

Logistics

- ▶ The importance of taking breaks
- ▶ Forming social connections during lunch
- ▶ Establishing working agreements
- ▶ Accessing Wi-Fi
- ▶ Locating restrooms 



SCALDED AGILE® © Scaled Agile, Inc.

1-3

Notes:

Course goals

To perform the role of a SAFe® 4 Scrum Master you should be able to:

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



SCALDED AGILE® © Scaled Agile, Inc.

1-4

Notes:

Course map

- ▶ Lesson 1: Introducing Scrum in SAFe
- ▶ Lesson 2: Characterizing the role of the Scrum Master
- ▶ Lesson 3: Experiencing PI Planning
- ▶ Lesson 4: Facilitating Iteration execution
- ▶ Lesson 5: Finishing the PI



SCALED AGILE® © Scaled Agile, Inc.

1-5

Notes:



Introduction: Spell it out - SAFe Scrum Master

- ▶ **Step 1:** Introduce yourself to someone you don't know.
- ▶ **Step 2:** Choose one letter from the course title. Use it to explain what you hope to learn in this class.
 - Example: "I selected 'C' for 'Communication' because I want to know how to communicate better with my team."



SCALED AGILE® © Scaled Agile, Inc.

1-6

Notes:

Lesson 1

Introducing Scrum in SAFe

Learning objectives:

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



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

1.1 Examine basic Agile development concepts

Notes:

1.1 Examine basic Agile development concepts



Activity: Fast focus! – Too much Work in Process (WIP)

- ▶ **Step 1:** When the instructor says “Go,” write the numbers 1 to 26 as many times as you can until the instructor says “STOP.”
- ▶ **Step 2:** When the instructor says “Go,” write the letters A to Z as many times as you can until the instructor says “STOP.”
- ▶ **Step 3:** When the instructor says “Go,” write number/letter pairs (i.e. 1A, 2B, 3C) as many times as you can until the instructor says “STOP.”



PREPARE SHARE
 4 min 2 min

1-10

SCALED AGILE® © Scaled Agile, Inc.



Thought organizer



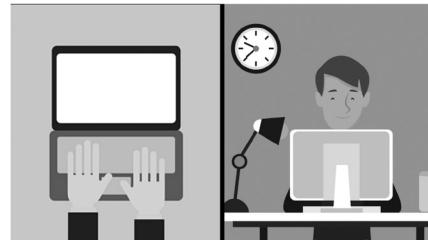
Summary: Fast focus! – Too much Work in Process (WIP)

► **Step 1:** At your tables, discuss:

- How many numbers did you write down? How many letters? How many number/letter pairs?

► **Step 2:** Share with the class:

- How many active projects are you currently juggling?
- How much of your day is actually spent adding value versus running from meeting to meeting?



SHARE



1-11

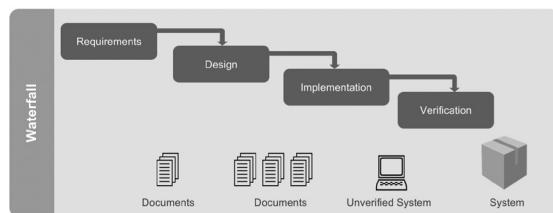
SCALED AGILE® © Scaled Agile, Inc.



Thought organizer

1.1 Examine basic Agile development concepts

Agile and Waterfall development



Waterfall Development

- ▶ Too much WIP
- ▶ Frequent context switching
- ▶ Increased overhead
- ▶ Delivery of value at the end



Agile Development

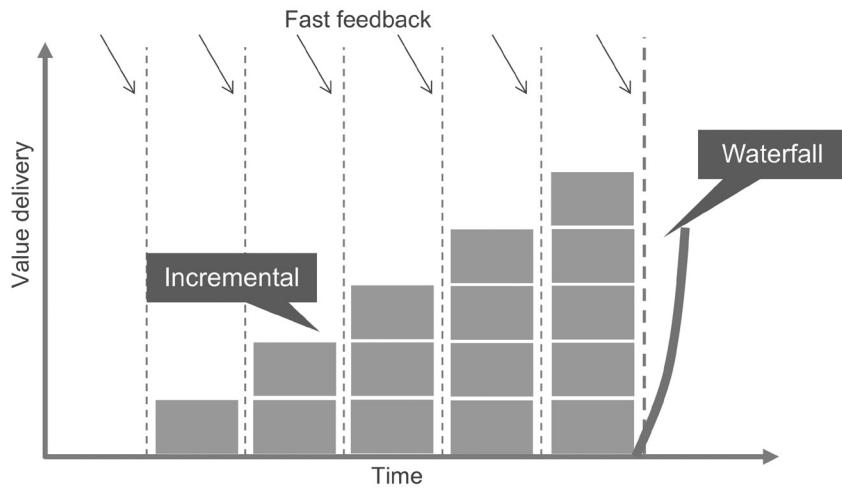
- ▶ Reduce and visualize WIP
- ▶ Incremental development
- ▶ Early and continuous value delivery

SCALED AGILE® © Scaled Agile, Inc.

1-12

Notes:

Agile development: Deliver value incrementally



SCALED AGILE® © Scaled Agile, Inc.

1-13

Notes:

Agile frameworks

Agile Development

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

Agile frameworks

- SAFe
- Scrum
- Crystal
- Kanban
- eXtreme Programming (XP)
- Feature-Driven Development

Practices

- Timeboxing
- User Stories
- Daily Stand-Ups
- Frequent Demos
- Test-Driven Development
- Information Radiators
- Retrospectives
- Continuous Integration

Notes:

1.1 Examine basic Agile development concepts



Activity: Manifesto for Agile Software Development

- **Step 1:** Locate the activity in your Workbooks
- **Step 2:** Individually fill in the following value statements using the appropriate phrases

Value statements

- 1) _____ over processes and tools
- 2) Working software over _____
- 3) Customer collaboration over _____
- 4) _____ over following a plan

Phrases

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

PREPARE



1-15

SCALED AGILE® © Scaled Agile, Inc.

Value Statements

- 1) _____ over processes and tools
- 2) Working software over _____
- 3) Customer collaboration over _____
- 4) _____ over following a plan

Phrases

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

1.1 Examine basic Agile development concepts

The Agile Manifesto

We are uncovering better ways of developing software by doing it and helping others do it.

Through this work we have come to value:

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan

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

 agilemanifesto.org

1-16

Notes:

The Agile Manifesto Principles

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

 agilemanifesto.org/principles.html

1-17

Notes:



The Agile Manifesto Principles

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.

SCALED AGILE® © Scaled Agile, Inc.



agilemanifesto.org/principles.html

1-18

Notes:



Discussion: Adopting Agile principles and practices

► **Step 1:** Pair with someone you haven't met yet and discuss the following:

- Do the Agile values and principles align with the culture in your organization?
- Are there any contradictions?
- Which principle or practice stands out to you?
- What are some of the biggest areas where Agile challenges traditional development?



► **Step 2:** Share with the class

PREPARE SHARE



1-19

SCALED AGILE® © Scaled Agile, Inc.



Thought organizer

1.2 Explore Scrum basics

Notes:



Video: The roots of Scrum

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

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

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



Notes:

Scrum values

The three pillars of Scrum: *the process, the workflow, progress support the Scrum values.*

Scrum Values

Courage Commitment Focus Respect Openness



SCALED AGILE® © Scaled Agile, Inc.

1-22

Notes:



Discussion: Scrum values create transparency

- ▶ **Step 1:** As a team, pick a Scrum value and discuss it in a context of a work project.
- ▶ **Step 2:** On a flip chart sheet, write down how it increases transparency in: the process, the workflow, and the work progress.
- ▶ **Step 3:** Share with the class how Scrum's focus on transparency compares to typical waterfall projects?



SCALED AGILE® © Scaled Agile, Inc.

1-23

Notes:

Scrum and SAFe terminology

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

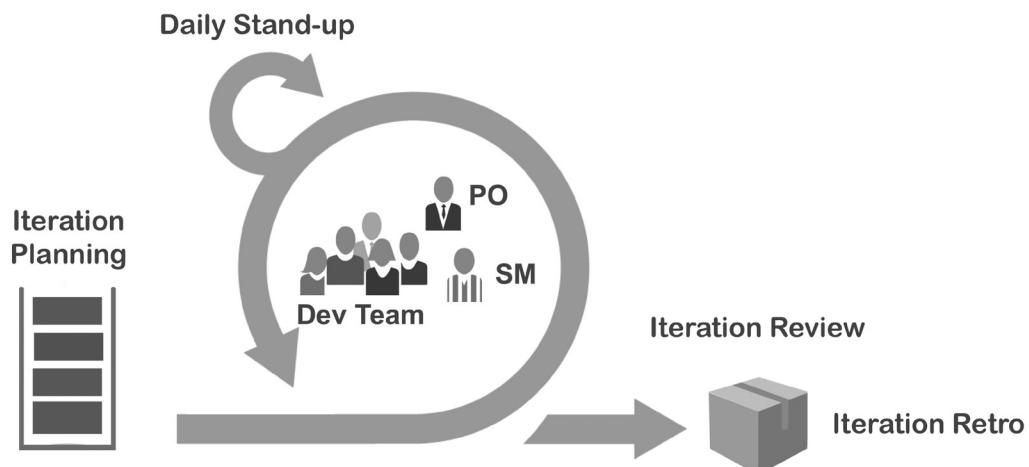
SCALED AGILE® © Scaled Agile, Inc.

1-24

Notes:

Teams execute iterations with Scrum

Scrum is built on transparency, inspection, and adaptation.



SCALED AGILE® © Scaled Agile, Inc.

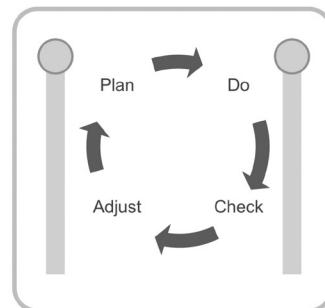
1-25

Notes:

Iteration basics

- ▶ **Definition:** Iterations are a single development cycle where each Agile team defines, builds, integrates and tests the Stories from their iteration *backlog*.
- ▶ **Duration:** Each Iteration is the same length, running back to back
 - SAFe advises two weeks
- ▶ **Goal:** To deliver working software/hardware at the end of each Iteration

Plan-Do-Check-Adjust Cycle of an Iteration



Avoid adding scope once the Iteration has begun

SCALED AGILE® © Scaled Agile, Inc.

1-26

Notes:

The Team Backlog organizes the team's work

- ▶ It is truly **all** things. If a piece of work is in the backlog, it might get done. If it isn't, there is no chance that it will be done.
- ▶ User Stories and enabler stories may be estimated (preferable), but estimates do not imply committed delivery.
- ▶ The Team Backlog:
 - Is created by the Agile Team
 - Is owned and prioritized by the team's Product Owner
 - Represents opportunities, not commitments
 - Contains user and enabler stories



Team Backlog



Product Owner

SCALED AGILE® © Scaled Agile, Inc.

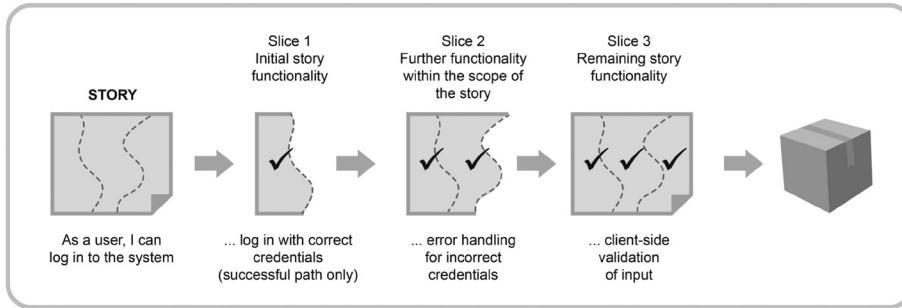
1-27

Notes:

Foundation for incremental development

Implementing Stories in vertical slices is key to incremental development:

- ▶ Enables a short feedback cycle
- ▶ Allows refinement of understanding of functionality
- ▶ Facilitates more frequent integration of working systems



SCALED AGILE® © Scaled Agile, Inc.

1-28

Notes:

1.3 Position an Agile Team in a SAFe Enterprise

SCALED AGILE® © Scaled Agile, Inc.

1-29

Notes:

The world's leading framework for enterprise agility.

SAFe® for Lean Enterprises is a knowledge base of proven, integrated principles, practices, and competencies for Lean, Agile, and DevOps



<http://www.scaledagileframework.com/>

SCALED AGILE® © Scaled Agile, Inc.

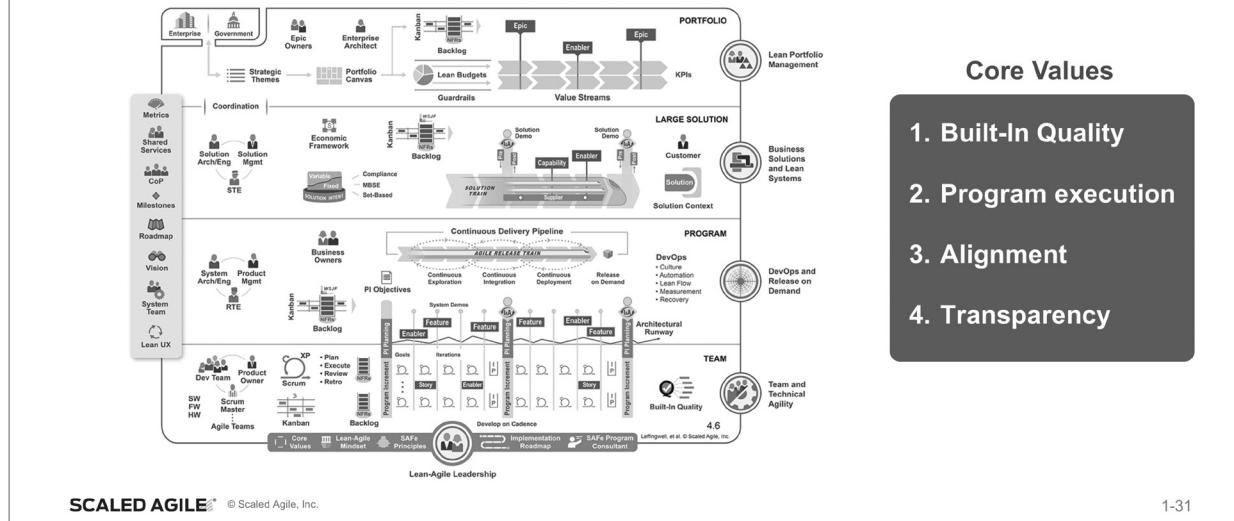
1-30

Notes:

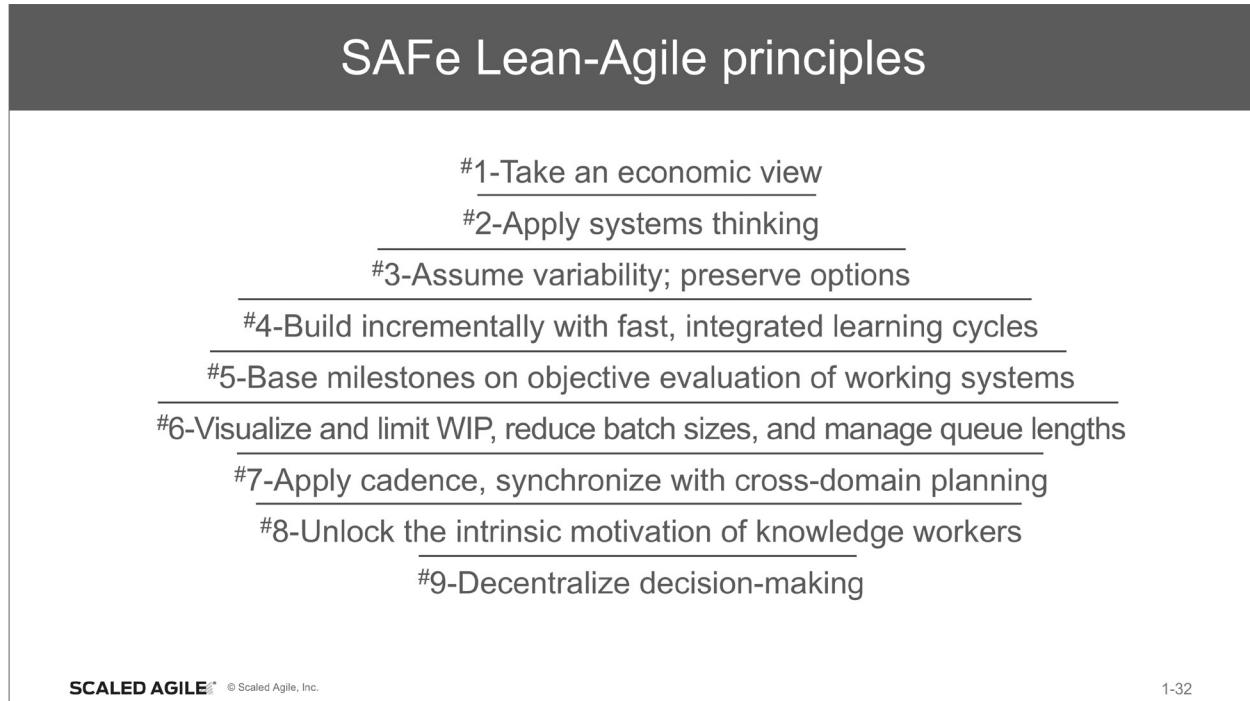
1.3 Position an Agile Team in a SAFe Enterprise

The Scaled Agile Framework® (SAFe)

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



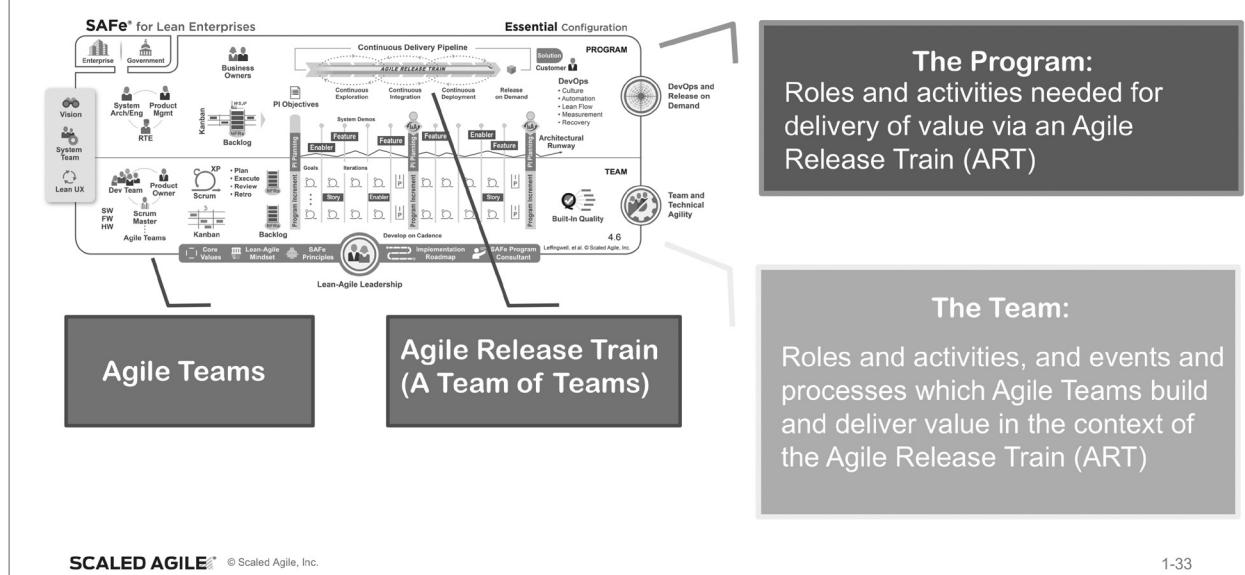
Notes:



Notes:

1.3 Position an Agile Team in a SAFe Enterprise

Positioning an Agile Team in a SAFe Enterprise



1-33

Notes:

What are Agile Teams?

- Agile teams are cross-functional, self-organizing entities that can *define, build and test*, and where applicable *deploy*, increments of value
- Optimized for communication and delivery of value
- Deliver value every two weeks



Agile Team

SCALED AGILE® © Scaled Agile, Inc.

1-34

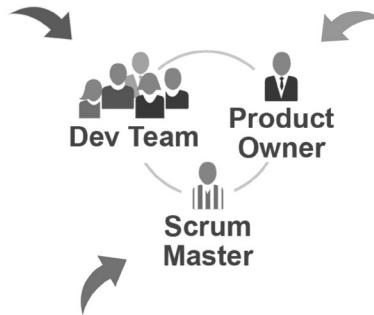
Notes:

1.3 Position an Agile Team in a SAFe Enterprise

Roles on the Agile Team

— Dev Team —

- Create and refine user Stories and acceptance criteria
- Define/Build/Test/Deliver Stories
- Develop and commit to Team PI Objectives and Iteration plans
- Three to nine members



— Product Owner —

- Defines and accepts Stories
- Acts as the customer for developer questions
- Works with Product Management to plan Program Increments (PI)

— Scrum Master —

- Coach the Agile team and facilitate team meetings
- Removes impediments; protects the team from outside influence
- Attends Scrum of Scrum meetings

SCALED AGILE® © Scaled Agile, Inc.

