

SAFe® Product Owner / Product Manager

Delivering value through effective Program Increment execution

SAFe® Course: Attending this course gives learners access to the SAFe® Product Owner / Product Manager exam and related preparation materials.



SCALED AGILE®

© Scaled Agile, Inc.

Exercise: Welcome!



- ▶ Find someone you don't know or haven't connected with in a long time
- ▶ Introduce yourself and share with them:
 - Two things you already know about the Product Owner and Product Manager roles
 - Two things you hope to learn about the role during this course
- ▶ After five minutes, team up with another pair and take turns introducing the person you just met



To perform the role of a SAFe® Product Owner / Product Manager, you should be able to:



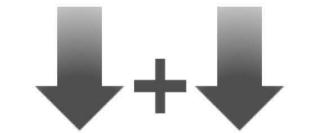
- ▶ Apply SAFe in the Lean enterprise
- ▶ Connect SAFe Lean-Agile principles and values to the PO/PM roles
- ▶ Collaborate with Lean Portfolio Management
- ▶ Explore Continuous Value with Program Increment Planning
- ▶ Execute the Program Increment and Deliver Continuous Value
- ▶ Articulate the Product Owner and Product Manager Roles
- ▶ Create your Role Action Plan

SAFe® Product Owner / Product Manager topics

1. Applying SAFe in the Lean Enterprise
2. Relating a Lean-Agile Mindset to the PO/PM Roles
3. Collaborating with Lean Portfolio Management
4. Continuously Explore Customer Needs
5. Executing the Program Increment
6. Defining the PO/PM Roles and Responsibilities
7. Creating your PO/PM Action Plan

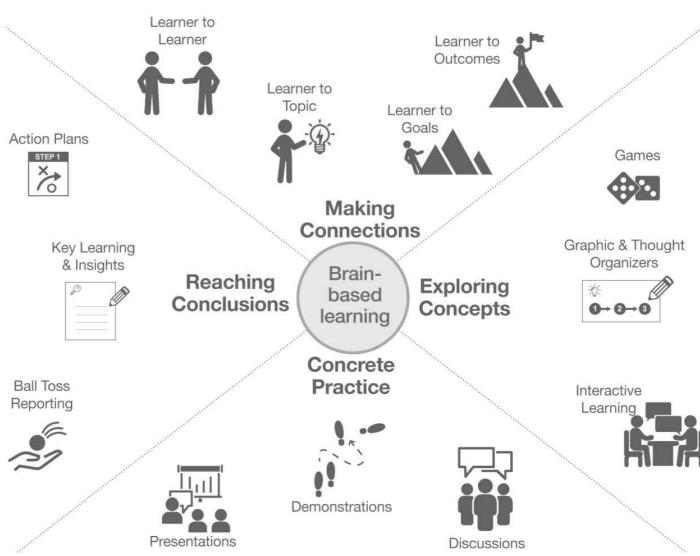
Logistics

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



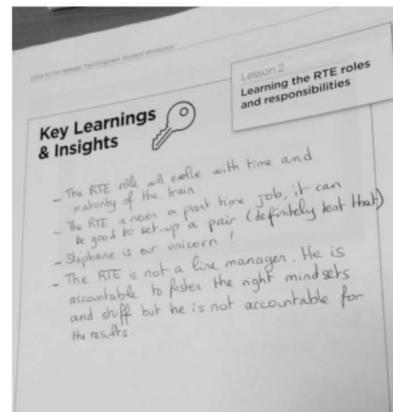
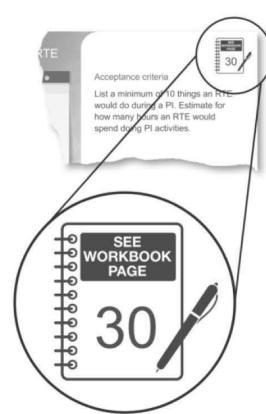
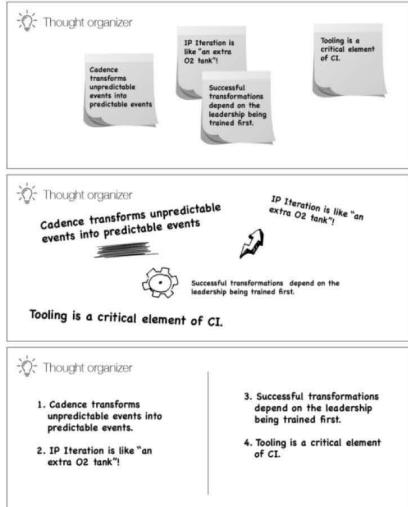
Elephant	North
Penguin	South
Giraffe	East
Dolphin	West
Zebra	(Back to top)
Wren	

Introducing active learning



Training from the Back of the Room, Sharon L. Bowman

Introducing the course workbook



Lesson 1

Applying SAFe in the Lean Enterprise

1. Applying SAFe in the Lean Enterprise
2. Relating a Lean-Agile Mindset to the PO/PM Roles
3. Collaborating with Lean Portfolio Management
4. Continuously Explore Customer Needs
5. Executing the Program Increment
6. Defining the PO/PM Roles and Responsibilities
7. Creating your PO/PM Action Plan

SAFe® Course: Attending this course gives learners access to the SAFe Product Owner/Product Manager exam and related preparation materials.

Learning objectives

1.1 Recognize the problem to be solved

1.2 Explore SAFe foundations

1.1 Recognize the problem to be solved

We thought we'd be developing like this.



But sometimes it feels like this.



Library of Congress

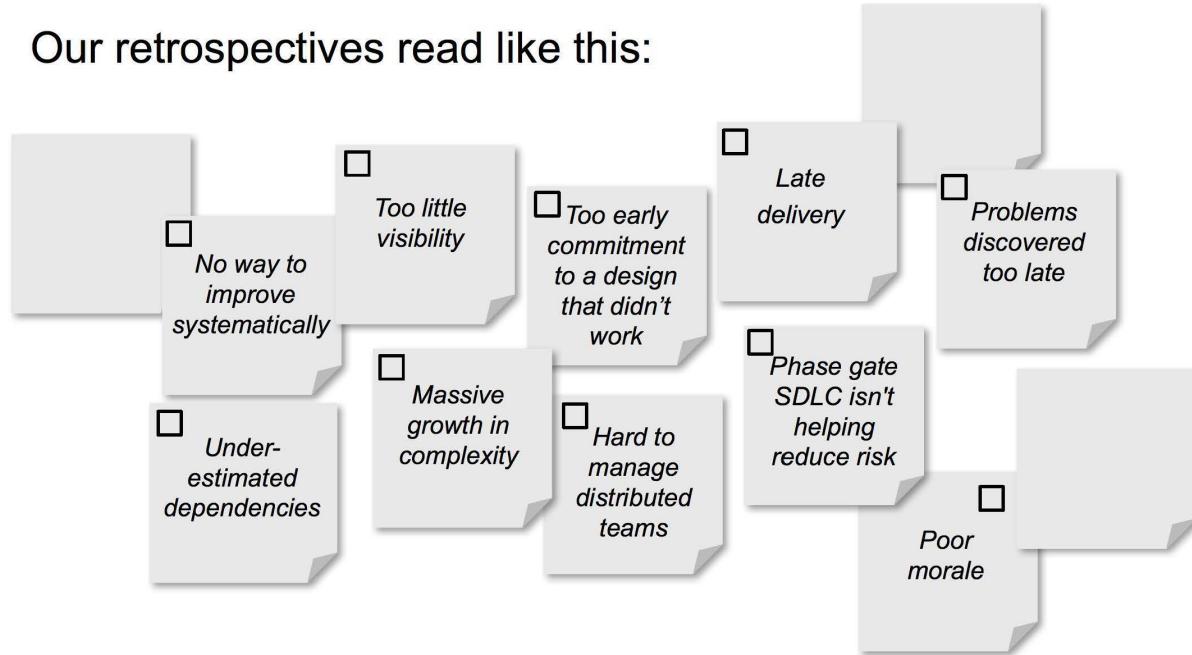
Exercise: Identifying problems



- At your table, review the following slide (also shown in your workbook on page 8)
- In your workbook, mark the problems you recognize in your organization that you would like to solve
- Add other issues you have to the empty sticky notes
- Discuss common problems at your table



Our retrospectives read like this:



Knowledge for people building the world's most important systems

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

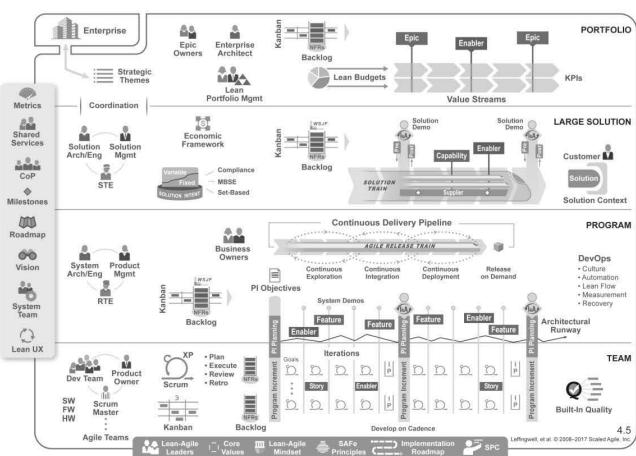
 scaledagileframework.com

SCALED AGILE® © Scaled Agile, Inc.

1.15

The Scaled Agile Framework® (SAFe®)

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



Core Values

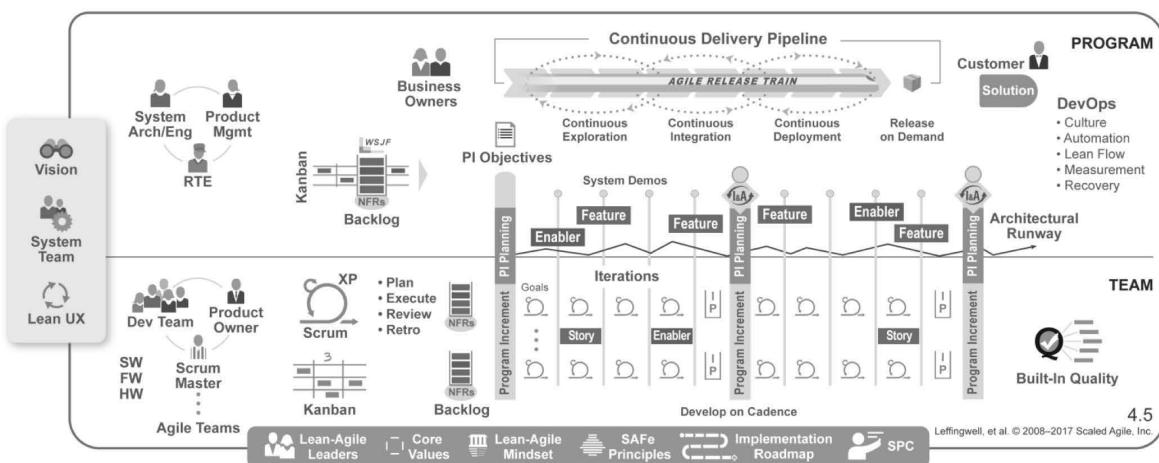
1. Built-In Quality
2. Program execution
3. Alignment
4. Transparency

SCALED AGILE® © Scaled Agile, Inc.

1.16

1.2 Explore SAFe Foundations

Essential SAFe provides the basis for success



Nothing beats an Agile Team

- ▶ Cross-functional, self-organizing entities that can define, build and test a thing of value
- ▶ Applies basic scientific practice: Plan—Do—Check—Adjust
- ▶ Delivers value every two weeks



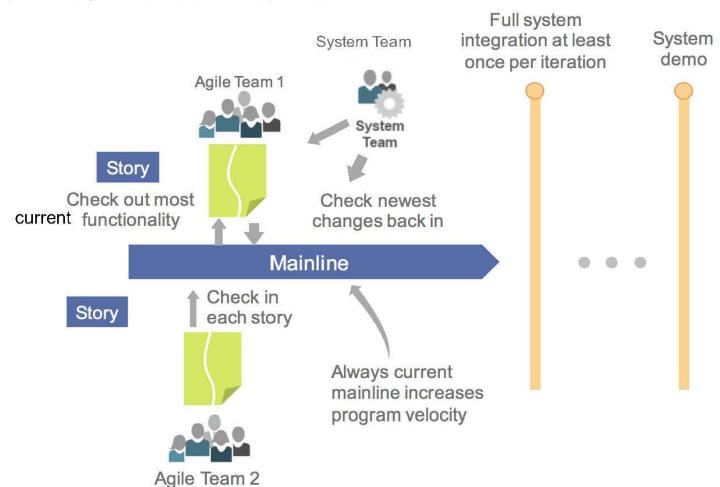
1.19

That integrates frequently

Integration points control product development.

— Dantar Oosterwal, *The Lean Machine*

- ▶ Avoid physical branching for software
- ▶ Frequently integrate hardware branches
- ▶ Use development by intention in for inter-team dependencies



© Scaled Agile, Inc.

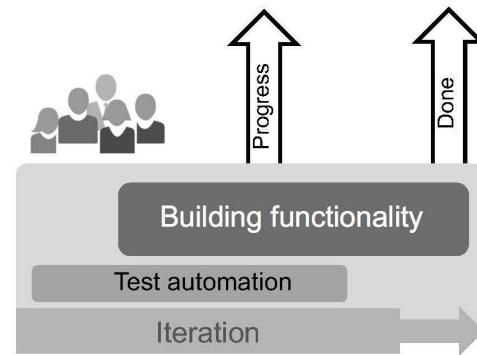
1.20

Applies test automation

Test automation supports rapid regression testing.

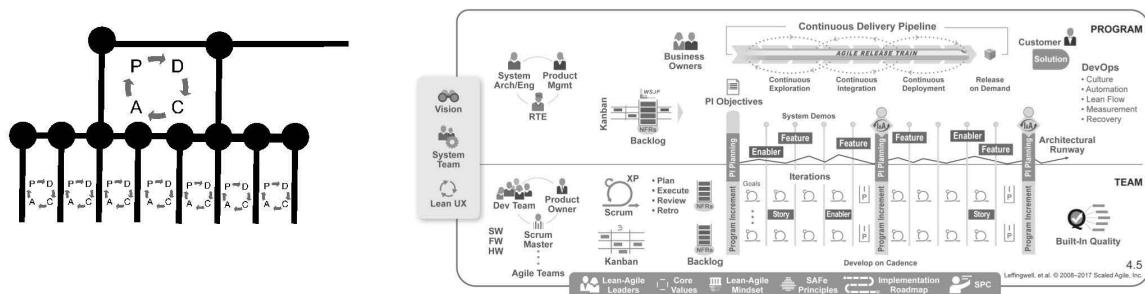
- ▶ Implemented in the same Iteration
- ▶ Maintained under version control
- ▶ Passing vs. not-yet-passing and broken automated tests are the *real* Iteration progress indicator

- ✓ Test 1
- Test 2
- ✓ Test 3
- Test 4
- ✗ Test 5
- ...
- ✓ Test 1
- ✓ Test 2
- ✓ Test 3
- ✓ Test 4
- ✓ Test 5
- ...



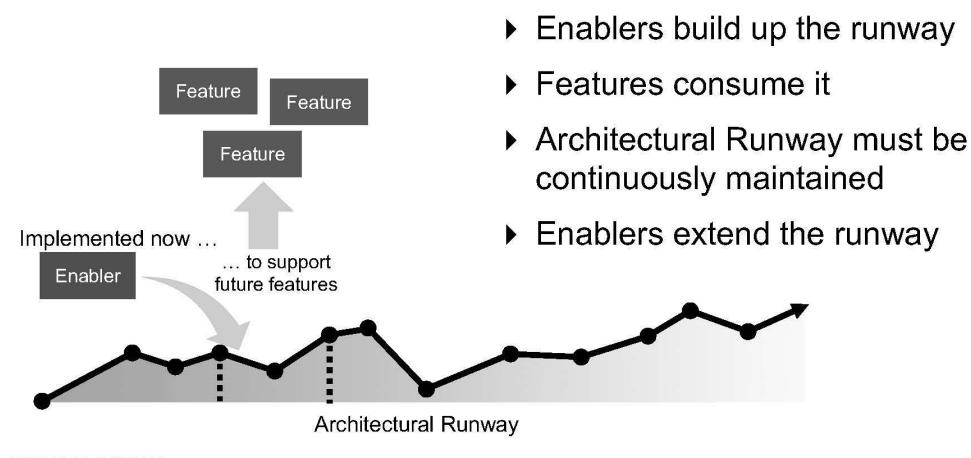
Except a team of Agile Teams

- ▶ Align 50-125 practitioners to a common mission
- ▶ Apply cadence and synchronization, Program Increments every 6-12 weeks
- ▶ Provide Vision, Roadmap, architectural guidance



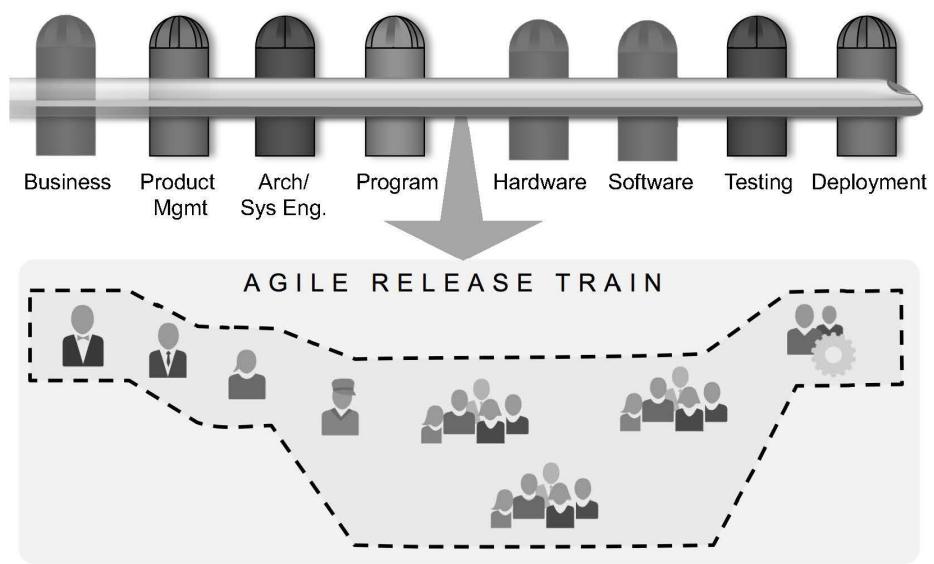
With some Architectural Runway

Architectural Runway—existing code, hardware components, etc. that technically enable near-term business features.



1.23

Bringing together the necessary people



Synchronizes with Program Increment Planning

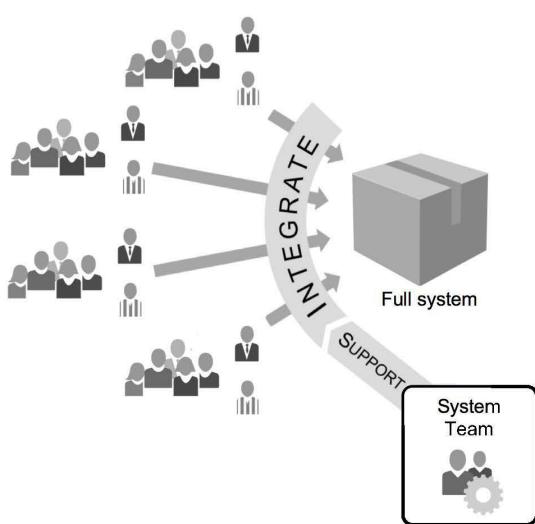
Future product development tasks can't be pre-determined. Distribute planning and control to those who can understand and react to the end results. — Michael Kennedy, *Product Development for the Lean Enterprise*

- ▶ All stakeholders face-to-face (but typically multiple locations)
- ▶ Management sets the mission, with minimum possible constraints
- ▶ Requirements and design emerge
- ▶ Important stakeholder decisions are accelerated
- ▶ Teams create—and take responsibility for—plans



For a short video PI planning example, see: <https://youtu.be/ZZAtl7nAB1M>

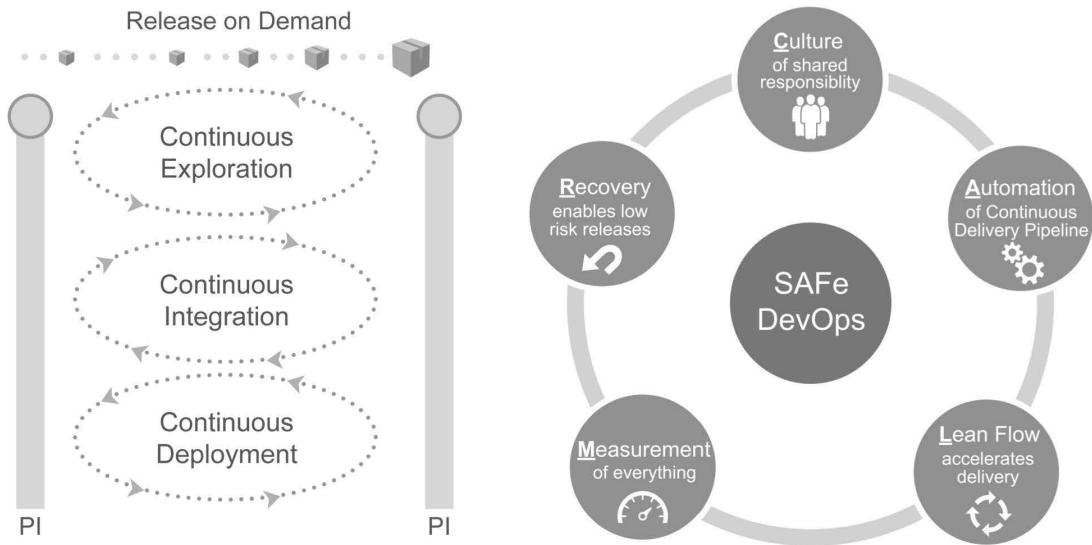
Demonstrates the full system every two weeks



- ▶ An integrated solution demo
- ▶ Objective milestone
- ▶ Demo from the staging environment, or the nearest proxy



Continuously delivers value to customers with DevOps

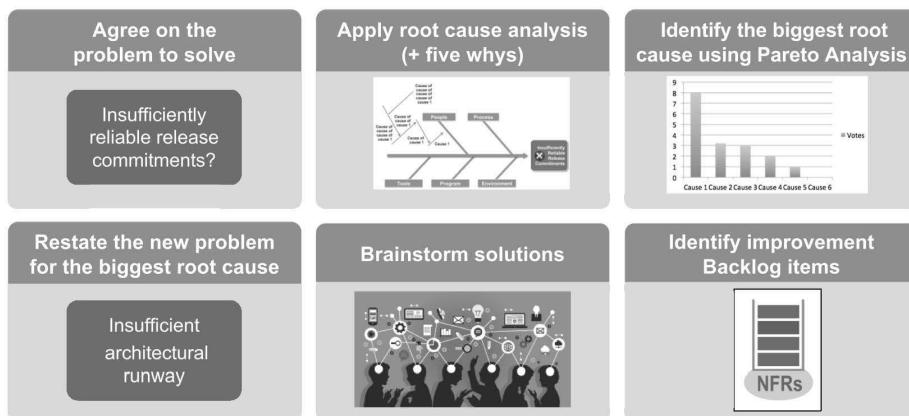


SCALED AGILE® © Scaled Agile, Inc.

1.27

Inspects and Adapts every PI

Every PI, teams systematically address the larger impediments that are limiting velocity.



SCALED AGILE® © Scaled Agile, Inc.

1.28

Exercise: Pique your interest?

SEE
WORKBOOK
PAGE
16

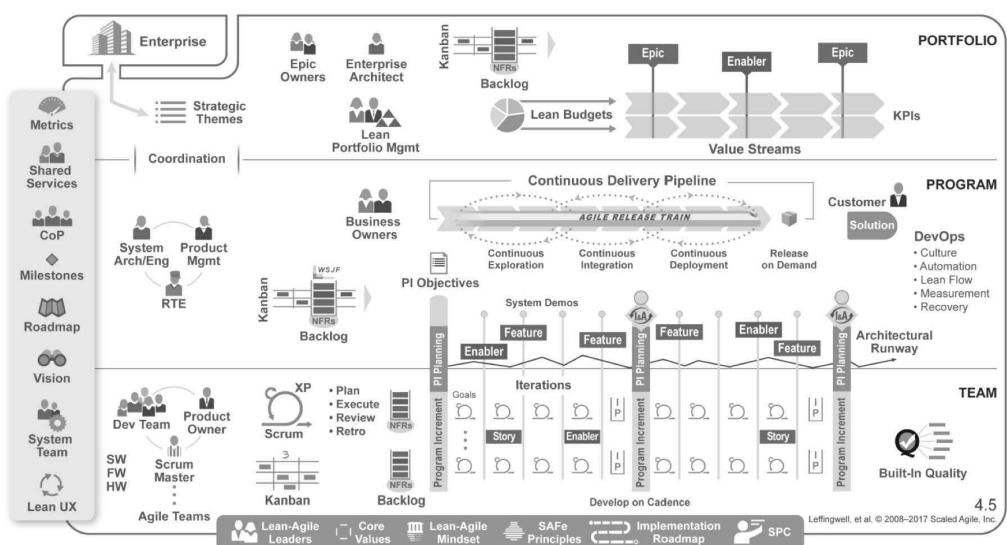
► What makes you curious about the Essential SAFe Foundation elements we have discussed?

► What would you like to learn more about? We will go deeper.

