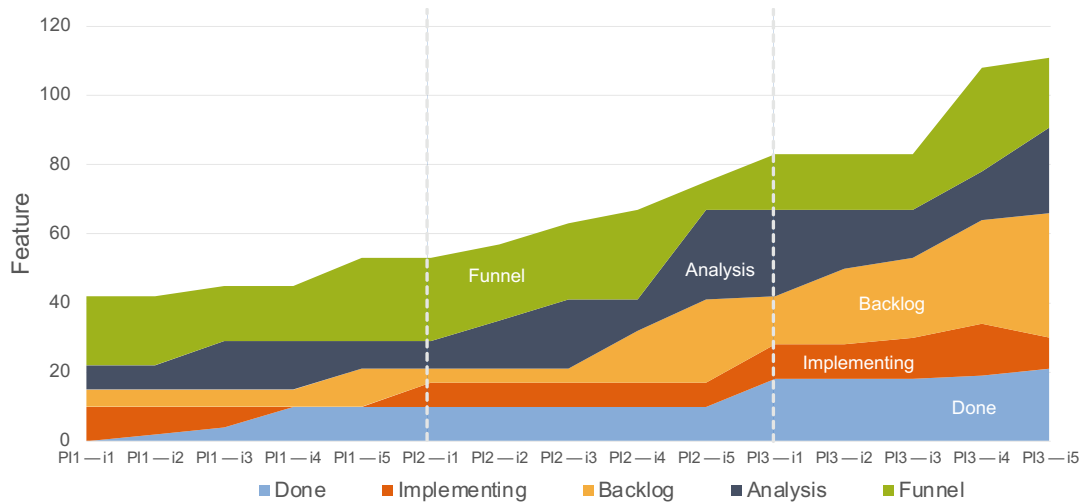


Activity: Reading cumulative flow diagrams

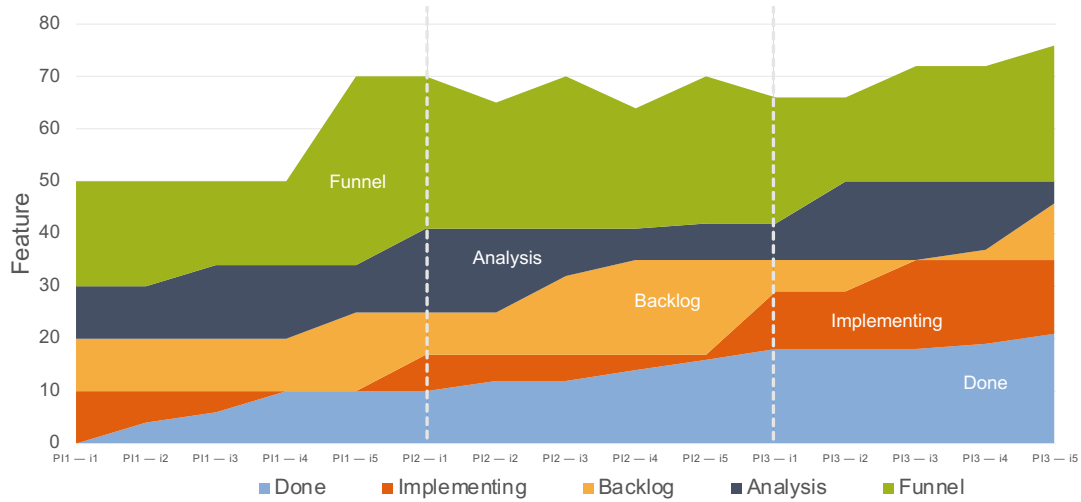


- ▶ **Step 1:** Working in your groups, review the two CFD example charts. Capture answers to the following questions:
 - What problems do you see?
 - How do you know these are problems?
- ▶ **Step 2:** Be prepared to share with the class.

Cumulative Flow Diagram – Example 1



Cumulative Flow Diagram – Example 2



Classes of service to adjust flow

- ▶ **Standard** - Operate normally. Adhere to WIP limits.
- ▶ **Fixed Date** - Adhere to WIP limits. Must be pulled from the backlog early enough.
- ▶ **Expedite** - Can violate WIP limits. No more than one item at a time.





Discussion: Classes of service



- ▶ **Step 1:** Provide examples where the three classes of service would apply in your context
- ▶ **Step 2:** Discuss what the potential sources of 'Fixed Date' and 'Expedite' items could be
- ▶ **Step 3:** Be prepared to share with the class

Standard

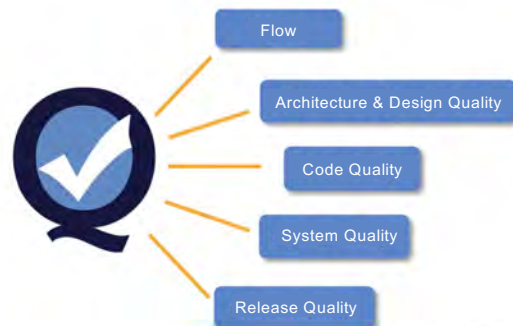
Fixed date

Expedite

5.3 Build quality in

Built-in Quality

- ▶ Ensures that every increment of the Solution reflects quality standards
- ▶ Is required for sustainably high development velocity
- ▶ Includes Continuous Integration, test-first, refactoring, pair work, collective ownership, and more (for software quality practices mostly inspired by XP)
- ▶ Is supported in hardware by early exploratory iterations, frequent system-level integration, design verification, MBSE, and set-based design



© Scaled Agile, Inc.

Emergent design and intentional architecture

Every team deserves to see the bigger picture.
Every team is empowered to design its part.

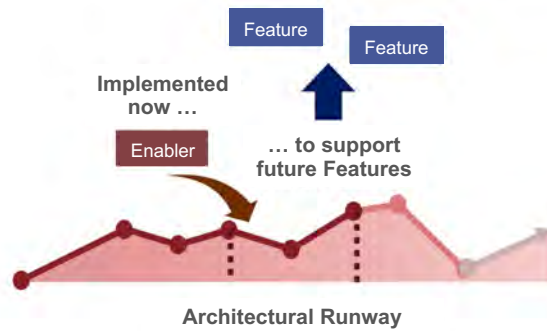
- ▶ **Emergent design** - Teams grow the system design as User Stories require
- ▶ **Intentional architecture** - Fosters team alignment and defines the Architectural Runway



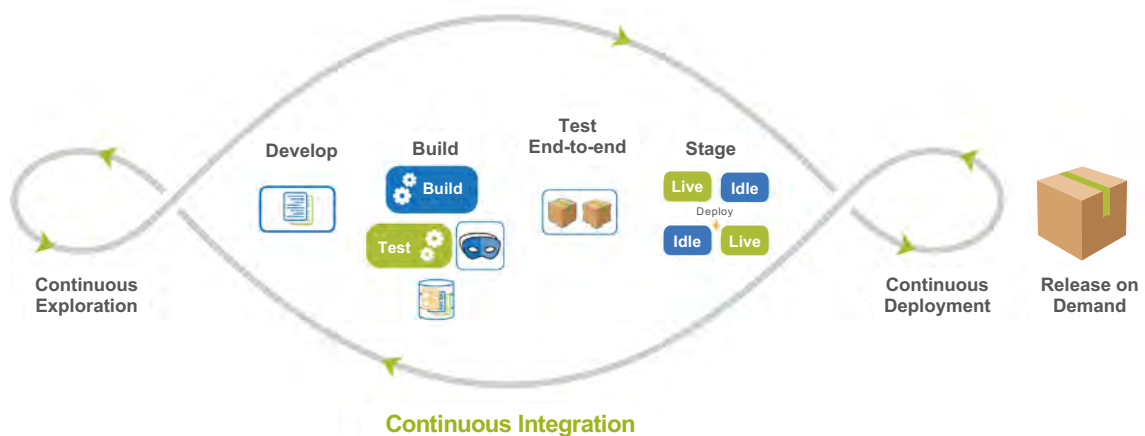
A balance between emergent design and intentional architecture is required for speed of development and maintainability.