1-35

Notes:

Nothing beats an Agile Team

- ▶ Teams use Scrum and Kanban for team agility
- ▶ Apply Built-in Quality practices for Technical Agility

Lean and Agile principles and practices

Behavior-Driven Development (BDD)

eXtreme Programming (XP)

Code quality

Design patterns and practices

Agile modeling



Agile Team Applies Built-In Quality

Built-in Quality practices

SCALED AGILE® © Scaled Agile, Inc.

1-36

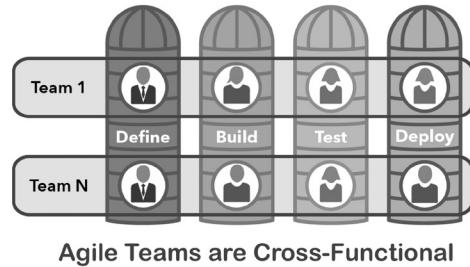
Notes:

1.3 Position an Agile Team in a SAFe Enterprise

Except a team of Agile Teams

Self-organizing, self-managing team of Agile teams operates on common principles:

- ▶ Delivers working, tested, full-system increments every two weeks
- ▶ Has common Iteration lengths and start/end dates
- ▶ Plans its work at periodic, largely face-to-face PI Planning events
- ▶ Develops on cadence and releases on demand



Agile Teams are Cross-Functional

SCALED AGILE® © Scaled Agile, Inc.

1-37

Notes:

The Agile Release Train (ART)

Each ART is a virtual organization of 5 – 12 teams (typically 50 – 125 people) that plans, commits, and develops and deploys together.

Agile Release Trains:

- ▶ Align teams to a common business and technology mission
- ▶ Deliver a continuous flow of value



SCALED AGILE® © Scaled Agile, Inc.

1-38

Notes:

1.3 Position an Agile Team in a SAFe Enterprise

Roles on the Agile Release Train



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 key stakeholders on the Agile Release Train.

AGILE RELEASE TRAIN

SCALED AGILE® © Scaled Agile, Inc.

1-39

Notes:

Program events

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

SCALED AGILE® © Scaled Agile, Inc.

1-40

Notes:

Lesson review

In this lesson you:

- ▶ Examined basic Agile development concepts
- ▶ Explored the basics of Scrum
- ▶ Positioned an Agile Team in a SAFe Enterprise



1-41

SCALED AGILE® © Scaled Agile, Inc.

Notes:



Toolbox: Focus on promoting and coaching transparency

- ▶ **Step 1:** Locate the Scrum Master Toolbox section in your Workbooks.
- ▶ **Step 2:** Begin adding tools to the Toolbox by brainstorming the following:
 - What are some of the key insights from this lesson?
 - What is your plan for promoting transparency in the process, the workflow, and the work progress?
 - What are some techniques you can apply for coaching the team in Scrum values?
- ▶ **Step 3:** Share one of your insights with the class.



PREPARE & SHARE



1-42

Notes:

Key Learnings & Insights



Lesson 1

Introducing Scrum in SAFe

Lesson 2

Characterizing the Role of the Scrum Master

Learning objectives:

- 2.1 Examine the responsibilities of the Scrum Master role
- 2.2 Explore the characteristics of an effective Scrum Master
- 2.3 Build high-performing teams
- 2.4 Facilitate effective SAFe Team events
- 2.5 Coach the Agile Team using powerful questions
- 2.6 Collaborate with other teams
- 2.7 Resolve team conflicts



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

2.1 Examine the responsibilities of the Scrum Master role

SCALED AGILE® © Scaled Agile, Inc.

2-3

Notes:



Discussion: The role of the Scrum Master in SAFe

- ▶ **Step 1:** Read the Scrum Master article
 - **Option 1:** Scan the QR Code with your mobile device
<https://www.scaledagileframework.com/scrum-master/>
 - **Option 2:** Read the article in your Workbook
- ▶ **Step 2:** As a team, discuss the role of the Scrum Master in SAFe and identify key responsibilities
- ▶ **Step 3:** Capture your team's key takeaways on a flip chart sheet



Scan me



2-4

SCALED AGILE® © Scaled Agile, Inc.

See the Scrum Master article on the next page >>>

Scrum Master

Good leaders must first become good servants. —Robert K. Greenleaf

Scrum Masters are servant leaders and coaches for an Agile Team. They help educate the team in Scrum, Extreme Programming (XP), Kanban, and SAFe, ensuring that the agreed Agile process is being followed. They also help remove impediments and foster an environment for high-performing team dynamics, continuous flow, and relentless improvement.



Although the Scrum Master role is mainly based on standard Scrum, Agile Teams—even those teams that are applying Kanban—establish this position to help the team meet its goals and coordinate activities with other teams. The Scrum Master role is taken by a team member whose primary responsibility is assisting the self-organizing, self-managing team achieve its goals. Scrum Masters do this by teaching and coaching team practices, implementing and supporting SAFe principles and practices, identifying and eliminating impediments, and facilitating flow.

Details

The Scrum Master role is a unique Agile team member who spends much of her time helping other team members communicate, coordinate, and cooperate; generally, this person assists the team in meeting their delivery goals. The Scrum Master is a servant leader who enables teams to self-organize, self-manage, and deliver via effective Lean-Agile practices. The Scrum Master supports and enforces the Scrum process and other rules that the team has agreed. The Scrum Master also helps the team coordinate with other teams on the Agile Release Train (ART) and communicates status to management as needed.

Responsibilities

An effective Scrum Master is a team-based servant leader who:

- Exhibits Lean-Agile leadership – Exhibits the behaviors of a Lean-Agile Leader with a Lean-Agile Mindset. Helps the team embrace SAFe Core Values, adopt and apply SAFe Principles, and implement SAFe practices.
- Supports the team rules – The rules of an Agile Team are lightweight, but they are rules nonetheless, and the Scrum Master is responsible for reinforcing them. These may include the rules of Scrum, Built-In Quality practices from Extreme Programming (XP), Work in Process (WIP) limits

2.1 Examine the responsibilities of the Scrum Master role

from Kanban, and any other process rules the team has agreed.

- Facilitates the team's progress toward team goals – The Scrum Master is trained as a team facilitator and is continuously engaged in challenging the old norms of development to improve performance in the areas of quality, predictability, flow, and velocity. They help the team focus on daily and Iteration Goals in the context of current Program Increment (PI) Objectives.
- Leads team efforts in relentless improvement – Helps the team improve and take responsibility for their actions; facilitates the team retrospective. Teaches problem-solving techniques and helps the team become better problem-solvers for themselves.
- Facilitates meetings – Facilitates all team meetings, including (where applicable) the Daily Stand-up, Iteration Planning, Iteration Review, and Iteration Retrospective.
- Supports the Product Owner – The Scrum Master helps the Product Owner in their efforts to manage the backlog and guide the team while facilitating a healthy team dynamic with respect to priorities and scope.
- Eliminates impediments – Many blocking issues will be beyond the team's authority or may require support from other teams. The Scrum Master actively addresses these issues so that the team can remain focused on achieving the objectives of the Iteration.
- Promotes SAFe quality practices – SAFe provides guidance to assist the teams in constantly improving the quality of their deliverables and meeting the Definition of Done (DoD). The Scrum Master helps foster the culture of technical discipline and craftsmanship that is the hallmark of effective Agile teams.
- Builds a high-performing team – Focuses on ever-improving team dynamics and performance. Helps the team manage interpersonal conflicts, challenges, and opportunities for growth. Escalates people problems to management where necessary, but only after internal team processes have failed to resolve the issue; helps individuals and teams through personnel changes.
- Protects and communicates – Communicates with management and outside stakeholders; helps protect the team from uncontrolled expansion of work.
- Responsibilities on the train – The Scrum Master helps coordinate inter-team cooperation and helps the team operate well on the train.

2.1 Examine the responsibilities of the Scrum Master role

- Coordinates with other teams – The Scrum Master is typically the representative in the Scrum of Scrums (SoS) meeting, and they pass information from that meeting back to the team (see Program Increment for more details). They often coordinate with the System Team, User Experience, Architecture, and Shared Services. It is important to note, however, that the responsibility for inter-team coordination cannot be delegated entirely to the Scrum Master; every team member shares responsibility in that regard.
- Facilitates preparation and readiness for ART events – Assists the team in preparation for ART activities, including PI Planning, System Demos, and the Inspect and Adapt.
- Supports estimating – Guides the team in establishing normalized estimates and helps the team understand how to estimate Features and Capabilities.

Sourcing the Role

The Scrum Master can be a part-time or full-time role, depending on the size of the team, the context, and other responsibilities. However, at Enterprise scale, it can be a challenge to sell the need for a full-time Scrum Master for each Agile team. After all, if the enterprise is organizing 100 new teams, it probably isn't economically or politically practical to take 100 full-time development team members and assign them to these new duties—duties that don't include development or testing. Nor is it economically viable to hire a full- or part-time consultant for each team to help them learn and master the new methods. That could kill the transformation before it even gets started, and before the teams have had a chance to prove the value of the role.

Therefore, SAFe takes a pragmatic approach and assumes, in general, that the Scrum Master is a part-time role. During initial SAFe adoption, however, the job may be more intensive. At this stage, the organization may find it beneficial to bring external consultants on board to coach the teams while they become experienced in Scrum and SAFe. These outside consultant Scrum Masters will often coach multiple teams in the organization.

Learn More

[1] www.scrumalliance.org.

[2] Leffingwell, Dean. *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise*. Addison-Wesley, 2011.

2.1 Examine the responsibilities of the Scrum Master role

Scrum Master

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



SCALED AGILE® © Scaled Agile, Inc.

2-5

Notes:

How a Scrum Master supports an Agile Team

- ▶ Facilitates team events
- ▶ Coaches the team and improve processes
- ▶ Removes impediments to the team's progress
- ▶ Fosters adoption of Agile technical practices
- ▶ Assists the PO in preparing and refining the backlog for PI and Iteration Planning



SCALED AGILE® © Scaled Agile, Inc.

2-6

Notes:

2.1 Examine the responsibilities of the Scrum Master role

The Scrum Master in a SAFe Enterprise

- ▶ Works with the RTE to ensure the train meets its overall PI Objectives
- ▶ Coordinates with other Scrum Masters, the System Team, and shared services in the ART PI Planning events
- ▶ Works with the teams throughout each Iteration and PI
- ▶ Participates in the Scrum of Scrums
- ▶ Fosters normalized estimating within the team
- ▶ Helps teams operate under architectural and portfolio governance, system integration, and System Demos



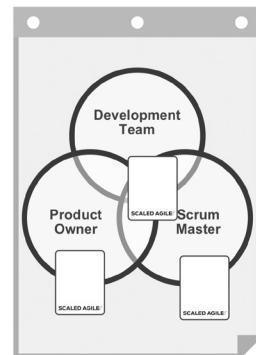
SCALED AGILE® © Scaled Agile, Inc.

2-7

Notes:

Card Game: Scrum roles and responsibilities

- ▶ **Step 1:** With your team, draw the following Venn diagram on a flipchart sheet.
- ▶ **Step 2:** Review *Role-Responsibility Cards*
- ▶ **Step 3:** Place them either in the role or at an intersection of the Venn diagram
- ▶ **Step 3:** Present your Venn diagram to the class



PREPARE SHARE



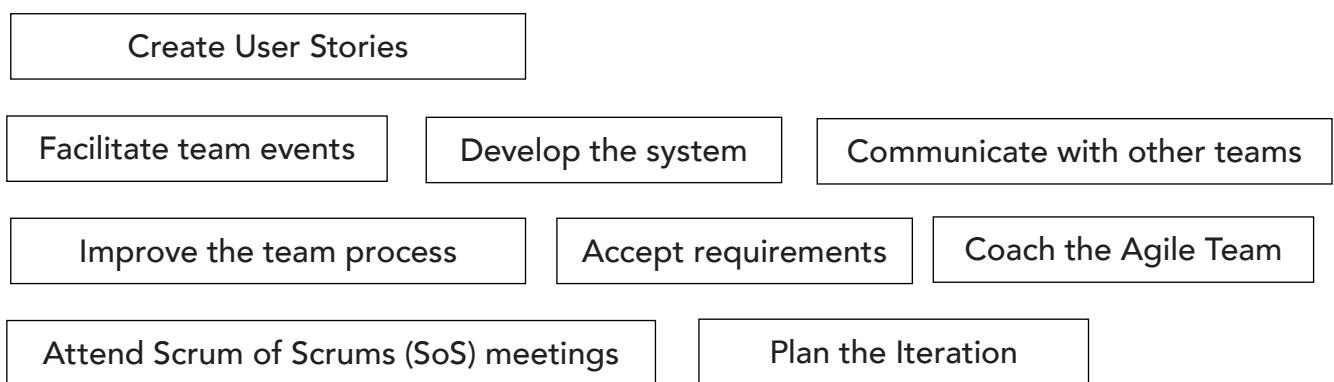
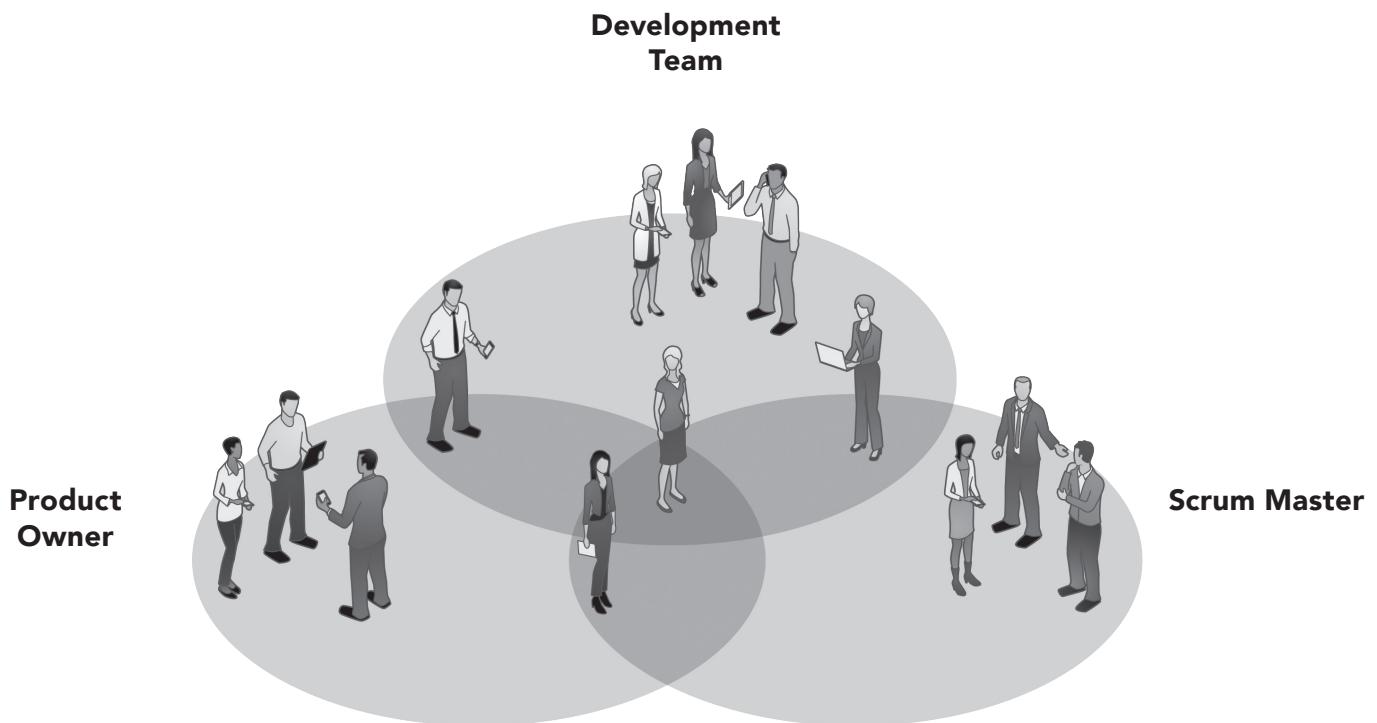
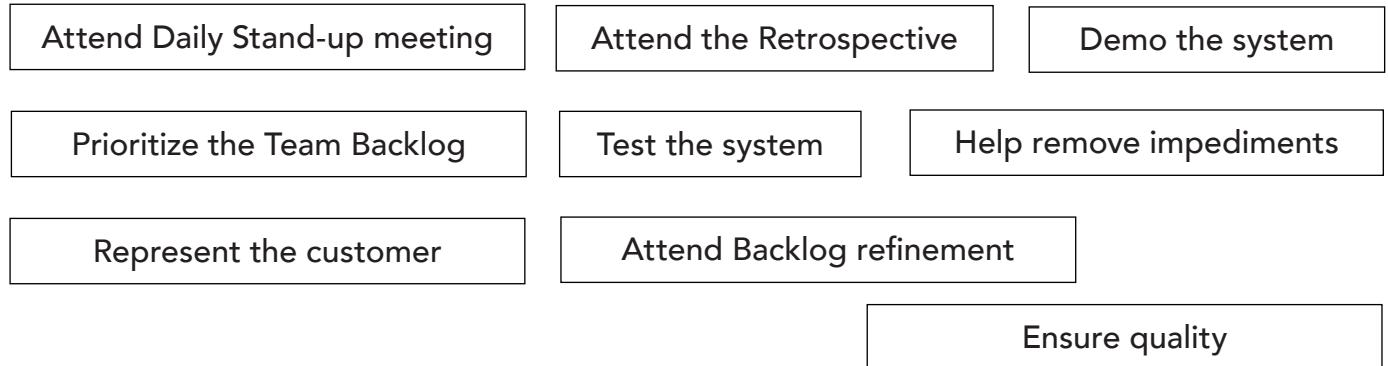
2-8

SCALED AGILE® © Scaled Agile, Inc.

See Scrum Roles and Responsibilities on the next page >>>

2.1 Examine the responsibilities of the Scrum Master role

Scrum Roles and Responsibilities



2.2 Explore the characteristics of an effective Scrum Master

Notes:

2.2 Explore the characteristics of an effective Scrum Master



Discussion: Effective Scrum Master in SAFe

- ▶ **Step 1:** Pair with someone you haven't met with
- ▶ **Step 2:** Discuss what traits and attributes you think an effective Scrum Master needs
- ▶ **Step 3:** Share with the class



SCALED AGILE® © Scaled Agile, Inc.

Notes:

Traits and Attributes of an Effective Scrum Master



2.2 Explore the characteristics of an effective Scrum Master

Act as a servant leader

- ▶ Listens to and supports team members in problem identification and decision-making
- ▶ Understands and empathizes with others
- ▶ Encourages and supports the personal development of each individual
- ▶ Persuades rather than uses authority
- ▶ Thinks beyond day-to-day activities
- ▶ Seeks to help without diminishing the commitment of others
- ▶ Is open and appreciates openness in others



Act as a Servant Leader

SCALED AGILE® © Scaled Agile, Inc.

2-11

Notes:

Servant leadership

A servant leader knows that his own growth comes from facilitating the growth of others who deliver the results.

Good leaders must first become good servants.

—Robert Greenleaf, father of Servant Leadership

SCALED AGILE® © Scaled Agile, Inc.

2-12

Notes:

2.2 Explore the characteristics of an effective Scrum Master

Scrum Master as servant leader

Trait in the context of SAFe
Listens to and supports team members in decision identification	<ul style="list-style-type: none">- As a good facilitator, encourages everyone to express their opinions- Is attentive to hesitant behavior and body language during Daily Stand-Up meetings, retrospectives, planning- Helps the team identify positive and negative changes during retrospectives
Understands and empathizes with others	<ul style="list-style-type: none">- Shares in celebrating every successful demo, feels bad about Iteration failures
Encourages and supports the personal development of each individual	<ul style="list-style-type: none">- Encourages team learning- Fosters collaborative practices: side-by-side programming, Continuous Integration, collective code ownership, short design sessions, specification workshops- Encourages rotation in technical areas of concern: functionality, components/layers, role aspects- Facilitates team decision-making rather than making decisions for the team
Persuades rather than uses authority	<ul style="list-style-type: none">- Asks questions that encourage the team to look at decisions from new perspectives- Articulates facts, helps the team see things they may have overlooked, helps them rethink

SCALED AGILE® © Scaled Agile, Inc.

2-13

Notes:

Scrum Master as servant leader

Trait in the context of SAFe
Thinks beyond day-to-day activities	<ul style="list-style-type: none">- Sets long-term operating goals for the team: Agile practices to master, new skills to acquire- Examines what is missing in order to make the environment better for everyone, prioritizes improvement activities and makes them happen
Seeks to help without diminishing the commitment of others	<ul style="list-style-type: none">- Facilitates ad hoc meetings (design discussions, story reviews with the PO, coding and unit testing approaches, critical bug-fix strategies)- Helps the team find access to external sources of information: subject matter experts, shared resources (architects, UX designers, tech writers)- Helps clarify and articulate rationale behind scope commitments- Helps team members prepare for Iteration Review and System Demo- Helps the team find techniques to be more collaborative
Is open and appreciates openness in others	<ul style="list-style-type: none">- Shows appreciation for team members who raise serious issues, even when delivery is jeopardized- Encourages and facilitates open communication among team members and with external colleagues- Encourages healthy conflict during team meetings- Gives open, honest opinions

Notes:



Activity: A day in the life of a Scrum Master

► **Step 1:** In your teams, use a flip chart sheet to brainstorm the typical daily activities that a Scrum Master as a Servant Leader would be involved in.