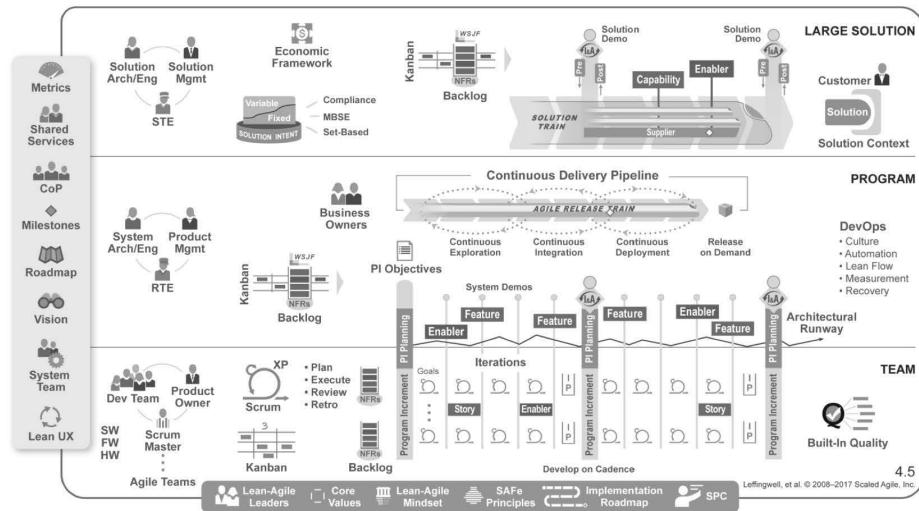
- Refer back to the Big Picture “Essential SAFe Configuration” at the front of your workbook. Write your curious questions on a sticky note, bring to the Parking Lot.
- Share your ideas with the class as you post
- Document these in your workbook



Portfolio SAFe adds Lean Portfolio governance



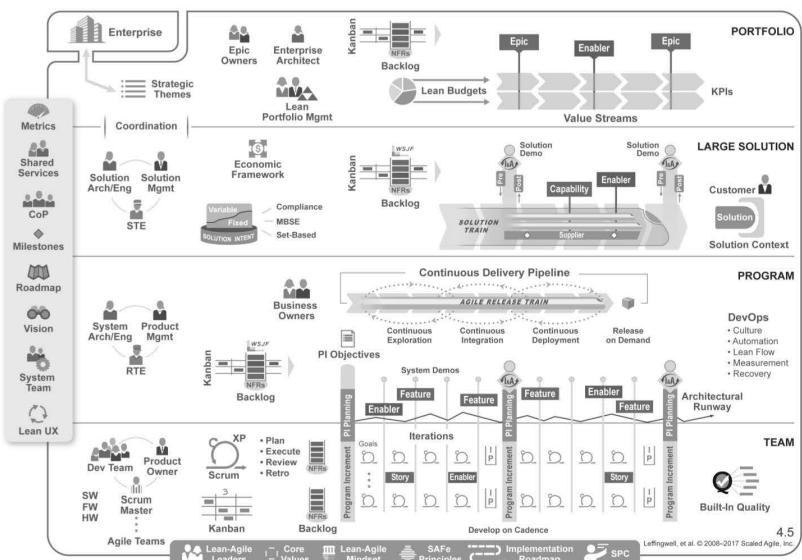
Large Solution SAFe is coordinated by a Solution Train



SCALED AGILE® © Scaled Agile, Inc.

1.31

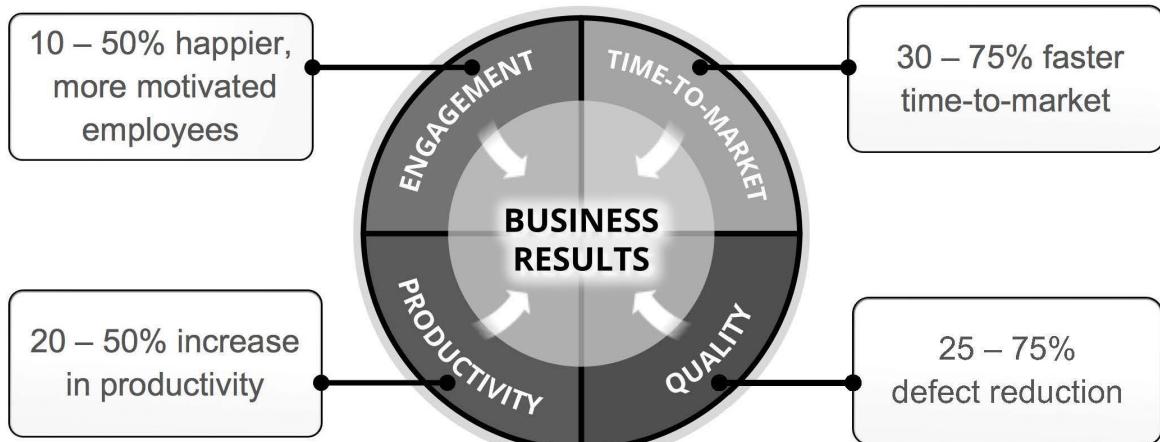
Full SAFe for large enterprises



SCALED AGILE® © Scaled Agile, Inc.

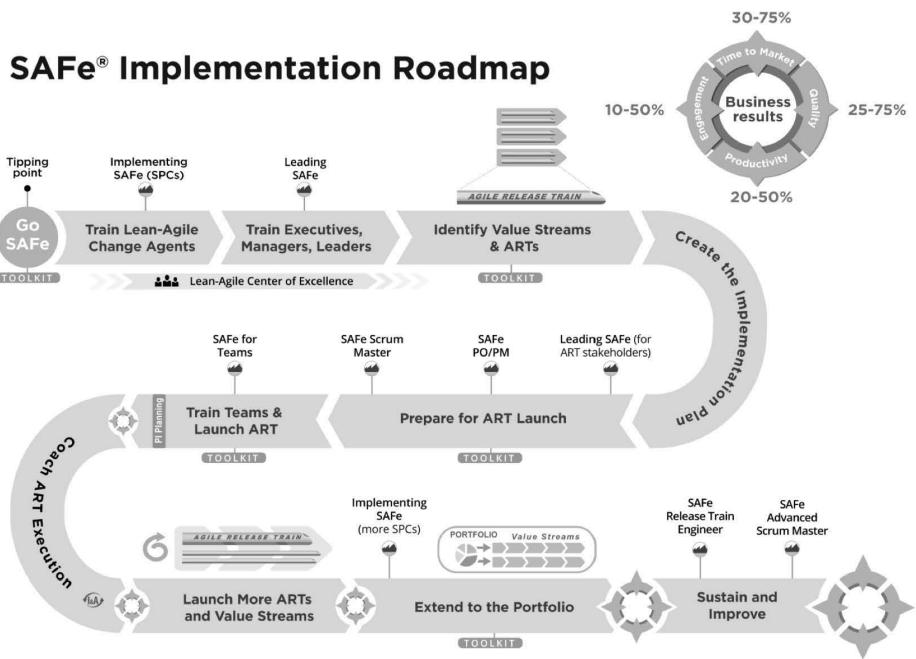
1.32

Achieve business results



See ScaledAgileFramework.com/case-studies

SAFe® Implementation Roadmap



SAFe POPM certification process

Certification is much more than just taking an exam.

Complete the course.



Follow your Learning Plan.



Leverage your study materials and practice tests, then pass the exam.



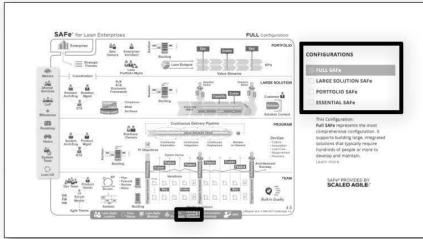
Continue your Learning Journey with the SAFe Community Platform and resources.



Lesson Summary

In this lesson, you:

- ▶ Recognized the problem to be solved
- ▶ Explored SAFe foundations



Suggested Scaled Agile Framework reading:

- “*Implementation Roadmap*” article
- *Four Framework configurations*

Exercise: This lesson's key learnings



Summarize key learnings and insights from this lesson in your workbook.



SAFe® Product Owner / Product Manager

Delivering value through effective Program Increment execution

SAFe® Course: Attending this course gives learners access to the SAFe® Product Owner / Product Manager exam and related preparation materials.



SCALED AGILE®

© Scaled Agile, Inc.

Exercise: Welcome!



- ▶ Find someone you don't know or haven't connected with in a long time
- ▶ Introduce yourself and share with them:
 - Two things you already know about the Product Owner and Product Manager roles
 - Two things you hope to learn about the role during this course
- ▶ After five minutes, team up with another pair and take turns introducing the person you just met



To perform the role of a SAFe® Product Owner / Product Manager, you should be able to:



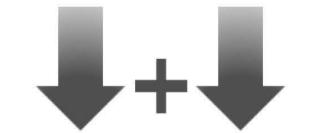
- ▶ Apply SAFe in the Lean enterprise
- ▶ Connect SAFe Lean-Agile principles and values to the PO/PM roles
- ▶ Collaborate with Lean Portfolio Management
- ▶ Explore Continuous Value with Program Increment Planning
- ▶ Execute the Program Increment and Deliver Continuous Value
- ▶ Articulate the Product Owner and Product Manager Roles
- ▶ Create your Role Action Plan

SAFe® Product Owner / Product Manager topics

1. Applying SAFe in the Lean Enterprise
2. Relating a Lean-Agile Mindset to the PO/PM Roles
3. Collaborating with Lean Portfolio Management
4. Continuously Explore Customer Needs
5. Executing the Program Increment
6. Defining the PO/PM Roles and Responsibilities
7. Creating your PO/PM Action Plan

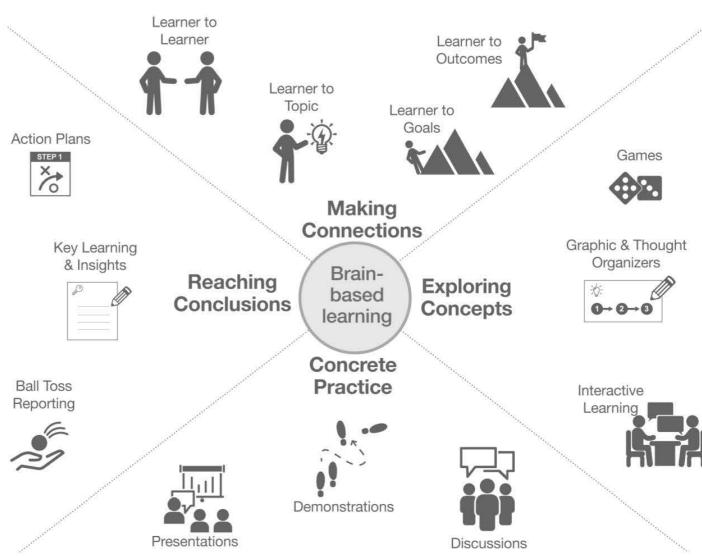
Logistics

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



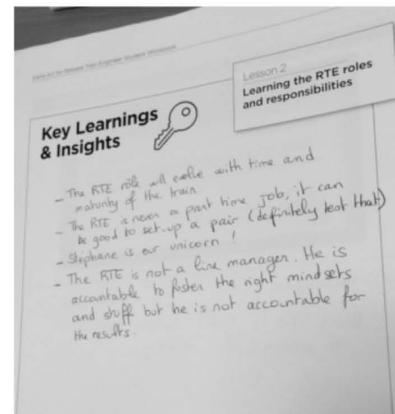
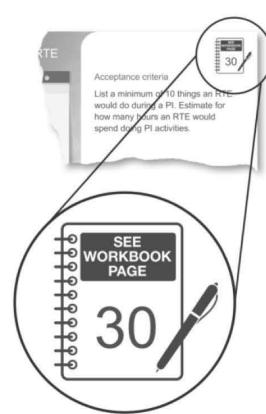
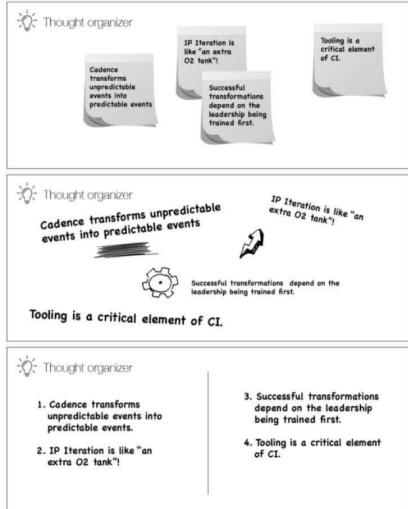
Elephant	North
Penguin	South
Giraffe	East
Dolphin	West
Zebra	(Back to top)
Wren	

Introducing active learning



Training from the Back of the Room, Sharon L. Bowman

Introducing the course workbook



SCALED AGILE® © Scaled Agile, Inc.

1.7

Lesson 1

Applying SAFe in the Lean Enterprise

1. Applying SAFe in the Lean Enterprise
2. Relating a Lean-Agile Mindset to the PO/PM Roles
3. Collaborating with Lean Portfolio Management
4. Continuously Explore Customer Needs
5. Executing the Program Increment
6. Defining the PO/PM Roles and Responsibilities
7. Creating your PO/PM Action Plan

SAFe® Course: Attending this course gives learners access to the SAFe Product Owner/Product Manager exam and related preparation materials.

SCALED AGILE® © Scaled Agile, Inc.

2.8

Learning objectives

1.1 Recognize the problem to be solved

1.2 Explore SAFe foundations

1.1 Recognize the problem to be solved

We thought we'd be developing like this.



But sometimes it feels like this.



Library of Congress

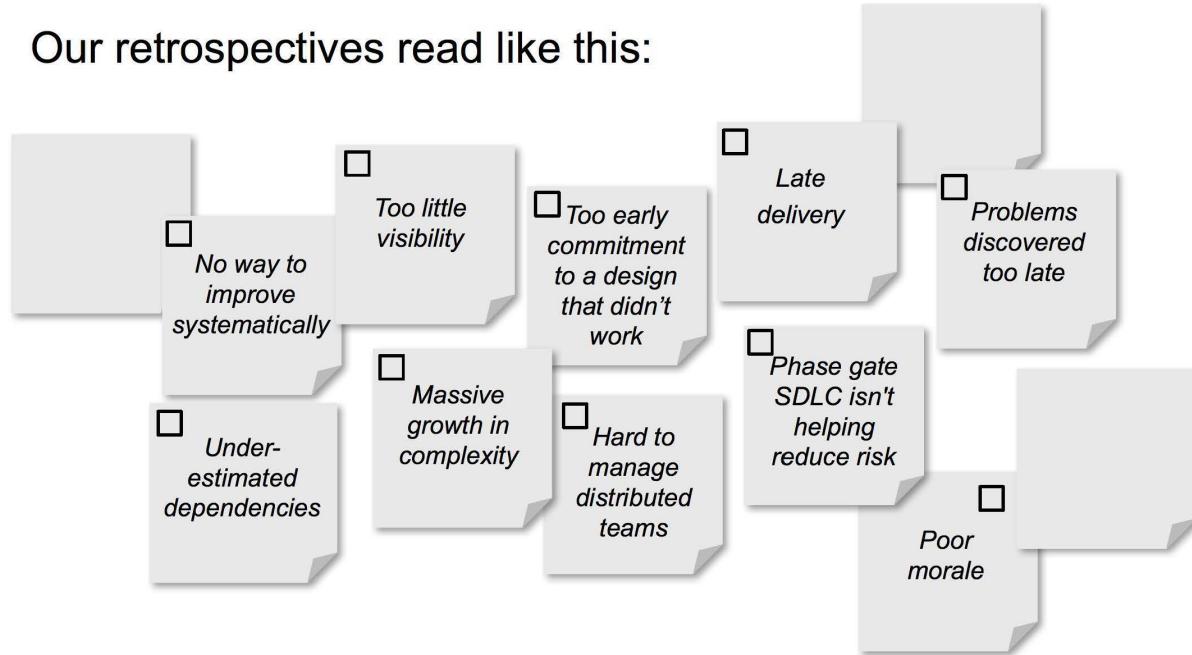
Exercise: Identifying problems



- At your table, review the following slide (also shown in your workbook on page 8)
- In your workbook, mark the problems you recognize in your organization that you would like to solve
- Add other issues you have to the empty sticky notes
- Discuss common problems at your table



Our retrospectives read like this:



Knowledge for people building the world's most important systems

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

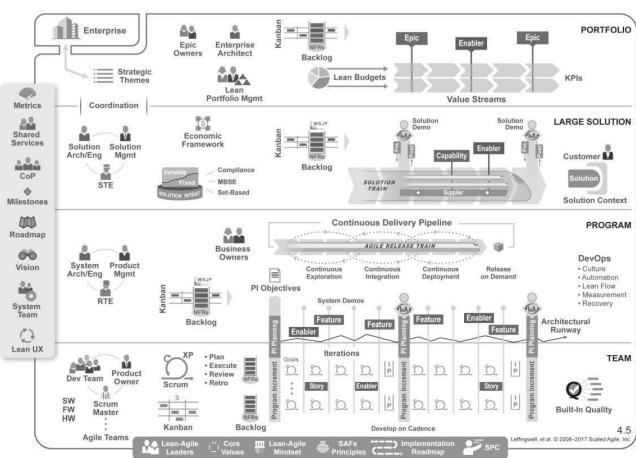
 scaledagileframework.com

SCALED AGILE® © Scaled Agile, Inc.

1.15

The Scaled Agile Framework® (SAFe®)

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



Core Values

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

SCALED AGILE® © Scaled Agile, Inc.

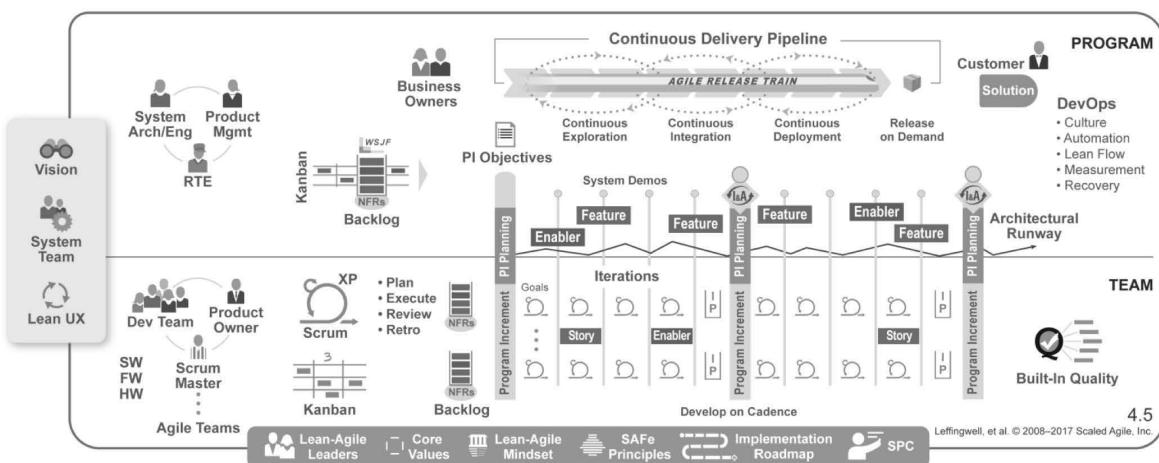
1.16

1.2 Explore SAFe Foundations

SCALED AGILE® © Scaled Agile, Inc.

1.17

Essential SAFe provides the basis for success



SCALED AGILE® © Scaled Agile, Inc.

1.18

Nothing beats an Agile Team

- ▶ Cross-functional, self-organizing entities that can define, build and test a thing of value
- ▶ Applies basic scientific practice: Plan—Do—Check—Adjust
- ▶ Delivers value every two weeks



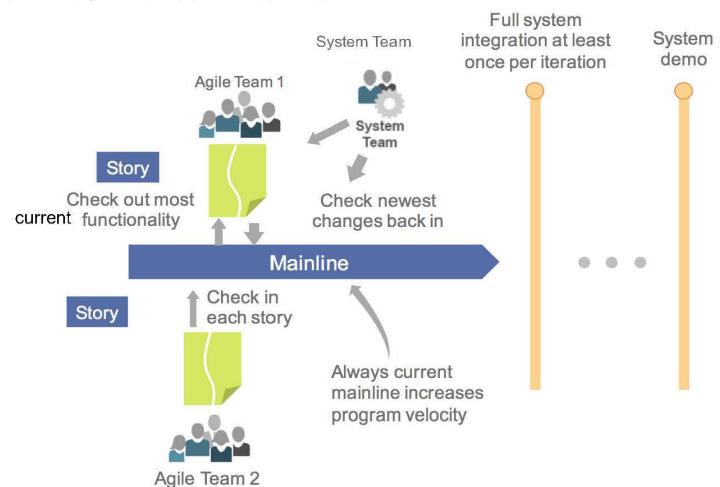
1.19

That integrates frequently

Integration points control product development.

— Dantar Oosterwal, *The Lean Machine*

- ▶ Avoid physical branching for software
- ▶ Frequently integrate hardware branches
- ▶ Use development by intention in for inter-team dependencies



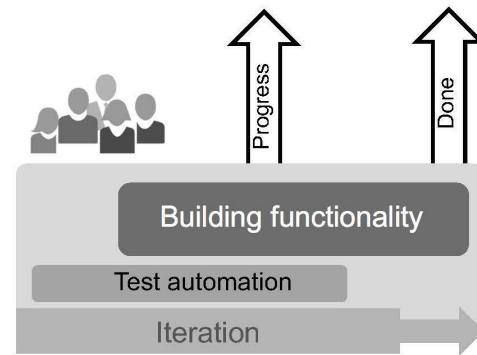
1.20

Applies test automation

Test automation supports rapid regression testing.

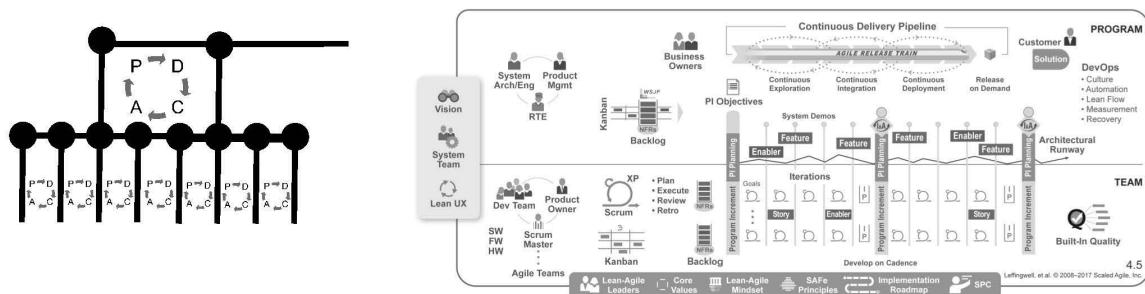
- Implemented in the same Iteration
- Maintained under version control
- Passing vs. not-yet-passing and broken automated tests are the *real* Iteration progress indicator

- ✓ Test 1
- Test 2
- ✓ Test 3
- Test 4
- ✗ Test 5
- ...
- ✓ Test 1
- ✓ Test 2
- ✓ Test 3
- ✓ Test 4
- ✓ Test 5
- ...



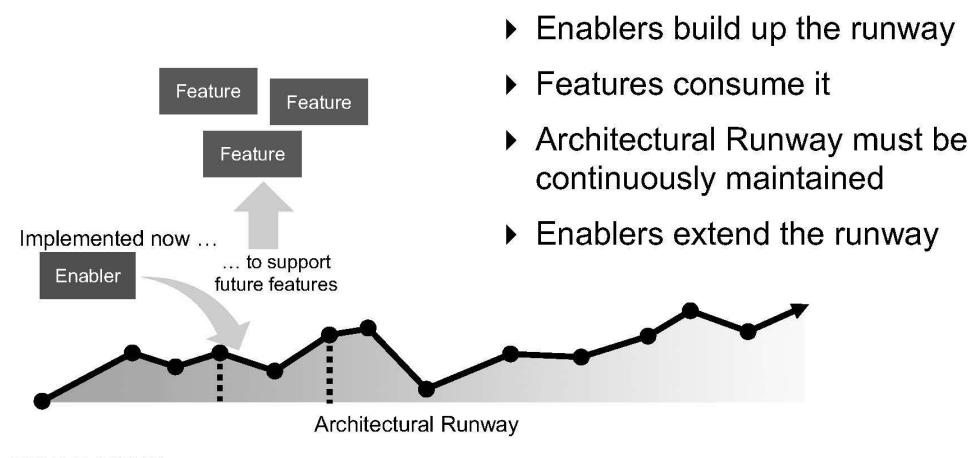
Except a team of Agile Teams

- Align 50-125 practitioners to a common mission
- Apply cadence and synchronization, Program Increments every 6-12 weeks
- Provide Vision, Roadmap, architectural guidance



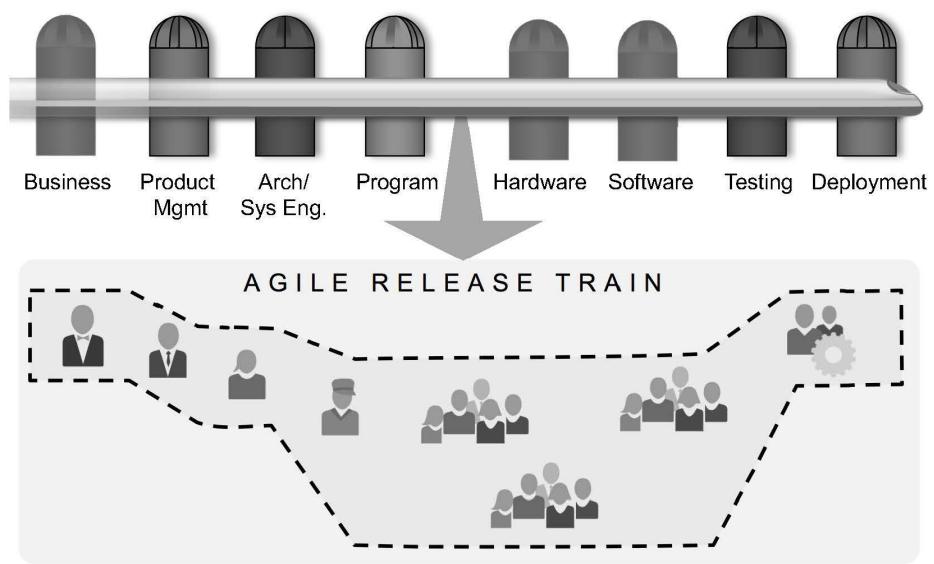
With some Architectural Runway

Architectural Runway—existing code, hardware components, etc. that technically enable near-term business features.