Architectural Runway

- ▶ Contains existing code, components, and technical infrastructure needed to implement near-term Features without excessive redesign and delay
- ▶ Supports the continuous flow of value through the Continuous Delivery Pipeline



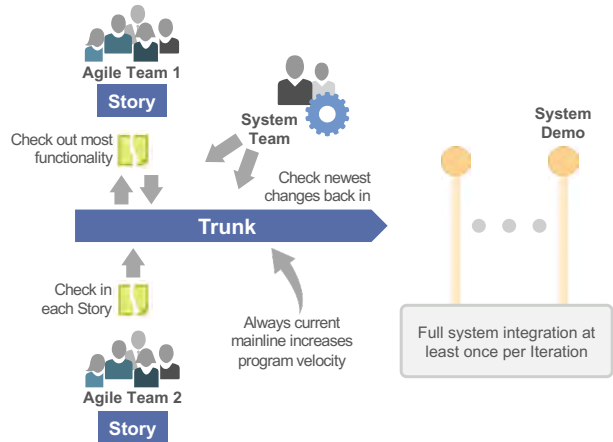
Continuous code integration



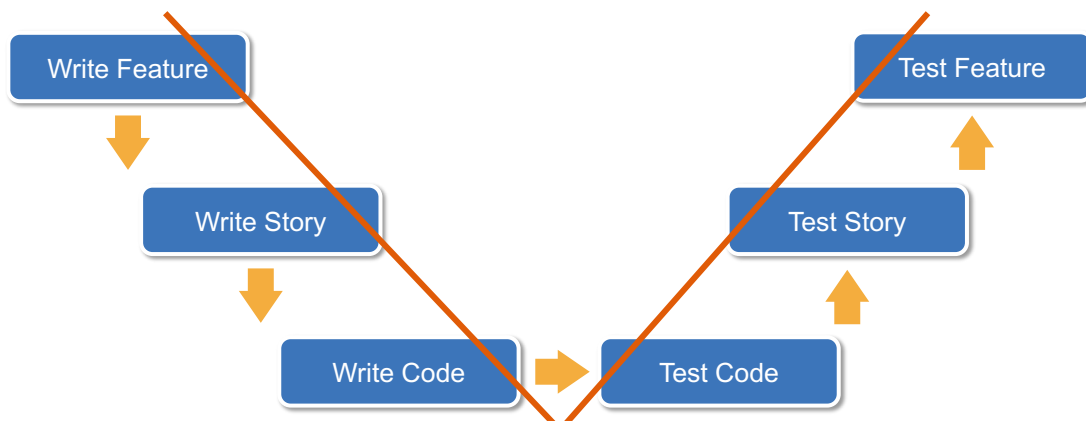
Continuous system integration

Teams continuously integrate assets, leaving as little as possible to the System Team.

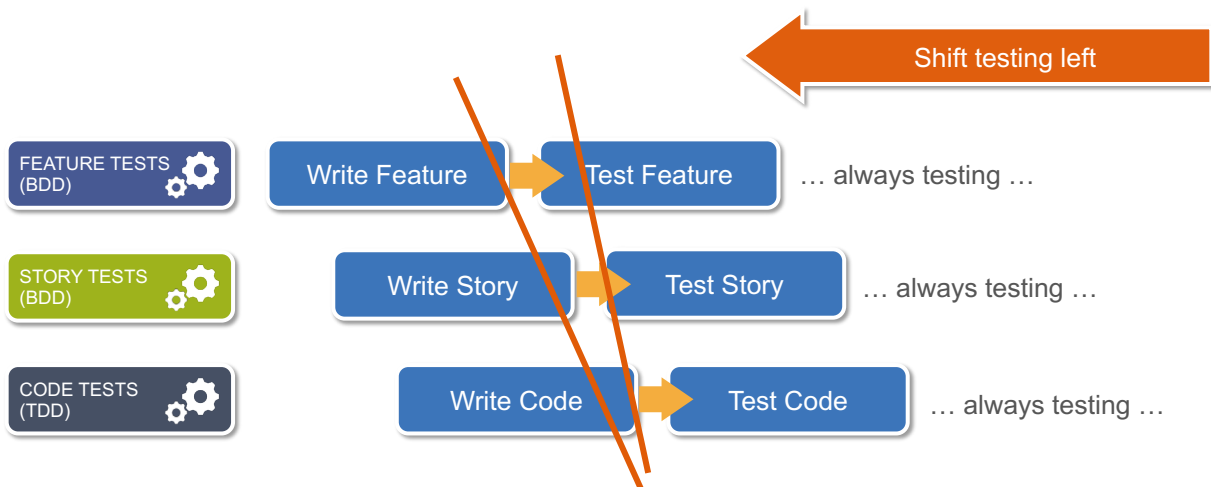
- ▶ Integrate every vertical slice of a User Story
- ▶ Avoid physical branching for software
- ▶ Frequently integrate hardware branches
- ▶ Use development by intention in case of inter-team dependencies
 - Define interfaces and integrate first; then add functionality



Traditional testing (V-Model) delays feedback



Shift testing left for fast and continuous feedback



Activity: Integration and ART velocity (part 1)



- ▶ **Step 1:** Each group is assigned **A** or **B**
- ▶ **Step 2:** Each group **A** is matched with a partner **B** group
- ▶ **Step 3:** Each group will build its own component, as shown in the picture.
- ▶ **Note:** The partnered teams **cannot** communicate with each other





Activity: Integration and ART velocity (part 2)

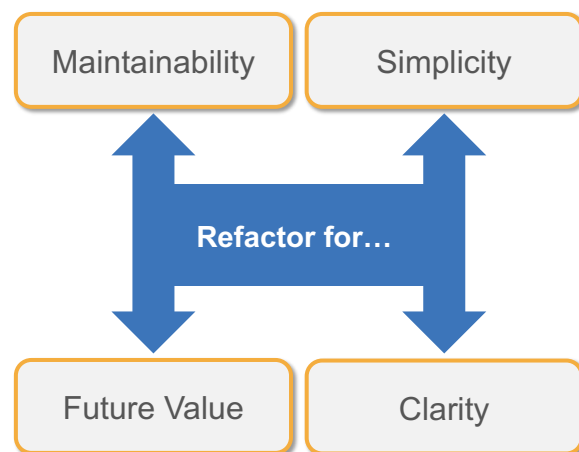


- ▶ **Step 4:** Once your group is done building your components, put them together for every pair of groups.
 - What do the results look like?
 - Is any rework required?
- ▶ **Step 5:** Based on what we've just seen, what impact does late cross-team integration have on the program's velocity?

Refactoring

Refactoring allows teams to maintain high velocity.

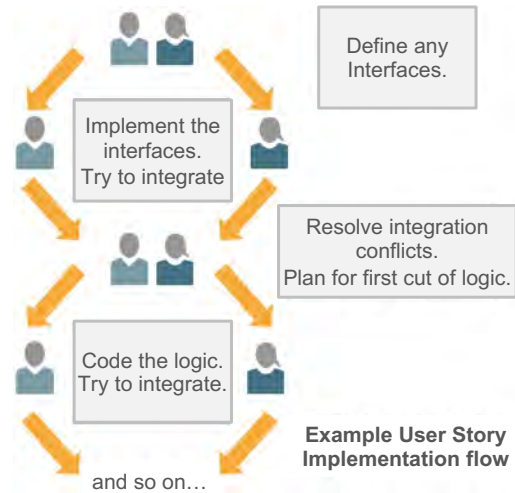
- ▶ It is impossible to predict requirements or design in detail
- ▶ Refactoring allows teams to quickly correct the course of action
- ▶ Emergent design is impossible without continuous refactoring
- ▶ Most User Stories will include some refactoring effort
- ▶ If technical debt is big—teams track and implement as separate backlog items—then it's time to refactor



Pair work

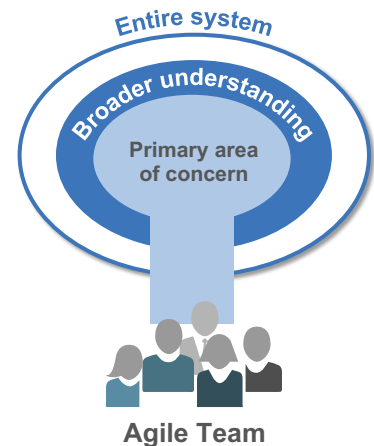
Improves system quality, design decisions, knowledge sharing, and team velocity