► **Step 2:** Include the following:

- A list of 10 activities
- Time estimates to each of the activities

► **Step 3:** Share with the class:

- What conclusions can you make about the Scrum Master role?
- Is this a full-time or part-time role?



A Day in the Life of a Scrum Master



SCALED AGILE® © Scaled Agile, Inc.

2-15



Thought organizer

2.3 Build high-performing teams

SCALED AGILE® © Scaled Agile, Inc.

2-16

Notes:

Common attributes of high-performing teams

- ▶ Self-organizing
- ▶ Effective decision-making
- ▶ Open and clear communication
- ▶ Valued diversity
- ▶ Mutual trust
- ▶ Healthy conflict
- ▶ Clear goals and purpose
- ▶ Concentration and focus
- ▶ Ownership and accountability
- ▶ Defined roles and responsibilities
- ▶ Aligned and collaborative
- ▶ Positive atmosphere
- ▶ Effective timely feedback
- ▶ Sufficient resources for local control
- ▶ Success focus over ‘failure avoidance’
- ▶ Abilities balanced with challenge
- ▶ Engagement
- ▶ Fun



SCALED AGILE® © Scaled Agile, Inc.

2-17

Notes:

Stages of high-performing teams

- 1 Forming
- 2 Storming
- 3 Norming
- 4 Performing
- 5 Adjourning



Source: Tuckman's stages of team dynamics (1977)

SCALED AGILE® © Scaled Agile, Inc.

2-18

Notes:



Discussion: Building a high-performing team

► **Step 1:** As a team discuss the following:

- Have you ever been on a high-performing team?
- What was it like?

► **Step 2:** Share with the class:

- As a Scrum Master what are some actions you can take in order to build a high performing team?
- How would you sustain a high-performing team?



Building a High-Performing Team



2-19

SCALED AGILE® © Scaled Agile, Inc.



Thought organizer

2.4 Facilitate effective SAFe Team events

SCALED AGILE® © Scaled Agile, Inc.

2-20

Notes:

Team events: Overview

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

SCALED AGILE® © Scaled Agile, Inc.

2-21

Notes:

2.4 Facilitate effective SAFe Team events

The challenge with meetings

- ▶ Meetings can be challenging because:
 - The purpose is not clear
 - There are no actionable outcomes
 - They may result in unproductive conflict
 - They may be boring
 - Conversation may divert from the agenda into deep discussion
- ▶ Such meetings add almost no value
- ▶ Ineffective meetings can (and should) be fixed



Challenge with Meetings

SCALED AGILE® © Scaled Agile, Inc.

2-22

Notes:

Running successful meetings

- ▶ Scrum Masters can benefit from the best practices for amazing meetings from companies like Apple and Google
- ▶ Prepare for every meeting, no matter how short
- ▶ Communicate a clear purpose and agenda
- ▶ Identify a Directly Responsible Individual (DRI) for maintaining agenda/action items
- ▶ Expect participants to know why they are attending, what contributions they will make, and expected outcomes
- ▶ Leave with clear action items
- ▶ Promote and keep to timeboxes
- ▶ Be prepared to challenge and be challenged
- ▶ Get participants moving and ensure active engagement



Successful Meetings

SCALED AGILE® © Scaled Agile, Inc.

2-23

Notes:

Running successful meetings

- ▶ Establish default decisions; decisions should never wait for a meeting
- ▶ Don't bring a problem without bringing at least one possible solution
- ▶ Review actions taken to meet commitments—enforce accountability
- ▶ Use “yes, and ...” instead of “no, but ...” to keep inputs positive and flowing
- ▶ Take frequent breaks
- ▶ Go the extra mile to bring remote participants into the discussion
- ▶ Maintain communication beyond the meeting



Successful Meetings

Notes:

2.4 Facilitate effective SAFe Team events



Discussion: Worst or best meeting ever

► **Step 1:** At your table share a personal experience about:

- The worst meeting you have attended or facilitated or
- The best meeting you have ever had

► **Step 2:** Discuss with your team and outline

- One reason for why the meeting was bad or
- One reason for what made the meeting great

► **Step 3:** Share with the class



2-25

SCALED AGILE® © Scaled Agile, Inc.



Thought organizer

2.5 Coach the Agile Team using powerful questions

SCALED AGILE® © Scaled Agile, Inc.

2-26

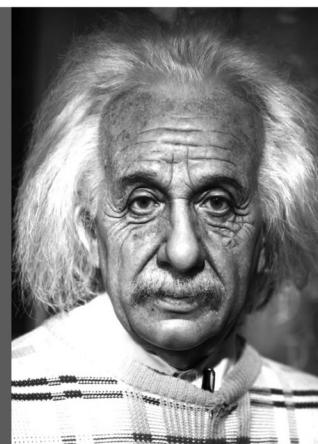
Notes:

Coaching with powerful questions

Coaches don't give people the answer. Instead, they guide people to the solution.

“ If I had an hour to solve a problem and my life depended on the solution, I would spend the first 55 minutes determining the proper question to ask. For once I know the proper question, I could solve the problem in less than five minutes. ”

— **Albert Einstein**



SCALED AGILE® © Scaled Agile, Inc.

2-27

Notes:

2.5 Coach the Agile Team using powerful questions

Shift towards new behaviors

Coaching sometimes requires a shift from old behaviors to new ones.

Move away from...	Move toward...
Coordinating individual contributions	Coaching the whole team to collaborate
Acting as a subject matter expert	Being a facilitator
Driving toward specific outcomes	Being invested in the team's overall performance
Knowing the answer	Letting the team find their own way
Directing	Guiding
Talking about deadlines and technical options	Focusing on business value delivery
Driving 'the right' (your) decisions	Doing the right thing for the business right now
Fixing problems rather than helping others fix them	Facilitating team problem-solving

Source: Lyssa Adkins, *Coaching Agile Teams*

2-28

Notes:

Why are questions powerful?

- ▶ They are thought-provoking
- ▶ They generate curiosity in the listener
- ▶ They channel focus
- ▶ They generate energy and forward movement
- ▶ They stimulate reflective conversation
- ▶ They surface underlying assumptions
- ▶ They invite creativity and new possibilities
- ▶ They inspire more questions
- ▶ They help reach for deep meaning



SCALED AGILE® © Scaled Agile, Inc.

2-29

Notes:

Powerful questions you can ask

Powerful questions like these can help connect ideas and generate deeper insights.

- ▶ What new connections are you making?
- ▶ What had real meaning for you from what you've heard?
- ▶ What surprised you?
- ▶ What challenged you?
- ▶ What's missing from this picture so far?
- ▶ What is it we're not seeing?
- ▶ What do we need more clarity about?
- ▶ What has been your major learning, insight, or discovery so far?
- ▶ What is the next level of thinking we need to do?
- ▶ What hasn't been said that would help us reach a deeper level of understanding and clarity?
- ▶ What would you do if success were guaranteed?



2-30

Notes:

2.5 Coach the Agile Team using powerful questions



Role Play: Powerful questioning

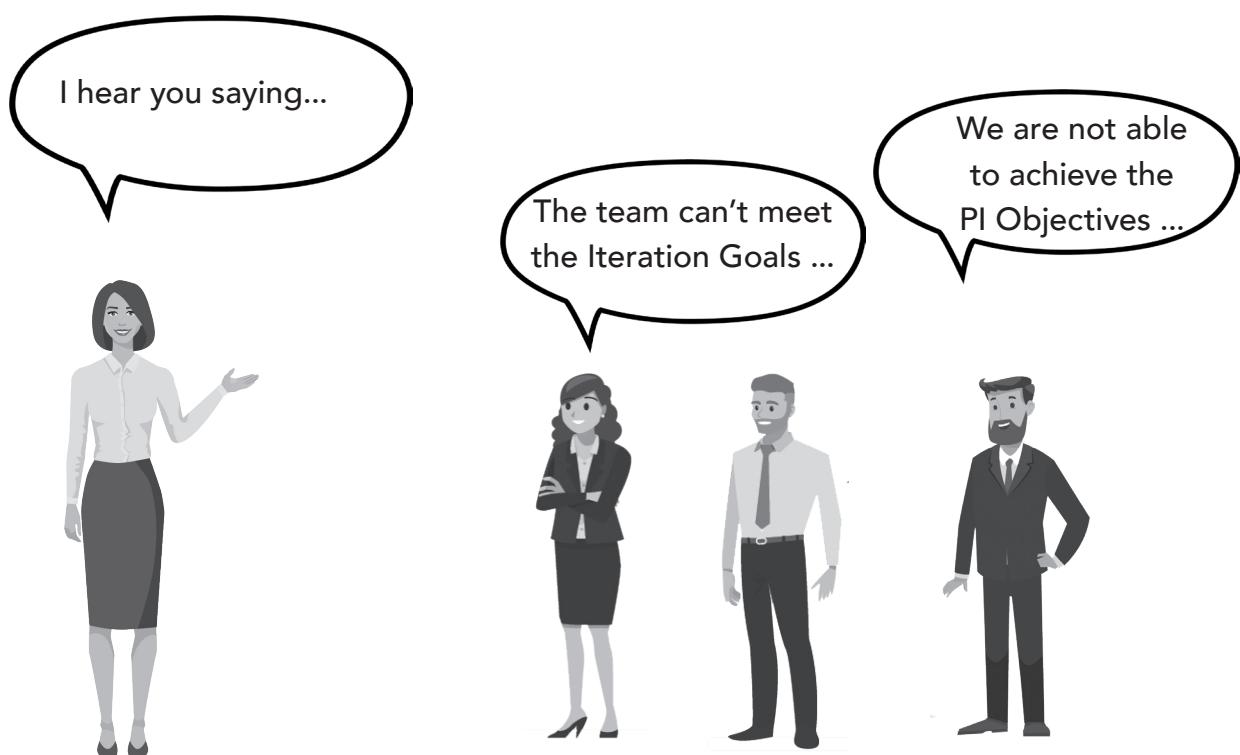
- ▶ **Step 1:** The people at your table are the team. Select one person to play the role of the Scrum Master.
- ▶ **Step 2:** The team brainstorms an issue with a project they are currently facing and presents to the Scrum Master.
- ▶ **Step 3:** The Scrum Master coaches the team to resolve the issue by only responding in one of two ways:
 - Reflective listening: *"I hear you saying ..."*
 - Asking a powerful question
- ▶ **Step 4:** As a Scrum Master, share your experience with the class:
 - Were you able to guide the team with powerful questioning rather than telling them what to do?



2-31

SCALED AGILE® © Scaled Agile, Inc.

Notes:



2.6 Collaborate with other teams

Notes:

Collaboration with other teams

The team should:

- ▶ Integrate their work often with other teams in the Program (at least multiple times per Iteration)
- ▶ Work with the System Team on automated system level tests
- ▶ Join their Daily Stand-up when important issues arise
- ▶ Join their demo or planning
- ▶ Work with the System Architect to better manage dependencies with other teams



Collaboration

Notes:

2.6 Collaborate with other teams

Actively engage with other Scrum Masters

- ▶ Work together with other Scrum Masters to organize and maintain Communities of Practice
- ▶ Actively participate in the Scrum of Scrums
- ▶ Coordinate the implementation of Program improvement backlog items
- ▶ Visit other teams' Scrum ceremonies and invite other teams to yours
- ▶ Self-organize with other Scrum Masters and the RTE to 'optimize the whole.'

Communities of Practice



SCALED AGILE® © Scaled Agile, Inc.

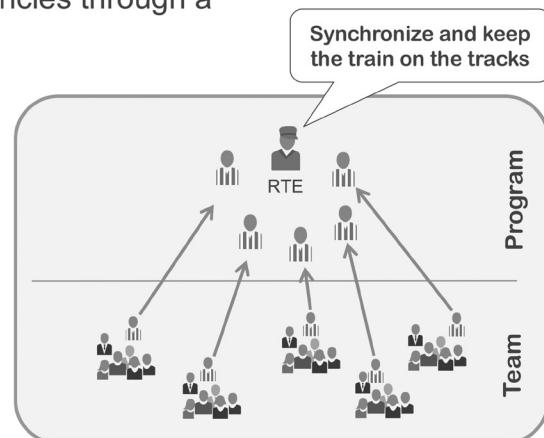
2-34

Notes:

Scrum of Scrums (SoS)

Programs continuously coordinate dependencies through a Scrum of Scrums (SoS).

- ▶ The SoS is a meeting for Scrum Masters and the Release Train Engineer to gain visibility into team progress and program impediments
- ▶ It is typically held twice per week
- ▶ It is timeboxed but is followed by a 'Meet After' for problem-solving



SCALED AGILE® © Scaled Agile, Inc.

2-35

Notes:

2.7 Resolve team conflicts

Notes:



Role Play: Dysfunction in action

- ▶ **Step 1:** Select one member of the team to play the role of the Scrum Master and another to play the role of a Developer. The rest play the Team.
- ▶ **Step 2:** Read the scenario in your Workbook and the parts for the Scrum Master, John (the Developer) and the Team.
- ▶ **Step 3:** Reenact the parts following the instructions.
- ▶ **Step 4:** Share your experiences with the class:
 - What dysfunctions do you see playing out in this team?
 - As a Scrum Master what do you do about it?



SCALED AGILE® © Scaled Agile, Inc.

2-37

Team Dysfunction During Retrospective

Scene 1

The Scrum Master of the team is facilitating a Team Retrospective. The team is gathered together to discuss the results of the Iteration.

SCRUM MASTER

It looks like we are unable to achieve the Iteration's goals. What happened?

TEAM

Well, a lot of time we spent on unplanned work dealing with defects from previous Iterations.

JOHN THE DEVELOPER

I can fix all the defects myself! Just let me work on my own and stay out of my way!

Team members exchange looks or silently look at the ground.

Some truths about teams

- ▶ Teams are far more productive than the same number of individuals
- ▶ Face-to-face communication is extremely efficient
- ▶ Teams work best when not interrupted
- ▶ Products are more robust when a team has all the cross-functional skills necessary
- ▶ When teams themselves make a commitment, they will probably figure out how to meet it
- ▶ Changes in team composition can impact productivity
- ▶ Peer pressure is a strong motivator



2-38

SCALED AGILE® © Scaled Agile, Inc.

Notes:

The five dysfunctions of a team

- ▶ Teamwork is the ultimate competitive advantage
- ▶ However, many teams are dysfunctional
- ▶ Absence of trust is the key problem that leads to the other four dysfunctions



Source: *Five Dysfunctions of a Team*, Patrick Lencioni
2-39

SCALED AGILE® © Scaled Agile, Inc.

Notes:

2.7 Resolve team conflicts

SAFe helps addressing the five dysfunctions

Inattention to Results	Results are empirically reviewed at the end of every Iteration and Release. Team retrospectives drive continuous improvement.
Avoidance of Accountability	Stakeholders, peer pressure, and review of results drive accountability.
Lack of Commitment	Teams make shared commitments to each other and to the external stakeholders.
Fear of Conflict	Scrum creates safe environment for conflict; the Scrum Master encourages discussion of disagreements. Shared commitment avoids individual conflict that occurs when objectives are not aligned.
Absence of Trust	The environment is safe. The team shares commitment and goals, displays hyper-transparency, and engages in retrospectives.

SCALED AGILE® © Scaled Agile, Inc.

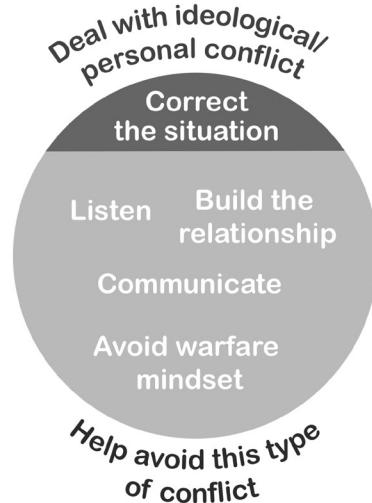
2-40

Notes:

Avoiding ideological/personal conflicts

A leader should spend far more time helping things go right than dealing with things that are going wrong.

- ▶ Help others see their teammates as human beings with their own needs, cares, worries, and objectives (instead of as obstacles)
- ▶ Help the team set a common vision, goals, and values
- ▶ Start gradually, dealing with long-term tension within the team
- ▶ Educate the team on achieving consensus
- ▶ Build ‘relentless collaboration’
- ▶ Master proven conflict-resolution techniques



Source: Adapted from The Arbinger Institute's *The Anatomy of Peace*
2-41

SCALED AGILE® © Scaled Agile, Inc.

Notes:

Resolving conflicts

In any system that is brought together for a purpose, there is no such thing as real conflict, only unexamined assumptions.

— E. Goldratt's Theory of Constraints

- ▶ Meet with the conflicting parties
- ▶ Identify exactly what each party wants
- ▶ Identify why each party needs what they want
- ▶ Obtain agreement that the common goal is correct
- ▶ Dig deeper and review the assumptions
- ▶ Challenge each of the assumptions
- ▶ Find out what the common goal is that ties these reasons together

SCALED AGILE® © Scaled Agile, Inc.

2-42

Notes:

Working agreements facilitate conflict management

Working Agreements: Example

As a participant on this team, I agree and acknowledge that:



- ▶ I am committed to the team's objectives and goals
- ▶ I respect other people's opinions, even when they contradict or conflict with mine
- ▶ If we cannot reach agreement, I will seek and support a consensus decision
- ▶ I will at all times avoid blocking my team from moving forward
- ▶ Whether or not the team decision coincides with mine, I will do my best to support it

SCALED AGILE® © Scaled Agile, Inc.

2-43

Notes:

Achieving consensus

- ▶ Define why reaching consensus is important in this situation.
- ▶ Let people exchange thoughts. Begin with someone who disagrees and then ask someone who agrees to give his or her perspective.
- ▶ Decompose the disagreement. Identify precisely what parts of the idea they disagree with. Can a portion be removed or modified?
- ▶ If that doesn't work, ask those who disagree to propose a modification to the idea or exchange alternative ideas.
- ▶ Continue exchanging thoughts and finding alternatives until you reach consensus or decide consensus is not possible. If consensus isn't possible, make a majority decision and clarify that everyone will support this decision.

Notes:



Role Play: Resolving team conflicts

- ▶ **Step 1:** Select one member of the team to play the role of the Scrum Master and another to play the role of John (The Developer). The rest play the Team.
NOTE: You can assume the same roles as previously played or switch roles.
- ▶ **Step 2:** Read the scenario in your Workbook and the parts for the Scrum Master, John (The Developer) and the Team in your Workbook.
- ▶ **Step 3:** Reenact the script.
- ▶ **Step 4:** As a Scrum Master, how would you handle the escalating conflict between John and the team. What tools or techniques would you use to resolve the conflict?
- ▶ **Step 5:** Share your experiences with the class.

SCALED AGILE® © Scaled Agile, Inc.



2-45

Team Conflict

Scene 1

You meet with the team. Everyone is present but John (The Developer) who comes in late to the meeting with a negative attitude.

SCRUM MASTER

Thank you for coming together as a team. Is everyone here? Where is John?

TEAM

Yeah, about John. John is not doing his work. He has been offensive and extremely difficult to work with.

JOHN THE DEVELOPER

What?! I have been working after hours and all my work is done! You are being unfair as a team!

The team continues to argue with John. The tension in the team room is rising.

Lesson review

In this lesson you:

- ▶ Examined the responsibilities of the Scrum Master role
- ▶ Discussed the characteristics of an effective Scrum Master
- ▶ Explored how to build high-performing teams
- ▶ Explored how to facilitate effective SAFe team events
- ▶ Practice coaching the Agile Team using powerful questions
- ▶ Discussed ways to collaborate with other teams
- ▶ Practice resolving team conflicts



2-46

SCALED AGILE® © Scaled Agile, Inc.

Notes:



Toolbox: Being an effective Scrum Master

- ▶ **Step 1:** Locate the Scrum Master Toolbox section in your Workbooks
- ▶ **Step 2:** Add more tools or techniques to the Toolbox by reflecting on the following:
 - What are some tools or techniques for conflict resolution?
 - What are some tools or techniques you can implement for coaching the Agile Team?
- ▶ **Step 3:** Share one of your insights with the class



PREPARE & SHARE



2-47

SCALED AGILE® © Scaled Agile, Inc.

Notes:

Key Learnings & Insights



Lesson 2

Characterizing the Role of
the Scrum Master

Lesson 3

Experiencing PI Planning

Learning objectives:

- 3.1 Prepare to experience PI Planning
- 3.2 Create and review draft PI plans
- 3.3 Finalize plans and establish business value
- 3.4 Review final plans and commit to a set of PI Objectives
- 3.5 Facilitate an effective PI Planning



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

3.1 Prepare to experience PI Planning

SCALED AGILE® © Scaled Agile, Inc.

3-3

Notes:



Video: PI Planning



DURATION



3-4

SCALED AGILE® © Scaled Agile, Inc.