1.23

Bringing together the necessary people



Synchronizes with Program Increment Planning

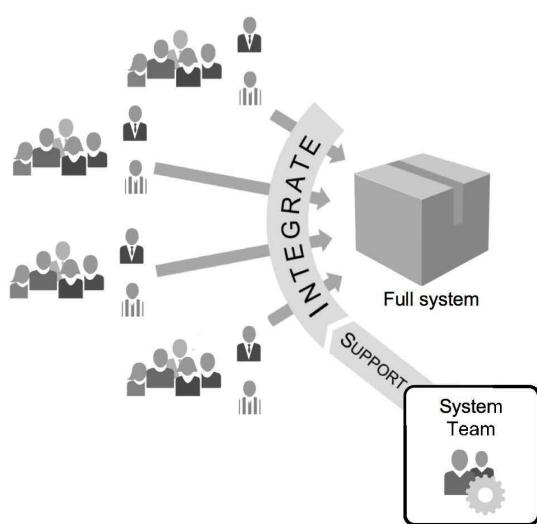
Future product development tasks can't be pre-determined. Distribute planning and control to those who can understand and react to the end results. — Michael Kennedy, *Product Development for the Lean Enterprise*

- ▶ All stakeholders face-to-face (but typically multiple locations)
- ▶ Management sets the mission, with minimum possible constraints
- ▶ Requirements and design emerge
- ▶ Important stakeholder decisions are accelerated
- ▶ Teams create—and take responsibility for—plans



For a short video PI planning example, see: <https://youtu.be/ZZAtl7nAB1M>

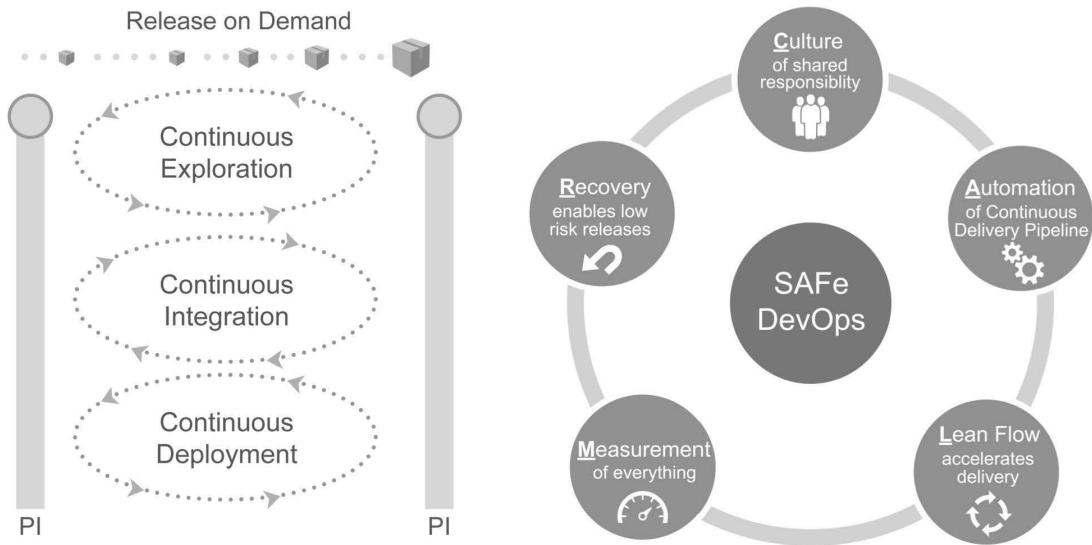
Demonstrates the full system every two weeks



- ▶ An integrated solution demo
- ▶ Objective milestone
- ▶ Demo from the staging environment, or the nearest proxy



Continuously delivers value to customers with DevOps

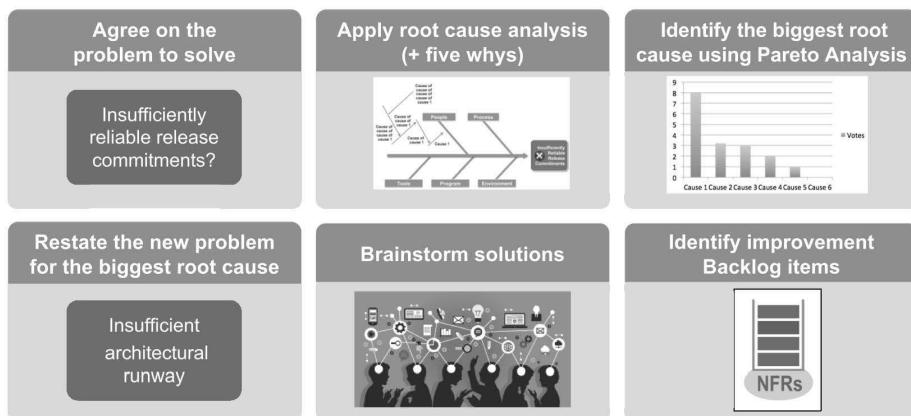


SCALED AGILE® © Scaled Agile, Inc.

1.27

Inspects and Adapts every PI

Every PI, teams systematically address the larger impediments that are limiting velocity.



SCALED AGILE® © Scaled Agile, Inc.

1.28

Exercise: Pique your interest?

SEE
WORKBOOK
PAGE
16

► What makes you curious about the Essential SAFe Foundation elements we have discussed?

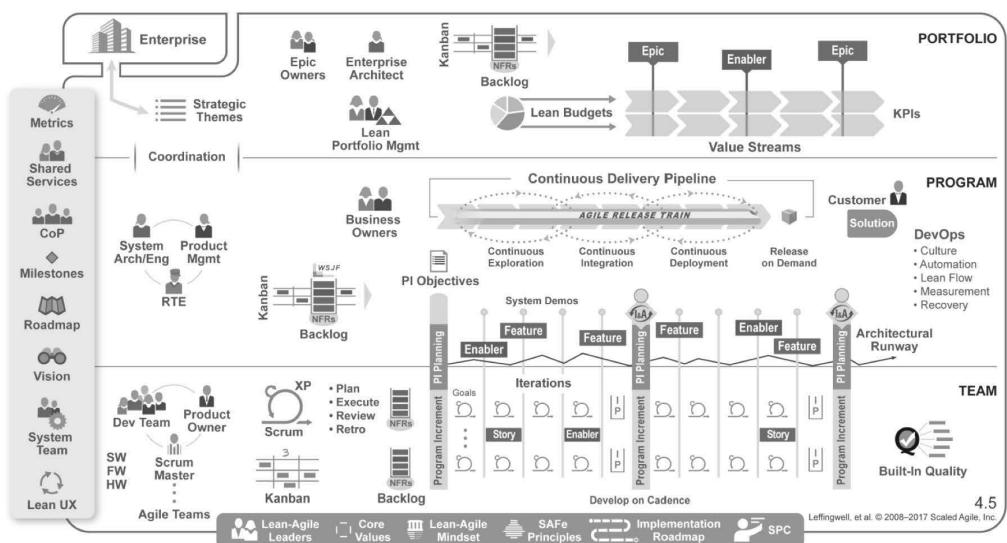
► What would you like to learn more about? We will go deeper.

- Refer back to the Big Picture “Essential SAFe Configuration” at the front of your workbook. Write your curious questions on a sticky note, bring to the Parking Lot.
- Share your ideas with the class as you post
- Document these in your workbook

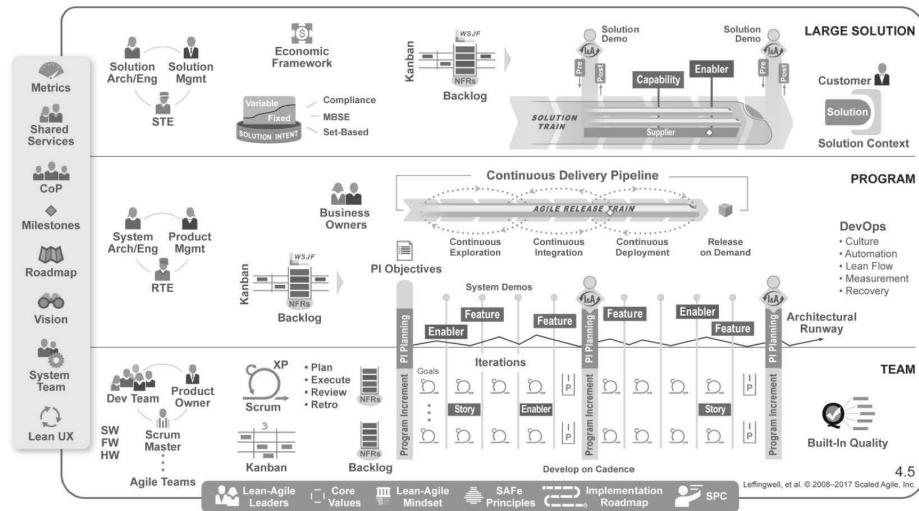
PREPARE
3 min

SHARE
3 min

Portfolio SAFe adds Lean Portfolio governance



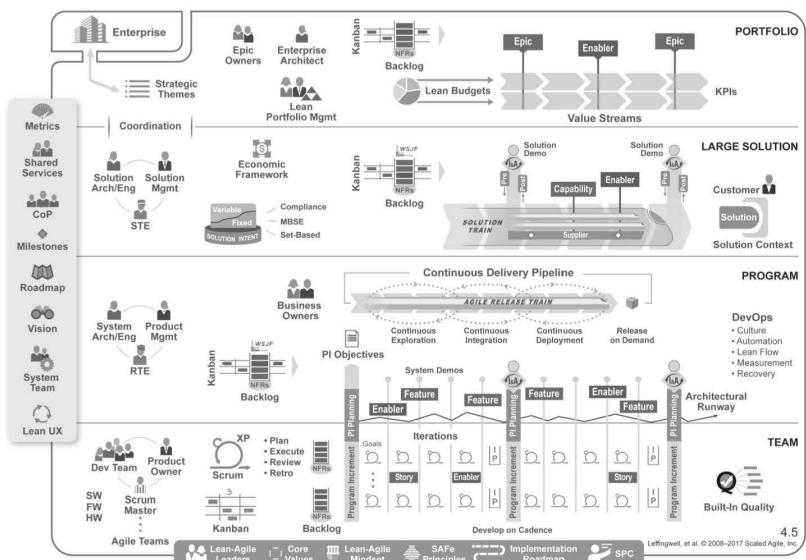
Large Solution SAFe is coordinated by a Solution Train



SCALED AGILE® © Scaled Agile, Inc.

1.31

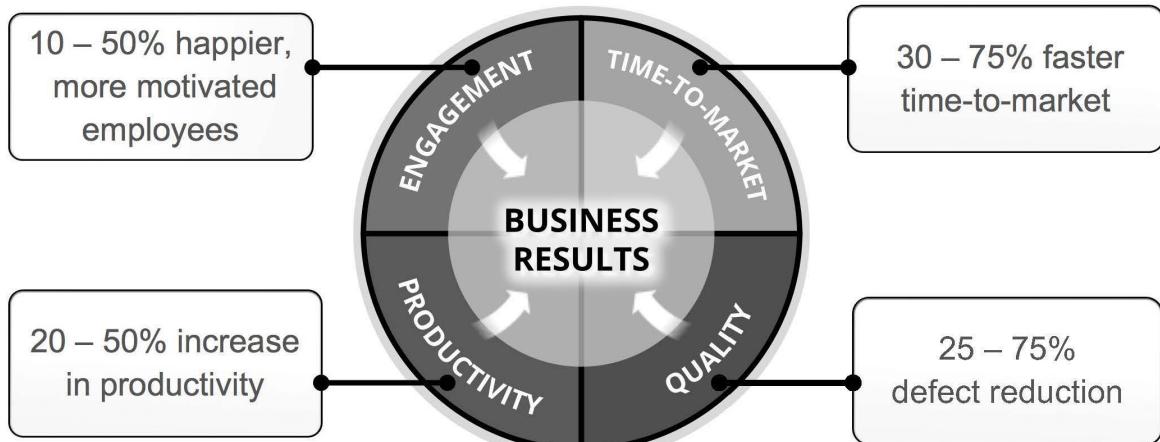
Full SAFe for large enterprises



SCALED AGILE® © Scaled Agile, Inc.

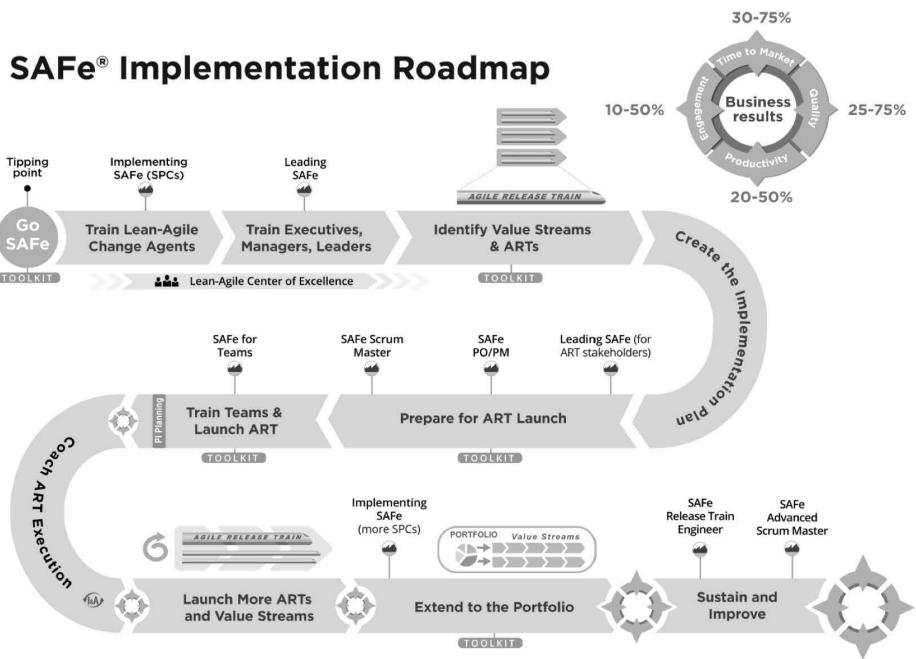
1.32

Achieve business results



See ScaledAgileFramework.com/case-studies

SAFe® Implementation Roadmap



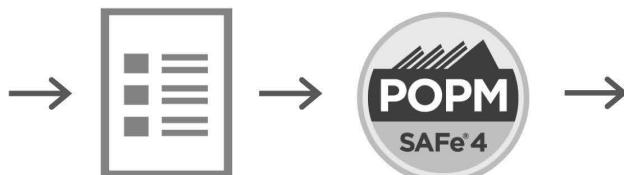
SAFe POPM certification process

Certification is much more than just taking an exam.

Complete the course.



Follow your Learning Plan.



Leverage your study materials and practice tests, then pass the exam.

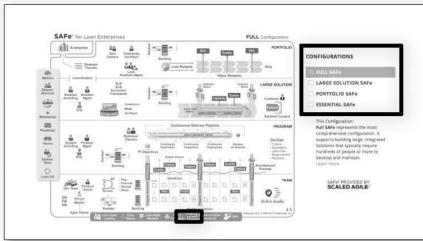
Continue your Learning Journey with the SAFe Community Platform and resources.



Lesson Summary

In this lesson, you:

- ▶ Recognized the problem to be solved
- ▶ Explored SAFe foundations



Suggested Scaled Agile Framework reading:

- “*Implementation Roadmap*” article
- *Four Framework configurations*

Exercise: This lesson's key learnings



Summarize key learnings and insights from this lesson in your workbook.



Lesson 2

Relating a Lean-Agile Mindset to the PO/PM Roles

1. Applying SAFe in the Lean Enterprise
2. Relating a Lean-Agile Mindset to the PO/PM Roles
3. Collaborating with Lean Portfolio Management
4. Continuously Explore Customer Needs
5. Executing the Program Increment
6. Defining the PO/PM Roles and Responsibilities
7. Creating your PO/PM Action Plan

SAFe® Course: Attending this course gives learners access to the SAFe Product Owner/Product Manager exam and related preparation materials.

2.1

Learning objectives

- 2.1 Connect the Product Owner and Product Manager roles
- 2.2 Embrace a Lean-Agile Mindset

2.1 Connect the Product Owner and Product Manager Roles

Traditional vs. Agile

PO and PM responsibilities differ between the traditional predictive approach and the Agile empirical approach.

PO/PM responsibilities	Traditional	Agile
Understand Customer need	Up front and discontinuous	Constant interaction
Document requirements	Fully elaborated in MRD/PRD	Coarsely documented in Vision and elaborated in Solution Intent
Scheduling	Plan delivery way later	Continuous near-term Roadmap
Prioritize requirements	Not at all or one-time only in PRD	Reprioritize every Program Increment (PI)
Validate requirements	NA—QA responsibility?	Accept every Iteration and PI; more frequent Releases
Manage change	Discourage change—weekly CCB meetings	Adapt and adjust at every PI and Iteration boundary
Assess status	Milestone document review	See working system every Iteration and every PI
Assess likelihood of release date	Defect trends or crystal ball, developers' words?	Release dates are fixed; manage scope expectations

Product Manager and Product Owner

In the Enterprise, a single person cannot handle product strategy and market launch while being dedicated to Agile Team(s) during development.

Product Manager

- ▶ Market/Customer facing and identifies market needs
- ▶ Collocated with and reports into marketing/business
- ▶ Owns Vision and Roadmaps, pricing, licensing, ROI, and Program Backlog
- ▶ Drives PI Objectives and content via prioritized Features and Enablers
- ▶ Establishes Feature acceptance criteria, accepts Features into the system



Product Owner

- ▶ Solution, technology, and team facing
- ▶ Collocated with and reports into development
- ▶ Contributes to Vision and Program Backlog; owns Team Backlog and implementation
- ▶ Drives Iteration Goals and Iteration content via prioritized Stories
- ▶ Establishes Story acceptance criteria, accepts Stories into the baseline

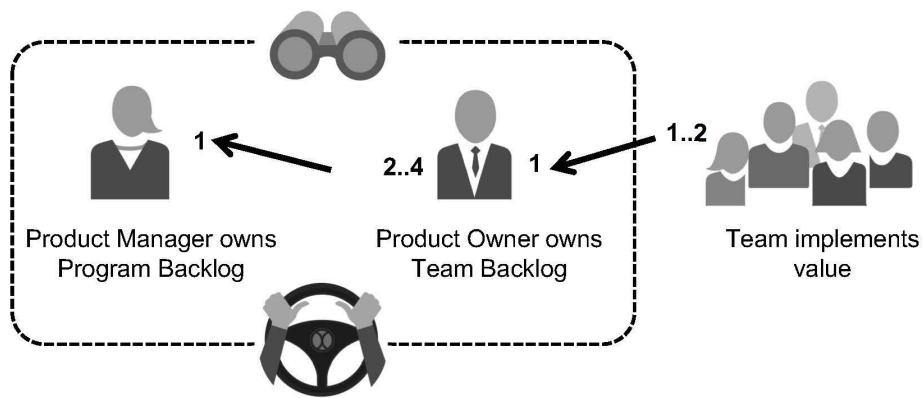


Drives the Releases and PIs

Drives the Iteration

The PO/PM team steers the train

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



PO, PM, and Team activities

Product Manager <i>Drives PIs and Releases</i>	Product Owner <i>Drives the iteration</i>	Agile Teams <i>Drives program execution</i>
		
<ul style="list-style-type: none">▶ Market/Customer facing and identifies market needs▶ Collocated with marketing/business▶ Owns Vision and Roadmaps, pricing, licensing, ROI, and Program Backlog▶ Drives PI Objectives and Release content via prioritized Features and Enablers▶ Establishes Feature acceptance criteria	<ul style="list-style-type: none">▶ Solution, technology, and team facing▶ Collocated with development▶ Owns Team Backlog and implementation; Contributes to Vision & Program Backlog▶ Drives Iteration Goals and Iteration content via prioritized Stories▶ Establishes Story acceptance criteria, accepts Stories	<ul style="list-style-type: none">▶ Integrate as a train▶ Coordinates dependencies and fosters collaboration▶ Builds quality in▶ Implements and delivers value

Release content governance

Product Manager	Product Owner	Team
 <ul style="list-style-type: none">▶ Owns Program Backlog▶ Defines Features, PIs, and Releases▶ Owns pricing, licensing, and ROI▶ Collaborates on Enablers	 <ul style="list-style-type: none">▶ Owns Team Backlog(s)▶ Defines Iterations and Stories▶ Accepts Iteration increments▶ Includes refactors and redesigns in backlog	 <ul style="list-style-type: none">▶ Contribute to intentional architecture, owns emergent design▶ Own Story estimates and implementation of value▶ Integrate with other teams

Exercise: Short and Tweet



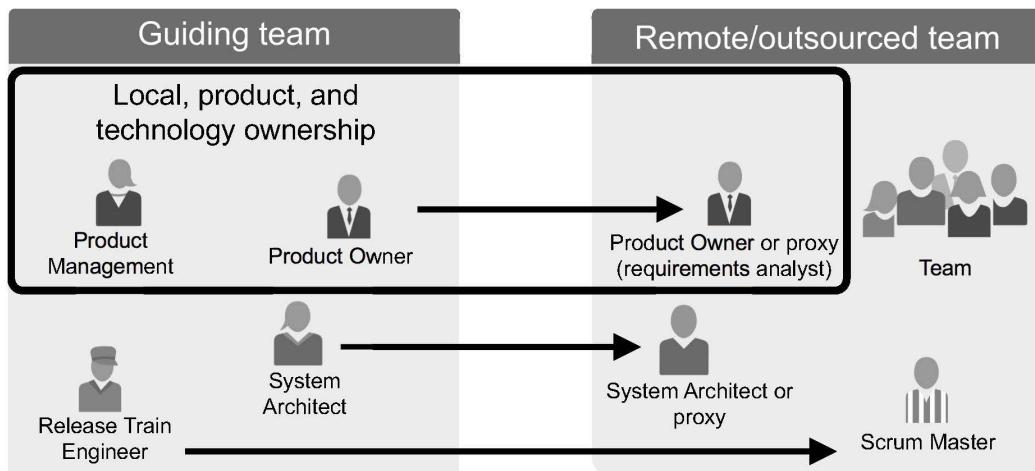
In pairs, create a 140-character description about the relationship between the Product Owner and Product Manager.

- ▶ Discuss the relationship between the roles of the Product Owner and Product Manager
- ▶ Be prepared to share your potential tweet



Product Owners/Product Managers and distributed teams

Product Owners and Product Managers can be located in the primary location, creating additional responsibilities.



Other ART roles



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



Product Management owns, defines, and prioritizes the program backlog.



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



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



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

Exercise: Facilitating PO/PM collaboration



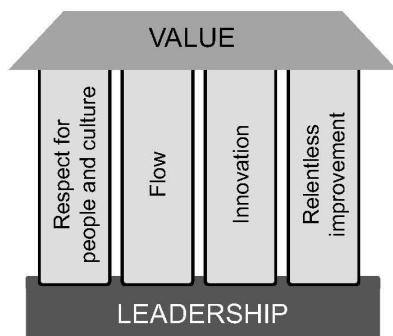
- Are your Product Owners and Product Managers effectively communicating?
 - What is going well?
 - What is going not so well?
- Be prepared to discuss



2.2 Embrace a Lean-Agile Mindset

Embrace Lean-Agile values

House of Lean



Value in the shortest sustainable lead time

Agile Manifesto

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- Individuals and interactions** over processes and tools
- Working software** over comprehensive documentation
- Customer collaboration** over contract negotiation
- Responding to change** over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

Agile Manifesto

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

 agilemanifesto.org/principles.html

SCALED AGILE® © Scaled Agile, Inc.

2.15

Agile Manifesto

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

 agilemanifesto.org/principles.html

SCALED AGILE® © Scaled Agile, Inc.

2.16

Exercise: Countdown 3, 2, 1!



- ▶ Discuss the **three** most important Agile Manifesto principles to the PO and PM roles, write them on a flip chart or whiteboard
- ▶ On the index card, write down **two** other Agile Manifesto Principles that you find important to practice for success in your organization
- ▶ Share the **one** Agile Manifesto Principle that will help you the most right now back in your organization, explain why



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

SAFe Lean-Agile Principles detail

#1

Take an Economic View

If the solution doesn't meet the customer's or systems builder's economic goals, then the long-term viability of the solution is suspect.

#2

Apply Systems Thinking

Systems thinking takes a holistic approach to solution development; one that incorporates design, development, deployment, and maintenance of the system itself.

#3

Assume variability; preserve options

Variability is not inherently bad or good. Rather, it is the economics associated with the timing and type of variability that determines the outcomes.

SAFe Lean-Agile Principles detail (cont.)

#4

Build incrementally with fast, integrated learning cycles

Cadence-based integration points become the primary focus of the systems builder via a development process and a solution architecture that is designed in part for that specific purpose.

#5

Base milestones on objective evaluation of working systems

The system can be measured and assessed, and evaluated by the relevant stakeholders frequently, and throughout the solution development life cycle.

#6

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

To achieve the sustainably shortest lead time, Lean systems builders strive to achieve a state of continuous flow, whereby new system capabilities move quickly from concept to cash.

SAFe Lean-Agile Principles detail (cont.)

#7

Apply cadence, synchronize with cross-domain planning

Cadence makes routine that which can be routine, so the intellectual capacity of knowledge workers can be devoted to managing the variable parameters.

#8

Unlock the intrinsic motivation of knowledge workers

Lean-Agile Leaders operate within a relatively new, fundamental truth—the “management” of knowledge workers is an oxymoron.

#9

Decentralize decision-making

Decentralized decision-making reduces delays, improves product development flow and throughput, and enables faster feedback and more innovative solutions.

Exercise: Divide and conquer teach back



In groups, learn about three SAFe Principles together, and then teach it back.

- #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



Exercise: Focus on #9 - Let's play Delegation Poker!



Centralize decisions that have economies of scale, decentralize all others. Use this game to learn how to decentralize.

- ▶ Delivers value in the shortest sustainable lead time
- ▶ Reduces delays
- ▶ Improves product development flow and throughput
- ▶ Enables faster feedback and more innovative solutions
- ▶ Increases empowerment

See Appendix B for
Delegation Poker cards you
can cut out and use



Source: <https://management30.com/product/delegation-poker/>

Exercise: Your Best Practices!