- ▶ Broader and less constraining than pair programming
- ▶ Collaborative effort of any two team members: dev/dev, dev/PO, dev/tester, etc.
- ▶ Team members spend 20% to 80% of their time pairing
- ▶ Pairs should be spontaneous and purposefully rotate over time



Collective ownership

- ▶ Addresses bottlenecks, increases velocity, and encourages shared contribution
- ▶ Fosters Feature orientation
- ▶ Supports the ART by using:
 - Design simplicity
 - Communities of practice
 - Pair work
 - Joint specification and design workshops
 - Frequent integration of the entire system
 - Standards
- ▶ Facilitates shared understanding of system behavior by using Collective test ownership

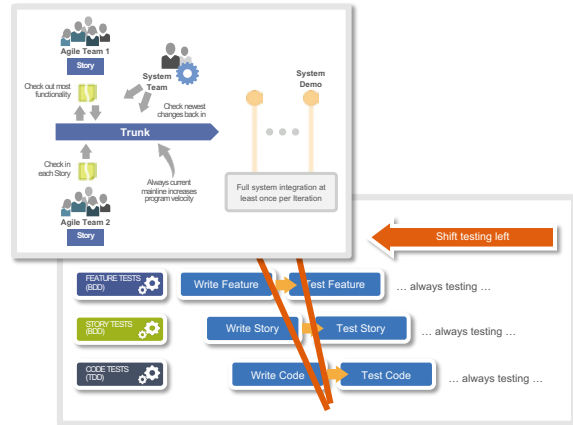




Activity: Facilitate the adoption of integration and testing



- ▶ **Step 1:** Identify current problems in your group's experience with integration and test automation
- ▶ **Step 2:** Build a realistic plan for enhancing your team's integration and testing ability
- ▶ **Step 3:** Discuss how this plan connects with the team's definition of done



5.4 Foster engineering craftsmanship

Foster adoption of technical practices

A Scrum Master facilitates the adoption of technical practices through the following actions:

- ▶ Helps the team refine the definition of done
- ▶ Creates transparency and urgency around continuous system integration
- ▶ Encourages small, automated acceptance tests at the beginning and evolves from there
- ▶ Encourages team members to coach each other in test-driven design (TDD), behavior-driven design (BDD), refactoring
- ▶ Helps the team adopt a 'thinking backward' approach by asking what is the expected behavior of the functionality that we are about to code?
- ▶ Helps facilitate the power of human-readable acceptance tests
- ▶ Encourages pairing and peer review



Encourage learning

Scrum Masters create an environment for continuous learning



Encourage learning: Inside-Outs

Team Inside-Outs	A team member prepares a short presentation or flip chart talk for their team.	Frequency: Once every 1–2 Iterations
		Duration: 30–60 minutes
		Example: We will soon start using Hibernate for data persistence. John has experience and is willing to share his knowledge.
Your role	Help kick-start the first 2–3 Inside-Outs and help participants prepare	
	Maintain the Inside-Out schedule	
	Invite shared resources (System Architect, User Experience, infrastructure, etc.) or people from other teams to discuss useful topics	

Encourage learning: Book and Coffee Breaks

Book and Coffee Breaks (BCBs)	A normal coffee break with 3–4 people discussing a book on a new technology, practice, or domain topic that the team is trying to master.	Frequency: 3–4 times per Iteration
		Duration: 30–60 minutes
		Example: The team is about to build its first crawler and Andrew reads them some excerpts from Soumen Chakrabarti's book <i>Mining the Web</i>
Your role	Lead a few BCBs and acquaint people with the format	

Encourage learning: Coding Dojo

Coding (and Testing) Dojo	A session where developers and/or automated test engineers gather to discuss programming and testing challenges. One or two people sit at the computer and project onto a screen. As they code, people comment out loud. After 5–8 minutes, people rotate.	Frequency: Once every 1–2 Iterations
		Duration: 60–90 minutes
Your role	Arrange facilities and equipment	
	Help brainstorm fun, challenging exercises (could be a spike, a script for retrieving data, or even code in one of the main modules)	
	Similarly, testers will enjoy learning how to write test scripts	

Encourage learning: Communities of Practice

Communities of Practice (CoPs)	Communities of practice are self-organizing groups that form to discuss new topics, challenges and best practices.	Frequency: Once every 1–2 Iterations
		Duration: 30–60 minutes
		Format: Any of the formats previously discussed (Inside-Out, BCB, Dojo)
		Example: An automated testing CoP gathers to attend Ivan's presentation on creating FIT tests for complex branching scenarios.
Your role	Work with other Scrum Masters and the Release Train Engineer to create and maintain the CoPs	
	Unite people from different teams in the program around the same process objectives or activities like unit testing, automated acceptance testing, or system design	

5.5 Facilitate collaboration with Architects, System Team, and Operations

System Architects, Team, Operations



System Architects – Provide architectural guidance to teams, collaborate on new technical research, address technical questions from team members



System Team – Assist the ART with frequent system integration and testing and development of infrastructure support



Operations – Enable the Continuous Delivery Pipeline through infrastructure and process support



Discussion: Collaboration with special teams



- ▶ **Step 1:** Consider one of the following events:
 - PI Planning
 - Iteration execution events (DSU, Iteration Planning, Iteration Review, Iteration Retro)
 - Inspect & Adapt
 - System Demo
- ▶ **Step 2:** Determine what kind of collaboration with Architects, System Team, and Operations would be useful in your context. Pro tip: Be specific, for example, will you need to meet regularly for release updates?
- ▶ **Step 3:** Be prepared to share with the class



Available Platform Resources:

- Distributed I&A Problem-Solving Workshop Checklist
- PI Planning Facilitator Guide
- Daily Stand-Up Facilitator Checklist
- Iteration Planning Facilitator Checklist
- Iteration Retrospective Facilitator Checklist
- Iteration Review Facilitator Checklist
- Backlog Refinement Facilitator Checklist

SAFe resources for Scrum Masters

Additional guides and checklists on the SAFe Community Platform to support flow, Metrics, and events:

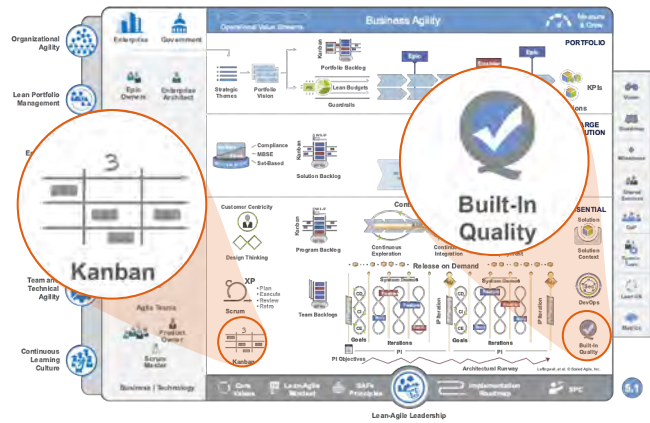
- ▶ Iteration Event Facilitator Checklists
- ▶ Iteration performance Metrics Spreadsheet
 - Also available in the Iteration Execution Toolkit
- ▶ Video Playlist: Kanban for Teams
- ▶ SAFe Collaborate Template: Build your own Kanban board



Lesson review

In this lesson you:

- ▶ Built your Kanban board
- ▶ Measured and optimized flow
- ▶ Explored building quality in
- ▶ Discussed fostering engineering craftsmanship
- ▶ Facilitated collaboration with Architects, System Team, and Operations



<http://www.scaledagileframework.com/team-kanban/>

<http://www.scaledagileframework.com/built-in-quality/>

Lesson notes



Enter your notes below:

Reminder: If using a digital workbook, save your PDF often so you don't lose any of your notes.

Lesson 6

Building High-Performing Teams

SAFe® Course - Attending this course gives students access to the SAFe Advanced Scrum Master exam and related preparation materials.



Lesson Topics

- 6.1** Foster collaboration on the team
- 6.2** Facilitate cross-team collaboration
- 6.3** Build trust with stakeholders
- 6.4** Develop team skill sets
- 6.5** Build an improvement Roadmap



6-2

Learning Objectives

At the end of this lesson, you should be able to:

- ▶ Develop practices to foster collaboration on the team
- ▶ Apply practices to facilitate cross-team collaboration
- ▶ Establish practices to build trust with stakeholders
- ▶ Develop T-shaped team skills
- ▶ Build an improvement Roadmap

6.1 Foster collaboration on the team

Collaboration

Simply following Scrum (or Lean-Agile) processes doesn't make an Agile Team a team.