Notes:

3.1 Prepare to experience PI Planning

PI Planning

Cadence-based PI Planning meetings are the heartbeat of the Agile Enterprise

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



Development Team

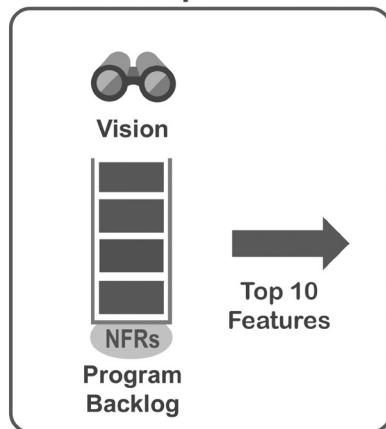
SCALED AGILE® © Scaled Agile, Inc.

3-5

Notes:

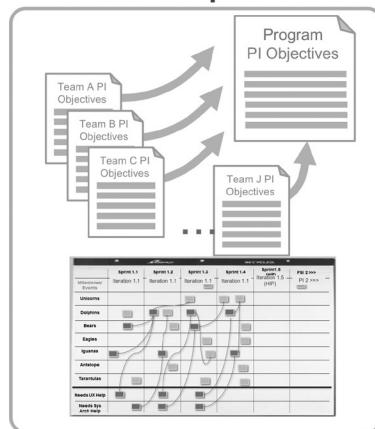
The PI Planning process

Input



Vision and top 10 Features

Output



Team and Program PI Objectives and Program Board

SCALED AGILE® © Scaled Agile, Inc.

3-6

Notes:

3.1 Prepare to experience PI Planning

What is a Feature?

- ▶ ‘Feature’ is an industry-standard term familiar to marketing and Product Management
- ▶ Fits in one Program Increment for one ART
- ▶ Includes *acceptance criteria*
- ▶ Describes larger system behaviors that fulfill users’ needs
- ▶ Expressed in plain language in a simple *feature and benefit (FAB) matrix*

Features acceptance criteria: Example for a Network Router

Feature:
- In-service software update
Acceptance Criteria:
- Nonstop routing availability
- Automatic and manual update support
- Rollback capability
- Support through existing admin tools
- All enabled services are running after the update

SCALED AGILE® © Scaled Agile, Inc.

Features and Benefits (FAB) Matrix

Feature	Benefit Hypothesis
In-service software update	Significantly reduced planned downtime
Hardware VPN acceleration	High-performance encryption for secure WAN
Traffic congestion management	Improve overall quality of service across different protocols
Route optimization	Improve quality of service due to faster and more reliable connectivity

3-7

Notes:

3.1 Prepare to experience PI Planning



Activity: Feature writing

- ▶ **Step 1:** Consider the following: Your team is tasked with coming up with a new *Personal Assistance Mobile App*
- ▶ **Step 2:** As a team, brainstorm five Features you would like to see on the new app and write them down on a flipchart sheet
- ▶ **Step 3:** Choose at least two Features and write down the details including:

- Title and description
- Acceptance criteria
- Benefit hypothesis



SCALED AGILE® © Scaled Agile, Inc.

3-8



Thought organizer

3.1 Prepare to experience PI Planning

How are Features implemented?

During PI Planning, features are broken down into smaller *Stories*.

- ▶ Stories are short description for a small piece of desired functionality, written in the user's language
- ▶ Fit in one Iteration for one team

Business Feature:
Sound Simulation
Benefit Hypothesis:
Improve the realism of the ride

User Stories: Enabler Stories:

As a rider I want to hear acceleration, braking, skidding, and crashing sounds so that I will feel immersed in the ride experience.

Test the two optional sound systems in the ride tunnel so we can select a system.

As a radar operator, I want to hear indicator sounds as we approach obstacles so that I can react better.

Record or buy some realistic sounds so we can test them with actual users.

3-9

SCALED AGILE® © Scaled Agile, Inc.

Notes:

User Stories: Focus on the user

- ▶ 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, device, or system 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 to get 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 save on paper.

SCALED AGILE® © Scaled Agile, Inc.

3-10

Notes:

3.1 Prepare to experience PI Planning

Using personas to better understand users

- ▶ Personas are detailed fictional characters acting as a representative user



Jane

- Law-abiding driver
- obeys all traffic signs
- Wants to save on gas

Mileage sensitive

As Jane, I want to travel at the legal limit and operate in an energy saving manner so that I do not get a ticket and I save money



Bob

- Impatient driver
- Ignores traffic signs if they slow him down

Time sensitive

As Bob, I want to travel at the maximum speed the roadway and my vehicle safely allows so that I arrive quickly

SCALED AGILE® © Scaled Agile, Inc.

3-11

Notes:

INVEST in a good story

I

Independent

▶ Write stories that can be developed separately

N

Negotiable

▶ Write stories in which scope can be negotiated

V

Valuable

▶ Write stories that are valuable to the Customer

E

Estimable

▶ Write stories that can be estimated

S

Small

▶ Write stories that can fit in an Iteration

T

Testable

▶ Write stories that are testable

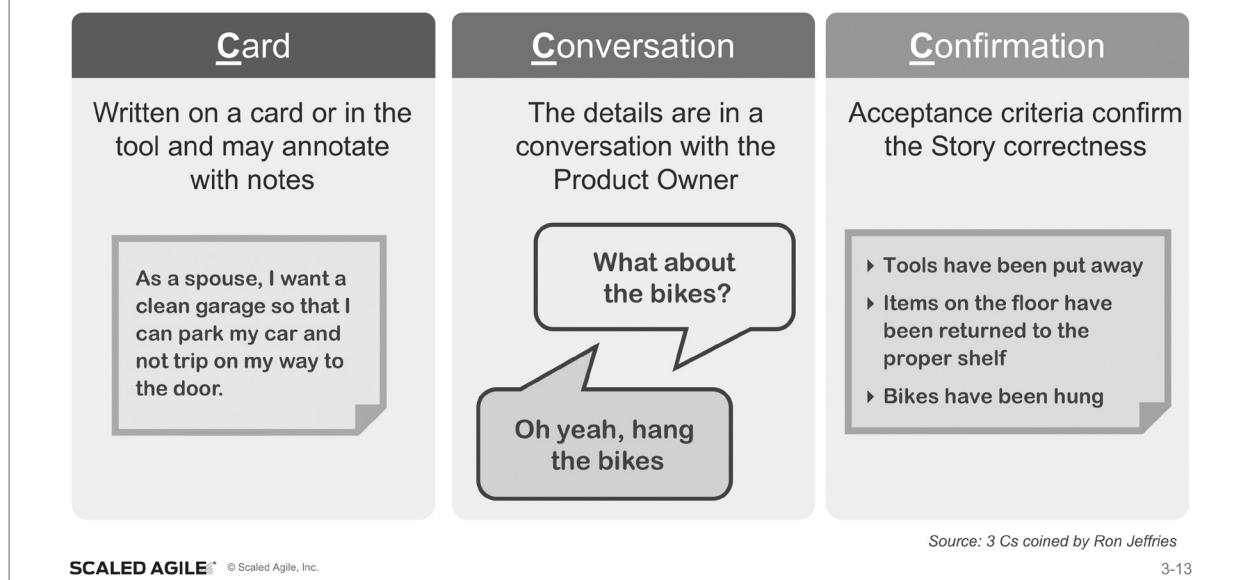
SCALED AGILE® © Scaled Agile, Inc.

3-12

Notes:

3.1 Prepare to experience PI Planning

Writing good stories: The 3Cs



SCALED AGILE® © Scaled Agile, Inc.

Source: 3 Cs coined by Ron Jeffries

3-13

Notes:

Acceptance criteria and acceptance tests

Acceptance criteria

- ▶ Provide the details of the Story from a testing point of view
- ▶ Define acceptable behavior

Story Acceptance Criteria

Given car is moving
When speed is set
Then speed is close to set speed

Acceptance tests

- ▶ Derive from acceptance criteria
- ▶ Define specific pass/fail behavior

Story Acceptance Test

Given car is moving at 10 mph
When speed is set to 30 mph
Then car is at 29 mph in less than 5 seconds
And car speed is no lower than 29 mph and no higher than 31 mph

SCALED AGILE® © Scaled Agile, Inc.

3-14

Notes:

3.1 Prepare to experience PI Planning



Activity: Story writing

- ▶ **Step 1:** Review the Features you wrote for the *Personal Assistant Mobile App*
- ▶ **Step 2:** As a team, using personas and the 3C's guidance, on a flipchart, write 5 to 7 Stories with acceptance criteria
- ▶ **Step 3:** Share an example with the class



SCALED AGILE® © Scaled Agile, Inc.



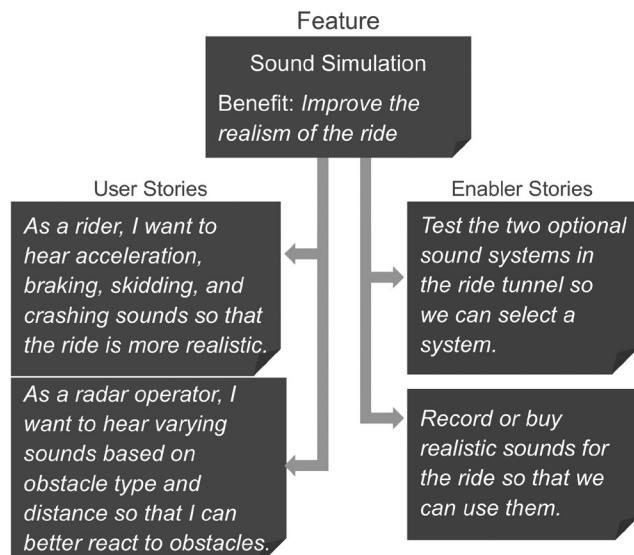
Thought organizer

3.1 Prepare to experience PI Planning

Enabler Stories

- ▶ May include any of the following:

- Refactoring and Spikes
- Building or improving infrastructure
- Verification of system qualities



SCALED AGILE® © Scaled Agile, Inc.

3-16

Notes:

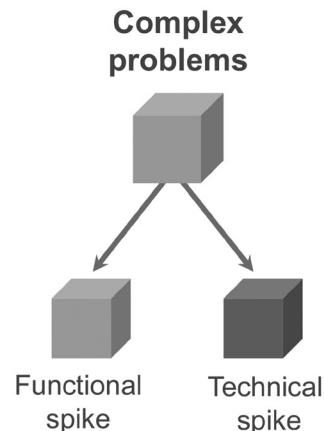
Spikes and Refactors

Refactors are a systematic approach to improving the system without changing observable system behavior

- ▶ Example: Improving maintainability, performance, or scalability

Spikes are research activities to reduce risk, understand a functional need, increase estimate reliability, or define a technical approach

- ▶ Technical spikes - Researching a technical approach or unknown
- ▶ Functional spikes - Researching how a user might use or interact with the system



SCALED AGILE® © Scaled Agile, Inc.

3-17

Notes:

3.1 Prepare to experience PI Planning

Estimate Stories with relative Story points

- ▶ A Story point is a singular number that represents:
 - Volume: how much is there?
 - Complexity: how hard is it?
 - Knowledge: what do we know?
 - Uncertainty: what's not known?
- ▶ Story points are relative. They are not connected to any specific unit of measure.

How big is it?



Guidance: Compared with other Stories an 8-point Story should take relatively 4x longer than a 2-point Story.

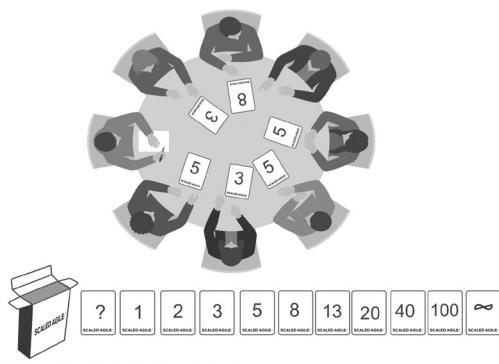
SCALED AGILE® © Scaled Agile, Inc.

3-18

Notes:

Apply Estimating Poker for fast, relative estimating

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



SCALED AGILE® © Scaled Agile, Inc.

Steps

- 1** Each estimator gets a deck of cards
- 2** Reads a job
- 3** Estimators privately select cards
- 4** Cards are turned over
- 5** Discuss differences
- 6** Re-estimate

Mike Cohn, *Agile Estimating and Planning*, 2005

3-19

Notes:

3.1 Prepare to experience PI Planning

Estimation is a whole-team exercise

- ▶ Increases accuracy by including *all* perspectives
- ▶ Builds understanding
- ▶ Creates shared commitment



The whole team estimates Stories

Warning: Estimation performed by a Manager, Architect, or select group, negates these benefits.

SCALED AGILE® © Scaled Agile, Inc.

3-20

Notes:

The Scrum Masters' role in facilitating estimations

Best Approaches

- Encourage everyone to participate
- Ensure relative estimates are used
- Focus the discussion around the contested items
- Identify subject matter experts who need to be present
- Keep time spent estimating Stories to a minimum

Common Anti-Patterns

- Pressure by stakeholders to lower estimations
- Only a few people participate
- Not using the adjusted Fibonacci scale



SCALED AGILE® © Scaled Agile, Inc.

3-21

Notes:

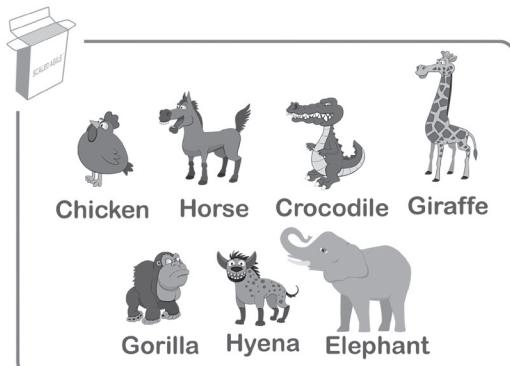
3.1 Prepare to experience PI Planning



Activity: Facilitating estimation

Use Estimating Poker to relatively estimate the mass of a set of animals.

- ▶ **Step 1:** Pick a person on the team to be in the role of a Scum Master and facilitate the estimation
- ▶ **Step 2:** As a team at your table, identify the smallest animal and mark it as 1. Estimate the remaining animals using values 1, 2, 3, 5, 8, 13, 20, 40, 100
- ▶ **Step 4:** Scrum Masters for each team share the experience



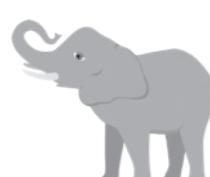
SCALED AGILE® © Scaled Agile, Inc.

Relative Size Estimating

Think in relative sizing of these animals. Which one would be smallest? Mark it as 1.

At your table, you will find a deck of Estimating Poker cards. As a team, use the cards to estimate the remainder of the animals.

If you identify the Hyena as 1. How would you relatively estimate the horse for example?



3.2 Create and review draft PI plans

SCALED AGILE® © Scaled Agile, Inc.

3-23

Notes:



PI Planning in Action



DURATION



3-24

SCALED AGILE® © Scaled Agile, Inc.

Notes:

3.2 Create and review draft PI plans



Activity: Identify team names and roles for PI Planning simulation

- ▶ **Step 1:** Your team is your table. Create a team name.
- ▶ **Step 2:** Name a Scrum Master for your team (it can be the same person you chose for facilitating the estimation)
- ▶ **Step 3:** Name a Product Owner for your team.
- ▶ **Step 4:** Make sure the team name and the names of the people selected are visible to all other teams.



PREPARE



3-25

SCALED AGILE® © Scaled Agile, Inc.

Notes:

3.2 Create and review draft PI plans



Activity: Identify program roles for PI Planning simulation

- ▶ **Step 1:** Get excited about the upcoming PI simulation!
- ▶ **Step 2:** Make sure all program roles have been assigned

Example: Your Instructor will be the RTE, a volunteer will be the Product Manager, etc.

Simulation role	Assigned to
Executive	Volunteer
Product Manager	Volunteer
System Architect, UX and Development Manager	Volunteer

SCALED AGILE® © Scaled Agile, Inc.



3-26

What is PI Planning?

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

The Agile Manifesto states, "The most efficient and effective method of conveying information to and within a development team is a face-to-face conversation."

SAFe takes this to the next level with PI planning, a routine, face-to-face event, with a standard agenda that includes a presentation of business context and vision followed by team planning breakouts—where the teams create their Iteration plans and objectives for the upcoming PI.

In the next few hours you will be immersed in a PI Planning simulation. With your teams, you will estimate your starting velocity and you will plan a short Program Increment with two iterations. You will get to observe a Scrum of Scrums event and you will present a summary of your team's draft PI Objectives.

Later, your trainer will demonstrate how business value is assigned to the objectives, how program risks are managed and you will recognize the value of the confidence vote.

Get excited!

There is no magic in SAFe . . . except maybe for PI Planning.
- Authors



3.2 Create and review draft PI plans



Simulation: Why are we here?

Presented by RTE

Alignment to a common mission

We are here to gain alignment and commitment around a clear set of prioritized objectives. I will now review the agenda for the next two days of the PI Planning Event.

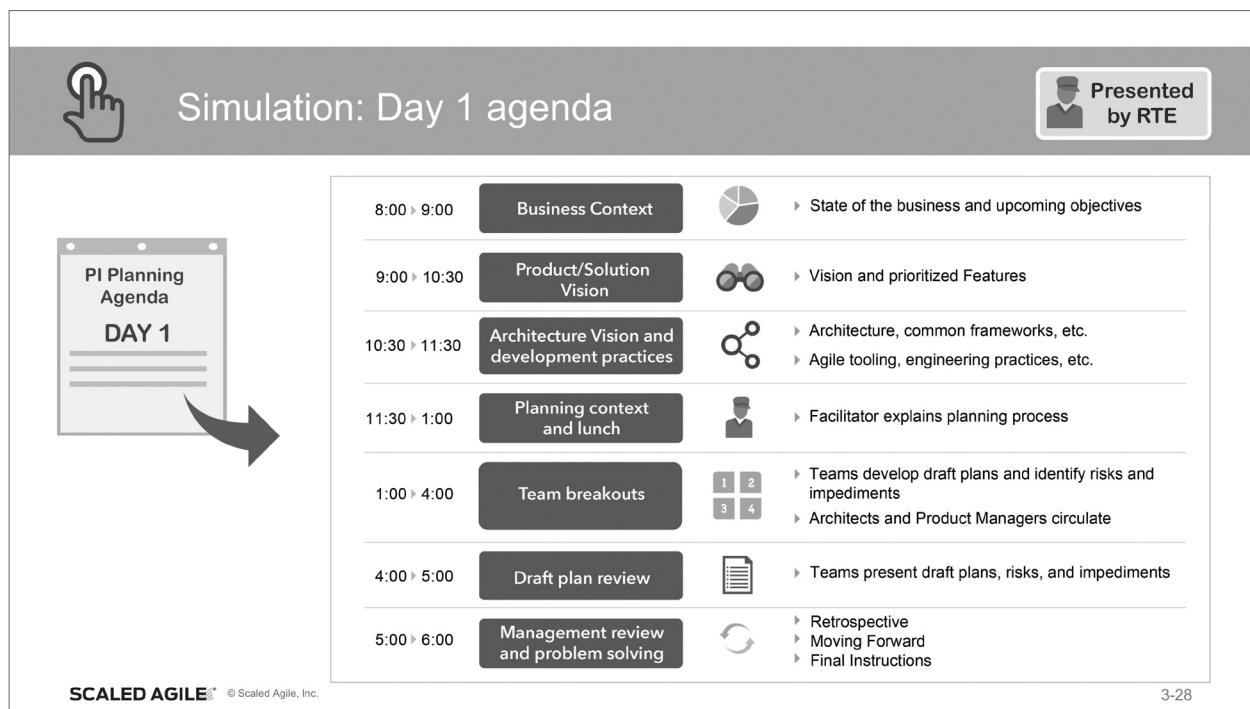
PI Planning Agenda

SCALED AGILE® © Scaled Agile, Inc.

3-27

A man in a suit stands next to a large presentation board. He is pointing at the board with his right hand. The board has a title 'Alignment to a common mission' and a text block about the purpose of the event. To the right of the board is a placeholder for a 'PI Planning Agenda' document, which is currently empty. The bottom left corner of the slide features the Scaled Agile logo.

Notes:



Simulation: Day 1 agenda

Presented by RTE

PI Planning Agenda DAY 1

Time	Activity	Icon	Description
8:00 ▶ 9:00	Business Context	Pie chart	▶ State of the business and upcoming objectives
9:00 ▶ 10:30	Product/Solution Vision	Binoculars	▶ Vision and prioritized Features
10:30 ▶ 11:30	Architecture Vision and development practices	Network	▶ Architecture, common frameworks, etc. ▶ Agile tooling, engineering practices, etc.
11:30 ▶ 1:00	Planning context and lunch	Person	▶ Facilitator explains planning process
1:00 ▶ 4:00	Team breakouts	1, 2, 3, 4	▶ Teams develop draft plans and identify risks and impediments ▶ Architects and Product Managers circulate
4:00 ▶ 5:00	Draft plan review	Document	▶ Teams present draft plans, risks, and impediments
5:00 ▶ 6:00	Management review and problem solving	Refresh	▶ Retrospective ▶ Moving Forward ▶ Final Instructions

SCALED AGILE® © Scaled Agile, Inc.