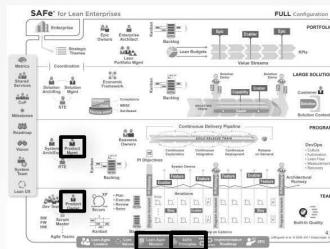
- ▶ Using the table on page 39 of your workbook, evaluate your organization on all nine SAFe Principles, in terms of how you are “Living the Values”
- ▶ Write one or two actions you will take to increase your intentions to live the SAFe values and principles



Lesson summary

In this lesson, you:

- ▶ Connected the Product Owner and Product Manager roles
- ▶ Embraced a Lean-Agile Mindset



Suggested Scaled Agile Framework reading:

- "Product Management" article
- "Product Owner" article
- "SAFe Principles" article
- "Lean Agile Mindset" article

Exercise: This lesson's key learnings



Summarize key learnings and insights from this lesson in your workbook.



Lesson 3

Collaborating with Lean Portfolio Management

1. Applying SAFe in the Lean Enterprise
2. Relating a Lean-Agile Mindset to the PO/PM Roles
- 3. Collaborating with Lean Portfolio Management**
4. Continuously Explore Customer Needs
5. Executing the Program Increment
6. Defining the PO/PM Roles and Responsibilities
7. Creating your PO/PM Action Plan

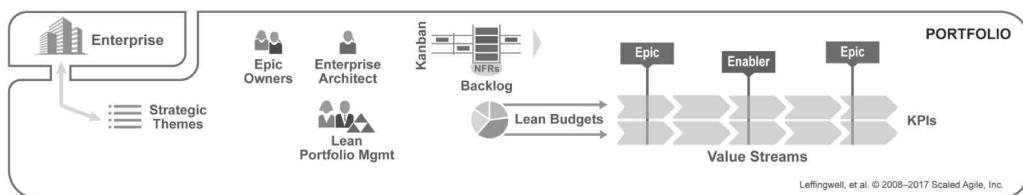
SAFe® Course: Attending this course gives learners access to the SAFe Product Owner/Product Manager exam and related preparation materials.

Learning objectives

- 3.1 Collaborate with the Lean Portfolio Management function
- 3.2 Detail the role of the Epic Owner
- 3.3 Use Lean Startup cycles to foster innovation
- 3.4 Develop Epic Hypothesis Statements
- 3.5 Contribute to a Lean Business Case
- 3.6 Visualize flow using a Portfolio Kanban
- 3.7 Elaborate Features for the Minimal Viable Product (MVP)

3.1 Collaborate with the Lean Portfolio Management function

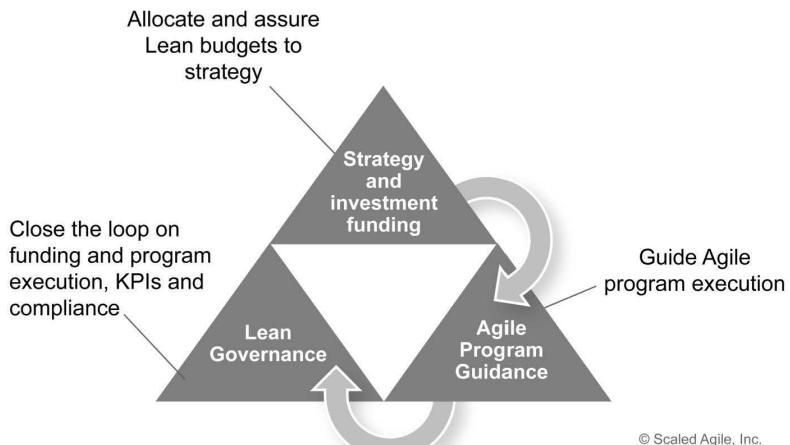
SAFe Portfolio Level



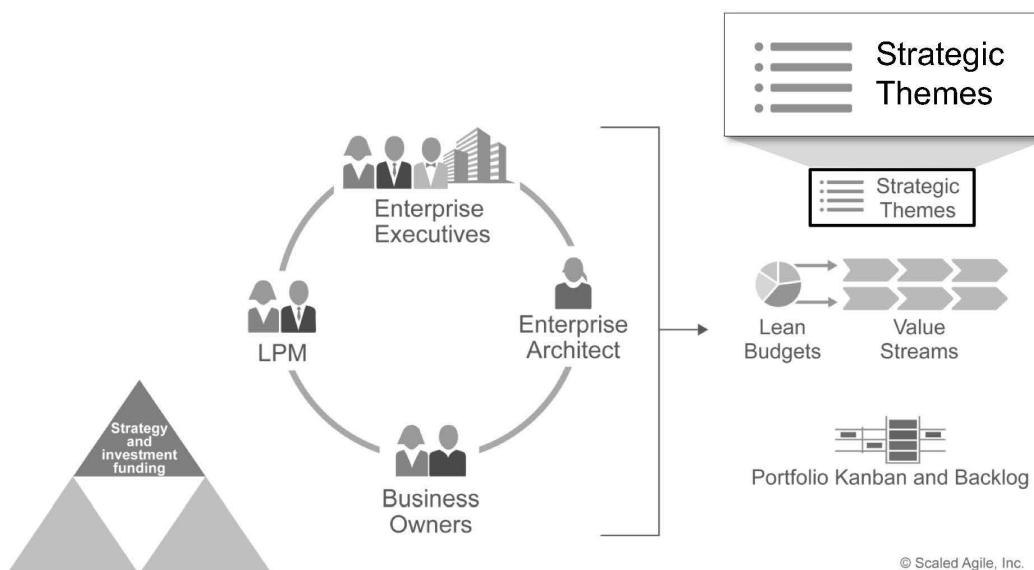
- ▶ Organized around the flow of value
- ▶ Lean Budgets empower decision makers
- ▶ Kanban system provides portfolio visibility and WIP limits
- ▶ Enterprise Architect guides larger technology decisions
- ▶ Objective metrics support governance and improvement
- ▶ Value delivery via Epics
- ▶ Strategic Themes

Lean Portfolio Management (LPM) function

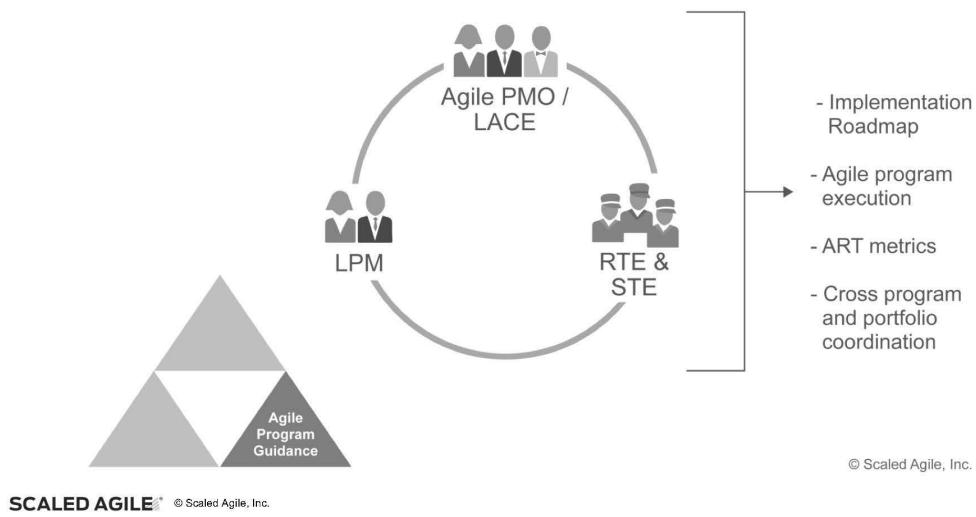
LPM is a function that is responsible for strategy and investment funding, Agile program execution, KPIs, and compliance.



Strategy and investment funding collaboration



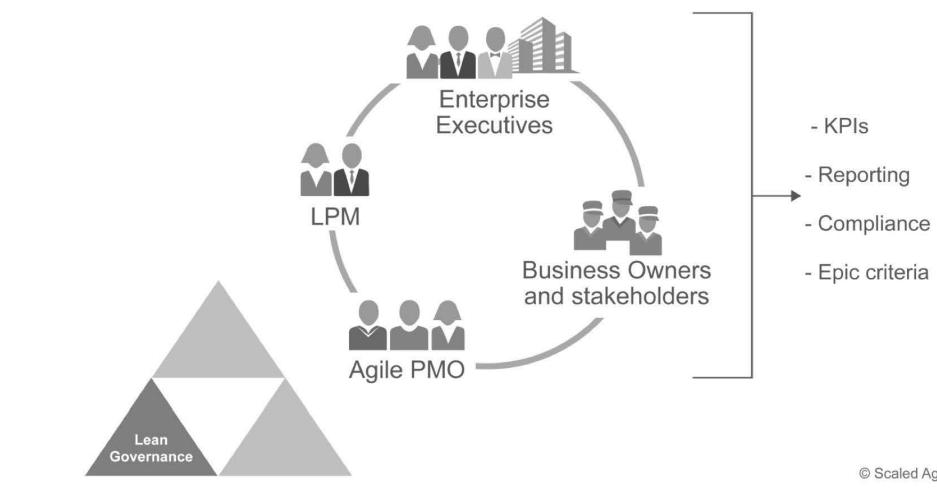
Agile Program Guidance collaboration



SCALED AGILE® © Scaled Agile, Inc.

3.7

Lean Governance collaboration



SCALED AGILE® © Scaled Agile, Inc.

3.8

Moving to a Lean Portfolio Management function

SAFe provides transformational patterns to move from traditional mindsets to a Lean Portfolio Collaboration.

From Traditional Approach	To Lean Approach
#1 Centralized control	➡ Decentralized decision-making
#2 Project overload	➡ Demand management; continuous value flow
#3 Detailed project plans	➡ Lightweight, Epic-only Lean Business Cases
#4 Centralized annual planning	➡ Decentralized, rolling-wave planning
#5 Work breakdown structure	➡ Agile estimating and planning
#6 Project-based funding and control	➡ Lean budgeting and self-managing Agile Release Trains
#7 Waterfall Milestones	➡ Objective, fact-based measures and Milestones

Exercise: Double up activity

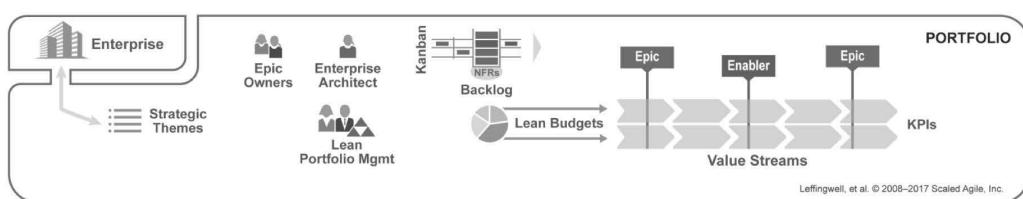


- ▶ Brainstorm your two biggest potential cultural impediments you expect to encounter
- ▶ Take turns by tossing a ball around the team. Share your two concerns with the group, along with any ideas for resolution you have
- ▶ Each participant asks for input from at least two group members to give them ideas based on their own organizational experiences



3.2 Detail the role of the Epic Owner

The role of the Epic Owner



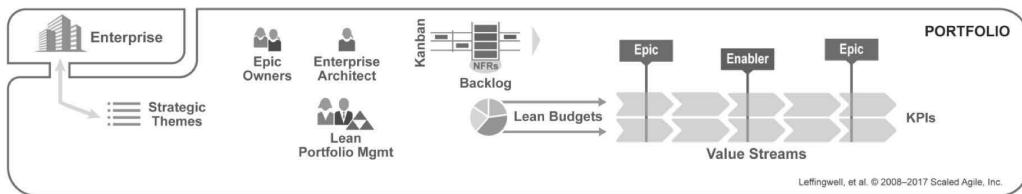
Leffingwell, et al. © 2008–2017 Scaled Agile, Inc.

The Epic Owner is responsible for driving individual Epics from identification through the analysis process of the Portfolio Kanban system and on to the go/no-go decision, then all the way through implementation.

- When an Epic is accepted for implementation, the Epic Owner works with the ART development teams and Product Management to initiate the development activities



The role of the Epic Owner (Cont.)

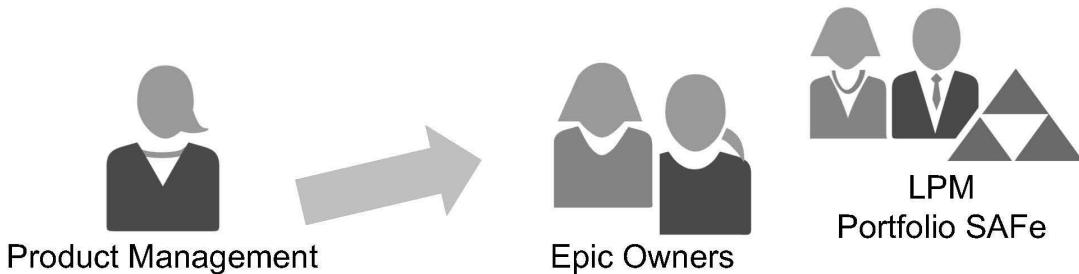


- ▶ In SAFe, this is a role rather than a job title. It can be Product Manager, Program Manager, Architect, Engineer, etc.
- ▶ Once successfully initiated, the Epic Owner may have some ongoing responsibilities for stewardship and follow-up



This could be you!

- ▶ In certain situations, the Product Manager or Product Owner might serve as an Epic Owner, or collaborate with an Epic Owner, and execute against a hypothesis
- ▶ It's important to understand how these close roles interact

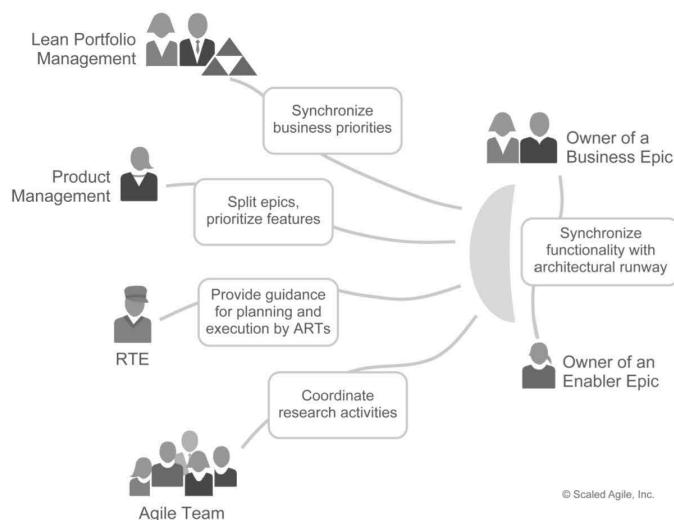


Responsibilities: Preparing the Epic

- ▶ Work with stakeholders and subject matter experts to define the Epic and its hypothesis statement, establish the cost of delay, and identify business sponsors
- ▶ Work with development teams to size the Epic and provide input for economic prioritization
- ▶ Define Epic outcomes hypothesis and MVP
- ▶ Guide the Epics through the Portfolio Kanban system and create the Lean Business Case
- ▶ Present the Epic, including the business case, to the LPM for go/no-go decision

Epic

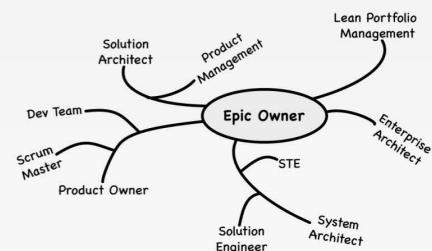
The collaborative nature of the Epic Owner role



Exercise: Your key collaborators in your context



- With your group, use a white board or flip chart to identify your key collaborators
- Using the previous slides on pages 47-49 in your workbook as a guide, identify key roles that need to collaborate with the Epic Owner
- Use your business context to create a Epic Owner mind map



Exercise: Share your stakeholder mind map



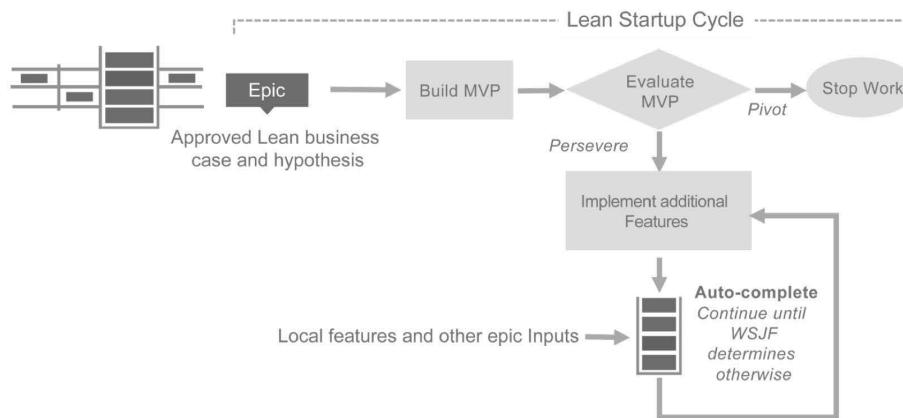
- In table groups, share your Epic stakeholder mind map
- Discuss your collaboration model and how you would engage your stakeholders



3.3 Use Lean Startup Cycles to foster innovation

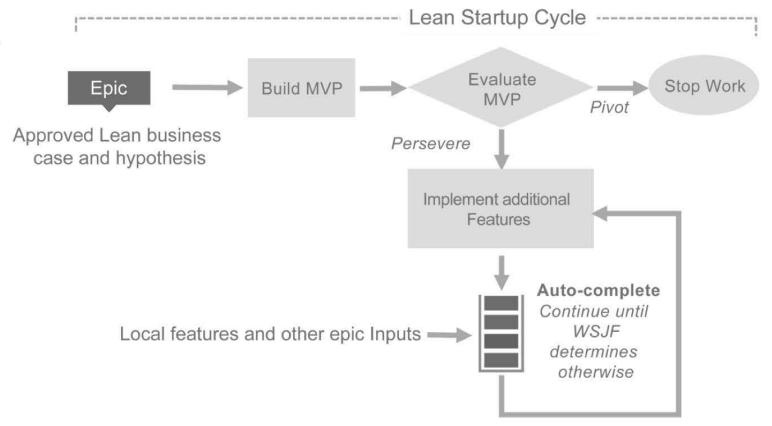
Foster innovation with the Lean Startup Cycle

Big-Up-Front Design (BUFD), along with big-up-front financial commitment, is a poor way to foster innovation.



Foster innovation with the Lean Startup Cycle (Cont.)

Epics deserve additional investment ... but not a fully committed investment up front.



SCALED AGILE® © Scaled Agile, Inc.

3.21

Exercise: Compare and contrast



- ▶ Compare the Lean Startup Cycle with your current Epic process
- ▶ In your workbook, detail the patterns and habits that may need to change in your enterprise
- ▶ Share the one pattern or habit that you would propose changing first



SCALED AGILE® © Scaled Agile, Inc.

3.22

3.4 Develop Epic Hypothesis Statements

Epic Hypothesis Statement example template

Epic Hypothesis Statement

For our community of banking consumers, **who** often login to our site through multiple devices or partner web sites, **the** Single Sign On Epic **is a** mechanism for the consumer **that** allows a secure login through partner websites and mobile devices and provides seamless integration through our valuable partner network.

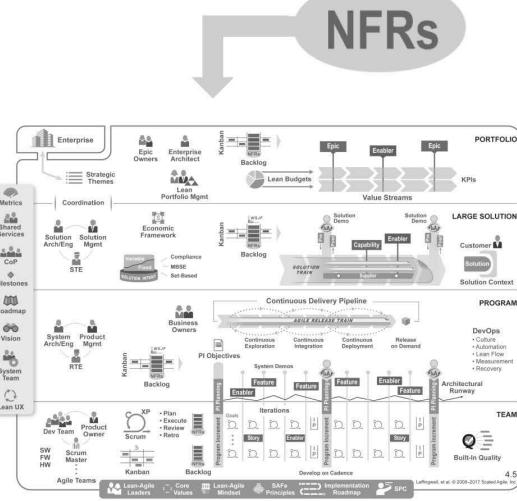
Unlike our competitors who redirect consumers to their site for separate logins, **our** single sign on experience will provide an integrated solution for our customers that will allow a unified banking experience.

Outcomes hypothesis:	Consumers will use single sign-on through our top three partner browser and mobile app capabilities
Leading indicators:	Multiple device access per user account increases, patterns of top devices become visible, separate login per account decreases. Baseline created
NFRs:	No more than 2 seconds latency, tested with all current SSO protocols

NFRs are key architecture concerns

Nonfunctional Requirements (NFRs) are system qualities that support end-user functionality and system goals.

- ▶ Nonfunctional requirements are associated with backlogs at all four configurations of SAFe
- ▶ Sometimes known as the “ilities”—reliability, usability, scalability, maintainability, etc.
- ▶ You may need to create backlog items to implement or evolve an NFR



SCALED AGILE® © Scaled Agile, Inc.

3.25

NFR types

Accessibility	Documentation	Operability
Audit and Control	Disaster Recovery	Performance
Availability	Efficiency	Reliability
Backup	Environmental Protection	Response Time
Capacity, Current and Forecast	Escrow	Robustness
Certification	Extensibility	Safety
Compliance	Failure Management	Scalability
Config. Management	Fault Tolerance	Supportability
Dependency on Third Party	Interoperability	Testability
Deployment	Maintainability	Usability

Source: Wikipedia

SCALED AGILE® © Scaled Agile, Inc.

3.26

Exercise: Draft an Epic Hypothesis Statement

SEE
WORKBOOK
PAGE
57

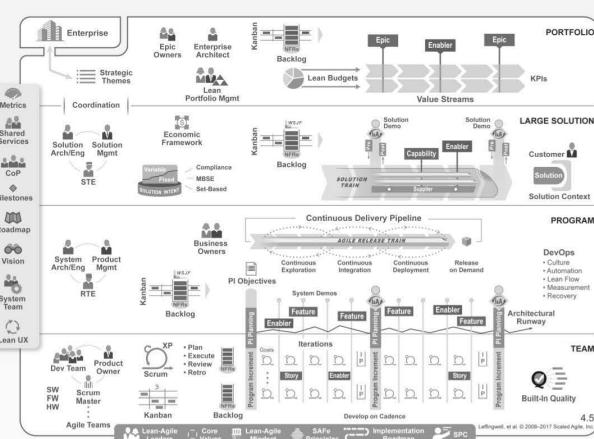
- ▶ In your context, think about something that is of truly Epic nature, a large project, with possible multiple ARTs
- ▶ Articulate the Epic using the Epic Hypothesis Statement format
- ▶ Share a statement with the class

Epic Hypothesis Statement	
For	<customers>
who	<do something>
the	<solution>
is a	<something – the "how">
that	<provides this value>
Unlike	<competitor, current solution, or non-existing solution>
our solution	<does something better – the "why">
Outcomes hypothesis:	:
Leading indicators:	:
NFRs:	:

PREPARE SHARE

Exercise: Top 3 to 5 Portfolio Epic NFRs

SEE
WORKBOOK
PAGE
58



Portfolio
NFRs

PREPARE SHARE

3.5 Contribute to a Lean Business Case

Epics deserve a Lean Business Case

Investment in Epics is a serious matter; analyses and informed decision-making are crucial.

- ▶ “Just the right amount” of analysis
- ▶ Avoid over-specificity
- ▶ Define the outcomes hypothesis
- ▶ Understand implementation impact
- ▶ Develop incremental implementation strategy
- ▶ Gain approval

SCALED AGILE® Lean Business Case		
Import Products, Programs and Services (Identify products, programs, services, teams, departments, etc. that will be impacted by this epic)		
Impact on Sales, Distribution, Deployment (Describe any impact on how the product is sold, distributed, or deployed)		
Analysis Summary (Brief summary of the analysis that has been formed to support this epic)		
Estimated Ship Point (MSP) (Estimated day point for the MSP of the epic)		
Type of Return (Type of return for the epic, e.g., Adding, Reducing, Reversing, Reversing and Adding)		
Type of Return (Market share, increased revenue, improved productivity, new market service, etc.)		
Outcomes Hypothesis (Provide recommendations for where the epic should go)		
Estimated Development Timeline (Start Date / Estimated start date)		
Incremental Implementation Strategy (Epics are defined as a single whole, but each epic usually details its potential strategies. Many parts of this guide discuss any concerns for sequencing the epic and managing its checkpoints.)		
Sequencing and Dependencies (Describe any concerns for sequencing the epic and managing its checkpoints)		
Identify potential resources or checkpoints for review		
Attachments:		
SCALED AGILE® Lean Business Case		
Epic Name: _____ Future Epic Name: _____ Epic Owner: _____ (Give name for the Epic) (Give the Epic owner's name) (Give the name of the Epic Owner)		
Epic Description: (Consider using the Epic Hypothesis Statement in the Epic article as a starting point for a description of the epic.)		
Outcomes Hypothesis: (Describe the outcomes of the epic will be measured for, for example, 50% increase in adoption under 25, Availability increases from 97% to 99.7%, Demographic switch 50% of Revenue related)		
Leading Indicators: (Identify key metrics or monitoring metrics to provide leading indicators of the outcome hypothesis, for example, a revenue change in purchase demographics within 90 days of feature release)		
In Scope	Out of Scope	Nonfunctional Requirements
• -	• -	• -
• -	• -	• -
• -	• -	• -
Minimum Viable Product (MVP) Features		
Additional Potential Features		
Sponsors: (List key business sponsors who will be supporting the initiative)		
Users and Markets Affected: (Describe the user community and any markets affected)		

Exercise: Start your own Lean Business Case



- ▶ Using the Lean Business Case template in your workbook (on page 60-61), create a business case for an Epic in your own context
- ▶ You can also download a Lean Business case template on the Scaled Agile Framework website at:
www.scaledagileframework.com/epic/

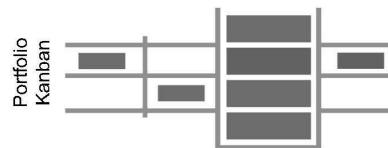
A screenshot of the Scaled Agile Lean Business Case template. It includes sections for Product Overview, Stakeholders, Business Requirements, Technical Requirements, and a Summary section.