Poor collaboration often leads to:

- ▶ Low velocity
- ▶ Poor product quality
- ▶ Low morale, low engagement, lack of commitment, poor working environment, and lack of trust
- ▶ Missed commitments and poor results

"Rather than moving in defined, highly structured stages, the real process is born out of the team members' interplay."

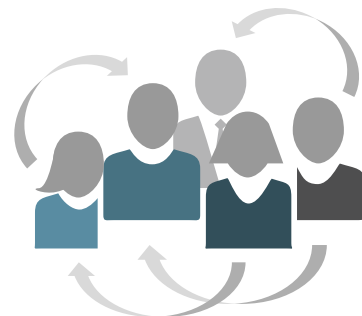
— Hirotaka Takeuchi and Ikujiro Nonaka

Fostering collaboration

Fostering collaboration is one of the most important tasks of a Scrum Master.

Weak collaboration often exists:

- ▶ Between developers and testers (late testing, poor quality, low velocity)
- ▶ Among developers (technical debt, poor knowledge sharing, too much WIP)
- ▶ Between the PO and the rest of the team (unnecessary rework due to misunderstood acceptance criteria, low velocity)
- ▶ With other teams (uncontrolled dependencies, sense of false progress)



Identify team member responsibilities

- ▶ It can be useful both for the team themselves, but also for other teams, to understand the responsibilities of the individual team members.
- ▶ Primary responsibilities are typically those things where you are directly responsible for the outcome.
- ▶ Secondary responsibilities tend to be those where you are contributing your expertise and time.

Team Member	Role	Primary Responsibilities	Secondary Responsibilities

This activity can be found in the Team Formation Toolkit



Video: You Can't Force the Mandate

Duration





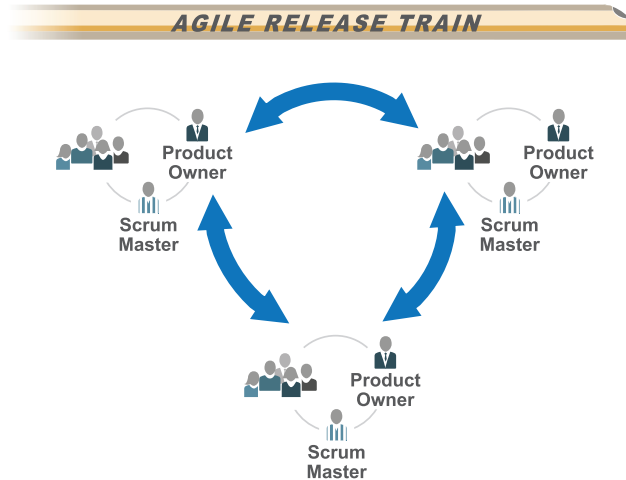
- ▶ **Step 1:** Consider your actual team environment:
 - What examples of collaboration would be helpful in your environment beyond basic Scrum events?
 - What problems would it solve?



6.2 Facilitate cross-team collaboration

Collaboration across teams is key

Agile Release Trains are built with a goal to foster team alignment and collaboration.



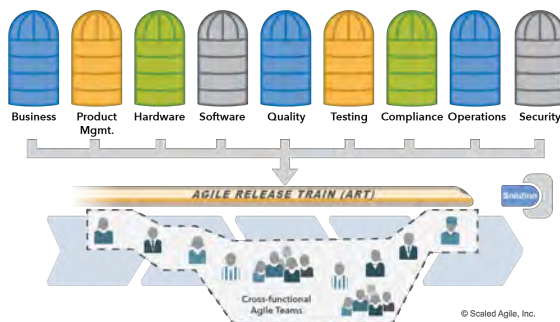
Define key interactions with other teams

Successful collaboration is key to ART success.

Three common interaction modes are:

- **Collaboration:** Working together with other teams for a prolonged period.
- **'As-a-service':** Providing or receiving a service to/from another team.
- **Facilitation:** Helping and mentoring another team

(Team Topologies, Skelton & Pais)



The ART is a team of Agile teams



Activity: Cross-Team Collaboration



- ▶ **Step 1:** Consider the following scenario:
 - *PI Planning resulted in quite a few dependencies with other teams, some of which must be fulfilled in the same iteration. One dependency occurs in Iteration 2. As Iteration 2 approaches, the team feels increasingly uncomfortable about being able to accomplish this critical dependency, because another team must provide their own part of the functionality first.*
- ▶ **Step 2:** Working in your groups, brainstorm possible solutions to the problem.
- ▶ **Step 3:** Capture at least two solutions. For each solution, identify potential advantages or disadvantages.
- ▶ **Step 4:** Be prepared to share with the class.

6.3 Build trust with stakeholders



Discussion: Building trust



► **Step 1:** Consider the scenario:

- *Agile doesn't work without trust. However, you haven't been able to establish a relationship of trust with Product Management. They require frequent status reports about your team's current Stories. The relationship is clearly broken.*

► **Step 2:** Discuss:

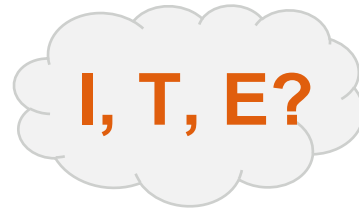
- What specific activities in SAFe do you think would help you establish trust with these stakeholders?
- What adjustments (if any) would you make to leverage those activities to build maximum trust?

6.4 Develop team skill sets

Help team members develop new skills

Narrow specialization of skills on the team is not supportive of any fluctuations in flow.

- ▶ Consider moving from an **I**-shaped skill set model to a **T**, or even an **E**-shaped skill set
- ▶ **T-shaped example:** A Java developer can do a bit of database development, a bit of configuration management, and has rudimentary knowledge in building webpages
- ▶ **E-shaped example:** A Python developer, who also knows Java very well, has deep knowledge of SQL and databases



Discussion: Developing skills

Prepare



Share



- ▶ **Step 1:** Working in your groups, discuss the following:
 - What opportunities for building T-shaped skill sets would you consider?
 - How would you achieve that?
 - Would you take on building E-shaped skill sets or is that too much to accomplish?
- ▶ **Step 2:** Be prepared to share with the class.

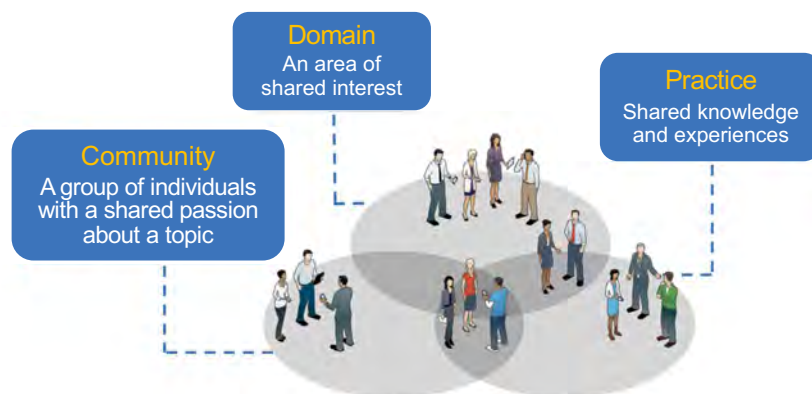


6.5 Build an improvement Roadmap

What are Communities of Practice (CoP)?

“Communities of practice are groups of people who share a common concern or a passion for something they do and learn how to do it better as they interact regularly.”