3-28

A slide titled 'Simulation: Day 1 agenda'. It features a 'PI Planning Agenda DAY 1' icon on the left with an arrow pointing to a detailed agenda table on the right. The table lists activities from 8:00 to 6:00 with corresponding icons and descriptions. The bottom left corner contains the Scaled Agile logo.

Notes:

3.2 Create and review draft PI plans

Simulation: Day 2 agenda



**PI Planning Agenda
DAY 2**

8:00 » 9:00 Planning adjustments  ▶ Planning adjustments made based on previous day's management meeting

9:00 » 11:00 Team breakouts  ▶ 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  ▶ If necessary, planning continues until commitment is achieved

After commitment Planning retrospective and moving forward  ▶ Adjustments made based on challenges, risks, and impediments

SCALED AGILE® © Scaled Agile, Inc.

3-29

Notes:

Simulation: Briefings



Executive 

Product Manager 

System Architect 

SCALED AGILE® © Scaled Agile, Inc.

3-30



Remember to refer to the PI Planning Briefings handout at your table.

3.2 Create and review draft PI plans

Simulation: Planning guidance

Presented by RTE



Expect this first PI Planning to feel a bit chaotic. Future PI Planning meetings will become more routine.

Product Owners: You have the content authority to make decisions at the user Story level

Scrum Masters: Your responsibility is to manage the timebox, the dependencies, and the ambiguities

Development Team: Your responsibility is to define users Stories, plan them into the Iteration, and work out interdependencies with other teams

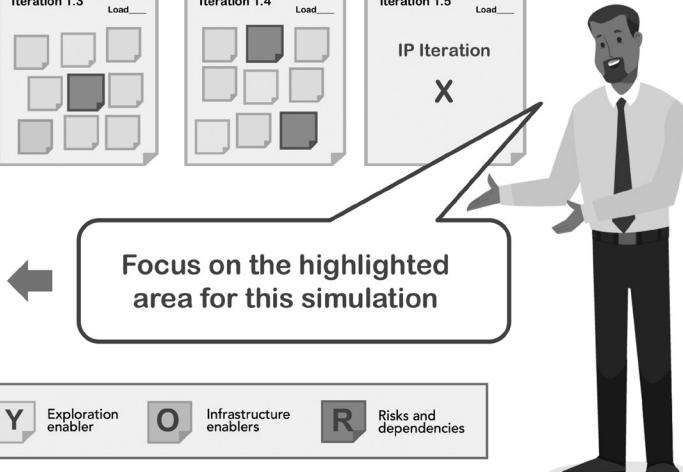
SCALED AGILE® © Scaled Agile, Inc.

3-31

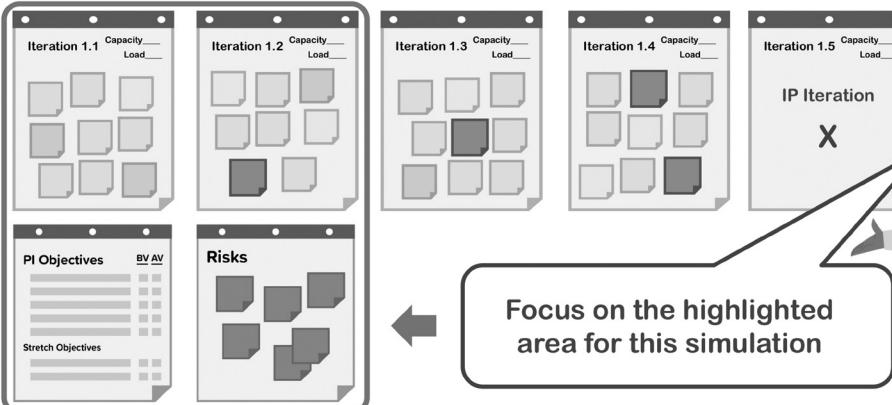
Notes:

Simulation: Planning requirements

Presented by RTE



Focus on the highlighted area for this simulation



Legend:

- G User story
- P Maintenance
- Y Exploration enabler
- O Infrastructure enablers
- R Risks and dependencies

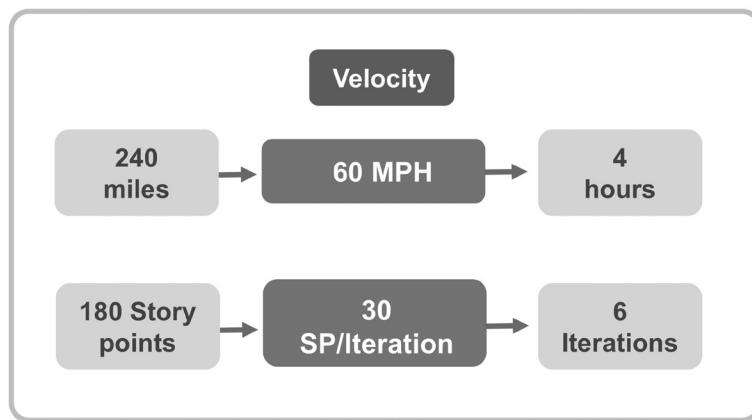
SCALED AGILE® © Scaled Agile, Inc.

Notes:



Simulation: Using historical data to calculate velocity

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



SCALED AGILE® © Scaled Agile, Inc.

3-33

Notes:



Simulation: Calculating Iteration capacity

Calculating Iteration capacity before *velocity* is established:

- ▶ For every full-time development team member, give the team 8 points (adjust for part-timers).
- ▶ Subtract 1 point for every team member vacation day and holiday.
- ▶ Find a small Story that would take about a half-day to develop and a half-day to test and validate. Call it a 1.
- ▶ Estimate every other Story relative to that one.
- ▶ Never look back (don't worry about recalibrating).



Example:

A 7-person team composed of 3 developers, 2 testers, 1 Product Owner, and 1 Scrum Master

Exclude The Scrum Master, Product Owner, and vacation time from the calculation.

Calculated Capacity:

$5 * 8 \text{ points} = 40 \text{ points per Iteration}$

Estimating velocity and calculating capacity: A brief introduction

Agile teams use story points to relatively estimate user stories in story points. With relative estimating, the size (effort) for each backlog item is compared to other stories. For example, an eight-point story is four times the effort as a two-point story. The team's velocity for an iteration is equal to the sum of all the stories completed in the prior iteration. Knowing a team's velocity assists with planning and helps limit Work in Process (WIP)—teams don't take on more stories than their prior velocity would allow. Velocity is also used to estimate how long it takes to deliver Features or Epics, which are also forecasted in story points.

Keep in mind, velocity is based on historical data of the team's completed story points. For the purpose of this PI Planning simulation you will be referring to calculating Iterations capacity, since velocity is not established yet.

More on velocity and capacity is coming up next.



Activity: Calculate capacity

- ▶ **Step 1:** Review the instructions and example provided in your Workbook
- ▶ **Step 2:** Calculate your own capacity for the next two, 2-week Iterations
 - The first Iteration starts Monday
 - Use your real availability
- ▶ **Step 3:** Make sure you have your team's capacity calculated

PREPARE



3-35

SCALED AGILE® © Scaled Agile, Inc.

What is Velocity?

The team's velocity for an iteration is equal to the sum of the points for all the completed stories that met their Definition of Done (DoD). As the team works together over time, their historical trend of average completed story points per iteration builds a reliable picture of the team's velocity.

What is Capacity?

Capacity is the portion of the team's velocity that is actually available for any given iteration. Vacations, training, and other events can make team members unavailable to contribute to an iteration's goals for some portion of the iteration. This decreases the maximum potential velocity for that team for that iteration.

Example:

Assuming a six-person team composed of three developers, two testers, and one PO, with no vacations or holidays, the estimated initial velocity = 5×8 points = 40 points/iteration. (Note: Adjusting a bit lower may be necessary if one of the developers and testers is also the Scrum Master.)

Using this example, and knowing the number of people on your team (at your table) estimate initial velocity.

Iteration 1

Team's Capacity:

Iteration 2

Team's Capacity:

The Scrum Master's role in team breakout #1

Best Approaches

- Ensure the team has a draft plan to present Identify as many risks and dependencies as possible for the management review
- Secure subject matter experts and Program stakeholders as needed by the team
- Facilitate the coordination with other teams for dependencies

Common Anti-Patterns

- No plan or partial plan at the end of the timebox
- Too much time is spent analyzing each Story
- Shared Scrum Masters and Product Owners are not available enough
- Part-time Scrum Masters don't have time to plan as part of the team



SCALED AGILE® © Scaled Agile, Inc.

3-36

Notes:



Activity: Team breakout #1

You will be planning a short Program Increment with two Iterations.

- ▶ **Step 1:** Setup team area. Enter the capacity for each Iteration.
- ▶ **Step 2:** Pick up a Feature from the Product Manager
- ▶ **Step 3:** Estimate the stories using Story Points
- ▶ **Step 4:** Load the stories into the Iterations
- ▶ **Step 5:** Write the PI Objectives using clear statements
- ▶ **Step 6:** Identify the stretch objectives
- ▶ **Step 7:** Identify any Program Risks and dependencies



SCALED AGILE® © Scaled Agile, Inc.

3-37

As a Scrum Master, remind the team that when estimating stories to consider that a story point is a singular number that represent:

- Volume: how much is there?
- Complexity: how hard is it?
- Knowledge: what do we know?
- Uncertainty: what's not known?

Compared with other Stories an 8-point Story should take relatively 4 times longer than a 2-point Story.

PI Objectives:

Objectives are business summaries of what each team intends to deliver in the upcoming PI. They often map directly to the Features in the backlog, but not always.

Stretch Objectives:

Stretch objectives are used to identify work that can be variable within the scope of a PI. Stretch objectives are not the way for stakeholders to load the teams with more work than they can do. It's not extra stuff to do, just in case time permits.

3.2 Create and review draft PI plans



Activity: Scrum of Scrums (SoS) sync

- ▶ **Step 1:** Observe the SoS sync, conducted by The RTE
- ▶ **Step 2:** Each team's Scrum Master provides the team's current status and addresses the questions from the RTE
- ▶ **Step 3:** RTE holds a 'Meet After' the sync (limited to 1–2 topics for the simulation)

SoS Sync Question	Team 1	Team 2
Have you identified the capacity for each Iteration in the PI?		
Have you identified most of the Stories for the first two Iterations and begun estimating?		
Have you begun resolving dependencies with other teams?		
Are you discussing trade-offs and conflicting priorities with your Business Owners?		
Have you identified any program risks?		
Will you be ready to start writing PI Objectives in the next 15 minutes?		
Is there anything you need to discuss with other Scrum Masters? If so, stay for the 'Meet After'		

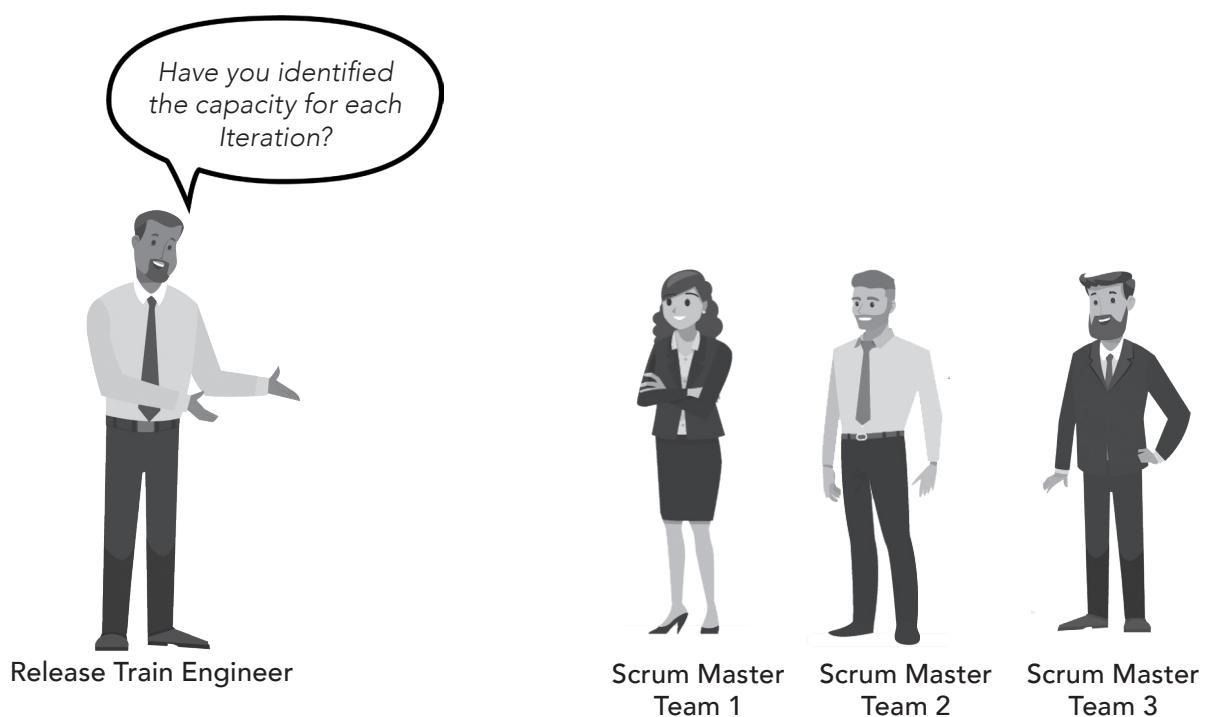
DURATION



3-38

SCALED AGILE® © Scaled Agile, Inc.

Scrum of Scrums (SoS) Sync



3.2 Create and review draft PI plans



Activity: Draft plan review

- ▶ **Step 1:** Present the summary of your team's first two Iterations and one or more draft PI Objectives
- ▶ **Step 2:** Make sure that you have included the following:
 - Capacity and load for each Iteration
 - Draft PI Objectives
 - Program risks and impediments



SCALED AGILE® © Scaled Agile, Inc.

Notes:

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? Team assignments?
- ▶ Where are the bottlenecks?
- ▶ What features must be de-scoped?
- ▶ What decisions must we make between now and tomorrow to address these issues?



SCALED AGILE® © Scaled Agile, Inc.

3-40

Notes:

3.3 Finalize plans and establish business value

SCALED AGILE® © Scaled Agile, Inc.

3-41

Notes:

Activities during Day 2

DAY 1

8:00 ▶ 9:00	Business Context	
9:00 ▶ 10:30	Product/Solution Vision	
10:30 ▶ 11:30	Architecture Vision and development practices	
11:30 ▶ 1:00	Planning context and lunch	
1:00 ▶ 4:00	Team breakouts	
4:00 ▶ 5:00	Draft plan review	
5:00 ▶ 6:00	Management review and problem solving	

SCALED AGILE® © Scaled Agile, Inc.

DAY 2

8:00 ▶ 9:00	Planning adjustments	
9:00 ▶ 11:00	Team breakouts	
11:00 ▶ 1:00	Final plan review and lunch	
1:00 ▶ 2:00	Program risks	
2:00 ▶ 2:15	PI confidence vote	
2:15 ▶ ???	Plan rework if necessary	
After commitment	Planning retrospective and moving forward	

3-42

Notes:

3.3 Finalize plans and establish business value

Make planning adjustments

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

Possible changes:

- ▶ Business priorities
- ▶ Adjustment to vision
- ▶ Changes to scope
- ▶ Movement of people



Teams adjusting for change in scope

SCALED AGILE® © Scaled Agile, Inc.

3-43

Notes:

Team breakout #2

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

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

Objectives for PI 1	Business Value	Actual Value
Structured location and validation of locations <ul style="list-style-type: none">• Navigate autonomously from distribution center to top 5 most frequent destinations• Park at 1 building that requires parallel parking• Reduce GPS signal loss by 25%• Build and demonstrate proof of concept for next generation vehicle navigation systems	7	
Stretch Objectives for PI 1 <ul style="list-style-type: none">• Spike: conduct hijack testing of the vehicle sensors	8	
	10	
	7	
	4	

SCALED AGILE® © Scaled Agile, Inc.

3-44

Notes:

3.3 Finalize plans and establish business value



Activity: Setting business value

The instructor will demonstrate assigning business value for one team's objectives.

- ▶ **Step 1:** Bring the Business Owners to one team's draft plans
- ▶ **Step 2:** The Business Owners will set value on a scale of 1-10 for each identified objective
- ▶ **Step 3:** Observe the discussion that would take place, illustrating the larger purposes and thought processes around assigning business value

Objectives for PI 1	Business Value	Actual Value
Structured location and validation of locations		
• Navigate autonomously from distribution center to top 5 most frequent destinations	7	
• Park at 1 building that requires parallel parking	8	
• Reduce GPS signal loss by 25%	10	
• Build and demonstrate proof of concept for next generation vehicle navigation systems	7	
Stretch Objectives for PI 1		
• Spike: conduct hijack testing of the vehicle sensors	4	



3-45

SCALED AGILE® © Scaled Agile, Inc.

Notes:



Discussion: Facilitating Team Breakout #2

Scenario: During the second team breakout, the Business Owners come to your team. The team has picked up several items that are meant to reduce technical debt and build a testing automation infrastructure.

As the Business Owners are from the business side of the Enterprise, they rank all of these objectives as 4 or lower. You can see that the team becomes upset.

- ▶ **Step 1:** As a team, discuss the following:
 - What can the Scrum Master do?
 - How can the Scrum Master help avoid this problem before it happens?
- ▶ **Step 2:** Share with the class



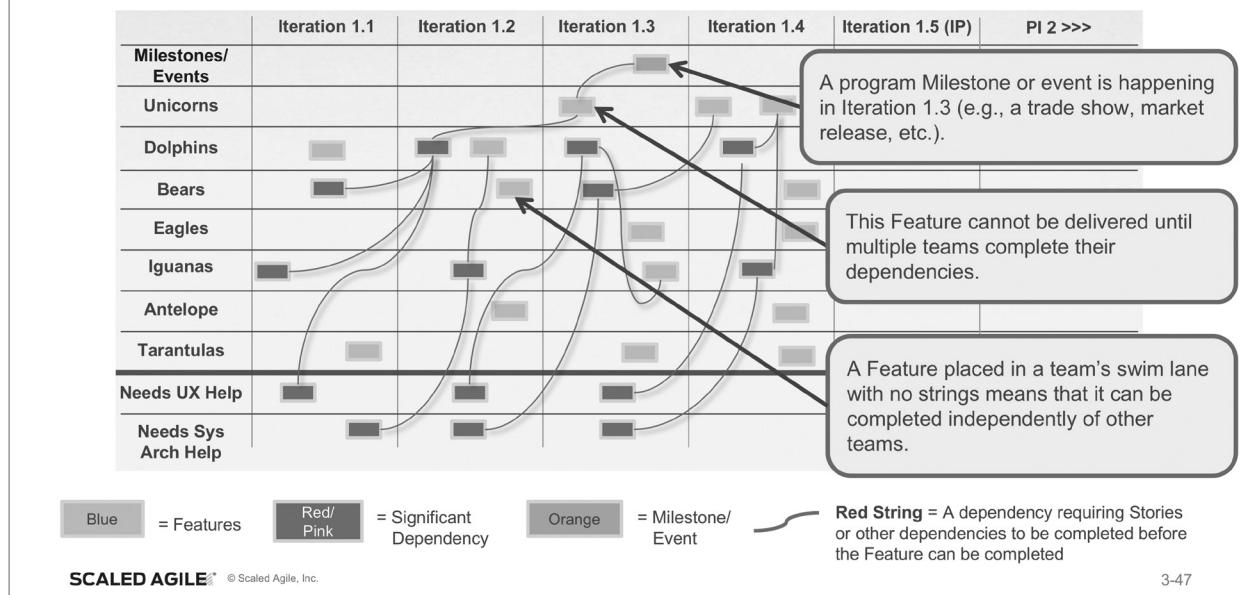
3-46

SCALED AGILE® © Scaled Agile, Inc.

Notes:

3.3 Finalize plans and establish business value

PI Planning output: The Program Board



Notes:

3.4 Review final plans and commit to a set of PI Objectives

SCALED AGILE® © Scaled Agile, Inc.

3-48

Notes:

Final plan review

Teams and Business Owners peer review all final plans.



Used with permission of SEI Global Wealth Services

SCALED AGILE® © Scaled Agile, Inc.

3-49

Notes:

3.4 Review final plans and commit to a set of PI Objectives

Building the final plan

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



Used with permission of Discount Tire Corporation

SCALED AGILE® © Scaled Agile, Inc.

3-50

Notes:

Addressing program risks



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

ROAMing risks:

Resolved Has been addressed. No longer a concern.

Owned Someone has taken responsibility.

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

Mitigated Team has plan to adjust as necessary.



SCALED AGILE® © Scaled Agile, Inc.

3-51

Notes:

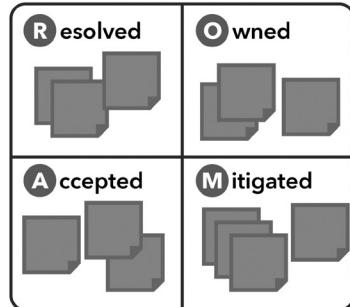
3.4 Review final plans and commit to a set of PI Objectives



Activity: Manage program risks

The Instructor will demonstrate ROAMing one to two risks for one team

- ▶ **Step 1:** Pick one to two risk examples.
- ▶ **Step 2:** Read them in front of all teams and stakeholders.
- ▶ **Step 3:** Ask if anyone can own, help mitigate, or resolve the risks. Otherwise, accept 'as-is'.
- ▶ **Step 4:** Put each risk into a corresponding quadrant of the ROAM sheet for the program.