3.6 Visualize flow using a Portfolio Kanban

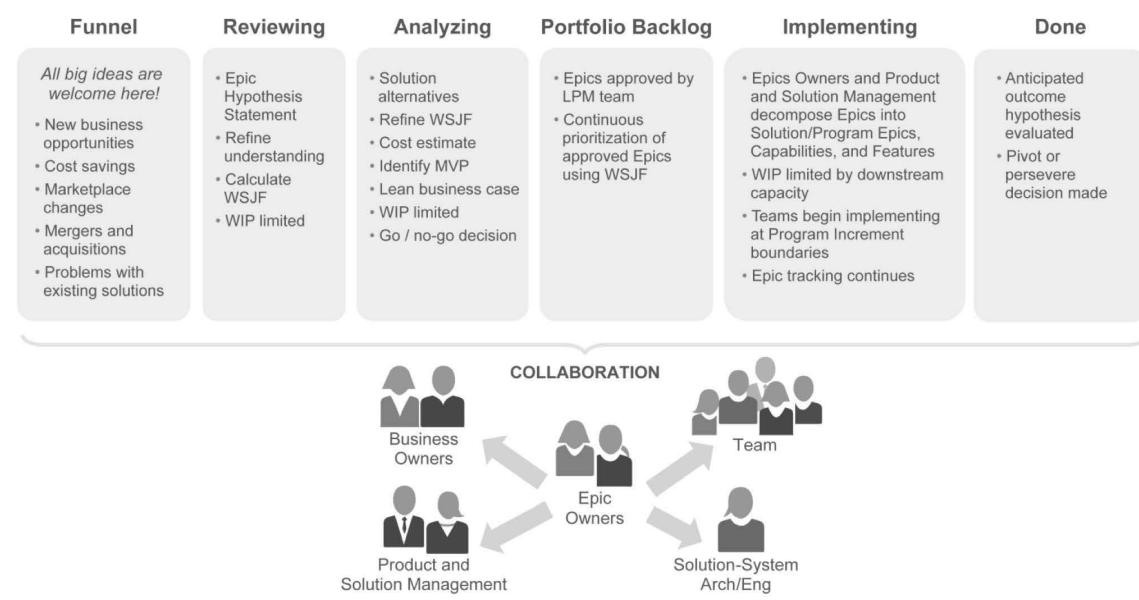
The Portfolio Kanban system

The Portfolio Kanban system manages the flow of Epics

- ▶ Makes largest business initiatives visible
- ▶ Brings structure to analysis and decision-making
- ▶ Provides WIP limits to ensure that the teams analyze responsibly
- ▶ Helps prevent unrealistic expectations
- ▶ Helps drive collaboration among the key stakeholders
- ▶ Provides a transparent and quantitative basis for economic decision-making



Prototypical Portfolio Kanban system



The “go/no go” Decision

When the Epic Owner or analyst thinks the analysis is sufficiently thorough, it is presented to the decision authority.

No Go



If the Epic requires more elaboration, it may be returned for more analysis ... NO GO = rejected!

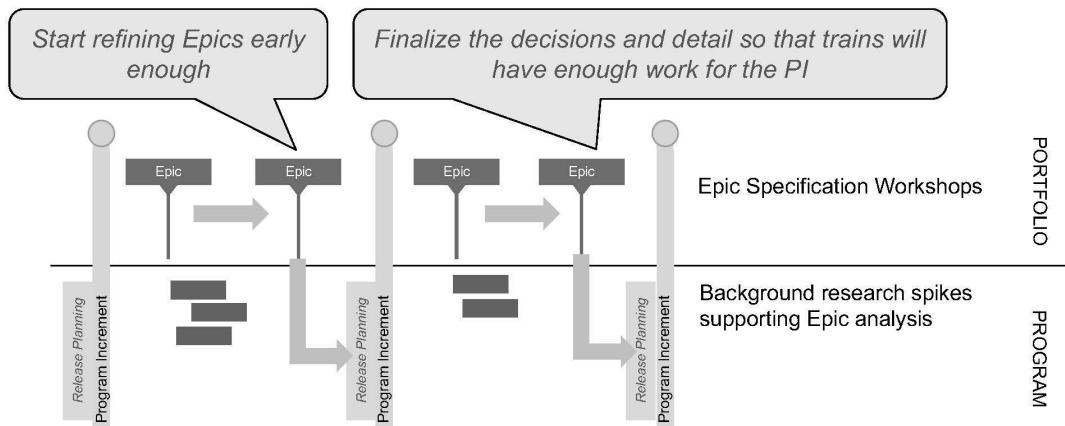
Go



When approved, the Epic is moved to the next state, which is the Portfolio Backlog, where it becomes a candidate for the Implementation state

Portfolio Kanban and cadence

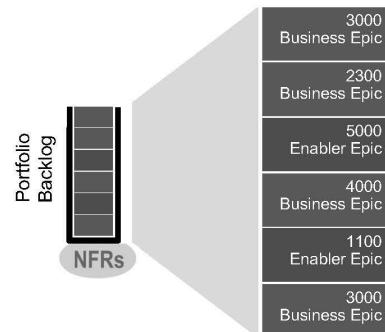
The Portfolio Kanban system is tightly connected to the cadence of the Agile Release Trains.



Portfolio Backlog

The Portfolio Backlog holds Epics approved for implementation.

- ▶ These Epics have made it through the Portfolio Kanban with 'go' approval
- ▶ Low-cost holding pattern for upcoming implementation work
- ▶ Sizing estimates are in story points
- ▶ Avoid excess WIP, await implementation capacity



Exercise: Place the Epic



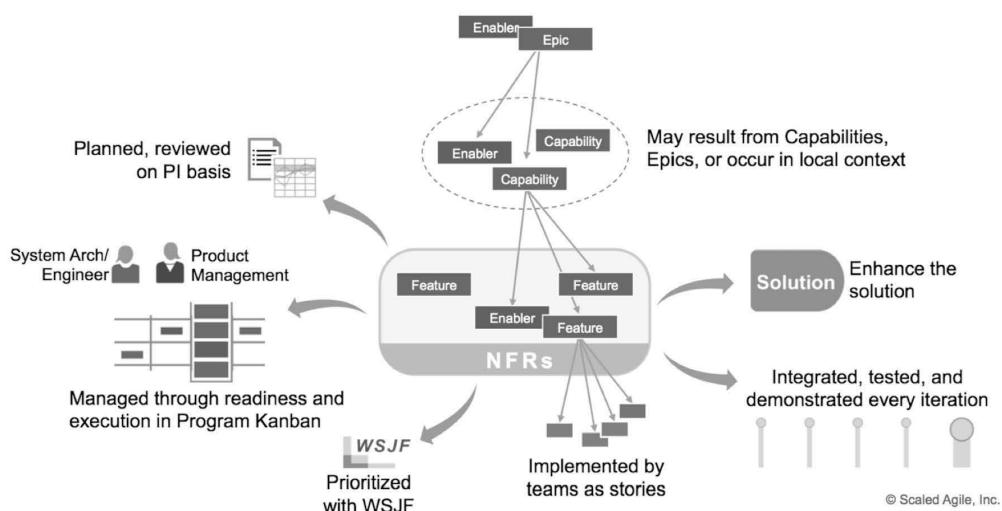
Where would you place your Epic in the Epic Kanban?

- ▶ Based on how much you know about your Epic, place your Epic within the appropriate state in the Kanban



3.7 Elaborate Features for the Minimal Viable Product (MVP)

Features in a SAFe context



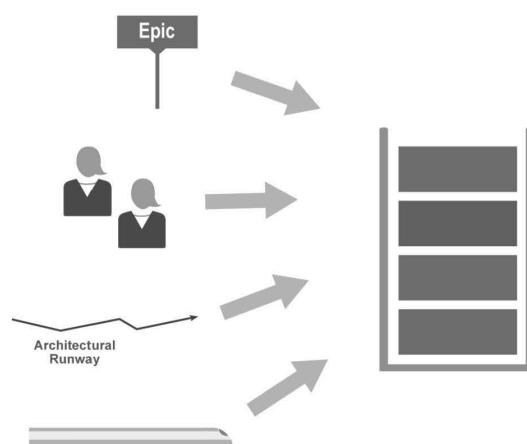
Describing Features

Feature	Benefit Hypothesis
Multi-factor Authentication	Enhance user security 98% by using multiple layers of identification: knowledge (something you know), possession (something you have), and inheritance (something you are)
Route Optimization	Improved Quality of Service 50% by using faster and more reliable connectivity
Hardware VPN Acceleration	Triple the number of encrypted packets delivered by secure WAN

Creating and Managing Features

Where do Features come from?

- Through preparation of Epics for MVP
- Product Managers
- Architectural Runway
- Emerge locally from the ART
- Other?



Create MVP Features with the Benefit Hypothesis



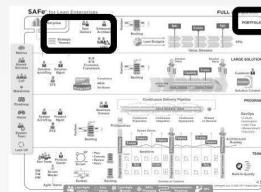
- ▶ Use the workbook template and describe some Features in the Feature/Benefit hypothesis format. Use an Epic you defined and create Features to support the MVP.
 - Ask yourself, should I consider anything from a Architectural Runway or Team perspective?
 - Does your Feature benefit hypothesis align with the Epic Hypothesis?
- ▶ Pair with someone else in the room to share and get feedback on your benefit hypothesis



Lesson summary

In this lesson, you:

- ▶ Collaborated with the Lean Portfolio Management function
- ▶ Detailed the role of the Epic Owner
- ▶ Used Lean Startup cycles to foster innovation
- ▶ Developed Epic hypothesis statements
- ▶ Contributed to a Lean Business Case
- ▶ Visualized flow using a Portfolio Kanban
- ▶ Built Features to support the Epic MVP



Suggested Scaled Agile Framework reading:
“Portfolio Level” and
“Strategic themes” abstracts

Exercise: This lesson's key learnings



Summarize key learnings and insights from this lesson in your workbook.



Lesson 4

Continuously Explore Customer Needs

1. Applying SAFe in the Lean Enterprise
2. Relating a Lean-Agile Mindset to the PO/PM Roles
3. Collaborating with Lean Portfolio Management
- 4. Continuously Explore Customer Needs**
5. Executing the Program Increment
6. Defining the PO/PM Roles and Responsibilities
7. Creating your PO/PM Action Plan

SAFe® Course: Attending this course gives learners access to the SAFe Product Owner/Product Manager exam and related preparation materials.

Learning objectives

- 4.1 Explore customer needs
- 4.2 Synthesize information for the Vision and Roadmap
- 4.3 Visualize Feature and Enabler flow using a Program Kanban
- 4.4 Prioritize the Program Backlog
- 4.5 Estimate and forecast the backlog

4.1 Explore customer needs

Moving the Epic from Portfolio Backlog to Implementing

As the Epic moves from the Portfolio Backlog to implementing, responsibility shifts.

Epic Owner drives ← → *Epic Owner shepherds*

Funnel

- All big ideas are welcome here!*
- New business opportunities
 - Cost savings
 - Marketplace changes
 - Mergers and acquisitions
 - Problems with existing solutions

Reviewing

- Epic Hypothesis Statement
- Refine understanding
- Calculate WSJF
- MVP limited

Analyzing

- Solution alternatives
- Refine WSJF
- Cost estimate
- Identify MVP
- Lean business case
- WIP limited
- Go / no-go decision

Portfolio Backlog

- Epics approved by LPM team
- Continuous prioritization of approved Epics using WSJF
- WIP limited
- Go / no-go decision

Implementing

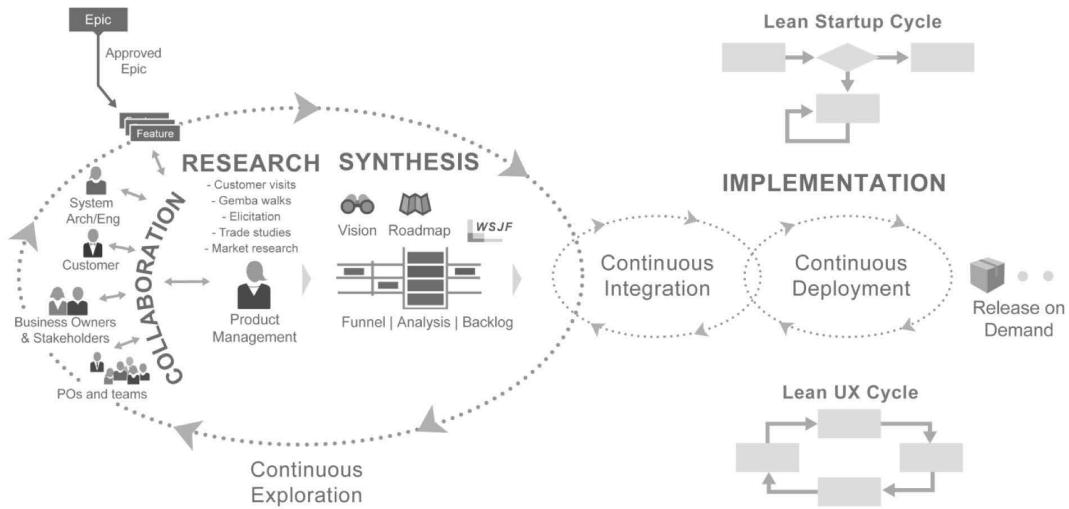
- Epics Owners and Product and Solution Management decompose Epics into Solution/Program Epics, Capabilities, and Features
- WIP limited by downstream capacity
- Teams begin implementing at Program Increment boundaries
- Epic tracking continues

Done

- Anticipated outcome hypothesis evaluated
- Pivot or persevere decision made

PO/PM stakeholders ← → *PO/PM drives*

Continuous Exploration



SCALED AGILE® © Scaled Agile, Inc.

4.5

Exercise: Continuous Exploration collaboration?



Select an approved Epic Hypothesis Statement and explore what collaboration and research the PM needs to engage in.

- ▶ Who are the PM key collaboration stakeholders?
- ▶ What are some key research resources and methods the PM can use?
- ▶ Be prepared to share



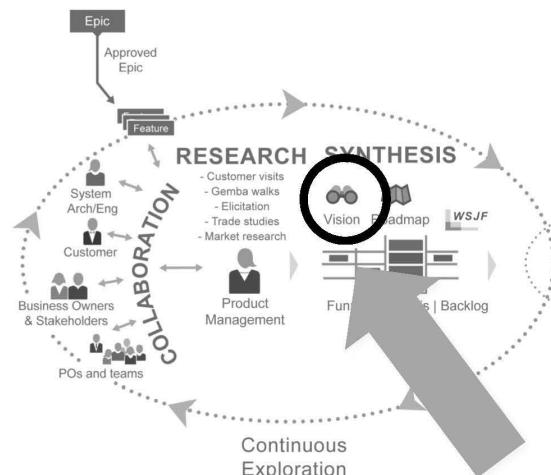
SCALED AGILE® © Scaled Agile, Inc.

4.6

4.2 Synthesize information for the Vision and Roadmap

Collaboration and Research Synthesis

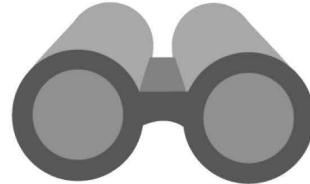
The Program Vision communicates strategic intent, and readies the Agile Release Train for launch.



Prepare the Vision

The Vision is a description of the future state of the Solution.

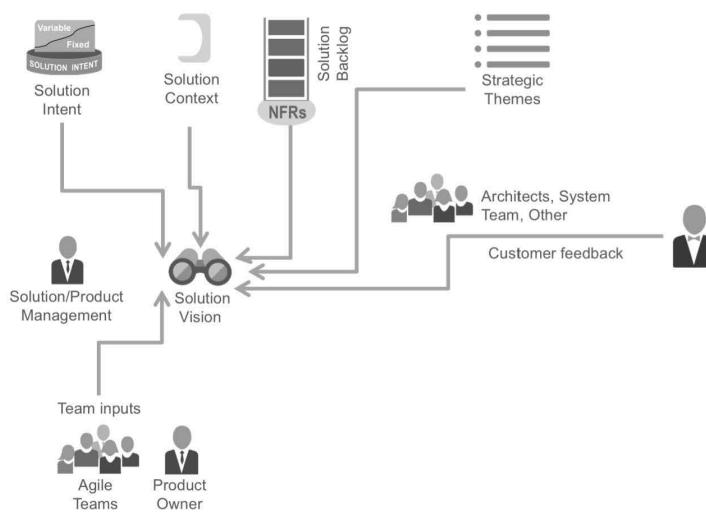
- ▶ Where are we headed with this Product or Solution?
- ▶ What problem does it solve?
- ▶ What Features and benefit hypothesis does it provide?
- ▶ For whom does it provide them?
- ▶ What Nonfunctional Requirements (performance, reliability, platforms, etc.) does the Solution deliver?



Common formats:

- ▶ Rolling-wave briefings
- ▶ Vision document
- ▶ Preliminary data sheet
- ▶ Draft press release

Inputs to the Solution and Program Vision



The Vision inspires action

It provides a longer-term context:

- ▶ How will our future Solution solve the larger Customer problems?
- ▶ How will it differentiate us?
- ▶ What is the future context in which our Solutions will operate?
- ▶ What is our current business context, and how must we evolve to meet this future state?



For a fun John Deere Vision video,
see <http://tinyurl.com/p5uloc5>

Vision: A postcard from the future



- Aspirational, yet realistic and achievable
- Motivational enough to engage others on the journey

Result: The teams start thinking about how to apply their strengths in order to get there

Switch: How to Change Things When Change Is Hard,
Chip Heath and Dan Heath, Broadway Books, 2010

Exercise: Visioning – Postcard from the future

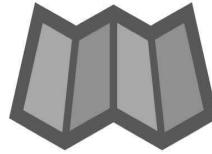


- ▶ In your group, pick one product or service that is currently being developed
- ▶ Imagine you are the customer. You have just received the product or service and are excited about using it!
- ▶ As the customer, write a postcard to the team describing:
 - What Features you like and why they are important to you
 - The quality of the product or service
- ▶ Share your postcard during a 'gallery walk'



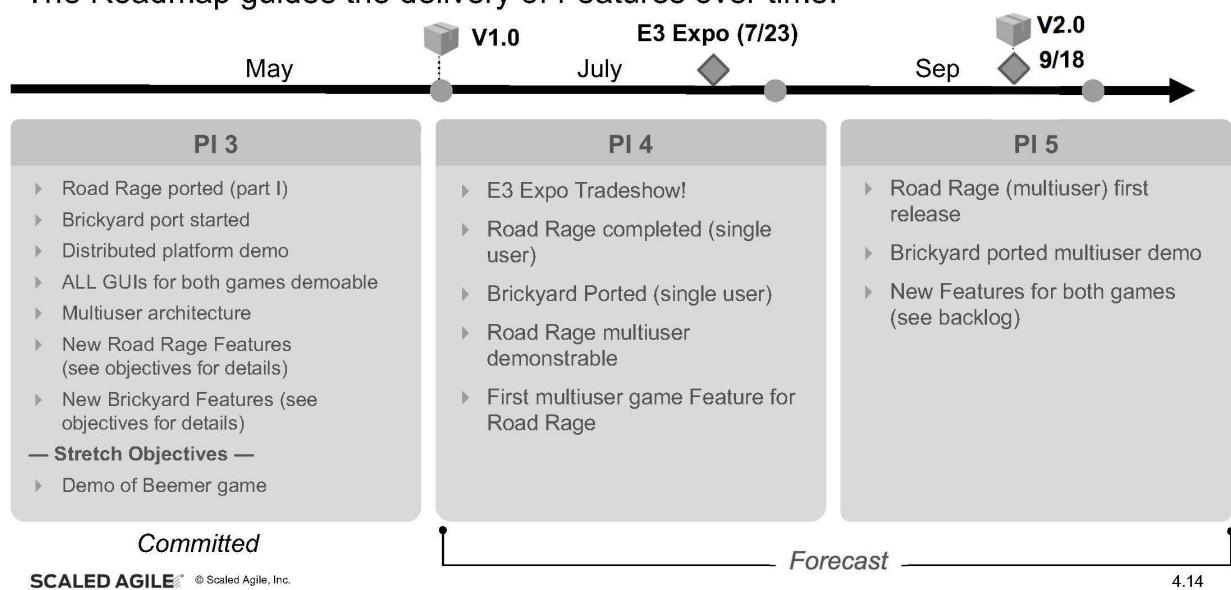
The SAFe Roadmap and its attributes

- ▶ Often derived from the Vision
- ▶ Created by the Product Manager
- ▶ Exists on the Spanning Palette in the SAFe Big Picture – and so can exist at all levels of the framework
- ▶ General plan of when Business and Enabler Features will be delivered over the next 3 PIs
- ▶ Only the first PI is committed; the others are a forecast which will be adjusted based on the learning from initial PI



Example Program Increment Roadmap

The Roadmap guides the delivery of Features over time.



Exercise: Construct a Roadmap

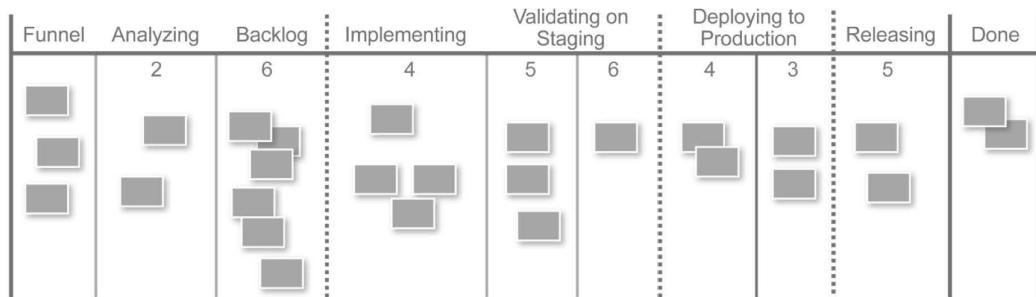


- At your table, use the template in your workbook to create a Roadmap of Features based on the Epic you previously created:
 - Pull in MVP Features
 - Identify additional Features from local context
 - Arrange them in the Roadmap
- Discuss how you will communicate your Roadmap to your customers and stakeholders
- Explore this question: Are Features always derived from Epics?
- Be prepared to share your Roadmap and communication plan



4.3 Visualize Feature and Enabler flow using a Program Kanban

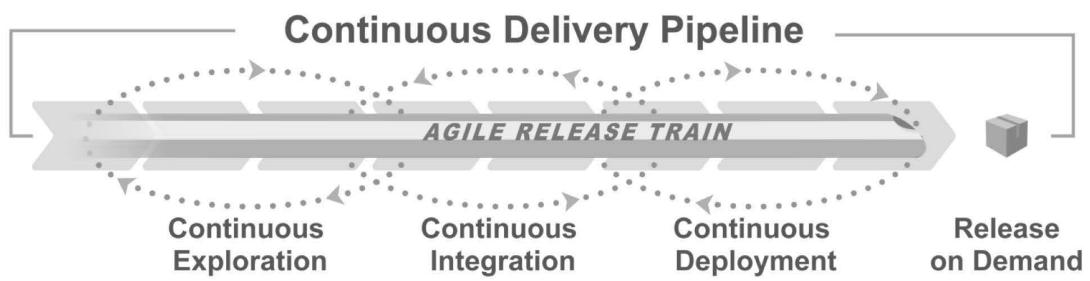
Kanban Systems



SCALED AGILE® © Scaled Agile, Inc.

4.17

Continuous Delivery Pipeline Learning Cycle

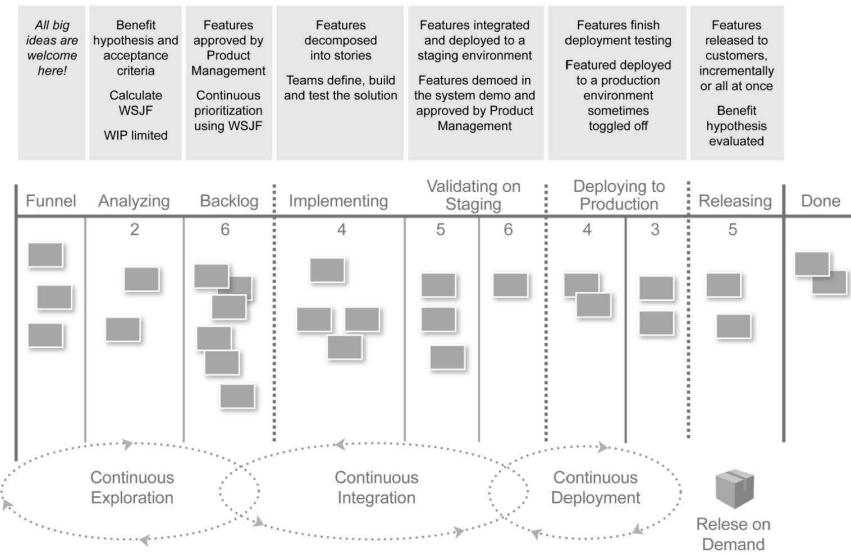


© Scaled Agile, Inc.

SCALED AGILE® © Scaled Agile, Inc.

4.18

The Program Kanban facilitates flow through the Pipeline



SCALED AGILE® © Scaled Agile, Inc.

4.19

Exercise: Create your Kanban and WIP limits



- ▶ Organize into groups of 2 to 4 people
- ▶ Choose one person's context and Features. Using the Kanban board in the workbook, fill in the initial WIP Limits in the circles
- ▶ Be prepared to present your thought process

PREPARE SHARE
5 min 5 min

SCALED AGILE® © Scaled Agile, Inc.