—Étienne Wenger, Communities of Practice: Learning, Meaning and Identity



Short-term benefits of CoPs

Benefits to the organization



Improves business outcomes

- Arena for problem-solving
- Quick answers to questions
- Reduced time and costs
- Improved quality of decisions
- More perspectives on problems
- Coordination/synergy across units

Benefits to community members



Improves experience of work

- Help with challenges
- Access to expertise
- Improved contribution to the team
- Increased confidence in approach
- Fun of being with colleagues
- More meaningful participation
- Sense of belonging

Longer-term benefits of CoPs

Benefits to the organization



Develops organizational Capabilities

- Be able to execute a strategic plan
- Gain credibility with clients
- Increase retention of talent
- Exploit unplanned Capabilities
- Enable competitive benchmarking
- Leverage advances in technology
- Harness the power of social networks

Benefits to community members



Fosters professional development

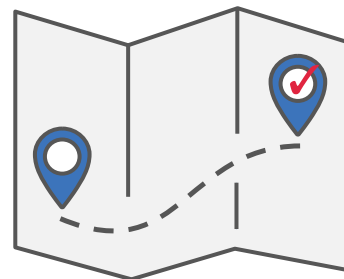
- Forum for expanding skills/expertise
- Network for staying current
- Enhanced professional reputation
- Increased marketability
- Strong sense of professional identity



Activity: Improvement Roadmap



- ▶ **Step 1:** Build an improvement Roadmap for your team for the next PI
- ▶ **Step 2:** Share with the class:
 - What new practices would you adopt and advance?
 - How could a Community of Practice (CoP) help?
 - How does your Roadmap relate to the team's definition of done?



Improvement Roadmap

Build a high-performing team

Use the following resources from the SAFe Community Platform to build your high-performing team:

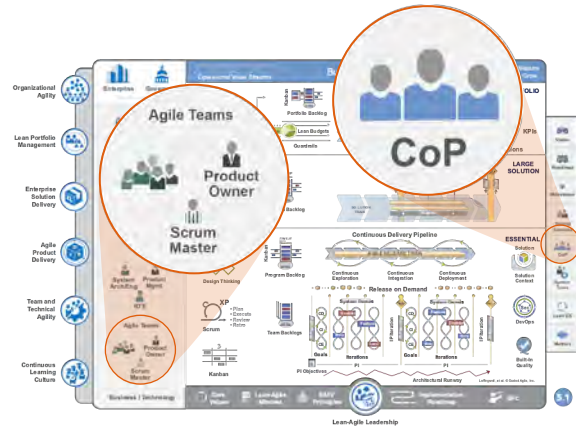
- ▶ Team Formation Toolkit
- ▶ Agile Team Charter Template (also included in the Team Formation Toolkit)
- ▶ SAFe Collaborate Template: Team Member Responsibilities
- ▶ SAFe Collaborate Template: Identify Key Stakeholders



Lesson review

In this lesson you:

- ▶ Developed practices to foster collaboration on the team
- ▶ Explored applying practices to facilitate cross-team collaboration
- ▶ Established practices to build trust with stakeholders
- ▶ Developed T-shaped team skills
- ▶ Built an improvement Roadmap



<https://www.scaledagileframework.com/agile-teams/>
<https://www.scaledagileframework.com/communities-of-practice/>

Lesson notes



Enter your notes below:

Reminder: If using a digital workbook, save your PDF often so you don't lose any of your notes.

Lesson 7

Improving Program Performance

SAFe® Course - Attending this course gives students access to the SAFe Advanced Scrum Master exam and related preparation materials.



Lesson Topics

7.1 Explore the Inspect and Adapt process

7.2 Apply problem-solving workshop



7-2

Learning Objectives

At the end of this lesson, you should be able to:

- ▶ Practice the Inspect and Adapt process
- ▶ Conduct the problem-solving workshop

7.1 Explore the Inspect and Adapt process

Inspect and Adapt event: Overview

- ▶ Three parts of Inspect and Adapt (I&A):

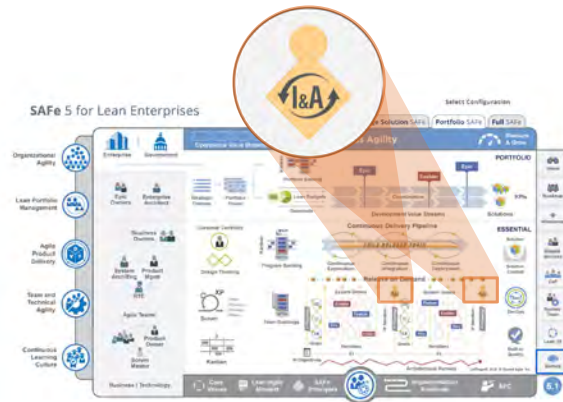
1. The PI System Demo

2. Quantitative and Qualitative Measurement

3. Problem-Solving Workshop

- ▶ **Timebox:** 3 – 4 hours per PI

- ▶ **Attendees:** Teams and stakeholders



<https://www.scaledagileframework.com/inspect-and-adapt/>

PI System Demo

At the end of the PI, teams demonstrate the current state of the Solution to the appropriate stakeholders.

- ▶ Often led by Product Management, Product Owners, and the System Team
- ▶ Attended by Business Owners, program stakeholders, the RTE, Scrum Masters, and Agile Teams
- ▶ Suggested timebox: 45–60 minutes



Team performance assessment

- ▶ During the IP Iteration, the PI Objectives for all teams are assigned an actual business value from 1 to 10.
- ▶ Review and rate your PI achievements:
 - How well did you do against your stated objectives, including timeliness, content, and quality?
 - Rate on a scale of 1 to 10, 10 being max total business value.
- ▶ The RTE aggregates and averages the scores across all objectives for a program percent achievement score
- ▶ Suggested timebox: 45–60 minutes

Team PI Performance Report

Objectives for PI 3	BV	ABV
1. Structured locations and validation of locations	7	7
2. Build and demonstrate a proof of concept for context images	8	8
3. Implement negative triangulation by: tags, companies and people	8	6
4. Speed up indexing by 50%	10	5
5. Index 1.2 billion more web pages	10	8
Uncommitted Objectives		
6. Fuzzy search by full name	7	0
7. Improve tag quality to 80% relevance	4	4
Totals:		
% Achievement: 90%	43	38

Team Predictability Measure



7-7

Team PI performance report

Team performance is based on the actual business value assignments provided by the Business Owners.

- ▶ Planned total does not include uncommitted objectives
- ▶ Actual total includes uncommitted objectives
- ▶ Percent achievement equals actual total/planned total
- ▶ A team can achieve greater than 100% (as a result of uncommitted objectives achieved)
- ▶ Effort required for uncommitted objectives is included in the load (i.e., not extra work the team does on weekends)
- ▶ Individual team totals are rolled up into the program predictability report

Objectives for PI 3	BV	ABV
1. Structured locations and validation of locations	7	7
2. Build and demonstrate a proof of concept for context images	8	8
3. Implement negative triangulation by: tags, companies and people	8	6
4. Speed up indexing by 50%	10	5
5. Index 1.2 billion more web pages	10	8
Uncommitted Objectives		
6. Fuzzy search by full name	7	0
7. Improve tag quality to 80% relevance	4	4
Totals:		
% Achievement: 90%	43	38

7-8

Program performance metrics

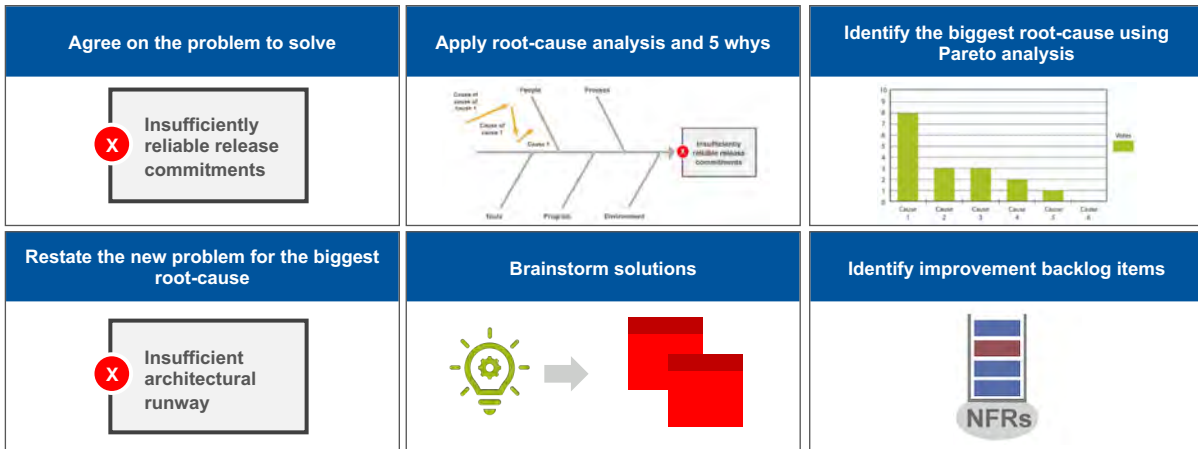
- ▶ Summarize and discuss any other program Metrics that the team has agreed to collect
- ▶ Suggested timebox: 45–60 minutes