PREPARE
 10 min

3-52

SCALED AGILE® © Scaled Agile, Inc.

Notes:

Confidence vote: Team and Program



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

A commitment with two parts:

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



No confidence



Little confidence



Good confidence



High confidence



Very high confidence



SCALED AGILE® © Scaled Agile, Inc.

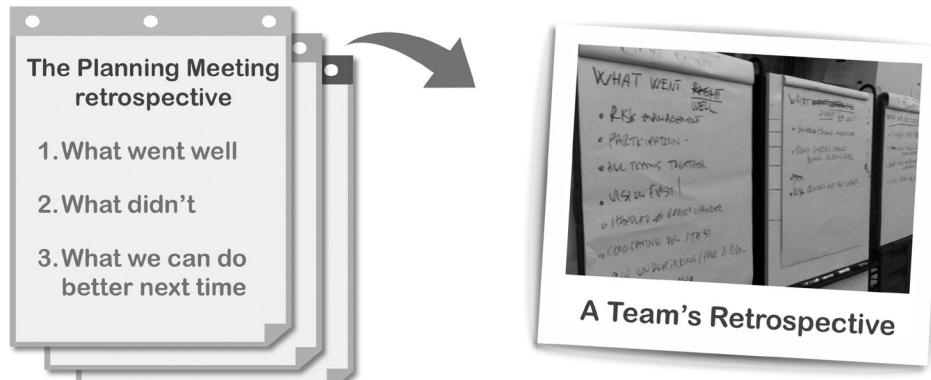
3-53

Notes:

3.4 Review final plans and commit to a set of PI Objectives

Run a planning meeting retrospective

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



Add the items to your Program Backlog and take action

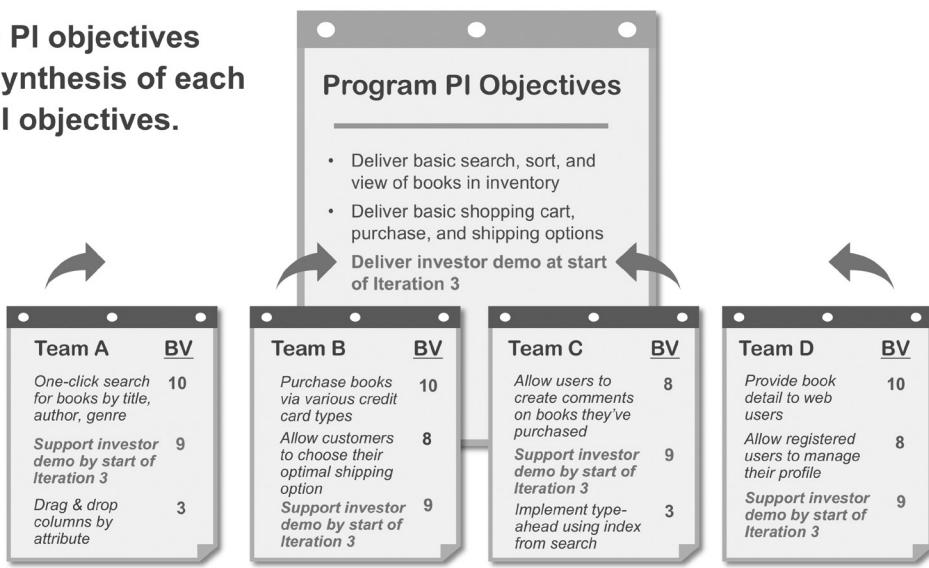
SCALED AGILE® © Scaled Agile, Inc.

3-54

Notes:

RTE take-a-way: Integrated PI objectives

Program PI objectives
are the synthesis of each
team's PI objectives.



SCALED AGILE® © Scaled Agile, Inc.

3-55

Notes:



Discussion: Simulation debriefing

- ▶ **Step 1:** Think about your experience during the PI Planning simulation.
- ▶ **Step 2:** Share the most exciting moments and some new insights. What have you learned?



PI Planning is about alignment

DISCUSS



3-56

SCALED AGILE® © Scaled Agile, Inc.



Thought organizer

3.5 Facilitate an effective PI Planning

SCALED AGILE® © Scaled Agile, Inc.

3-57

Notes:



PI Planning Event: Multiple Perspectives



DURATION



3-58

SCALED AGILE® © Scaled Agile, Inc.

Notes:

3.5 Facilitate an effective PI Planning



Discussion: Facilitating PI Planning

► **Step 1:** With your team discuss:

- What challenges do you see coming up during PI Planning?
- How can you in the role of the Scrum Master help solve these challenges?

► **Step 2:** Share with the class

PREPARE SHARE



3-59

SCALED AGILE® © Scaled Agile, Inc.



Thought organizer

3.5 Facilitate an effective PI Planning

The Scrum Master's role in PI Planning

Best Approaches

- Maintain the timebox
- Make sure the team builds a plan they can commit to
- Ensure that the team is honest in their confidence vote
- Facilitate the coordination with other teams, but don't do it for the team
- Act as a request buffer for a team that has a lot of dependencies
- Manage the program board
- Facilitate the retrospective

Common Anti-Patterns

- Pressure is put on the team to overcommit
- Team under-commits due to fear of failure
- Over-planning ahead of time to make it more efficient loses the essence of PI Planning
- The plan, rather than the alignment, become the goal



SCALED AGILE® © Scaled Agile, Inc.

3-60

Notes:

Lesson review

In this lesson you:

- ▶ Experienced PI Planning
- ▶ Created and reviewed draft PI Plans
- ▶ Finalized plans and established business value
- ▶ Reviewed final plans and committed to a set of PI Objectives
- ▶ Explored the Scrum Master's role in facilitating effective PI Planning



3-61

SCALED AGILE® © Scaled Agile, Inc.

Notes:



Toolbox: Being a Scrum Master during PI Planning event

- ▶ **Step 1:** Locate the Scrum Master Toolbox section in your Workbooks
- ▶ **Step 2:** Add more tools or techniques to the Toolbox by reflecting on the following:
 - What are some tools or techniques for preparing for PI Planning?
 - What are some tools or techniques for aligning the team towards achieving the PI Objectives?
- ▶ **Step 3:** Share one of your insights with the class



PREPARE & SHARE



3-62

SCALED AGILE® © Scaled Agile, Inc.

Notes:

Key Learnings & Insights



Lesson 3
Experiencing PI Planning

Lesson 4

Facilitating Iteration execution

Learning objectives:

- 4.1 Plan the Iteration
- 4.2 Track the Iteration progress
- 4.3 Refine the Backlog
- 4.4 Facilitate the Iteration Review
- 4.5 Facilitate relentless improvement
- 4.6 Support DevOps and Release on Demand



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

4.1 Plan the Iteration

SCALED AGILE® © Scaled Agile, Inc.

4-3

Notes:



Running an Effective SAFe Iteration Planning Meeting

Running an Effective SAFe® Iteration Planning Meeting

SCALED AGILE®
Provider of SAFe®

DURATION



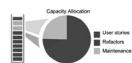
4-4

SCALED AGILE® © Scaled Agile, Inc.

Notes:

Iteration Planning flow

1 Establishing capacity



2 Story analysis and estimating



3 Detailing Stories



4 Developing Iteration goals



5 Committing to Iteration goals



Iteration Planning

- Timebox: Four hours
- This meeting is by and for the team
- SMEs may attend as required

SCALED AGILE® © Scaled Agile, Inc.

4-5

Notes:

Establishing capacity

- ▶ Team applies 'capacity allocation' to the team backlog
- ▶ Team quantifies capacity to perform work in the upcoming iteration
- ▶ Each team member determines their availability, acknowledging time off and other potential duties
- ▶ The PO in collaboration with the teams select the highest-priority backlog items for each 'slice' of the capacity allocation to implement in an iteration

SCALED AGILE® © Scaled Agile, Inc.

4-6

Notes:

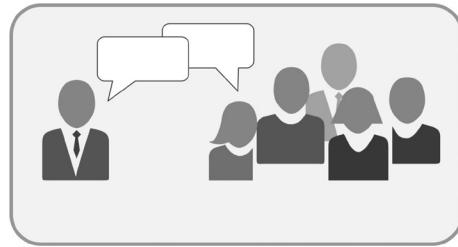
Story analysis and estimating



- ▶ The Product Owner presents Stories in order of priority

- ▶ Each Story

- Is discussed and analyzed by the team
 - Has its acceptance criteria defined and refined
 - Is estimated



- ▶ The process continues until the estimation of the Stories has reached the velocity of the team

SCALED AGILE® © Scaled Agile, Inc.

4-7

Notes:

Detailed Stories



Detailing stories is mostly used by beginner teams. Team members discuss:

- ▶ Who would be the best person to accomplish it
- ▶ How long would it take approximately
- ▶ What are the dependencies it may have to other stories



Story Detailing

SCALED AGILE® © Scaled Agile, Inc.

4-8

Notes:

Iteration Goals



Iteration Goals provide clarity, commitment, and management information.

They serve three purposes:

1. Align team members to a common purpose
2. Align Agile 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

Notes:

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 the concern.

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 do the work.
They are committed to other teams, the program, and the stakeholders.

Notes:



Role Play: Iteration Planning

- ▶ **Step 1:** Select a Scrum Master for your team
- ▶ **Step 2:** Review the *Team Backlog* in your Workbook
- ▶ **Step 3:** As a team, plan how you will execute an Iteration
- ▶ **Step 4:** As a Scrum Master, facilitate the Iteration planning for the team
- ▶ **Step 5:** Using a flip chart sheet outline your Iteration Plan

PREPARE



4-11

SCALED AGILE® © Scaled Agile, Inc.

See Team Backlog on the next page >>>

4.1 Plan the Iteration

Team Backlog

Item #	Size	Required*	Backlog Item
1		x	Estimate the pages in the course workbook.
2			Calculate the square root of 54,289 without a computer or calculator.
3			Add the following numbers with a calculator and be certain the answer is correct: 1, 2, 3, 5, 8, 13, 20, 40, 100
4			Accurately count the pages in the course workbook.
5			Introduce yourself to every person on your Scrum team and write down their children's names.
6			Add the following numbers without a calculator and be certain the answer is correct: 1,2,3,5,8,13,21,40,100
7		x	Calculate how tall your Scrum team is when each member is stacked vertically.
8			Write a program, without Excel, that accepts 10 numbers from a user and displays the total as each number is entered.
9			Calculate the cubic volume of the room to within approximately 30%.
10			Calculate the cubic volume of this room to within 10%.
11			Estimate the snowfall in Oulu, Finland this winter in centimeters.
12			Estimate the number of words in the workbook.
13			Estimate the cubic meters of snowfall in Oulu, Finland this winter.
14			Obtain an accurate count of the number of words in the course workbook.
15		x	Calculate how tall ALL the Scrum teams are when stacked vertically. (Note: Don't double count the PO!)

The Scrum Master's role in Iteration Planning

Best Approaches

- Maintain timebox
- Ensure that the team commits to the Iteration Goals
- Verify that the PO or other managers don't influence the team to overcommit
- Challenge the team to exceed their previous accomplishments
- Ensure that improvement items from the retrospective are put into effect
- Ensure time is allocated for technical debt activities

Common Anti-Patterns

- Delving too deep into technical discussions
- Commitment is unrealistic
- Velocity and load are exactly the same
- Scrum Master is more focused on technical hat than facilitator's hat
- The team under-commits due to fear of failure
- No time is reserved for support activities



SCALED AGILE® © Scaled Agile, Inc.

4-12

Notes:

4.2 Track the Iteration progress

SCALED AGILE® © Scaled Agile, Inc.

4-13

Notes:



Video: Running an Effective SAFe Daily Stand-up

Running an Effective SAFe® Daily Stand-up

SCALED AGILE®
Provider of SAFe®

DURATION



4-14

SCALED AGILE® © Scaled Agile, Inc.

Notes:

Daily Stand-up patterns

Basic Scrum pattern meeting agenda

Each person answers:

1. What did I do yesterday to advance the Iteration Goals?
2. What will I do today to advance the Iteration Goals?
3. Are there any impediments that will prevent the team from meeting the Iteration Goals?

The meet-after agenda

1. Review topics the Scrum Master wrote on the meet-after board.
2. Involved parties discuss, uninvolved people leave.

SCALED AGILE® © Scaled Agile, Inc.

4-15

Notes:

Daily Stand-up anti-patterns

Poor daily stand-ups may be a symptom of a deeper problem that requires a more systematic approach.



Potential root causes:

- ▶ Poor collaboration of the team members during the Iteration (e.g., Vijay does not know and doesn't care about what Ken is working on, and vice versa)
- ▶ Lack of collective code ownership
- ▶ Infrequent verification and integration during the Iteration (e.g., we are working on something, and we think it's good)
- ▶ Perpetual, unresolved conflict within the team

SCALED AGILE® © Scaled Agile, Inc.

4-16

Notes:

4.2 Track the Iteration progress



Role Play: Facilitating the Daily Stand-up

- ▶ **Step 1:** Choose a Scrum Master and have him/her read your team project
- ▶ **Step 2:** Pick up a secret identity card but don't show it to others
- ▶ **Step 3:** Run a Daily Stand-up playing your role as assigned by the card
 - If the Scrum Master calls you on your specific behavior, you can stop
- ▶ **Step 4:** Share your experience with the class:
 - Scrum Master, how was it for you?
 - Team members, what insights do you have for the Scrum Master?
 - How can we deal with those behaviors when they come up in Daily Stand-up meetings?



SCALED AGILE® © Scaled Agile, Inc.

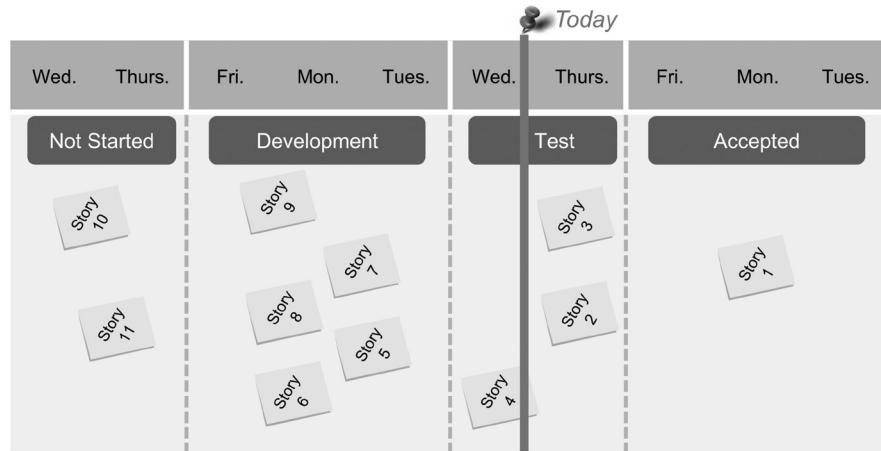


Thought organizer

4.2 Track the Iteration progress

Using the team board to track progress

The board acts as a Big Visible Information Radiator (BVIR).



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

SCALED AGILE® © Scaled Agile, Inc.

4-18

Notes:

Setting WIP limits

WIP limits improve the flow of work. Some steps have no WIP limits, while others serve as buffers and have minimal as well as maximal WIP.



SCALED AGILE® © Scaled Agile, Inc.

4-19

Notes:

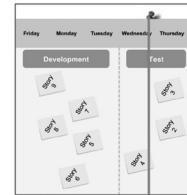
4.2 Track the Iteration progress



Activity: Work-In-Process (WIP) constraints

- **Step 1:** Considering the Big Visible Information Radiator (BVIR), discuss:

- What would be the effect of a 3-Story WIP constraint on development and testing?



- **Step 2:** Consider the scenario: You're a developer and you just finished Story 6. What would you do if:

- There is no WIP constraint?
- The 3-Story WIP constraint is in place?
- Which scenario has the highest throughput?



SCALED AGILE® © Scaled Agile, Inc.

4-20

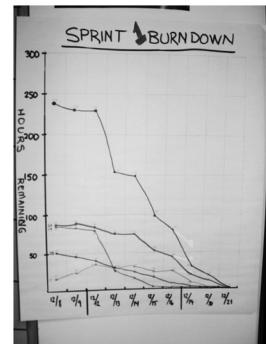


Thought organizer

4.2 Track the Iteration progress

Iteration burn-down

- ▶ Many Scrum teams use Iteration burn-down charts
- ▶ Burn-downs count the remaining effort (Stories, tasks, etc.)
- ▶ As we don't advocate tasks in SAFe, we prefer burn-ups and CFDs
- ▶ Burn-down charts provide several other challenges:
 - Focus on tasks completed vs. Stories completed
 - Hard to distinguish between work added and not done

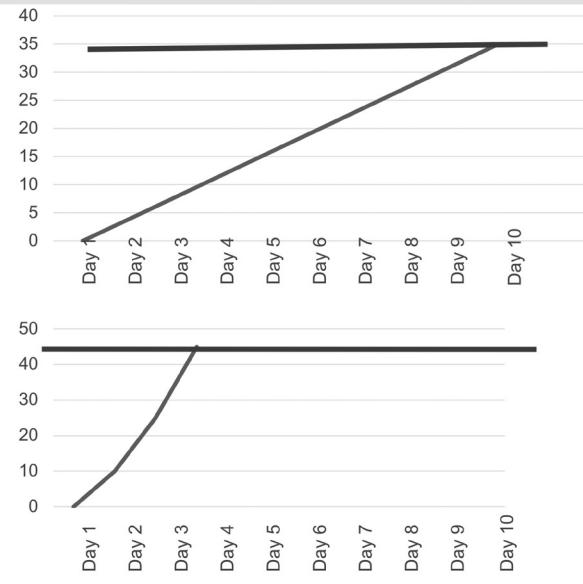


SCALED AGILE® © Scaled Agile, Inc.

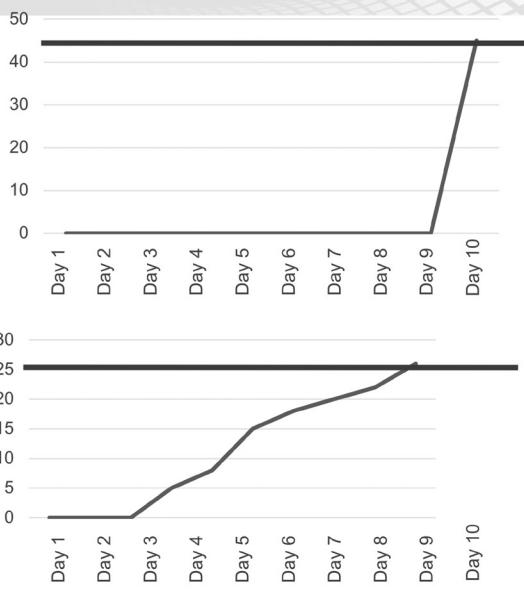
4-21

Notes:

Example burn-up charts



SCALED AGILE® © Scaled Agile, Inc.



4-22

Notes:

4.2 Track the Iteration progress



Role Play: Iteration Execution

- ▶ **Step 1:** Execute the Iteration that you planned earlier by completing as many backlog items as possible
- ▶ **Step 2:** Select a Scrum Master and a Product Owner

IMPORTANT: As a Scrum Master, ensure you are available to the team. As a PO make sure to refer to the *Acceptance Criteria for Product Owners Only* manual in the Workbook

- ▶ **Step 3:** Share your experience with the class



4-23

SCALED AGILE® © Scaled Agile, Inc.

Notes:

The Scrum Master's role in tracking Iteration progress

Best Approaches

- Facilitate mid-PI re-planning.
- Encourage the team to point out as early as possible if they think they will miss Iteration goals or PI Objectives. Communicate to and from the Scrum of Scrums.
- Encourage the use of engineering practices.
- Make sure defects are not pushed to the IP Iteration.
- Facilitate preparation for the next PI.
- Support release activities.

Common Anti-Patterns

- Team gets no input from Scrum of Scrums
- Teams are unwilling to change or add objectives mid-PI
- Scrum Master does all the synchronization, so team is incapable of doing it themselves



SCALED AGILE® © Scaled Agile, Inc.

4-24

Notes:

4.3 Refine the Backlog

Notes:



Discussion: Backlog refinement

► **Step 1:** As a team discuss the split in responsibility for backlog refinement between the Product Owner and the Scrum Master:

- How can you, as the Scrum Master, help facilitate this process?
- A common problem is that the Dev Teams are spending too much time refining Stories. How would you facilitate this problem?
- What should the Scrum Master do if a Story does not have acceptance criteria?

► **Step 2:** Share with the class.



Thought organizer

The backlog refinement event

- ▶ Timebox: 1 – 2 hours weekly.
- ▶ Helps the team ‘sleep’ on new Stories prior to Iteration Planning.
- ▶ Provides time to identify dependencies and issues that could impact the next Iteration. Ensures that we have a ready backlog for Iteration planning.
- ▶ Agile Team members are in attendance and actively engaged, subject matter experts and other teams’ members are invited as needed.