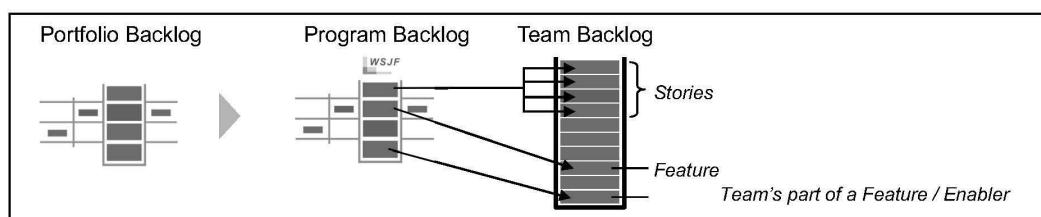
4.20

4.4 Prioritize the Program Backlog

Manage and prioritize the flow of work

Product Managers manage the flow of work from the Portfolio Kanban through the Portfolio Backlog and into the Program Backlog.

- ▶ Decompose Portfolio Epics from the Portfolio Backlog (as well as program-defined Epics) into Features
- ▶ Ensure that there are enough Features in the backlog, clearly defined with acceptance criteria, ready to be pulled by Agile Teams at all times
- ▶ Prioritize the Program Backlog using WSJF



Non-economic-based prioritization

Product Owners, Product and Solution Managers will often face non-economic-based prioritization.



HiPPo – Highest-paid person makes the decision.

“The Senior VP said we should do this project.”



Anti-pattern



Squeaky Wheel – The person who yells the loudest or makes the biggest promise of revenue.

“Fund my project and we will make a billion dollars!”



ROI – Making a decision based exclusively on an ROI metric (e.g., NPV, payback, etc.). Requires a sensitivity analysis to be relevant.

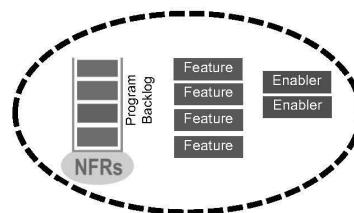
“The NPV indicates we will make a 30% profit.”

Prioritize Features for optimal ROI

In a flow system, job sequencing by Product Owners and Product Managers are key to economic outcomes.

To prioritize based on Lean economics, you need to know two things:

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

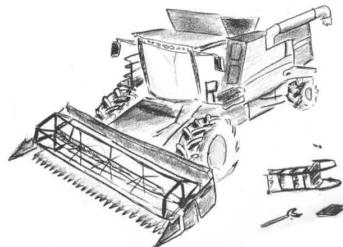


If you only quantify one thing, quantify the cost of delay.

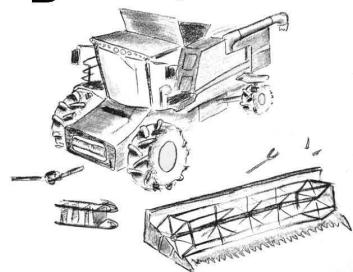
—Donald G. Reinertsen, *Principles of Product Development Flow*

Example with equal Cost of Delay: Which job first?

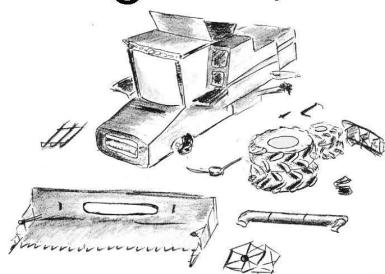
A \$\$, 1 day



B \$\$, 3 days

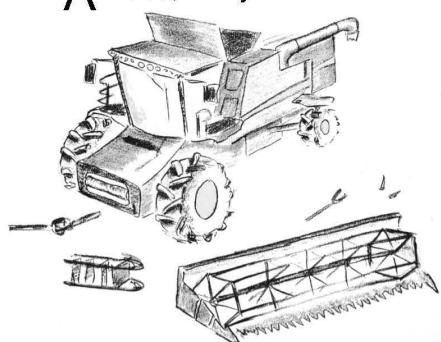


C \$\$, 10 days

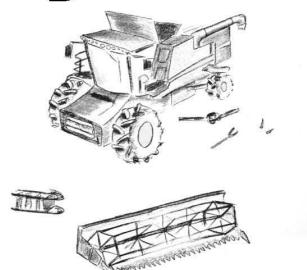


Example with equal duration: Which job first?

A \$\$\$, 3 days



B \$\$, 3 days

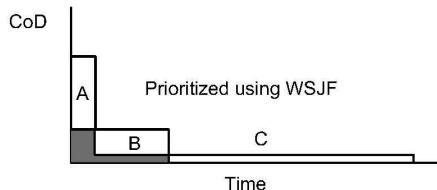


C \$, 3 days

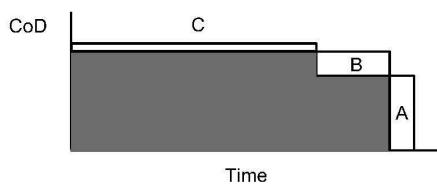


General case: Any CoD and duration

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



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



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

— Dark area: Total cost of delay

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

4.27

Components of Cost of Delay

User and business value

- Relative value to the customer or business
- ▶ They prefer this over that
 - ▶ Revenue impact?
 - ▶ Potential penalty or other negative impact?

Time criticality

- How user/business value decays over time
- ▶ Is there a fixed deadline?
 - ▶ Will they wait for us or move to another solution?
 - ▶ What is the current effect on customer satisfaction?

Risk reduction and opportunity enablement (RR & OE)

- What else does this do for our business
- ▶ Reduce the risk of this or future delivery?
 - ▶ Is there value in the information we will receive?
 - ▶ Enable new business opportunities?

Calculate WSJF with relative estimating

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

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

Exercise: Prioritizing the Program Backlog



- ▶ Select three Features from the “Create Features with benefits hypothesis” exercise and prioritize them using the WSJF template in your workbook
- ▶ Do one column at a time. Start by picking the smallest item and giving it a “1.” There must be at least one number “1” in each column of the template.
- ▶ Be prepared to share your WSJF prioritization



Exercise: Duration discussion



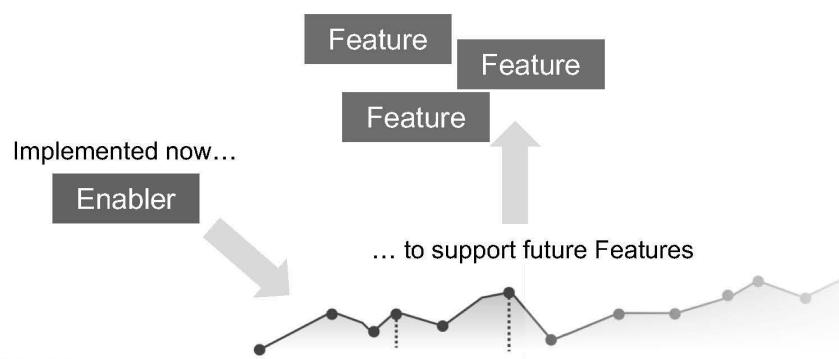
As a class, consider and discuss these three questions:

1. Is job size always a good proxy for duration?
2. When might it NOT be the case?
3. How would you adjust duration based on that case?



Partner with System Architect/Engineering

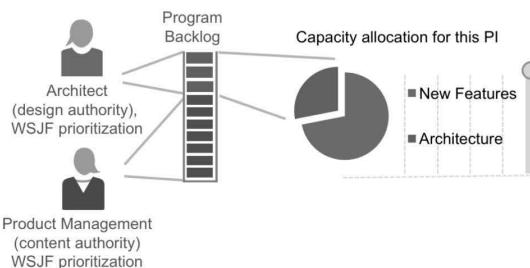
- ▶ Support Enabler items that provide sufficient Architectural Runway
- ▶ Work with System and Solution Architects/ Engineering to sequence technical infrastructures that will enable delivery of new business functionality



Define Features by type of service

Types of service provide a way to separate concerns, such that we can deliver the right mix of new Features and architecture evolution.

1. Determine how much service is to be allocated to each class
2. Establish policies to determine how work is performed for each defined class

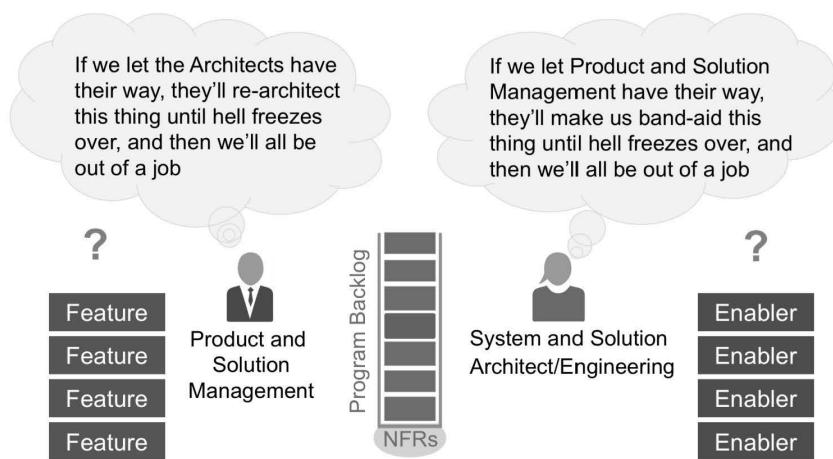


Architecture policies

1. We agree on the percentage of resources to be devoted to new Feature development vs. architecture at each boundary.
2. We agree that the architect has design authority and prioritizes the work in that class.
3. We agree that content authority (Product Management) prioritizes work in that class.
4. We agree to jointly prioritize our work based on economics.
5. We agree to collaborate so as to sequence work in a way that maximizes Customer value.

How much architecture?

Product Management collaborates with System Architects to build Architectural Runway.



Exercise: Draft your Capacity Allocation policy



-
-
-
- ▶ Consider how you would use Capacity Allocation in your enterprise
- ▶ In your workbook, draft a Capacity Allocation policy that you could bring back for discussion with your key collaborators
- ▶ Share your policies with a person next to you

PREPARE SHARE



4.5 Estimate and forecast the backlog

Relative estimating

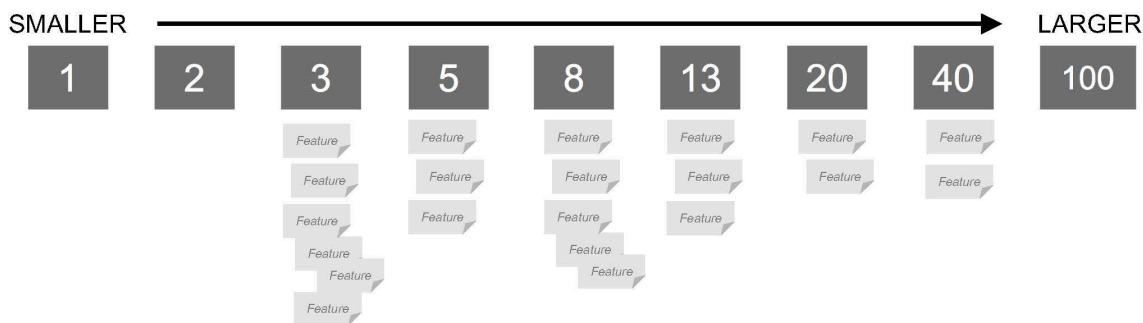
Agile Teams use story points and relative estimating to quickly arrive at estimates for size and duration for user stories

- ▶ Product Managers can use historical data to fairly quickly estimate the size of Features in story points as well
- ▶ Feature estimates can then be rolled up into epic estimates in the portfolio backlog
- ▶ Portfolio managers and other planners can use capacity allocation to estimate how long a portfolio epic might take under various scenarios

White Elephant sizing

When you need to estimate a lot of Features fast:

- ▶ Team members take turns putting Features under the size that they feel is appropriate or using their turn to move a Feature to a different estimate
- ▶ When all Features are estimated, the team reviews the Feature sizes and can make one final change



Exercise: Take a white elephant out for a run

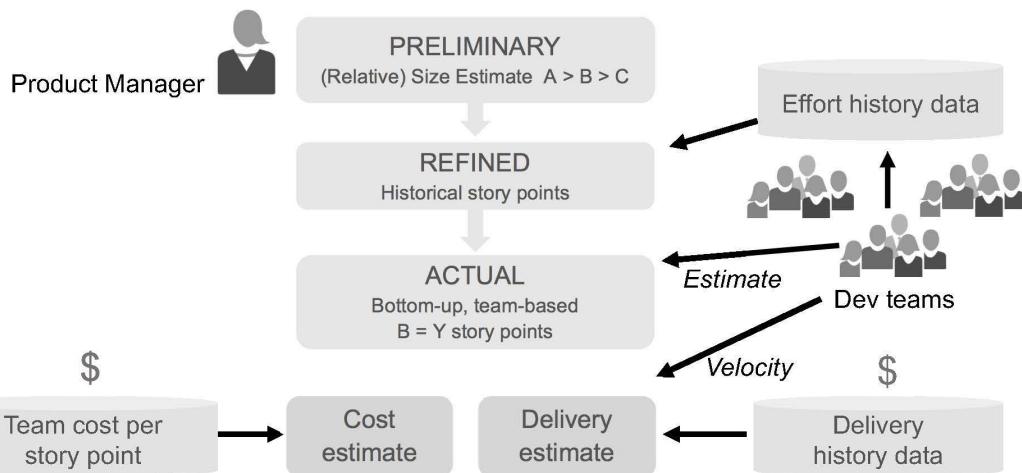


- ▶ Using the modified Fibonacci sequence (1, 2, 3, 5, 8, 13, 20, 40, 100), build a sizing board so that your group can estimate a series of Features
- ▶ Choose 3 to 5 Features in your group, and each person writes the Features on a sticky note
- ▶ Team members take turns putting the Features under the size that they feel is appropriate or using their turn to move a Feature to a different estimate
- ▶ When all Features are estimated, the team reviews the Feature sizes and can make one final change



Estimating Features effort

Estimating the effort needed to implement a Feature typically goes through a series of successive refinements.

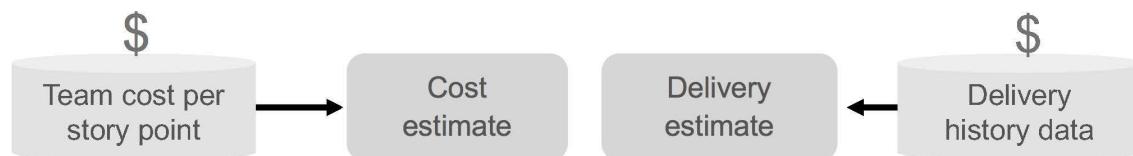


Estimating cost

Once the Feature has been estimated in story points, a cost estimate can be quickly derived.

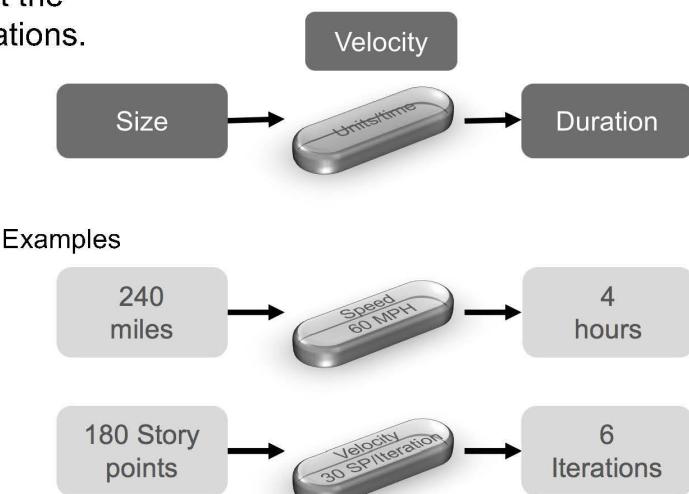
- ▶ Calculate the burdened cost for a team in an Iteration length
- ▶ Divide that by their PI velocity to get average cost per story point

Example: If a team has an average velocity of 40 points, and their cost is \$40,000 per Iteration, then each story point costs ~\$1,000



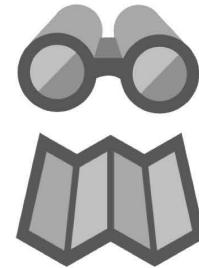
Using size to estimate duration

Establish velocity by looking at the average output of the last Iterations.



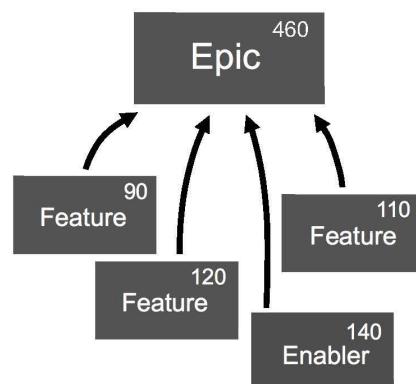
The business needs to forecast

- ▶ SAFe enhances enterprise adaptability, providing faster response to changing market opportunities
- ▶ Yet, the enterprise, its partners, and customers need to plan some sense of the future
- ▶ Estimating must:
 - Be fast and efficient as possible to be reasonably accurate
 - Support “what if” analysis of various implementation scenarios
- ▶ Traditional Work Breakdown Structure to task-level estimating binds the teams to waterfall practices



Estimating Epics in SAFe

1. Epics are broken down into potential Features during the Portfolio Kanban analysis stage
2. Potential Features are estimated in story points
 - Typically performed at the PM-System Architect level, based on history and relative size
 - Individual teams are engaged as necessary
3. Feature estimates are aggregated back into the Epic estimate as part of the lightweight business case



Exercise: Forecasting



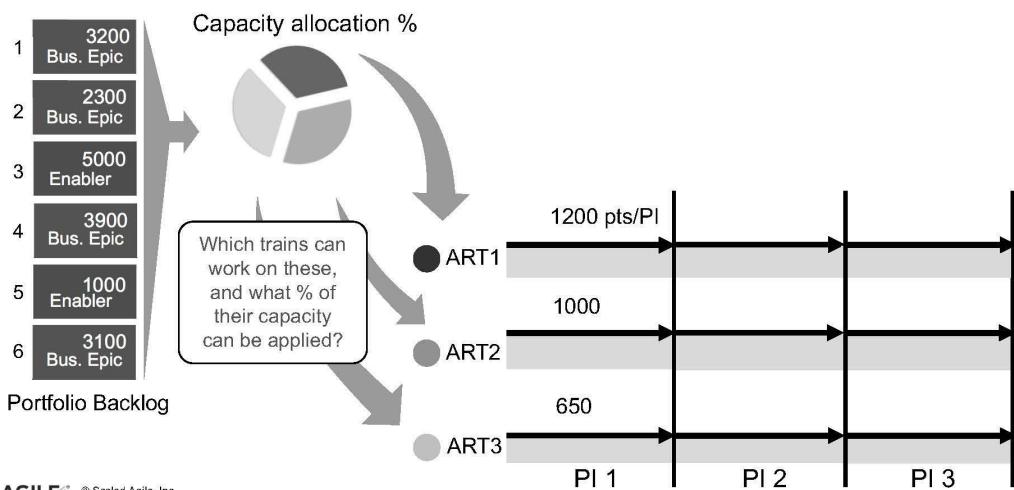
With your small group, look at the next slide (slide is also shown on page 102 in your workbook) and together, answer the following question:

- ▶ ART 2 is capable of doing Epic 5 by themselves, but they can only dedicate half of their capacity to that initiative. In which PI would you reflect this Epic on the Roadmap?



Forecasting from the Portfolio Backlog

Given knowledge of Epic sizes and ART velocities, applying “what if” capacity allocations informs decisions and forecasting.



Organizational readiness

How can you effectively ensure organizational readiness? Ask yourself these questions!

- ▶ Planning scope and context:
 - Is the scope (product, system, technology domain) of the planning process understood?
 - Do we know which teams need to plan together?
- ▶ Business alignment:
 - Is there reasonable agreement on priorities among the Business Owners?
- ▶ Agile teams:
 - Do we have Agile Teams?
 - Does each have dedicated developer and test resources and an identified Scrum Master and Product Owner?



Exercise: Are you ready to rock and roll?



The PM's responsibility is to enter PI Planning with a prioritized backlog.

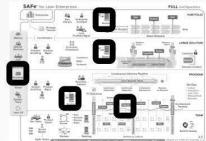
- ▶ What does a PM need to do to ensure org readiness and content readiness?
- ▶ How could the PO support the PM in org readiness and content readiness?



Lesson summary

In this lesson, you:

- ▶ Continuously explored customer needs
- ▶ Synthesized information for the Vision and Roadmap
- ▶ Visualized Feature and Enabler flow using a Program Kanban
- ▶ Prioritized the Program Backlog
- ▶ Estimated and forecasted the Backlog



Suggested Scaled Agile Framework reading: "Portfolio Backlog", "Program and Solution Backlogs", "Roadmap", and "Features and Capabilities" articles

Exercise: This lesson's key learnings



Summarize key learnings and insights from this lesson in your workbook.



Lesson 5

Executing the Program Increment

1. Applying SAFe in the Lean Enterprise
2. Relating a Lean-Agile Mindset to the PO/PM Roles
3. Collaborating with Lean Portfolio Management
4. Continuously Explore Customer Needs
5. Executing the Program Increment
6. Defining the PO/PM Roles and Responsibilities
7. Creating your PO/PM Action Plan

SAFe® Course: Attending this course gives learners access to the SAFe Product Owner/Product Manager exam and related preparation materials.

Learning objectives

- 5.1 Create alignment with PI Planning
- 5.2 Decompose Features into Stories
- 5.3 Plan the Iteration
- 5.4 Execute the PI
- 5.5 Release on Demand

5.1 Create alignment with PI Planning

Participate in PI planning

The Product Manager and Product Owners play key roles before, during, and after the PI Planning event.





Schroders PI Planning Event Kickoff

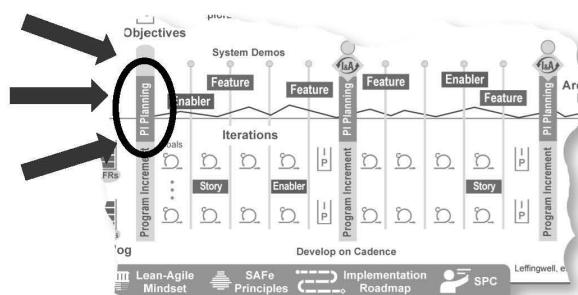
<https://vimeo.com/169066536>

SCALED AGILE® © Scaled Agile, Inc.

5.5

Inputs of PI Planning

- Inputs of PI Planning include:
- ▶ Business context
 - ▶ Roadmap and Vision
 - ▶ Top 10 Features from the single ART Program Backlog



SCALED AGILE® © Scaled Agile, Inc.

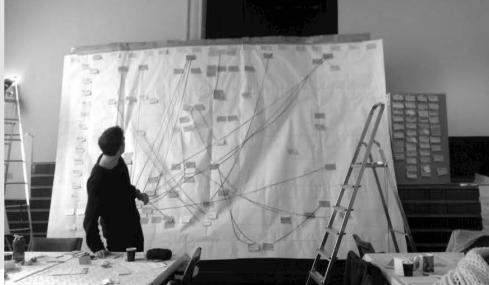
5.6

Successful PI Planning Outputs

A successful PI planning event delivers two primary outputs:

1. Committed PI Objectives (SMART)
2. Program Board

<u>Objectives for PI 1</u>	<u>Business Value</u>
<ul style="list-style-type: none">> Structured location and validation of locations> Build and demonstrate a proof of concept for context images> Implement negative triangulation by tags, companies, and people> Speed up indexing by 50%> Index 1.2 B more web pages> Extract and build URL abstracts	
<u>Stretch Objectives for PI 1</u>	
<ul style="list-style-type: none">> Fuzzy search by full name> Improve tag quality to 80% relevance	



5.7

SCALED AGILE® © Scaled Agile, Inc.

PI Planning preparation

Prior to the PI Planning event, the PM should collaborate with POs to:



Create/update the Vision.



Prepare and estimate the top 10 Features list from the Program Backlog.



Socialize the above artifacts with business, stakeholders, and architects to validate the Feature list and set expectations for the PI Planning meeting.

SCALED AGILE® © Scaled Agile, Inc.

5.8

What POs and PMs do during PI Planning – Day 1

- ▶ Communicate:
 - Program Vision
 - Present the top 10 Features
- ▶ Collaborate to decompose Features into Stories
- ▶ Negotiate scope
- ▶ Review draft PI plans and provide feedback
- ▶ Participate in management review of draft PI plans



You are here! Day 1 agenda

8:00-9:00	Business context		State of the business and upcoming objectives
9:00-10:30	Product/Solution Vision		Vision and prioritized Features
10:30-11:30	Architecture Vision and development practices		▶ Architecture, common frameworks, etc. ▶ Agile tooling, engineering practices, etc.
11:30-1:00	Planning context and lunch		Facilitator explains planning process
1:00-4:00	Team breakouts SoS	1 2 3 4	▶ Teams develop draft plans and identify risks and impediments ▶ Break Features into Stories ▶ Architects and Product Managers circulate
4:00-5:00	Draft plan review		Teams present draft plans, risks, and impediments
5:00-6:00	Management review and problem solving		Adjustments made based on challenges, risks, and impediments

Management review and problem-solving

At Day 1 end, management meets to make adjustments to scope and objectives based on the day's planning.

Common questions during the managers' review:

- ▶ What did we just learn?
- ▶ 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?



Used with permission of Hybris Software

Exercise: Problem Solving Workshop - PO and PM roles



Think about the Management Review and Problem-Solving Workshop.

- ▶ What activities are required for the PO?
- ▶ What activities are required for the PM?
- ▶ Capture ideas and share with the class



What POs and PMs do during PI Planning – Day 2

- ▶ Participate in final PI plan review
- ▶ Establish business value with Business Owners
- ▶ Accept Team Objectives
- ▶ Provide feedback on program risks
- ▶ Participate in confidence vote, rework (if applicable), and planning retrospective



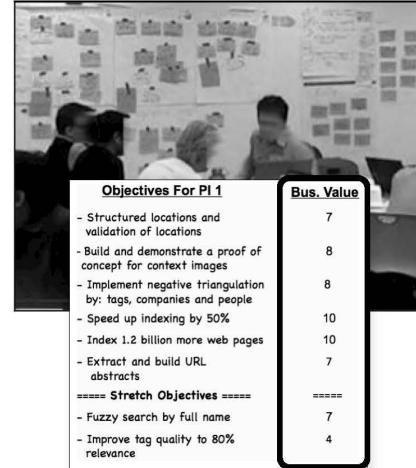
Day 2 agenda

8:00-9:00	Planning adjustments		Planning adjustments made based on previous day's management meeting
9:00-11:00	Team breakouts	1 2 3 4	<ul style="list-style-type: none">▶ Teams develop final plans and refine risks and impediments▶ Business Owners circulate and assign business value to team objectives
11:00-1:00	Final plan review and lunch		Teams present final plans, risks, and impediments
1:00-2:00	Program risks		Remaining program-level risks are discussed and ROAMED
2:00-2:15	PI confidence vote		Team and program confidence vote
2:15-???	Plan rework if necessary	1 2 3 4	If necessary, planning continues until commitment is achieved
After commitment	Planning retrospective and moving forward		<ul style="list-style-type: none">▶ Retrospective▶ Moving Forward▶ Final Instructions

Supporting the Team Objectives

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
- ▶ Stretch objectives provide the capacity and guard band needed to increase cadence-based delivery reliability



5.15

SCALED AGILE® © Scaled Agile, Inc.