Functionality	PI 1	PI 2	PI 3
Program velocity			
Predictability measure			
# Features planned			
# Features accepted			
# Enablers planned			
# Enablers accepted			
# Stories planned			
# Stories accepted			
Quality			
Unit test coverage %			
Defects			
Total tests			
% automated			
# NFR tests			

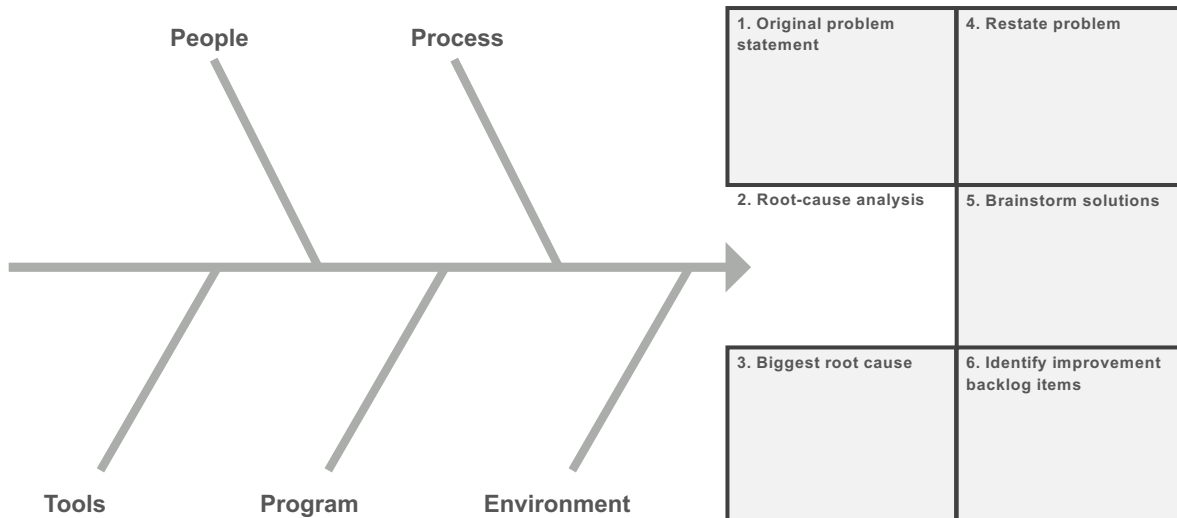
7.2 Apply problem-solving workshop

The problem-solving workshop overview

After a retrospective, teams use root cause analysis to address the larger impediments that are limiting velocity



Build the problem-solving board



Agree on the problem to solve

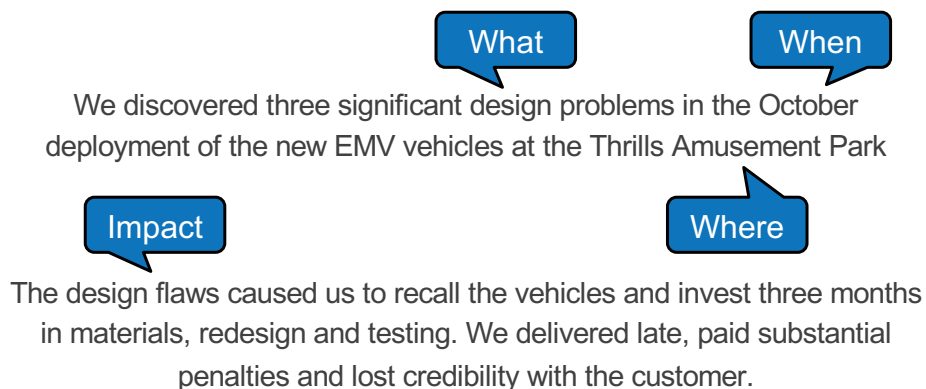
Clearly stating the problem is key to problem identification and correction

- ▶ You must define the problem or situation, so everyone involved in the workshop understands what to address
- ▶ A clearly defined problem focuses your investigation efforts and saves time
- ▶ A problem that is not well-defined may result in failure to reach the proper countermeasure



Anatomy of a well-defined problem

Think about the **What**, **When**, **Where**, and **Impact**



Concept contributed by Beth Miller

Prepare for the problem-solving workshop

The following resources are available on the SAFe Community platform:

- ▶ SAFe I&A Problem Solving Workshop Checklist
- ▶ Video Playlist: Inspect and Adapt Series
- ▶ SAFe Collaborate template: Root Cause Analysis and Problem-Solving Board

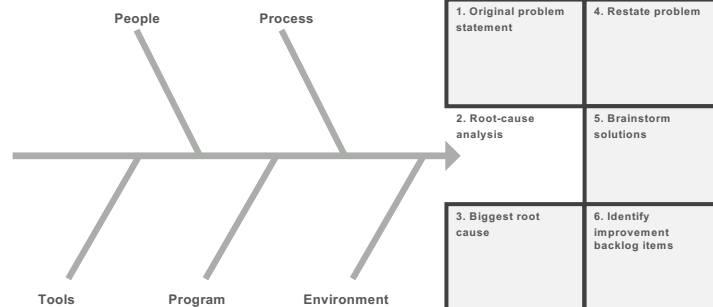


Activity: Agree on the problem

Duration



- ▶ **Step 1:** Working in your groups, build the problem-solving board
- ▶ **Step 2:** Choose one person's context to define the problem and create a clear problem statement
- ▶ **Step 3:** Capture the statement under the *Original problem statement* heading



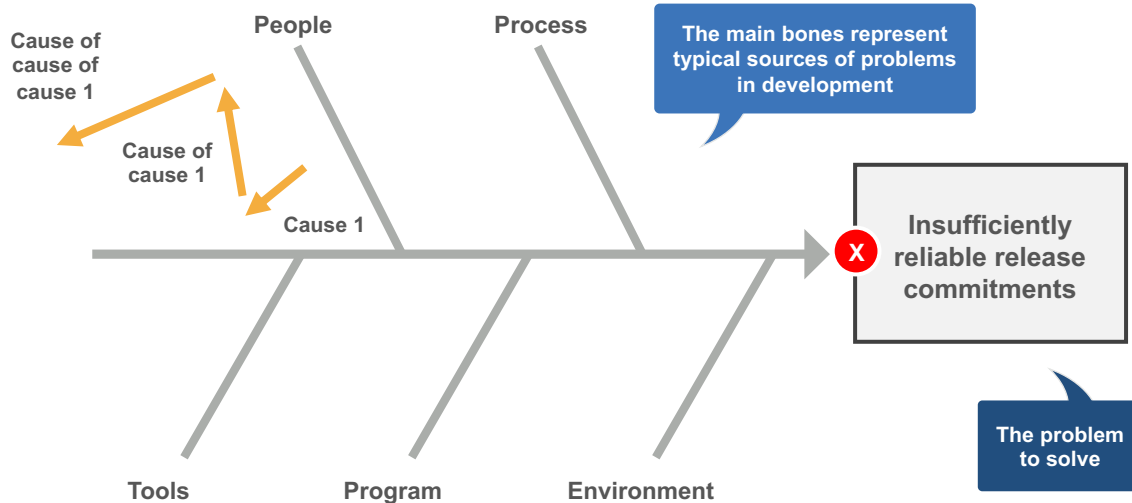
Finding the root cause: The Five Whys

“By repeating why five times, the nature of the problem, as well as its solution, becomes clear.”
 —Taiichi Ohno, father of the Toyota Production System

- ▶ The Five Whys is a proven problem-solving technique used to explore cause-and-effect relationships
- ▶ The key is to avoid assumptions and logic traps
- ▶ Instead, trace the chain of causality in direct increments from the effect to a root cause

The problem: My car will not start.	
Why?	The battery is dead.
Why?	The alternator is not functioning.
Why?	The alternator belt has broken.
Why?	The alternator belt was well beyond its useful service life.
Why?	I have not been maintaining my car according to the recommended service schedule (root cause)