Sample Backlog Refinement Event Agenda

1. The PO presents the set of candidate Stories for the next Iteration
2. The team discusses whether the set of candidate Stories should be reduced or increased; Stories are added or removed
3. The PO guides the team through the candidate Stories one by one:
 - a) The team discusses each Story, estimates it, and splits it if necessary
 - b) The PO clarifies or supplements the acceptance criteria
 - c) The team identifies dependencies on other teams
4. Action items are summarized for all Stories that still require external input or action

SCALED AGILE® © Scaled Agile, Inc.

4-27

Notes:

The Scrum Master’s role in backlog refinement

Best Approaches

- Maintain timeboxes
- Maintain the right level of a deep backlog vs. ready backlog for two Iterations
- Make sure all the team members participate
- Invite the right subject matter experts
- Hold the event at regular intervals

Common Anti-Patterns

- Arriving to the Iteration with non-ready Stories
- Not doing the backlog refinement consistently
- Team sees Stories for the first time during Iteration or PI Planning
- Feature estimations impact Story estimation



SCALED AGILE® © Scaled Agile, Inc.

4-28

Notes:

4.4 Facilitate the Iteration Review

SCALED AGILE® © Scaled Agile, Inc.

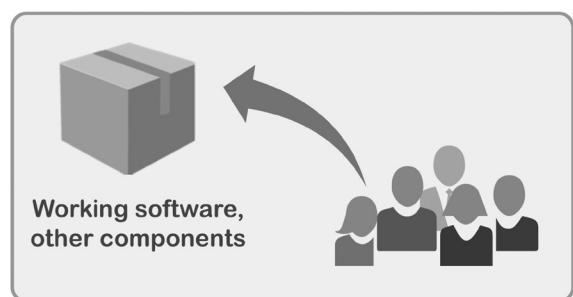
4-29

Notes:

The Iteration Review

- ▶ Provides the true measure of progress by showing working software functionality, hardware components, etc.
- ▶ Preparation for the review starts with planning
- ▶ Teams demonstrate every Story, Spike, Refactor, and NFR
- ▶ Attendees are the Team and its stakeholders

Demonstrating a working, tested team increment



SCALED AGILE® © Scaled Agile, Inc.

4-30

Notes:

Iteration Review guidelines

- ▶ **Timebox:** 1 to 2 hours.
- ▶ **Preparation:** Review preparation should be limited to 1 to 2 hours. Minimize presentation. Work from the repository of Stories.
- ▶ **Attendees:** If a major stakeholder cannot attend, the Product Owner should follow up individually.

Sample Iteration Review Agenda

1. Review business context and Iteration Goals
2. Demo and solicit feedback of each Story, spike, refactor, NFR
3. Discuss Stories not completed and why
4. Identify risks, impediments
5. Revise team backlog and Team PI Objectives as needed

SCALED AGILE® © Scaled Agile, Inc.

4-31

Notes:

Two views from the Iteration Review

The Iteration Review provides two views into the program, based on a working system:

1. How we did in the Iteration?
2. How we are doing in the PI?



Iteration Review

How we did in the Iteration:

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

How we're doing in the PI:

- ▶ Review of PI Objectives
- ▶ Review of remaining PI scope and reprioritizing if necessary

SCALED AGILE® © Scaled Agile, Inc.

4-32

Notes:

The Scrum Master's role in the Team and System Demos

Best Approaches

- Begin to consider how and what to demo in Iteration Planning
- Make sure the right participants are present
- Ensure that the team celebrates its accomplishments and that stakeholders acknowledge them
- Make sure different team members have the opportunity to demo
- Ensure that the team is ready for the System Demo and coordinates with the System Team

Common Anti-Patterns

- A lot of time is spent preparing for the demo
- Demo is mainly talk/slides as opposed to working software and/or hardware
- PO sees things for the first time in the Team Demo
- System Demo is not done because 'the Team Demo is enough'
- Team members are not invited to the System Demo, to save time
- Demos that are not interesting or relevant to Program-Level stakeholders



SCALED AGILE® © Scaled Agile, Inc.

4-33

Notes:

4.5 Facilitate relentless improvement

SCALED AGILE® © Scaled Agile, Inc.

4-34

Notes:

Relentless improvement

Agile Teams continuously adapt to new circumstances and improve the methods of value delivery.

- ▶ Understand where you are
- ▶ Foster the culture of ‘improving everywhere’
- ▶ Use retrospectives as summary points but not as limitations
- ▶ Support continuous learning
- ▶ Actively engage with other SMs to drive improvement on the Program



SCALED AGILE® © Scaled Agile, Inc.

4-35

Notes:

4.5 Facilitate relentless improvement

Improving everywhere

Address every area that surfaces as a constraint to the team's performance.

Examples:

- ▶ Testing: moving from manual to automated
- ▶ Communication with remote teams, subject matter experts, etc.
- ▶ The team's skill set
- ▶ Preparing and running the demo
- ▶ Nonfunctional Requirements (NFR) testing
- ▶ More efficient and disciplined design sessions

SCALED AGILE® © Scaled Agile, Inc.

4-36

Notes:

Iteration Metrics

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

SCALED AGILE® © Scaled Agile, Inc.

4-37

Notes:

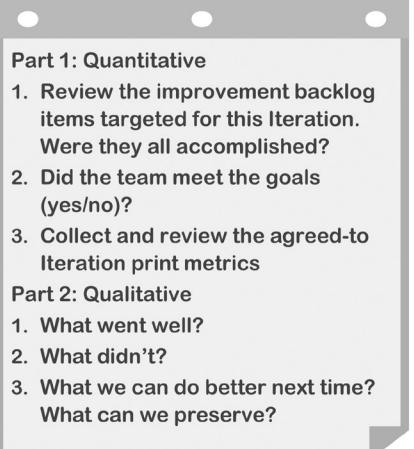
Iteration Retrospective

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

— Agile Manifesto

- ▶ **Timebox:** 30 to 60 minutes
- ▶ **Attendees:** Just the Agile Team
- ▶ **Preparation:** Pick 1 – 2 things that can be done better or preserved, target for next Iteration.
Enter improvement items into the Team Backlog

Sample Agenda

- 
- Part 1: Quantitative**
1. Review the improvement backlog items targeted for this Iteration. Were they all accomplished?
 2. Did the team meet the goals (yes/no)?
 3. Collect and review the agreed-to Iteration print metrics
- Part 2: Qualitative**
1. What went well?
 2. What didn't?
 3. What we can do better next time? What can we preserve?

SCALED AGILE® © Scaled Agile, Inc.

4-38

Notes:

Creative Iteration retrospectives

Simple
Three columns
and open
discussion

Appreciations
Has someone
helped you or
helped the team?

One word
to describe
the Iteration

Individually
write Post-Its
and then find
patterns as a
group

Rate the Iteration
on a scale of
1 – 5 and then
brainstorm how
to make the next
a 5

SCALED AGILE® © Scaled Agile, Inc.

4-39

Notes:

4.5 Facilitate relentless improvement



Presentation: Tune and Adjust

Use a sailboat as a metaphor for the team. Just like the sailboat, there are things that slow our team down, and things that propel it forward.



Tune and Adjust

► **Step 1:** On a flip chart sheet draw a sailboat with an anchor. By the anchor draw the things that slow the team down (impediments).

► **Step 2:** Around the sails draw things that propel the team forward

► **Step 3:** Present your poster to the class and discuss how can your team tune and adjust:

- What's working well (what are the propelling forces)
- What isn't working (what are the anchors)
- What can we do better next time (how can we tune and adjust)



4-40

SCALED AGILE® © Scaled Agile, Inc.

Notes:

The Scrum Master's role in the improvement

Best Approaches

- Encourage improvement between retrospectives
- Coach the team on problem-solving techniques
- Retrospective:
 - Start by reviewing the results of the previous retrospective
 - Make sure each person speaks
 - Make sure the meeting ends with actionable improvement Stories that are added to the backlog
 - Write down what people are saying exactly
 - Take program concerns to the RTE

Common Anti-Patterns

- The only focus is on what to improve and not what to preserve
- Focus on problems that are outside of the team's control
- Failure to achieve results
- Inviting people outside the team (especially management) to the retrospective



SCALED AGILE® © Scaled Agile, Inc.

4-41

Notes:

4.6 Support DevOps and Release on Demand

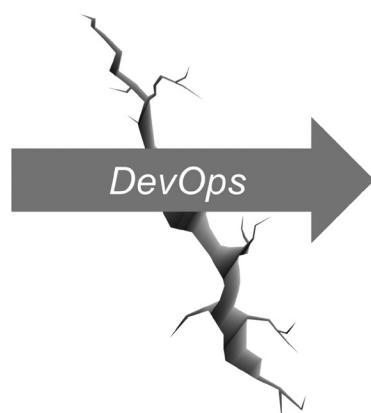
Notes:

What is DevOps?

An Agile approach to bridge the gap between development and operations to deliver value *faster and more reliably*.

Development:

- ▶ Create change
- ▶ Add or modify Features



Operations:

- ▶ Create stability
- ▶ Create or enhance services

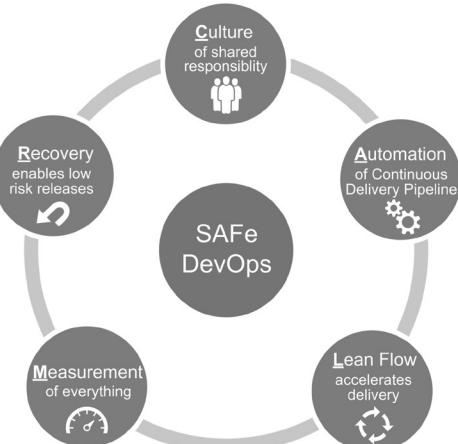
DevOps is a capability of every Agile Release Train

Notes:

DevOps enables Continuous Delivery and Release on Demand

- ▶ **Culture** Establish a culture of shared responsibility for development, deployment, and operations.
- ▶ **Automation** Automate the continuous delivery pipeline.
- ▶ **Lean flow** Keep batch sizes small, limit WIP, and provide extreme visibility.
- ▶ **Measurement** Measure the flow through the pipeline. Implement application telemetry.
- ▶ **Recovery** Architect and enable low risk releases. Establish fast recovery, fast reversion, and fast fix-forward.

A CALMR Approach to DevOps



SCALED AGILE® © Scaled Agile, Inc.

4-44

Notes:



Activity: Supporting DevOps and Release on Demand

- ▶ **Step 1:** Think about the five concepts necessary to Release on Demand: Culture, Automation, Lean Flow, Measurement, and Recovery and what might be some challenges associated with them
- ▶ **Step 2:** On a flip chart sheet or a whiteboard draw the CALMR approach and next to each of the five concepts write:
 - As a Scrum Master how would you support each concept?
 - What are some of the tools and techniques you can use to support the team applying those concepts?
- ▶ **Step 3:** Share with the class



SCALED AGILE® © Scaled Agile, Inc.



4-45

Notes:

Lesson review

In this lesson you:

- ▶ Planed the Iteration
- ▶ Tracked the Iteration progress
- ▶ Refined the Backlog
- ▶ Explored how to facilitate the Iteration Review
- ▶ Discussed how to relentlessly improve
- ▶ Discussed ways for supporting DevOps and Release on Demand



SCALED AGILE® © Scaled Agile, Inc.

4-46

Notes:



Toolbox: Facilitating Iteration Execution

- ▶ **Step 1:** Locate the Scrum Master Toolbox section in your Workbooks
- ▶ **Step 2:** Add more tools or techniques to the Toolbox by reflecting on the following:

- What are some tools or techniques for coaching and supporting the team in executing the Iteration
- What are some tools or techniques for facilitating Iteration Review?

- ▶ **Step 3:** Share one of your insights with the class



PREPARE & SHARE



4-47

SCALED AGILE® © Scaled Agile, Inc.

Notes:

Key Learnings & Insights



Lesson 4

Facilitating Iteration execution

Lesson 5

Finishing the PI

Learning objectives:

- 5.1 Coach the IP Iteration
- 5.2 Prepare the team for the Inspect and Adapt Workshop



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

5.1 Coach the IP Iteration

Notes:

Innovation and Planning Iteration

Provide sufficient capacity margin to enable cadence.

— Don Reinertsen, *Principles of Product Development Flow*

Definitions

Innovation: Opportunity for innovation spikes, hackathons, and infrastructure improvements

Planning: Provides for cadence-based planning and is an estimating guard band for cadence-based delivery

Common Anti-Patterns

- Planning work for the IP Iteration in PI Planning
- Leaving testing or bug fixing to the IP Iteration
- Leaving integration of the whole system to the IP Iteration



Notes:

5.1 Coach the IP Iteration

IP Iteration calendar

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

SCALED AGILE® © Scaled Agile, Inc.

5-5

Notes:

Without the IP Iteration ...

- ▶ Lack of delivery capacity buffer impacts predictability
- ▶ Little innovation, tyranny of the urgent
- ▶ Technical debt grows uncontrollably
- ▶ People burn out
- ▶ No time for teams to plan, demo, or improve together



SCALED AGILE® © Scaled Agile, Inc.

5-6

Notes:

5.2 Prepare the team for the Inspect and Adapt Workshop

SCALED AGILE® © Scaled Agile, Inc.

5-7

Notes:

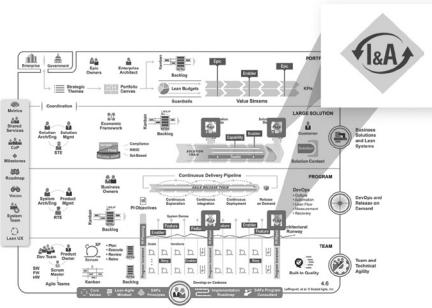
Improving results with Inspect and Adapt event

Three parts of Inspect and Adapt:

1. The PI System Demo
2. Quantitative Measurement
3. Problem-Solving Workshop

Timebox: 3 – 4 hours per PI

Attendees: Teams and stakeholders



SCALED AGILE® © Scaled Agile, Inc.

5-8

Notes:

1.

PI System Demo

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

- ▶ Often led by Product Management, POs, and the System Team
- ▶ Attended by Business Owners, program stakeholders, Product Management, RTE, Scrum Masters, and teams



SCALED AGILE® © Scaled Agile, Inc.

5-9

Notes:

Program performance reporting

As part of the PI System Demo, teams compare planned vs. actual PI Objectives.

- ▶ Teams meet with their Business Owners to self-assess the business value they achieved for each objective
- ▶ Each team's planned vs. actual business value is then rolled up to the Program-level in the Program Predictability Measure

Business Value		
Objectives for PI 3	Plan	Actual
▪ Structured locations and validation of locations	7	7
▪ Build and demonstrate a proof of concept for context images	8	8
▪ Implement negative triangulation by: tags, companies and people	8	6
▪ Speed up indexing by 50%	10	5
▪ Index 1.2 billion more web pages	10	8
▪ Extract and build URL abstracts	7	7
Stretch Objectives		
▪ Fuzzy search by full name	7	0
▪ Improve tag quality to 80% relevance	4	4
Totals	50	45
% Achievement: 90%		

SCALED AGILE® © Scaled Agile, Inc.

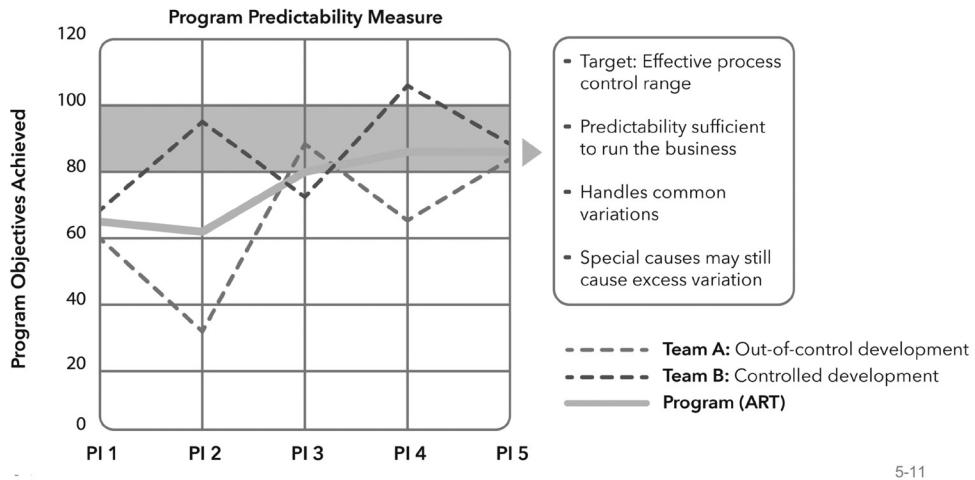
5-10

Notes:

Quantitative measurement

2.

The PI Predictability Measure shows whether achievements fall into an acceptable process control band.



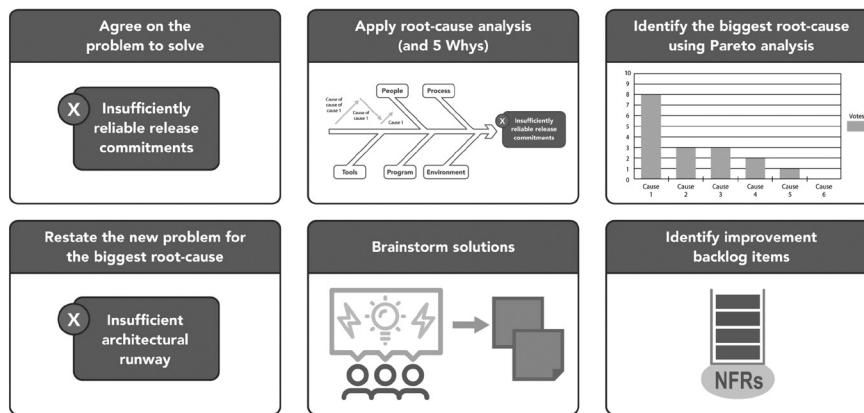
5-11

Notes:

The problem-solving workshop

3.

Teams conduct a short retrospective, then systematically address the larger impediments that are limiting velocity.



5-12

Notes:

The Scrum Master's role in Inspect and Adapt

Best Approaches

- Facilitate the team preparation for the PI System Demo
- Provide data
- Facilitate one of the teams in the Problem-Solving Workshop
- Help the RTE make sure improvement items are included during the PI
- If using ad-hoc teams for the I&A, then Scrum Masters may be participants rather than facilitators

Common Anti-Patterns

- Only the PO presents in the PI System Demo
- No actionable improvement Features are created
- Improvement items don't enter the PI Planning process
- Improvement items are not demoed in the PI System Demo



Notes:

Lesson review

In this lesson you:

- ▶ Explored how to coach the IP Iteration
- ▶ Discussed how to prepare the team for the Inspect and Adapt Workshop



5-14

SCALED AGILE® © Scaled Agile, Inc.

Notes:



Toolbox: Finishing the PI

► **Step 1:** Locate the Scrum Master Toolbox section in your Workbooks

► **Step 2:** Add more tools or techniques to the Toolbox by reflecting on the following:

- What are some tools or techniques for facilitating the IP Iteration?
- What are some tools or techniques for preparing the team for System Demo?

► **Step 3:** Share one of your insights with the class



PREPARE & SHARE



5-15

SCALED AGILE® © Scaled Agile, Inc.

Notes:

Feedback

http://bit.ly/SAFe_SSM



SCALED AGILE® © Scaled Agile, Inc.

5-16

Notes:

Key Learnings & Insights



Lesson 5

Finishing the PI Workshop

Lesson 6

Becoming a SAFe Scrum Master



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

Welcome to Role-based learning and certification

Scaled Agile's role-based learning and certification offerings focus on the skills, knowledge, and experience required to successfully perform the job.

As part of your SAFe learning journey, we encourage you to attend courses, read recommended books and articles, take advantage of videos and enablement, gain some real-world experience in the role, and then take the exams.

*We look forward to you becoming
a member of the Certified SAFe® Community!*



SCALED AGILE® © Scaled Agile, Inc.

6-2

Notes:

Become a Certified SAFe® Scrum Master in 1-2-3

1 Take the course

- Course completion is the first step towards SAFe certification!
- After course completion you will receive access to the SAFe Community Platform which also gives you access to study materials and the exam.



All course information is on SAI.com at:
www.scaledagile.com/certification/courses/safe-scrum-master/

SCALED AGILE® © Scaled Agile, Inc.

6-3

Notes:

Become a Certified SAFe® Scrum Master in 1-2-3

2 Study and learn

- An exam study guide and a practice test is available on the **SAFe Community Platform** in your Learning Plan to help you prepare for the exam and to further your learning.
- Full exam details such as time on exam, number of questions, passing score, and sample test are available on SAI's certification web page.



All course information is on SAI.com at:
www.scaledagile.com/certification-and-exam-information-ssm/

SCALED AGILE® © Scaled Agile, Inc.

6-4

Notes:

Become a Certified SAFe® Scrum Master in 1-2-3

3 Take the exam

- The exam is available in your Learning Plan after you complete the Course Feedback Survey.
- The exam is Web-based (single-browser), closed book, no outside assistance, and timed.
- First exam attempt is included as part of the course registration fee if the exam is taken within 30 days of course completion. Each retake attempt costs \$50.



All course information is on SAI.com at:
www.scaledagile.com/certification-and-exam-information-ssm/

SCALED AGILE® © Scaled Agile, Inc.

6-5

Notes:



Video resource: What's next?

Want to learn more about the next steps on your SAFe Journey?

Watch the short video at
www.vimeo.com/201877314



SCALED AGILE® © Scaled Agile, Inc.