Exercise: Supporting Team Objectives with Business Value



- ▶ Read the Guidance Article on “The Role of PI Objectives” (page 116 in your workbook)
- ▶ In your workbook, reason about how the POs and PMs will support the assignment of Business Value
- ▶ Be prepared to share your results

Objectives For PI 1		Bus. Value
- Structured locations and validation of locations		7
- Build and demonstrate a proof of concept for context images		8
- Implement negative triangulation by: tags, companies and people		8
- Speed up indexing by 50%		10
- Index 1.2 billion more web pages		10
- Extract and build URL abstracts		7
===== Stretch Objectives =====		=====
- Fuzzy search by full name		7
- Improve tag quality to 80% relevance		4



5.16

SCALED AGILE® © Scaled Agile, Inc.

Confidence vote: Team and Program Levels

After dependencies are resolved and risks are addressed, a confidence vote is taken at the Team and Program Levels.

'Fist of five' confidence vote

- ▶ Range of 1-5
- ▶ 1 = No confidence
- ▶ 5 = Very high confidence

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



SCALED AGILE® © Scaled Agile, Inc.

Used with permission of Standard Bank

5.17

Exercise: PO/PM day 1 and 2 PI Planning activities



Refer to your workbook as needed and summarize the things you need to be doing as a PO/PM:

- ▶ Before PI Planning in preparation
- ▶ During PI Planning on day 1
- ▶ During PI Planning on day 2



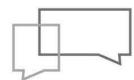
SCALED AGILE® © Scaled Agile, Inc.

5.18

What POs and PMs do after PI Planning



Update the Roadmap



Continue to collaborate with each other



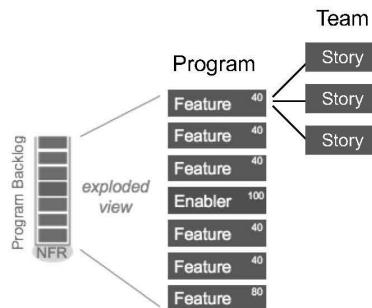
Prepare for the next PI!

5.2 Decompose Features into Stories

Feature decomposition

Feature breakdown into User Stories happens prior to, during, and after the PI Planning event.

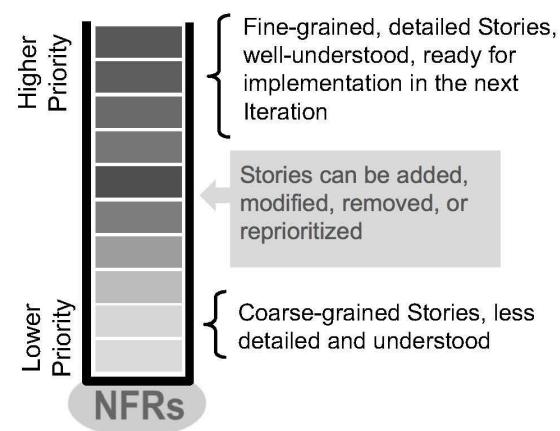
Before, During, and After



Features drive User Stories

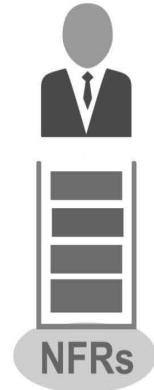
User Stories (from XP) replace traditional requirements expressions.

- ▶ The Team Backlog consists of backlog items, many of which are elaborated as User Stories that express the needs of the stakeholders
- ▶ User Stories are not requirements. They are negotiable expressions of intent.

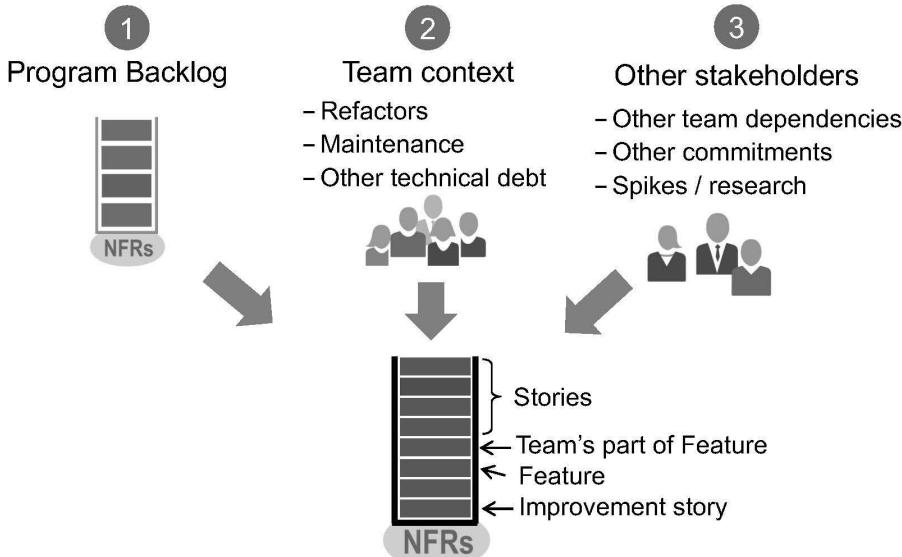


The Team Backlog

- ▶ Contains all the work the team needs to work on
- ▶ Contains User and Enabler Stories
 - User Stories provide customers with value
 - Enabler Stories build the infrastructure and architectures that makes User Stories possible
- ▶ Stories in the backlog are prioritized
- ▶ Stories for the next Iteration are more detailed than Stories for later Iterations
- ▶ Non-Functional Requirements (NFRs) are a constraint on the Backlog



Three primary sources of Team Backlog items



Backlog Item	Size
1. Login portal	5
2. Display minutes	3
3. Support ticket call	8
4. Remote login help	13
5. Update profile	2
6. Buy more time	8
7. Single sign-on	20
8. Log rotation	5
9. Timer display	3
10. Automatic logout	1
11. Usage warning	2
12. 300 logins/min	13
13. Location tracking	5
17. Intrusion detection	13
18. Update MySQL DB	20
19. Update Web Stack	13
20. Update Linux Kernel	8
21. Support Novel Auth	13
22. Support RADIUS Auth	8
23. Scan & Block Interface	40
24. AP Manager Interface	20

A Team Backlog

Backlogs bring stakeholder ideas together

Where Good Ideas Come From



<https://youtu.be/Mb0ssmoXG1I>

User Stories

- ▶ Containers for user or customer value
- ▶ Written using the following template:

As a <user role>, I want <activity> so that <business value>

- **User role** is the description of the person doing the action
- **Activity** is what they can do with the system
- **Business value** is why they want to do the activity

As a driver, I want to limit the amount of money before I fuel so that I can control my expenditure.

As a driver, I want a receipt after fueling so that I can expense the purchase.

As the Finance Department, we want to print receipts only for drivers who request them so that we can save on paper.

User Story guidelines — The 3 Cs

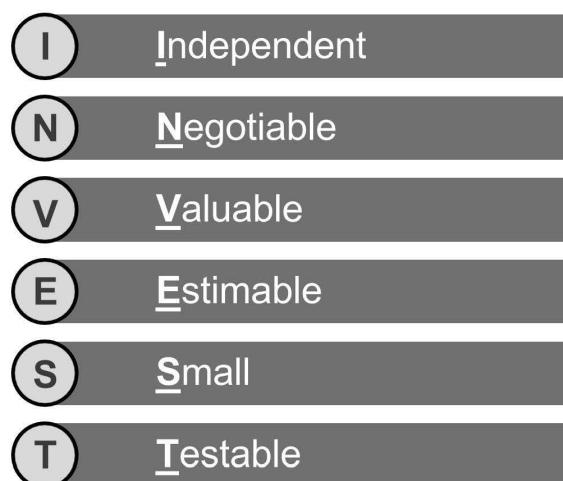
Card	Conversation	Confirmation
<p>Written on a card or in the tool and may annotate with notes</p> <p>As a spouse, I want a clean garage so that I can park my car and not trip on my way to the door</p>	<p>The details are in a conversation with the Product Owner</p> <p>What about the bikes?</p> <p>Oh yeah, hang the bikes</p>	<p>Acceptance criteria confirm the Story correctness</p> <ul style="list-style-type: none">▶ Tools have been put away▶ Items on the floor have been returned to the proper shelf▶ Bikes have been hung

SCALED AGILE® © Scaled Agile, Inc.

Source: 3 Cs coined by Ron Jeffries

5.27

INVEST in a good Story



SCALED AGILE® © Scaled Agile, Inc.

5.28

Enabler Stories

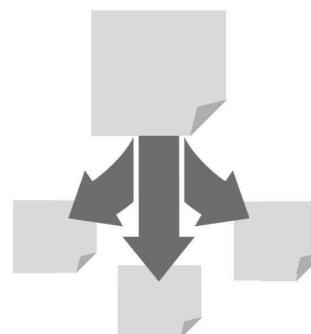
Enabler Stories build the groundwork for future User Stories.

There are generally four types of Enabler Stories:

1. Infrastructure – Build development and testing frameworks that enable a faster and more efficient development process
2. Architecture – Build the Architectural Runway, which enables smoother and faster development
3. Exploration – Build understanding of what the customer needs to understand prospective Solutions and evaluate alternatives
4. Compliance - Compliance enablers are used to schedule and manage specific compliance activities, including Verification and Validation (V&V), documentation and signoffs, and regulatory submissions and approvals.

10 patterns for breaking Features into Stories

- | | | | |
|---|--------------------------|----|------------------------|
| 1 | Work flow steps | 6 | Data 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 |



Exercise: Break Features into Stories



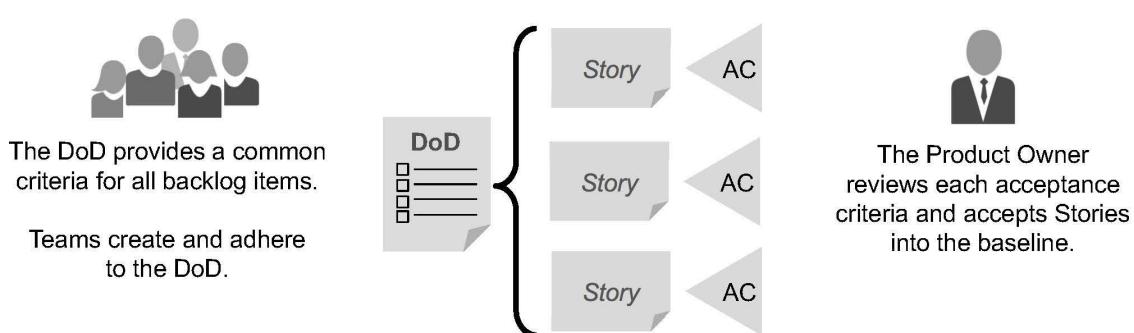
Using the information in your workbook (pages 127-131), break your Features into Stories that are small enough to fit into an Iteration.

- ▶ Break a Feature into at least five Stories
- ▶ Try to create Stories that are less than a week in size
- ▶ Identify spikes as needed
- ▶ You should have at least five Stories from your Feature
- ▶ See your workbook for examples



Accepting User Stories

A Story is accepted when it satisfies the Definition of Done (DoD) and is accepted by the Product Owner.



Acceptance criteria

- ▶ Acceptance criteria provide the details of the Story from a testing point of view
- ▶ Acceptance criteria are created by the team and the PO

As a driver, I want to limit the amount of money before I fuel so that I can control my expenditure.

Acceptance criteria:

1. The fueling process stops automatically on the exact value
2. I can stop fueling before the limit has been reached and will only be charged for the amount fueled

As a driver, I want to get a receipt after fueling so that I can expense the purchase.

Acceptance criteria :

1. Receipt includes: Amount fueled, Amount paid, Tax, Vehicle number, Date, Time

Exercise: Write acceptance criteria



- ▶ Write acceptance criteria for 2 to 3 Stories you have identified
- ▶ Make sure the acceptance criteria is testable
- ▶ Ensure that the Story meets the INVEST criteria to prepare for estimation



Estimation is a whole-team exercise

Agile Teams estimate Stories; POs provide clarification, but do not estimate the work

- ▶ Usually occurs during the backlog refinement event
- ▶ Increases accuracy by including all perspectives
- ▶ Builds understanding
- ▶ Creates shared commitment



Estimation performed by a manager, architect, or select group negates these benefits.

Estimate Stories with relative Story points

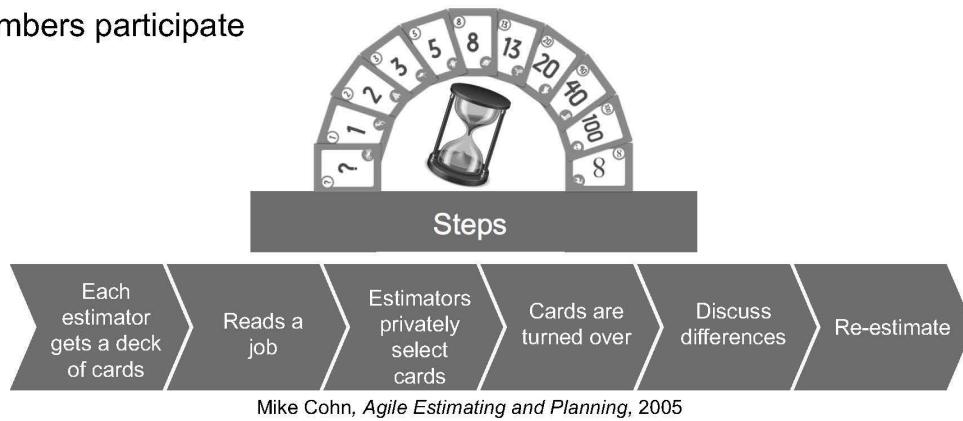
- ▶ A Story point is a singular number that represents:
 - Volume: how much is there?
 - Complexity: how difficult is it?
 - Knowledge: what do we know?
 - Uncertainty: what is not known?
- ▶ Story points are relative. They are not connected to any specific unit of measure.
- ▶ Compare with other stories (an 8-point story should take 4X longer than a 2-point story)

How big is it?



Apply Estimating Poker for fast, relative estimating

- ▶ Estimating Poker combines expert opinion, analogy, and disaggregation for quick but reliable estimates
- ▶ All team members participate



Exercise: Relative size estimating



Use Estimating Poker to relatively estimate a group of Stories.

- ▶ Use one set of Stories from your team
- ▶ Identify the smallest Story that takes about a day to develop and mark it as 1
- ▶ Estimate the remaining Story using the values 1, 2, 3, 5, 8, 13, 20, 40, 100



5.3 Plan the Iteration

Plan and commit

Purpose

Define and commit to what will be built in the Iteration

Process

- ▶ The Product Owner defines *what*
- ▶ The team defines *how* and *how much*
- ▶ Four hours max



Result

Iteration Goals and backlog of the team's commitment

Reciprocal commitment

- ▶ Team commits to delivering specific value
- ▶ Business commits to leaving priorities unchanged during the Iteration

Iteration Goals

Iteration Goals provide clarity, commitment, and management information.

They serve three purposes:

1. Align team members to a common purpose
2. Align Program Teams to common PI Objectives and manage dependencies
3. Provide continuous management information

Iteration Goals example

1. Finalize and push last-name search and first-name morphology
2. Index 80% of remaining data
3. Other Stories:
 - Establish search replication validation protocol
 - Refactor artifact dictionary schema

Commit to the Iteration Goals

A Team meets its commitment:

By doing everything they said they would do.

- or -

In the event that it is not feasible, they must immediately raise a red flag.

Commitment

Too much holding to a commitment can lead to burnout, inflexibility, and quality problems.



Adaptability

Too little commitment can lead to unpredictability and lack of focus on results.

Team commitments are not just to the work. They are committed to other teams, the program, and the stakeholders.

Exercise: Iteration Planning - PO and PM roles



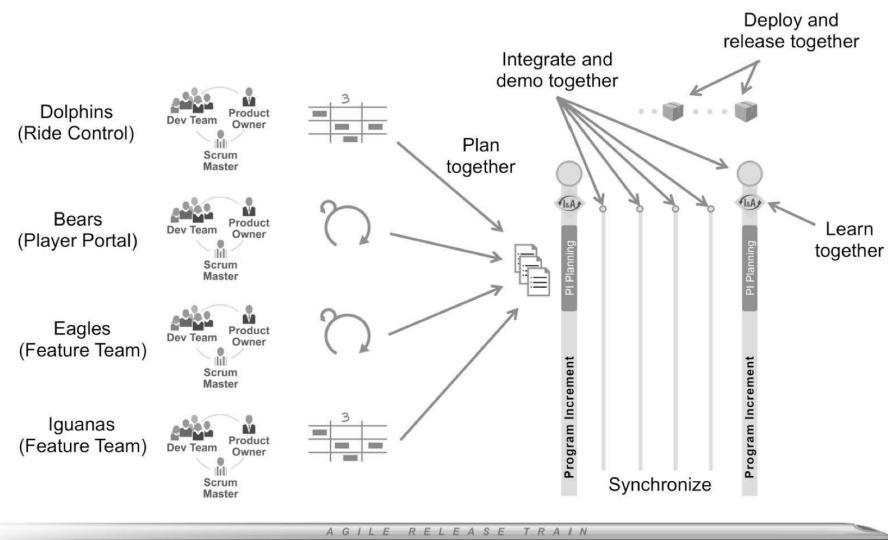
Iteration Planning

- ▶ What activities are required for the PO?
- ▶ What activities are required for the PM?
- ▶ Capture ideas and share with the class



5.4 Execute the PI

Agile Teams are on the Train

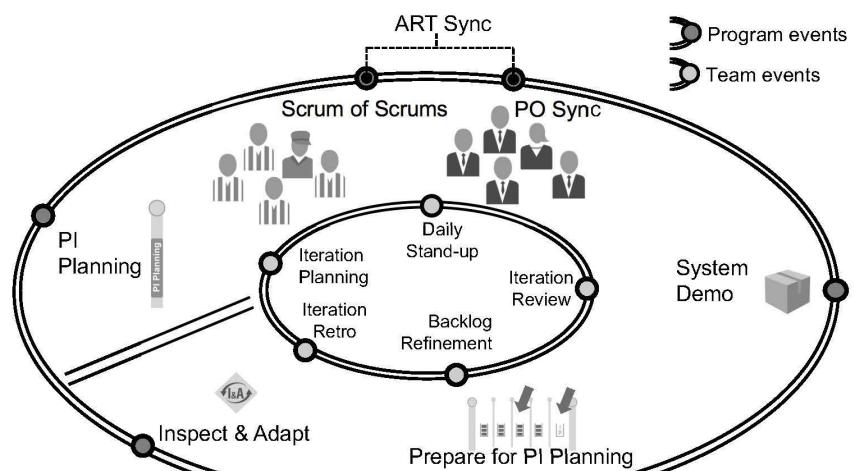


SCALED AGILE® © Scaled Agile, Inc.

5.45

Program Execution

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



SCALED AGILE® © Scaled Agile, Inc.

5.46

The Daily Stand-up (DSU)

The DSU is the key to team synchronization and self-organization.

The DSU (or Daily Scrum) is **not** a daily status meeting for management. It is used to:

- ▶ Share information about progress
- ▶ Coordinate activities
- ▶ Raise blocking issues
- ▶ Most importantly, ask, “*Will we still meet our Iteration Goals and our commitment?*”



- Every day at the same time in front of the team board
- Timebox of 15 minutes
- Not a problem-solving session
- Update the board

ART Sync

Programs coordinate dependencies through sync meetings.



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 min.
- ▶ Timeboxed, followed by a ‘Meet After’



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 min.
- ▶ Timeboxed, followed by a ‘Meet After’

Exercise: Roleplay the events



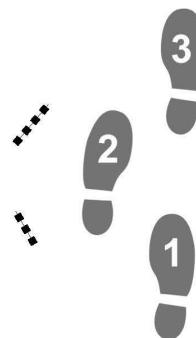
- ▶ Daily Stand Up and ART Sync
 - One half roleplays a typical Daily Stand-up
 - Other half roleplays a typical ART Sync



Sequencing Stories and Team Backlog Prioritization

Primary economic prioritization happens at the Program Backlog. Agile Teams sequence work for efficient execution of business priorities.

- ▶ The Product Owner and the team sequence work based on:
 - Story priorities inherited from Program Backlog priorities
 - Events, Milestones, Releases, and other commitments made during PI Planning
 - Dependencies with other teams
 - Local priorities
 - Capacity allocations for defects, maintenance, and refactors
- ▶ Initial sequencing happens during PI Planning
- ▶ Adjustments happen at Iteration boundaries



The Backlog Refinement Session

What it is:

The backlog refinement session is a preview and elaboration of upcoming Stories.

- ▶ Helps the team 'sleep' on new Stories prior to Iteration Planning
- ▶ Provides enough time to identify and resolve dependencies and issues that could impact the next Iteration
- ▶ Creates strong process foundation for test automation and Acceptance Test-Driven Development (ATDD)

How it's done:

It is is not a single event but a continuous effort.



- ▶ The PO synthesizes input from Mgmt, subject matter experts, Customers, and other stakeholders as needed
- ▶ The PO always has:
 - One to two Iterations of Stories
 - High-level acceptance criteria
 - A sense for capacity allocation and relative priorities within types

Exercise: Backlog Refinement



- ▶ At your table, discuss the backlog refinement session, preparation, and guidelines for the Team Backlog
- ▶ Consider the following questions
 - What does the PO/PM need to do to prepare?
 - What is the PO/PM's involvement in these activities?

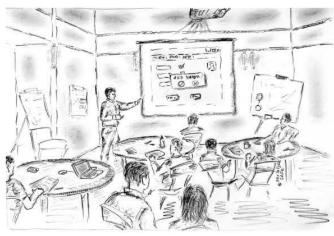


The Iteration Review: Two views

The Iteration Review provides measures of progress into the program:

1. How we did on the Iteration

- ▶ Did we meet the goals?
- ▶ Story by Story review



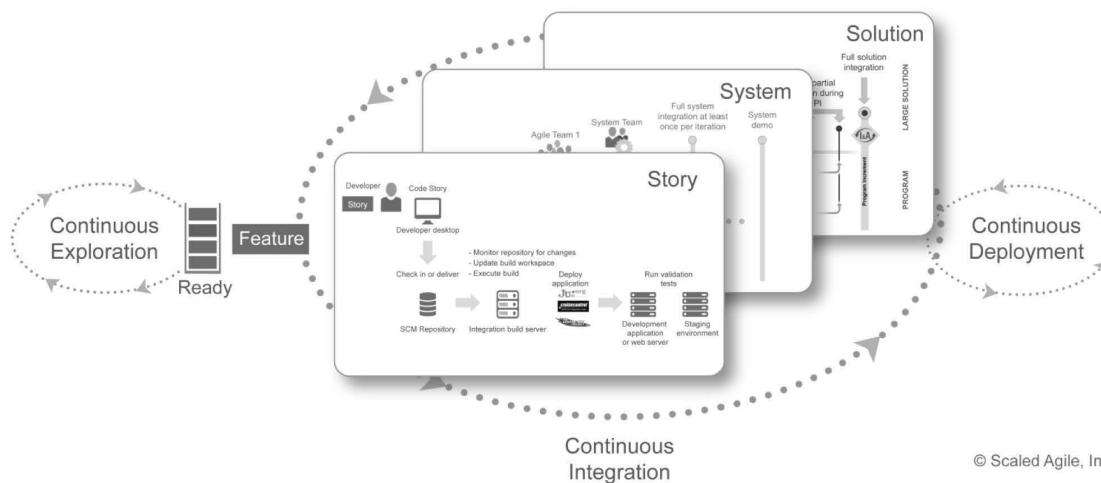
2. How we're doing on the PI

- ▶ Review of PI objectives
- ▶ Review remaining PI scope and reprioritize if necessary

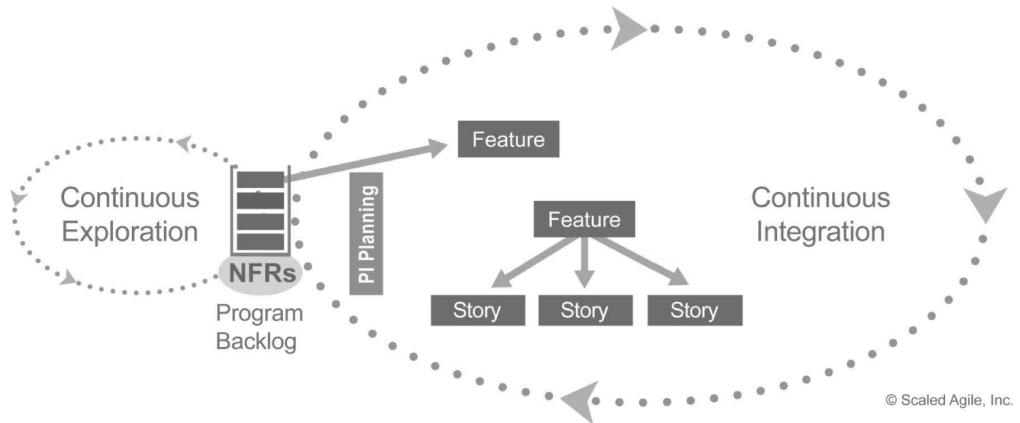
Attendees are the team and its stakeholders

Ad hoc demos – Don't wait for the Iteration Review.
Demo Stories when they are done.