Root cause analysis diagram



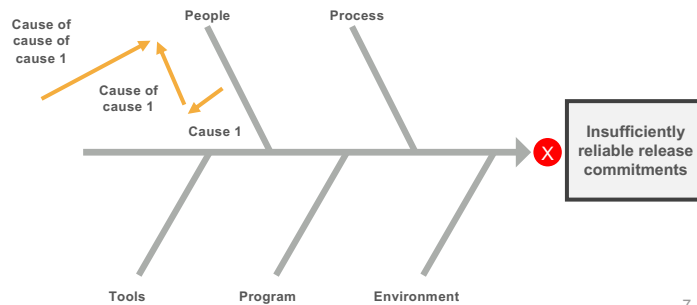


Activity: Root cause analysis

Duration



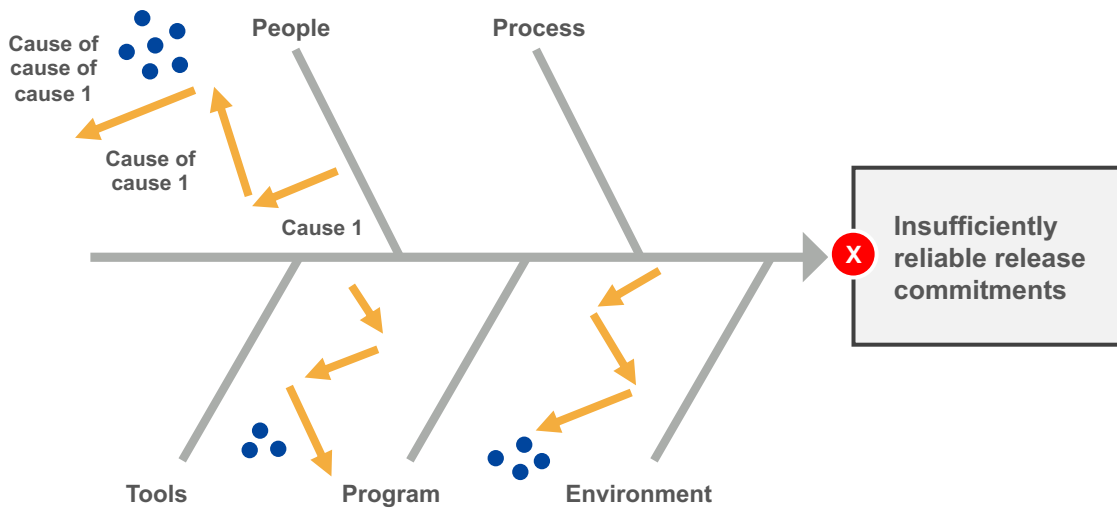
- ▶ **Step 1:** Working in your groups, brainstorm potential causes of the problem and write them down under the *Root Cause Analysis* heading on the problem-solving board
- ▶ **Step 2:** For each cause identified, use the Five Whys technique to get to a potential root cause



SCALED AGILE © Scaled Agile, Inc.

7-19

Vote on root causes

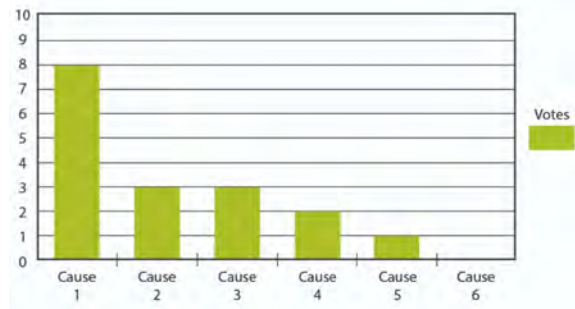


SCALED AGILE © Scaled Agile, Inc.

7-20

Pareto analysis – Identify the biggest root cause

- ▶ Also known as the 80/20 rule, the Pareto analysis is a statistical decision technique used to narrow down the number of actions that produce the most significant overall effect
- ▶ It uses the principle that 20% of root causes can cause 80% of problems
- ▶ It is useful where many possible sources and actions are competing



Activity: Restate the new problem

Duration



- ▶ **Step 1:** Working in your groups, each member will dot vote to identify the biggest root cause on your chart
- ▶ **Step 2:** Use Pareto analysis to identify the biggest root cause. Put it under the *Biggest Root Cause* heading on the problem-solving board
- ▶ **Step 3:** Restate the problem to address the identified root cause, including the economic impact of the problem under the *Restate problem* heading on the problem-solving board

Example:

We did not have the ability to measure or test the full electrical load on vehicles in real operating conditions.

Impact:

We had to upgrade the deployed power distribution system beyond what was specified. Major cost and schedule overrun.



Activity: Brainstorm solutions

Duration



- ▶ **Step 1:** Individually brainstorm and write ideas on sticky notes and put them up on the board
- ▶ **Step 2:** Working in your groups, discuss each idea as a group
- ▶ **Step 3:** Organize ideas into affinity groups
- ▶ **Step 4:** Dot vote to identify the top contenders to put in the *Identify Improvement Backlog Items* heading on the problem-solving board

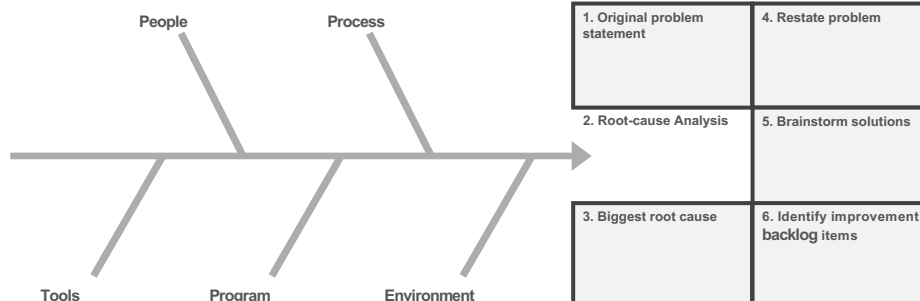


Discussion: Problem-solving workshop readout

Duration

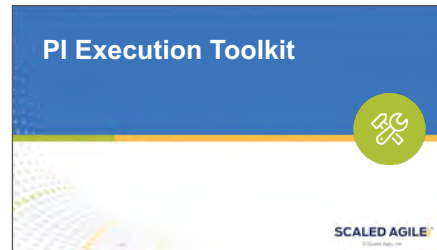


- ▶ **Step 1:** Select a person from your group to do a readout
- ▶ **Step 2:** Start the readout by stating the original problem and conclude with the proposed improvement backlog items



ART execution artifacts in the PI Execution Toolkit

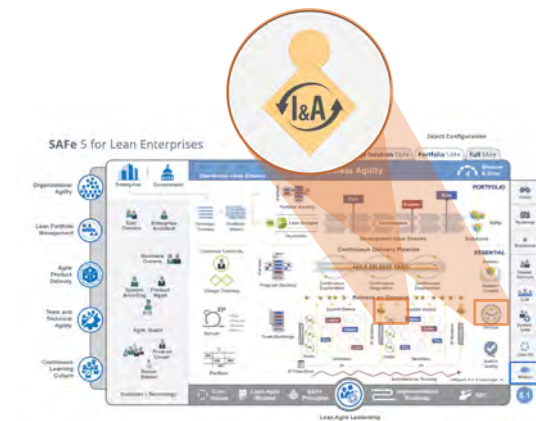
- ▶ SAFe ART Events and Activities presentation
- ▶ Inspect and Adapt Event template
- ▶ Program Performance Metrics spreadsheet
- ▶ Program Predictability Measure spreadsheet
- ▶ Self-Assessments and Metrics resources



Lesson review

In this lesson you:

- ▶ Explored how to improve program performance using an Inspect and Adapt event
- ▶ Experienced a problem-solving workshop



<https://www.scaledagileframework.com/inspect-and-adapt/>

Lesson notes



Enter your notes below:

Reminder: If using a digital workbook, save your PDF often so you don't lose any of your notes.

Lesson 8 Practicing SAFe

SAFe® Course - Attending this course gives students access to the SAFe® Advanced Scrum Master exam and related preparation materials.