DURATION



6-6

Notes:

Key Learnings & Insights



Lesson 6

Becoming a SAFe Scrum Master

SAFe Glossary

Visit the Scaled Agile Framework site
(www.scaledagileframework.com)
to download glossaries translated
into other languages

Guide to acronyms and abbreviations

ART	Agile Release Train	OpEx	Operating Expenses
BO	Business Owner	PDCA	Plan, Do, Check, Adjust
BV	Business Value	PI	Program Increment
BVIR	Big Visual Information Radiator	PM	Product Management
CapEx	Capital Expenses	PO/PM	Product Owner / Product Manager
CD	Continuous Deployment	PO	Product Owner
CE	Continuous Exploration	ROAM	Resolved, Owned, Accepted, Mitigated
CI	Continuous Integration	RR	Risk Reduction
CFD	Cumulative Flow Diagram	RTE	Release Train Engineer
CoD	Cost of Delay	S4T	SAFe® for Teams
CoP	Community of Practice	SAFe®	Scaled Agile Framework
DoD	Definition of Done	SA	SAFe® Agilist
DSU	Daily Stand-up	SBD	Set-Based Design
EA	Enterprise Architect	SM	Scrum Master
EO	Epic Owner	SMART	Specific, Measurable, Achievable, Realistic, Time-bound
FW	Firmware	SoS	Scrum of Scrums
HW	Hardware	SP	SAFe® Practitioner
I&A	Inspect and Adapt	SPC	SAFe® Program Consultant
IP	Innovation and Planning (iteration)	STE	Solution Train Engineer
KPI	Key Performance Indicator	SW	Software
LPM	Lean Portfolio Management	UX	User Experience
MBSE	Model-Based Systems Engineering	VS	Value Stream
MMF	Minimum Marketable Feature	VSE	Value Stream Engineer
MVP	Minimum Viable Product	WIP	Work in Process
NFR	Non-functional Requirements	WSJF	Weighted Shortest Job First
OE	Opportunity Enablement	XP	Extreme Programming

Agile Architecture

Agile Architecture is a set of values and practices that support the active evolution of the design and architecture of a system while implementing new system capabilities.

Agile Release Train (ART)

The Agile Release Train (ART) is a long-lived team of Agile teams, which, along with other stakeholders, develops and delivers solutions incrementally, using a series of fixed-length Iterations within a Program Increment (PI) timebox. The ART aligns teams to a common business and technology mission.

Agile Team

The SAFe Agile Team is a cross-functional group of 5 to 10 people who have the ability and authority to define, build, and test some element of Solution value—all in a short Iteration timebox. Specifically, the SAFe Agile Team incorporates the Dev Team, Scrum Master, and Product Owner roles.

Architectural Runway

The Architectural Runway consists of the existing code, components, and technical infrastructure needed to implement near-term features without excessive redesign and delay.

Built-In Quality

Built-In Quality practices ensure that each Solution element, at every increment, meets appropriate quality standards throughout development.

Business Owners

Business Owners are a small group of stakeholders who have the primary business and technical responsibility for governance, compliance, and return on investment (ROI) for a Solution developed by an Agile Release Train (ART). They are key stakeholders on the ART who must evaluate fitness for use and actively participate in certain ART events.

CapEx and OpEx

Capital Expenses (CapEx) and Operating Expenses (OpEx) describe Lean-Agile financial accounting practices in a Value Stream budget. In some cases, CapEx may include capitalized labor associated with the development of intangible assets—such as software, intellectual property, and patents.

Capabilities

A Capability is a higher-level solution behavior that typically spans multiple ARTs. Capabilities are sized and split into multiple features to facilitate their implementation in a single PI.

Communities of Practice (CoPs)

Communities of Practice (CoPs) are organized groups of people who have a common

interest in a specific technical or business domain. They collaborate regularly to share information, improve their skills, and actively work on advancing the general knowledge of the domain.

Compliance

Compliance refers to a strategy and a set of activities and artifacts that allow teams to apply Lean-Agile development methods to build systems that have the highest possible quality, while simultaneously assuring they meet any regulatory, industry, or other relevant standards.

Continuous Delivery Pipeline

The Continuous Delivery Pipeline (also referred to as 'pipeline') represents the workflows, activities, and automation needed to provide a continuous release of value to the end user.

Continuous Deployment (CD)

Continuous Deployment (CD) is the process that takes validated Features from Continuous Integration and deploys them into the production environment, where they are tested and readied for release. It is the third element in the four-part Continuous Delivery Pipeline of Continuous Exploration (CE), Continuous Integration (CI), Continuous Deployment, and Release on Demand.

Continuous Exploration (CE)

Continuous Exploration (CE) is the process of continually exploring the market and user needs, and defining a Vision, Roadmap, and set of Features that address those needs. It's the first element in the four-part Continuous Delivery Pipeline, preceding Continuous Integration (CI) Continuous Deployment (CD), and Release on Demand.

Continuous Integration (CI)

Continuous Integration (CI) is the process of taking features from the Program Backlog and developing, testing, integrating, and validating them in a staging environment where they are ready for deployment and release.

Core Values

The four Core Values of alignment, built-in quality, transparency, and program execution represent the fundamental beliefs that are key to SAFe's effectiveness. These guiding principles help dictate behavior and action for everyone who participates in a SAFe portfolio.

Customers

Customers are the ultimate buyer of every Solution. They are an integral part of the Lean-Agile development process and Value Stream and have specific responsibilities in SAFe.

Dev Team

The Dev Team is a subset of the Agile Team. It consists of the dedicated professionals who can develop and test a Story, Feature, or component. The Dev Team typically

includes software developers and testers, engineers, and other dedicated specialists required to complete a vertical slice of functionality.

DevOps

DevOps is a mindset, a culture, and a set of technical practices. It provides communication, integration, automation, and close cooperation among all the people needed to plan, develop, test, deploy, release, and maintain a Solution.

Develop on Cadence

Develop on Cadence is an essential method for managing the inherent variability of systems development in a flow-based system, by making sure important events and activities occur on a regular, predictable schedule.

Economic Framework

The Economic Framework is a set of decision rules that align everyone to the financial objectives of the Solution and guides the economic decision-making process. It contains four primary constructs: Lean Budgets, Epic funding and governance, decentralized decision-making, and job sequencing based on the Cost of Delay (CoD).

Enablers

Enablers support the activities needed to extend the Architectural Runway to provide future business functionality. These include exploration, infrastructure, compliance, and architecture development. They are captured in the various backlogs and occur at all levels of the Framework.

Enterprise

The Enterprise represents the business entity to which each SAFe portfolio belongs.

Enterprise Architect

The Enterprise Architect promotes adaptive design, and engineering practices and drives architectural initiatives for the portfolio. Enterprise Architects also facilitate the reuse of ideas, components, services, and proven patterns across various solutions in a portfolio.

Epic

An Epic is a container for a Solution development initiative large enough to require analysis, the definition of a Minimum Viable Product (MVP), and financial approval prior to implementation. Implementation occurs over multiple Program Increments (PIs) and follows the Lean startup ‘build-measure-learn’ cycle.

Epic Owners

Epic Owners are responsible for coordinating portfolio Epics through the Portfolio Kanban system. They define the epic, its Minimum Viable Product (MVP), and Lean business case, and when approved, facilitate implementation.

Essential SAFe configuration

The Essential SAFe configuration is the heart of the Framework and is the simplest starting point for implementation. It's the basic building block for all other SAFe configurations and describes the most critical elements needed to realize the majority of the Framework's benefits.

Features

A Feature is a service that fulfills a stakeholder need. Each feature includes a benefit hypothesis and acceptance criteria, and is sized or split as necessary to be delivered by a single Agile Release Train (ART) in a Program Increment (PI).

Foundation

The Foundation contains the supporting principles, values, mindset, implementation guidance, and leadership roles needed to deliver value successfully at scale.

Full SAFe configuration

The Full SAFe configuration is the most comprehensive version of the Framework. It supports enterprises that build and maintain large integrated solutions, which require hundreds of people or more, and includes all levels of SAFe: team, program, large solution, and portfolio. In the largest enterprises, multiple instances of various SAFe configurations may be required.

Innovation and Planning Iteration

The Innovation and Planning (IP) Iteration occurs every Program Increment (PI) and serves multiple purposes. It acts as an estimating buffer for meeting PI Objectives and provides dedicated time for innovation, continuing education, PI Planning, and Inspect and Adapt (I&A) events.

Inspect and Adapt (I&A)

The Inspect and Adapt (I&A) is a significant event, held at the end of each Program Increment (PI), where the current state of the Solution is demonstrated and evaluated by the train. Teams then reflect and identify improvement backlog items via a structured, problem-solving workshop.

Iteration

Iterations are the basic building block of Agile development. Each iteration is a standard, fixed-length timebox, where Agile Teams deliver incremental value in the form of working, tested software and systems. The recommended duration of the timebox is two weeks. However, one to four weeks is acceptable, depending on the business context.

Iteration Execution

Iteration Execution is how Agile Teams manage their work throughout the Iteration timebox, resulting in a high-quality, working, tested system increment.

Iteration Goals

Iteration Goals are a high-level summary of the business and technical goals that the Agile Team agrees to accomplish in an Iteration. They are vital to coordinating an Agile Release Train (ART) as a self-organizing, self-managing team of teams.

Iteration Planning

Iteration Planning is an event where all team members determine how much of the Team Backlog they can commit to delivering during an upcoming Iteration. The team summarizes the work as a set of committed Iteration Goals.

Iteration Retrospective

The Iteration Retrospective is a regular meeting where Agile Team members discuss the results of the Iteration, review their practices, and identify ways to improve.

Iteration Review

The Iteration Review is a cadence-based event, where each team inspects the increment at the end of every Iteration to assess progress, and then adjusts its backlog for the next iteration.

Large Solution Level

The Large Solution Level contains the roles, artifacts, and processes needed to build large and complex solutions. This includes a stronger focus on capturing requirements in Solution Intent, the coordination of multiple Agile Release Trains (ARTs) and Suppliers, and the need to ensure compliance with regulations and standards.

Large Solution SAFe configuration

The Large Solution SAFe configuration is for developing the largest and most complex solutions that typically require multiple Agile release trains and Suppliers, but do not require portfolio-level considerations. This is common for industries like aerospace and defense, automotive, and government, where the large solution—not portfolio governance—is the primary concern.

Lean Budgets

Lean Budgets is a set of practices that minimize overhead by funding and empowering Value Streams rather than projects while maintaining financial and fitness-for-use governance. This is achieved through objective evaluation of working systems, active management of Epic investments, and dynamic budget adjustments.

Lean Portfolio Management (LPM)

The Lean Portfolio Management (LPM) function has the highest level of decision-making and financial accountability for the products and Solutions in a SAFe portfolio.

Lean User Experience (Lean UX)

Lean User Experience (Lean UX) design is a mindset, culture, and a process that embraces Lean-Agile methods. It implements functionality in minimum viable increments and determines success by measuring results against a benefit hypothesis.

Lean and Agile Principles

SAFe is based on nine immutable, underlying Lean and Agile Principles. These tenets and economic concepts inspire and inform the roles and practices of SAFe.

Lean-Agile Leaders

Lean-Agile Leaders are lifelong learners who are responsible for the successful adoption of SAFe and the results it delivers. They empower and help teams build better systems by learning, exhibiting, teaching and coaching SAFe's Lean-Agile principles and practices.

Lean-Agile Mindset

The Lean-Agile Mindset is the combination of beliefs, assumptions, and actions of SAFe leaders and practitioners who embrace the concepts of the Agile Manifesto and Lean thinking. It's the personal, intellectual, and leadership foundation for adopting and applying SAFe principles and practices.

Metrics

Metrics are agreed-upon measures used to evaluate how well the organization is progressing toward the portfolio, large solution, program, and team's business and technical objectives.

Milestones

Milestones are used to track progress toward a specific goal or event. There are three types of SAFe milestones: Program Increment (PI), fixed-date, and learning milestones.

Model-Based Systems Engineering (MBSE)

Model-Based Systems Engineering (MBSE) is the practice of developing a set of related system models that help define, design, and document a system under development. These models provide an efficient way to explore, update, and communicate system aspects to stakeholders, while significantly reducing or eliminating dependence on traditional documents.

Nonfunctional Requirements (NFRs)

Nonfunctional Requirements (NFRs) define system attributes such as security, reliability, performance, maintainability, scalability, and usability. They serve as constraints or restrictions on the design of the system across the different backlogs.

Portfolio Backlog

The Portfolio Backlog is the highest-level backlog in SAFe. It provides a holding area for upcoming business and enabler Epics intended to create a comprehensive set of Solutions, which provides the competitive differentiation and operational improvements needed to address the Strategic Themes and facilitate business success.

Portfolio Kanban

The Portfolio Kanban is a method used to visualize, manage, and analyze the prioritization and flow of portfolio Epics from ideation to implementation and completion.

Portfolio Level

The Portfolio Level contains the principles, practices, and roles needed to initiate and

govern a set of development Value Streams. This is where strategy and investment funding are defined for value streams and their Solutions. This level also provides Agile portfolio operations and Lean governance for the people and resources needed to deliver solutions.

Portfolio SAFe configuration

The Portfolio SAFe configuration helps align portfolio execution to the enterprise strategy, by organizing Agile development around the flow of value, through one or more value streams. It provides business agility through principles and practices for portfolio strategy and investment funding, Agile portfolio operations, and Lean governance.

Pre-and Post-PI Planning

Pre- and Post-Program Increment (PI) Planning events are used to prepare for, and follow up after, PI Planning for Agile Release Trains (ARTs) and Suppliers in a Solution Train.

Product Management

Product Management has content authority for the Program Backlog. They are responsible for identifying Customer needs, prioritizing Features, guiding the work through the Program Kanban and developing the program Vision and Roadmap.

Product Owner (PO)

The Product Owner (PO) is a member of the Agile Team responsible for defining Stories and prioritizing the Team Backlog to streamline the execution of program priorities while maintaining the conceptual and technical integrity of the Features or components for the team.

Program Backlog

The Program Backlog is the holding area for upcoming Features, which are intended to address user needs and deliver business benefits for a single Agile Release Train (ART). It also contains the enabler features necessary to build the Architectural Runway.

Program Increment (PI)

A Program Increment (PI) is a timebox during which an Agile Release Train (ART) delivers incremental value in the form of working, tested software and systems. PIs are typically 8 – 12 weeks long. The most common pattern for a PI is four development Iterations, followed by one Innovation and Planning (IP) Iteration.

Program Increment (PI) Planning

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

Program Kanban

The Program and Solution Kanban systems are a method to visualize and manage the flow of Features and Capabilities from ideation to analysis, implementation, and release through the Continuous Delivery Pipeline.

Program Level

The Program Level contains the roles and activities needed to continuously deliver solutions via an Agile Release Train (ART).

Refactoring

Refactoring is the activity of improving the internal structure or operation of a code or component without changing its external behavior.

Release Train Engineer (RTE)

The Release Train Engineer (RTE) is a servant leader and coach for the Agile Release Train (ART). The RTE's major responsibilities are to facilitate the ART events and processes and assist the teams in delivering value. RTEs communicate with stakeholders, escalate impediments, help manage risk, and drive relentless improvement.

Release on Demand

Release on Demand is the process by which Features deployed into production are released incrementally or immediately to Customers based on market demand.

Roadmap

The Roadmap is a schedule of events and Milestones that communicate planned Solution deliverables over a timeline. It includes commitments for the planned, upcoming Program Increment (PI) and offers visibility into the deliverables forecasted for the next few PIs.

SAFe Implementation Roadmap

The SAFe Implementation Roadmap consists of an overview graphic and a 12-article series that describes a strategy and an ordered set of activities that have proven to be effective in successfully implementing SAFe.

SAFe Program Consultants (SPCs)

SAFe Program Consultants (SPCs) are change agents who combine their technical knowledge of SAFe with an intrinsic motivation to improve the company's software and systems development processes. They play a critical role in successfully implementing SAFe. SPCs come from numerous internal or external roles, including business and technology leaders, portfolio/program/project managers, process leads, architects, analysts, and consultants.

Scrum Master

Scrum Masters are servant leaders and coaches for an Agile Team. They help educate the team in Scrum, Extreme Programming (XP), Kanban, and SAFe, ensuring that the agreed Agile process is being followed. They also help remove impediments and foster an environment for high-performing team dynamics, continuous flow, and relentless

improvement.

ScrumXP

ScrumXP is a lightweight process to deliver value for cross-functional, self-organized teams within SAFe. It combines the power of Scrum project management practices with Extreme Programming (XP) practices.

Set-Based Design

Set-Based Design (SBD) is a practice that keeps requirements and design options flexible for as long as possible during the development process. Instead of choosing a single point solution upfront, SBD identifies and simultaneously explores multiple options, eliminating poorer choices over time. It enhances flexibility in the design process by committing to technical solutions only after validating assumptions, which produces better economic results.

Shared Services

Shared Services represents the specialty roles, people, and services that are necessary for the success of an Agile Release Train (ART) or Solution Train but that cannot be dedicated full-time.

Solution

Each Value Stream produces one or more Solutions, which are products, services, or systems delivered to the Customer, whether internal or external to the Enterprise.

Solution Architect/Engineer

The Solution Architect/Engineering role represents an individual or small team that defines a shared technical and architectural vision for the Solution under development. They participate in determining the system, subsystems, and interfaces, validate technology assumptions and evaluate alternatives, working closely with the Agile Release Train (ARTs) and Solution Train.

Solution Backlog

The Solution Backlog is the holding area for upcoming Capabilities and enablers, each of which can span multiple ARTs and is intended to advance the Solution and build its architectural runway.

Solution Context

Solution Context identifies critical aspects of the operational environment for a Solution. It provides an essential understanding of requirements, usage, installation, operation, and support of the solution itself. Solution context heavily influences opportunities and constraints for releasing on demand.

Solution Demo

The Solution Demo is where the results of development efforts from the Solution Train

are integrated, evaluated, and made visible to Customers and other stakeholders.

Solution Management

Solution Management has content authority for the Solution Backlog. They work with customers to understand their needs, prioritize Capabilities, create the Solution vision and roadmap, define requirements, and guide work through the Solution Kanban.

Solution Train

The Solution Train is the organizational construct used to build large and complex Solutions that require the coordination of multiple Agile Release Trains (ARTs), as well as the contributions of Suppliers. It aligns ARTs with a shared business and technology mission using the solution Vision, Backlog, and Roadmap, and an aligned Program Increment (PI).

Spanning Palette

The Spanning Palette contains various roles and artifacts that may be applicable to a specific team, program, large solution, or portfolio context. A key element of SAFe's flexibility and configurability, the spanning palette permits organizations to apply only the elements needed for their configuration.

Spikes

Spikes are a type of exploration Enabler Story in SAFe. Defined initially in Extreme Programming (XP), they represent activities such as research, design, investigation, exploration, and prototyping. Their purpose is to gain the knowledge necessary to reduce the risk of a technical approach, better understand a requirement, or increase the reliability of a story estimate.

Stories

Stories are short descriptions of a small piece of desired functionality, written in the user's language. Agile Teams implement small, vertical slices of system functionality and are sized so they can be completed in a single Iteration.

Supplier

A Supplier is an internal or external organization that develops and delivers components, subsystems, or services that help Solution Trains provide Solutions to their Customers.

System Demo

The System Demo is a significant event that provides an integrated view of new Features for the most recent Iteration delivered by all the teams in the Agile Release Train (ART). Each demo gives ART stakeholders an objective measure of progress during a Program Increment (PI).

System Team

The System Team is a specialized Agile Team that assists in building and using the Agile development environment, including Continuous Integration, test automation, and Continuous Deployment. The System Team supports the integration of assets from Agile teams, performs end-to-end Solution testing where necessary, and assists with

deployment and release.

Team Backlog

The Team Backlog contains user and enabler Stories that originate from the Program Backlog, as well as stories that arise locally from the team's local context. It may include other work items as well, representing all the things a team needs to do to advance their portion of the system.

Team Kanban

Team Kanban is a method that helps teams facilitate the flow of value by visualizing workflow, establishing Work In Process (WIP) limits, measuring throughput, and continuously improving their process.

Team Level

The Team Level contains the roles, activities, events, and processes which Agile Teams build and deliver value in the context of the Agile Release Train (ART).

Test-First

Test-First is a Built-In Quality practice derived from Extreme Programming (XP) that recommends building tests before writing code to improve delivery by focusing on the intended results.

Value Stream Coordination

Value Stream Coordination provides guidance to manage dependencies and exploit the opportunities in a portfolio.

Value Streams

Value Streams represent the series of steps that an organization uses to build Solutions that provide a continuous flow of value to a Customer. SAFe value streams are used to define and realize Portfolio-level business objectives and organize Agile Release Trains (ARTs) to deliver value more rapidly.

Vision

The Vision is a description of the future state of the Solution under development. It reflects Customer and stakeholder needs, as well as the Feature and Capabilities, proposed to meet those needs.

Weighted Shortest Job First (WSJF)

Weighted Shortest Job First (WSJF) is a prioritization model used to sequence jobs (ex., Features, Capabilities, and Epics) to produce maximum economic benefit. In SAFe, WSJF is estimated as the Cost of Delay (CoD) divided by job size.