Continuous Integration

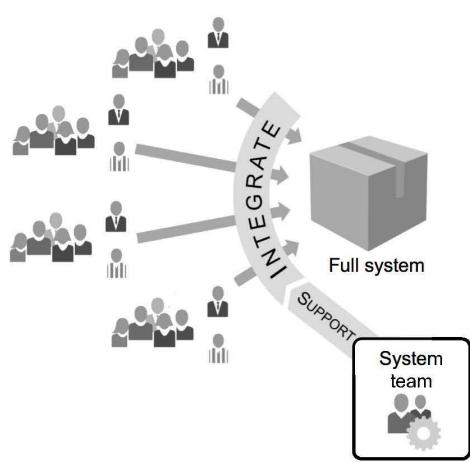


Continuously Integrate and Test Stories



System Demo every two weeks

Demonstrate the full Solution increment to stakeholders every Iteration.



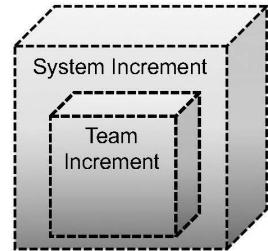
- ▶ An integrated Solution demo
- ▶ Happens after the teams' demo (may lag by as much as one Iteration, maximum)
- ▶ Demo from the staging environment, or the nearest proxy



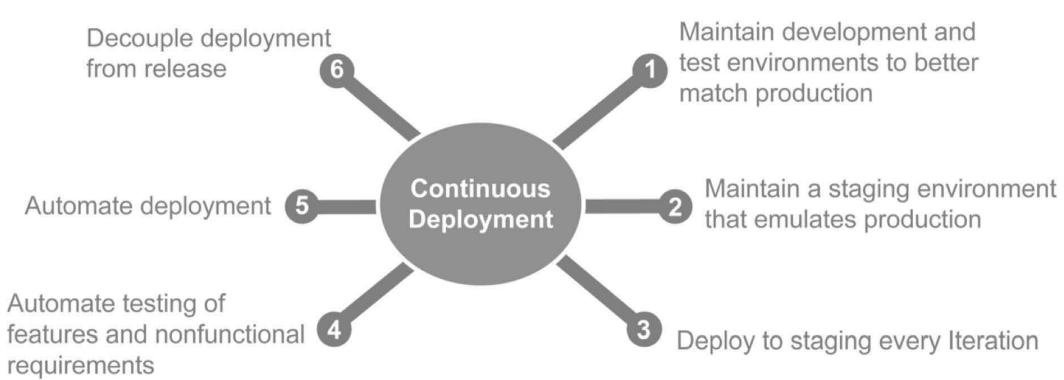
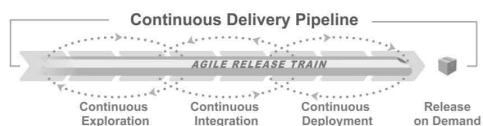
New system increment every two weeks

Every two weeks, teams evaluate the status of the new, integrated system increment.

- ▶ Features are functionally complete or ‘toggled’ so as not to disrupt demonstrable functionality
- ▶ New Features work together, and with existing functionality
- ▶ Architectural Runway work in process is scaffolded and toggled
- ▶ System is continually verified via Story and Feature acceptance tests
- ▶ All practical NFR testing is done continuously



Continuous Deployment



Create your DevOps culture with a CALMR approach



© Scaled Agile, Inc.

SCALED AGILE® © Scaled Agile, Inc.

5.59

Exercise: Create an action poster!



- ▶ Connect actions to Continuous Delivery Pipeline
 - One group selects Continuous Integration (pages 151-157)
 - One group selects Continuous Deployment (pages 158-162)
 - One group selects DevOps (pages 163-168)
- ▶ Using the articles in the workbook (starting on page 151), create a poster of actions you can take as a PO/PM to increase these capabilities within your enterprise



SCALED AGILE® © Scaled Agile, Inc.

5.60

Iteration Retrospective

At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

—Agile Manifesto

- ▶ 30 – 60 minutes
- ▶ Pick 1 – 2 things that can be done better, target for next Iteration
- ▶ Enter improvement items into the Team Backlog

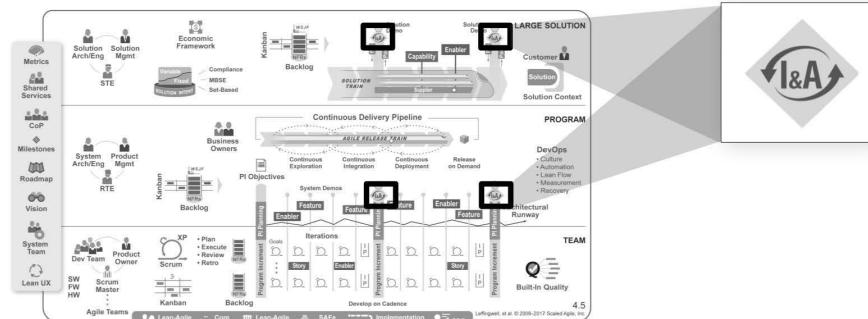
The Innovation and Planning (IP) Iteration

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
31	1	2	3	4	5	6
			Validation (if shipping)			
			Innovation / hackathon / spikes for next PI			
			PI Planning readiness			
7	8	9	10	11	12	13
		Continuing education		PI Planning		
				Business Context Product/Solution Vision Architecture Vision & Development Practices Planning Requirements & Lunch Team Breakouts Draft Plan Review Management Review & Problem Solving	Planning Adjustments Team Breakouts Final Plan Review & Lunch Program Risks PI Confidence Vote Plan Rework If Necessary After Commitment Planning Retrospective & Moving Forward	

Inspect and Adapt

Three parts:

1. The PI System Demo
 2. Quantitative measurement
 3. The problem-solving workshop
- ▶ Attendees: Teams and stakeholders
 - ▶ Timebox: 3 – 4 hours per PI



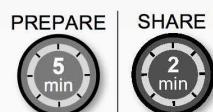
SCALED AGILE® © Scaled Agile, Inc.

5.63

Exercise: Reflecting on Program Increment events



- ▶ Map out your program events on a calendar, and socialize with another team
- ▶ Think about the PO and PM roles, and place the program events in the column in which that role would add the most value for them to attend/participate.
- ▶ Prepare to share with the class



SCALED AGILE® © Scaled Agile, Inc.

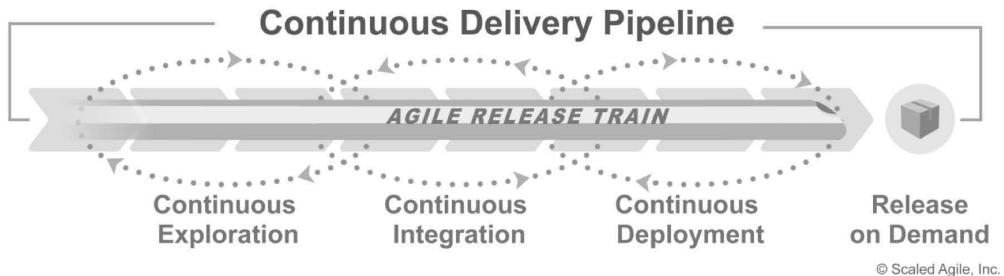
5.64

5.5 Release on Demand

SAFe Definition of Done

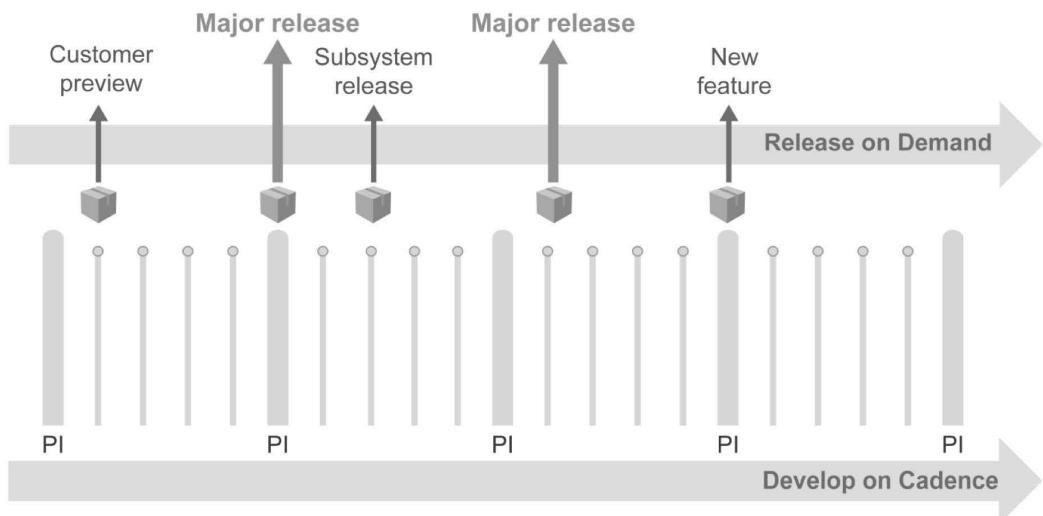
Team Increment	System Increment	Solution Increment	Release
<ul style="list-style-type: none">• Stories satisfy acceptance criteria• Acceptance tests passed (automated where practical)• Unit and component tests coded, passed, and included in the BVT• Cumulative unit tests passed• Assets are under version control• Engineering standards followed• NFRs met• No must-fix defects• Stories accepted by Product Owner	<ul style="list-style-type: none">• Stories completed by all teams in the ART and integrated• Completed features meet acceptance criteria• NFRs met• No must-fix defects• Verification and validation of key scenarios• Included in build definition and deployment process• Increment demonstrated, feedback achieved• Accepted by Product Management	<ul style="list-style-type: none">• Capabilities completed by all trains and meet acceptance criteria• Deployed/installed in the staging environment• NFRs met• System end-to-end integration, verification, and validation done• No must-fix defects• Included in build definition and deployment/transition process• Documentation updated• Solution demonstrated, feedback achieved• Accepted by Solution Management	<ul style="list-style-type: none">• All capabilities done and meet acceptance criteria• End-to-end integration and solution V&V done• Regression testing done• NFRs met• No must-fix defects• Release documentation complete• All standards met• Approved by Solution and Release Management

Release on Demand

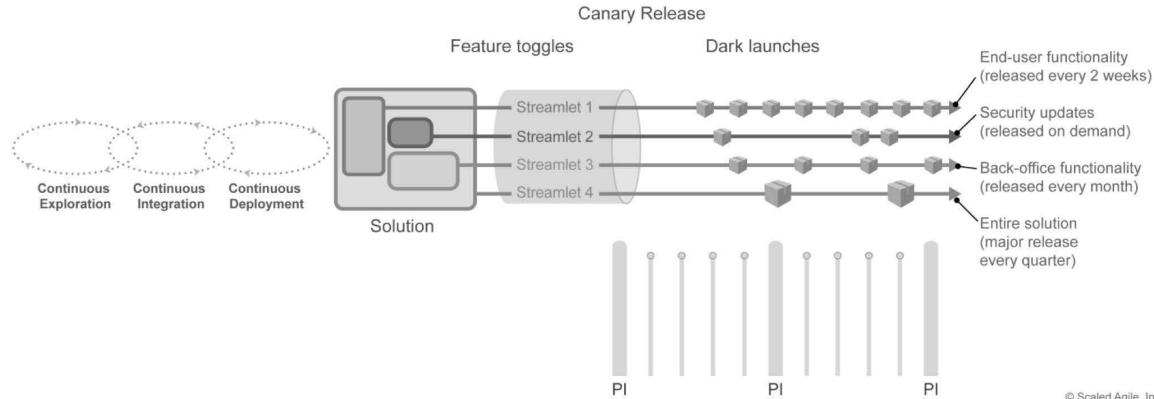


For more on the Continuous Delivery Pipeline, watch the “Introduction to Continuous Delivery Pipeline” video on the SAFe Community Platform

Decouple Cadence. Release on Demand



Architect the Solution for Incremental Release



Exercise: Release on Demand strategies



► Your Release on Demand strategies:

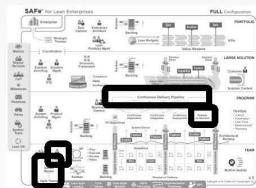
- What can you do to evolve your Definition of Done?
- What can you do to better foster architecting the solution for incremental release?
- What might you change in the future?



Lesson summary

In this lesson, you:

- ▶ Created alignment with PI Planning
- ▶ Decomposed Features into stories
- ▶ Explored planning the Iteration
- ▶ Discussed events to execute the PI
- ▶ Integrated Release on Demand strategies



Suggested Scaled Agile Framework reading:

*“Scrum Master”, “Agile Teams”, “Product Owner”,
“Continuous Delivery Pipeline”, and “Release on
Demand” articles*

Exercise: This lesson's key learnings



Summarize key learnings and insights from this lesson in your workbook.



Lesson 6

Defining the PO/PM Roles and Responsibilities

1. Applying SAFe in the Lean Enterprise
2. Relating a Lean-Agile Mindset to the PO/PM Roles
3. Collaborating with Lean Portfolio Management
4. Continuously Explore Customer Needs
5. Executing the Program Increment
6. Defining the PO/PM Roles and Responsibilities
7. Creating your PO/PM Action Plan

SAFe® Course: Attending this course gives learners access to the SAFe Product Owner/Product Manager exam and related preparation materials.

Learning objectives

- 6.1 Characterize the roles of the Product Owner and Product Manager
- 6.2 Examine other key program collaboration roles

6.1 Characterize the roles of the Product Owner and Product Manager

Product Owner responsibilities

- ▶ Ensures Stories and Enablers meet the acceptance criteria
 - Stories are aligned to Vision, Features, and PI Objectives
 - Has content authority for the Team Backlog
- ▶ Represents customers and stakeholders
- ▶ Participates in Iteration ceremonies as a team member
- ▶ Helps decompose Features into Stories and prioritizes the Team Backlog
 - Works with the System Architect and the Team to understand and prioritize Enablers
- ▶ Is open to negotiations that will occur
- ▶ Accepts the Stories as done



Candidates for the Product Owner role

- ▶ Business Analysts (BAs)
- ▶ Subject Matter Experts (SMEs)
- ▶ Project Managers
- ▶ Domain Experts
- ▶ Architects



Product Owner attributes

Focused on delivering stories and enablers to the train, tasked with helping the team *build the right things at the right time*.

- ▶ Ability to communicate
- ▶ Good business sense
- ▶ Technical foundation
- ▶ Trust
- ▶ Courage
- ▶ Content authority



The Product Owner in the Enterprise

- ▶ Establishes the sequence of backlog items based on program priorities, events, and dependencies with other teams
- ▶ Operates as part of an extended Product Management Team
- ▶ Understands how to operate with Epics, Capabilities, Features, and Stories
- ▶ Uses PI Objectives and Iteration Goals to communicate with management
- ▶ Coordinates with other Product Owners, the System Team, and shared resources in the ART PI Planning meetings
- ▶ Works with other Product Owners and Product Management throughout each Iteration and PI



Product Manager responsibilities

- ▶ Collaborates and sets expectations with Product Owners, stakeholders, customers, architects
- ▶ Can say 'no' in multiple directions: Stakeholders, Managers, LPM, and Product Owners
- ▶ Prioritizes Features and negotiates Enablers in the Program Backlog using WSJF
- ▶ Sets the Vision and Roadmap for the train
- ▶ Has significant people skills and the innate ability to navigate the political landscape
- ▶ Collaborates with the train to set scope
- ▶ Accepts the Features as done



Product Manager attributes

Focused on the business aspects and the market at large, tasked with *building the right Features at the right time.*

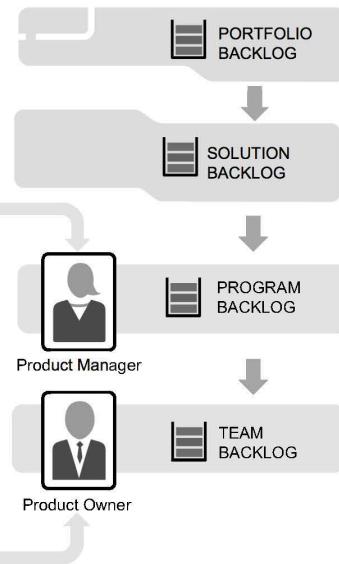
- ▶ Sense of balance
- ▶ Forward thinking
- ▶ Continuous exploration and understanding the customer's needs
- ▶ Solid understanding of current Solution
- ▶ Trust



PO and PM governance: Content authority

Product Manager – Program Backlog

- ▶ Has Product Backlog content authority. Works with the System Architect and Team to prioritize Enablers.
- ▶ Has content authority for Vision and Roadmap
- ▶ Helps drive PI Objectives
- ▶ Establishes Features and acceptance criteria



Product Owner – Team Backlog

- ▶ Has Team Backlog content authority. Works with the System Architect to prioritize Enablers.
- ▶ Drives Iteration Goals and content via prioritized Stories
- ▶ Establishes Story acceptance criteria
- ▶ Has authority for accepting Stories and Team increments
- ▶ Helps drive PI Objectives at the Team Level

Exercise: A day in the life of the PO and PM



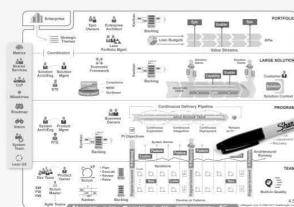
- ▶ Half the room will take the role of the PO, then the other half will take the role of the PM
- ▶ In your teams, use your flip chart to brainstorm the typical activities a PM or PO would be involved in. List a minimum of 10 things a PM or PO should do on a daily (or near-daily) basis.
- ▶ Add a time estimate to each item
- ▶ What conclusion can you make about the PM and PO roles?



Exercise: PO/PM content authority impact



- ▶ Use the Framework image on page 184 in your workbook to draw connections from the Product Owner and Product Manager to other framework elements, based on:
 - Content authority and impact on the enterprise
 - Collaboration
 - Problem-solving
 - Inputs/Outputs
 - Other ideas you have
- ▶ List the impacts and the challenges you may have in performing the content authority aspect of your role
- ▶ Be ready to present



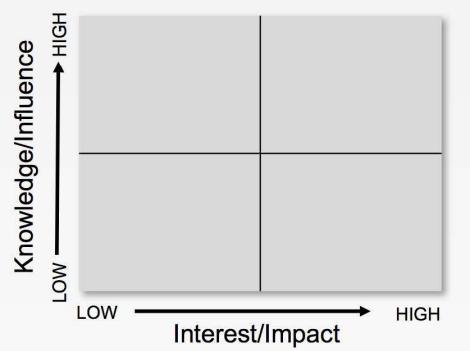
6.2 Examine other key program collaboration roles

Exercise: Who are the stakeholders in your environment?



Using the roles in your workbook, and the roles in your enterprise, create a potential collaboration model for your role in the enterprise.

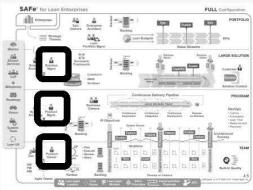
- ▶ As a table, review the SAFe roles on the top of the workbook page
- ▶ Identify the potential stakeholders for collaboration in your workbook
- ▶ Using the SAFe roles, and current roles within your enterprise, place the role in the appropriate section of the matrix



Lesson summary

In this lesson, you:

- ▶ Compared and contrasted the commonalities and differences between the Product Owner and Product Manager roles
- ▶ Defined the Product Owner and Product Manager roles



*Suggested Scaled Agile Framework reading:
“Product Owner” and the “Product and Solution Management”
articles*

Exercise: This lesson's key learnings



Summarize key learnings and insights from this lesson in your workbook.



Lesson 7

Creating your PO/PM Action Plan

1. Applying SAFe in the Lean Enterprise
2. Relating a Lean-Agile Mindset to the PO/PM Roles
3. Collaborating with Lean Portfolio Management
4. Continuously Explore Customer Needs
5. Executing the Program Increment
6. Defining the PO/PM Roles and Responsibilities
7. Creating your PO/PM Action Plan

SAFe® Course: Attending this course gives learners access to the SAFe Product Owner/Product Manager exam and related preparation materials.

Learning objectives

- 7.1 Develop a personal PO/PM action plan
- 7.2 Create a personal PO/PM improvement roadmap
- 7.3 Commit to the plans

7.1 Develop a personal PO/PM action plan

Exercise: Find your personal challenges



- ▶ Review the 'Key Learnings and Insights' you wrote after each lesson in your workbook
- ▶ From your 'Key Learnings and Insights' select three things that challenge you or cause you difficulties
- ▶ Write these items in your personal challenges list



Exercise: Define personal actions



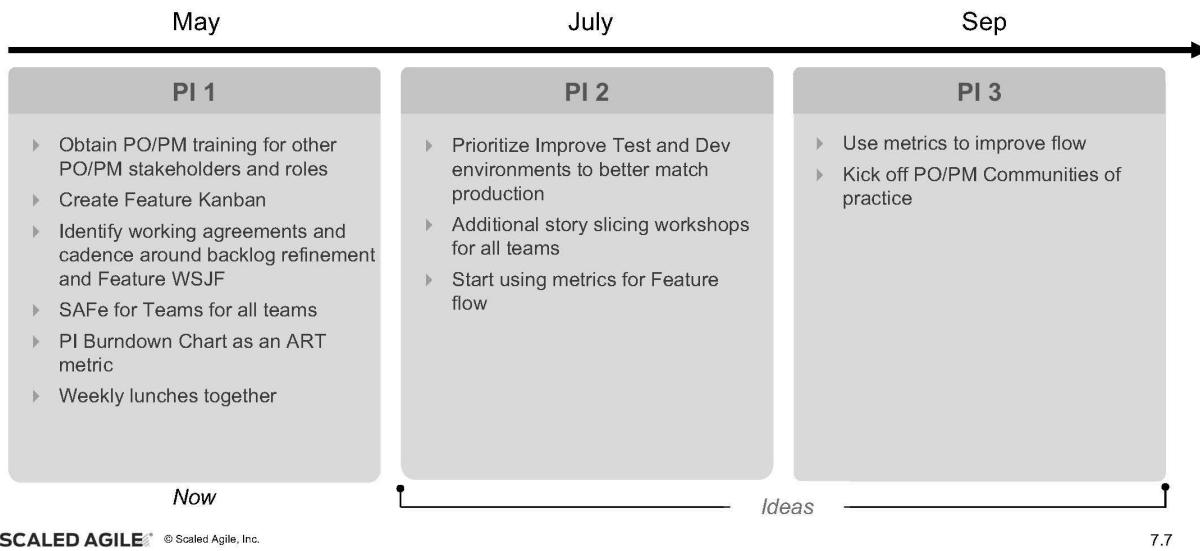
-
-
-
- ▶ In pairs, alternate discussing your three personal challenges
- ▶ With your teammate, identify one clearly-defined action for each of your challenges that you can implement in your work context
- ▶ Document your plans in your personal actions list
- ▶ Be prepared to share



7.2 Create a personal improvement roadmap

Improvement Roadmap

An Improvement Roadmap guides your evolution over time.



SCALED AGILE® © Scaled Agile, Inc.

Exercise: Find improvement opportunities



- ▶ Identify three action items in your organization where you believe you can immediately improve your PO/PM interactions
- ▶ Select areas where you see the most opportunity for improvement
- ▶ Write these items in your improvement list in your workbook



SCALED AGILE® © Scaled Agile, Inc.

7.8

Exercise: Define your improvement roadmap



- ▶ In pairs, alternate discussing your three improvement opportunities
- ▶ With your partner, brainstorm and define two sequential actions for each item in your improvement list that you can implement in your work context
- ▶ Document your improvement actions in your action roadmap list in your workbook



7.3 Commit to the plans

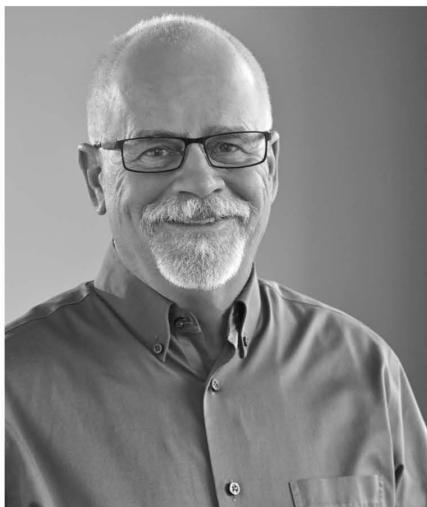
Exercise: Commit to the plan



- ▶ Organize into new pairs, take turns reviewing the personal and the improvement roadmap lists you created
- ▶ Commit to each other that you will do your best to act on the items in the list
- ▶ Schedule a date in two to eight weeks for a follow-up check-in to meet and talk about your progress against your plans. Agree to keep each other accountable.



Emphasize lifelong learning



Lean-Agile Leaders are lifelong learners who help teams build better systems through understanding and exhibiting the values, principles, and practices of Lean, systems thinking, and Agile development.

—Dean Leffingwell

PO/PM reading list

- ▶ *SAFe Distilled*, Richard Knaster, Dean Leffingwell
- ▶ *Tribal Unity*, Em Campbell-Pretty
- ▶ *The Lean Machine*, Dantar Oosterwald
- ▶ *The Goal*, Eliyahu Goldratt
- ▶ *Principles of Product Development Flow*, Don Reinertsen
- ▶ *Switch*, Chip Heath and Dan Heath
- ▶ *The Five Dysfunctions of a Team*, Patrick Lencioni
- ▶ *Agile Software Requirements*, Dean Leffingwell
- ▶ *Agile Retrospectives*, Esther Derby and Diana Larsen
- ▶ *Death by Meeting*, Patrick Lencioni
- ▶ *Lean Product and Process Development*, Allen Ward and Durward Sobeck II
- ▶ *That's Not How we Do It Here!* John Kotter and Holger Rathgeber

Lesson summary

In this lesson, you:

- ▶ Developed a personal action plan for your PO or PM role
- ▶ Created a personal improvement roadmap
- ▶ Committed to the plans

Exercise: This lesson's key learnings



Summarize key learnings and insights from this lesson in your workbook.