SCALED AGILE®



Video: SAFe Certification Benefits

Duration



<https://bit.ly/BenefitsSAFeCertification>

A Path Towards Certification



● Access exam study guides and practice tests



● Download your certificate of course completion



● Take the **Certification Exam**



● Showcase your **Digital Badge** and get recognized as Certified SAFe Professional

Becoming a
Certified
SAFe
Professional



Video: Welcome to the SAFe Community Platform



Welcome to the
SAFe® Community
Platform

SCALED AGILE®
Provider of SAFe®

<https://bit.ly/SAFeCommunityPlatform>

Access tools and resources for your SAFe practice and continue your professional development



- **SAFe ART and Team Events** and **Toolkits** to help practice SAFe



- Access on-demand, self-paced, Getting Started **e-learning** modules and resources



- **SAFe Community Video Hub**



- Organize and run virtual SAFe events in real time with **SAFe Collaborate**



- Measure Business Agility progress with the **SAFe Assessments**



- **SAFe Forums** and **FAQs** to get questions answered

Your
SAFe
Community
Membership



SAFe ART and Team Events

SAFe ART and Team Events: Tools to support RTEs, Scrum Masters, and coaches in scheduling, preparing for, and facilitating key SAFe events



The screenshot shows the 'SAFe ART & Team Events' page. At the top is a navigation bar with links: Home, Learn, Implement, Measure, Connect, Teach, Partner, Support. Below the navigation bar is the page title 'SAFe ART & Team Events'. The main content area is divided into two columns. The left column has a section titled 'Support for ART and Team Events' with a globe icon and text: 'Learn how to facilitate SAFe ART and Team events for both in-person and distributed teams. SAFe tools and guidance are added and updated regularly to help you prepare for and facilitate successful SAFe events regardless of whether they will be face-to-face or distributed. Use articles, videos, agendas and one-pagers, and more to support your ART and team events. Need help? Check out our FAQs.' The right column has a section titled 'What's on this page?' with a list of three items: 1. Videos, checklists, toolkits, and more to help guide you in preparing for and facilitating SAFe ART and Team events. 2. SAFe Collaborate templates specifically created for each SAFe event. 3. New to SAFe Collaborate? Find guidance for using that tool here as well. Below these columns is an 'Events' section with three cards: 'Program Increment (PI) Planning' (Find the tools, resources, and assets you need to help you facilitate a PI Planning event.), 'Inspect and Adapt (I&A)' (Find the tools, resources, and assets you need to help you facilitate an Inspect and Adapt event.), and 'Team Events' (Find the tools, resources, and assets you need to help you facilitate ART & Team Events.).

SAFe Toolkits



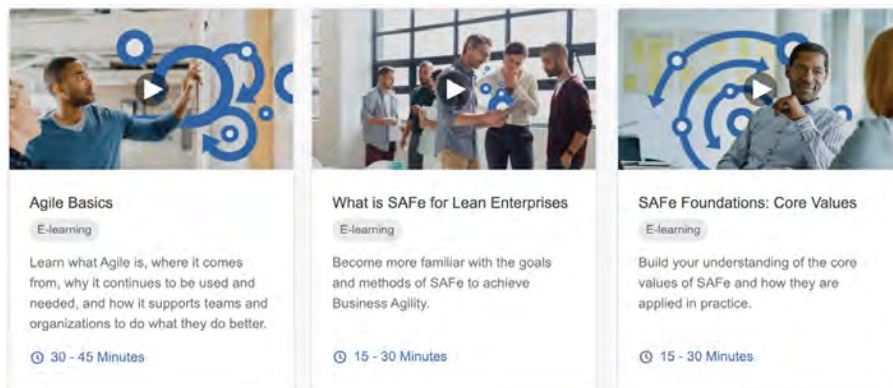
Explore ready-to-use templates and job resources to help execute SAFe events and workshops more effectively



E-learning resources



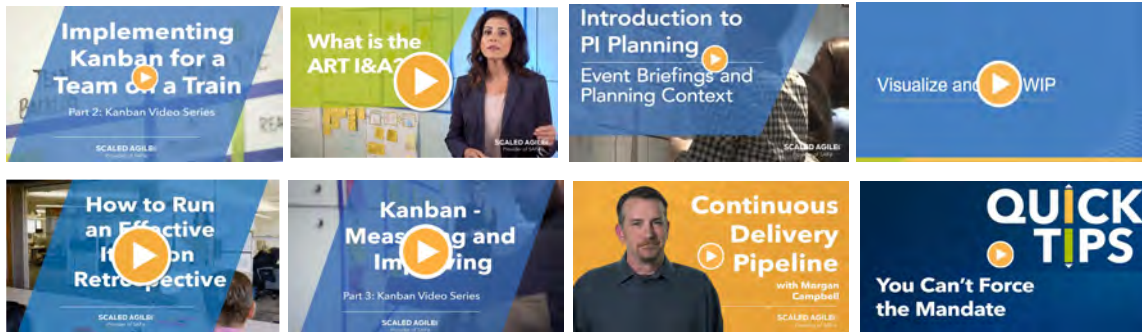
Discover and develop skills through self-paced, interactive e-learning modules to achieve your goals



Community Video Hub



Access videos to support your learning and grow your skills

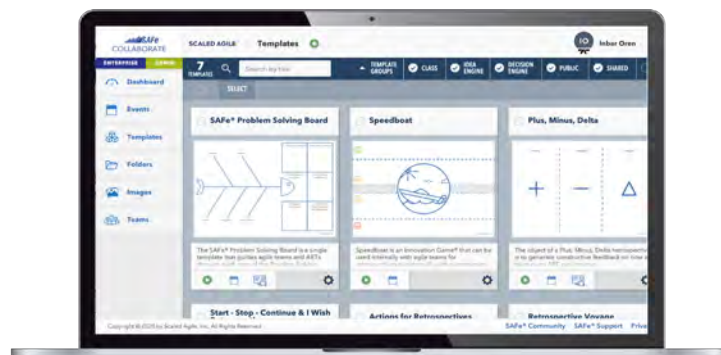


SAFe Collaborate



Organize and run virtual SAFe events in real time

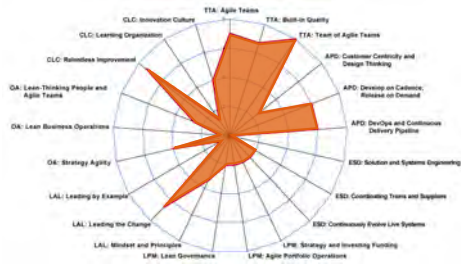
SAFe Collaborate SAFe Collaborate is a visual, cloud-based workspace tool for organizations to orchestrate virtual SAFe events and activities.



SAFe Assessments



Evaluate progress towards business agility with the SAFe assessments, Measure and Grow workshop and our assessment partners



Business Agility Assessment

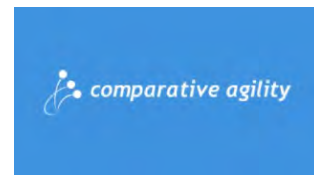
Measure and Grow Workshop Toolkit

SAFe Measure and Grow Workshop Toolkit

PDF

Find the tools and resources needed to facilitate successful Measure & Grow Workshops in your organization.

[Download](#)



SAFe Forums



Join the SAFe Release Scrum Masters Community Forum to connect with a community of Scrum Masters.

PUBLIC

SAFe Scrum Masters

[+ Join Group](#)

SAFe FAQs



When you need support, check the FAQ page for your question or contact SAI support directly.

The screenshot shows the SAFe Support page with a navigation bar containing 'Home' and 'Topics'. The main heading is 'Frequently Asked Questions'. Below this, there are four columns representing different user roles: 'I'm A Learner', 'I'm An Instructor', 'I'm A Partner', and 'General Support'. Each column has a corresponding image and a list of links. The 'I'm A Learner' column includes 'Certifications', 'Digital Badges', and 'Practice Tests'. The 'I'm An Instructor' column includes 'Exams' and 'Training'. The 'I'm A Partner' column includes 'Forums' and 'Remote Learner'. The 'General Support' column includes 'Resource Downloads', 'Course Administration', and 'Access to Materials'.

Join us on the SAFe Community Platform

We are here to help with
your SAFe role and
practice!

community.scaledagile.com/



Lesson notes



Enter your notes below:

Reminder: If using a digital workbook, save your PDF often so you don't lose any of your notes.

SAFe Glossary

**SAFe Glossary:**

Visit the Scaled Agile Framework site (<http://bit.ly/SAFeGlossary>) to download glossaries translated into other languages