Lesson 4

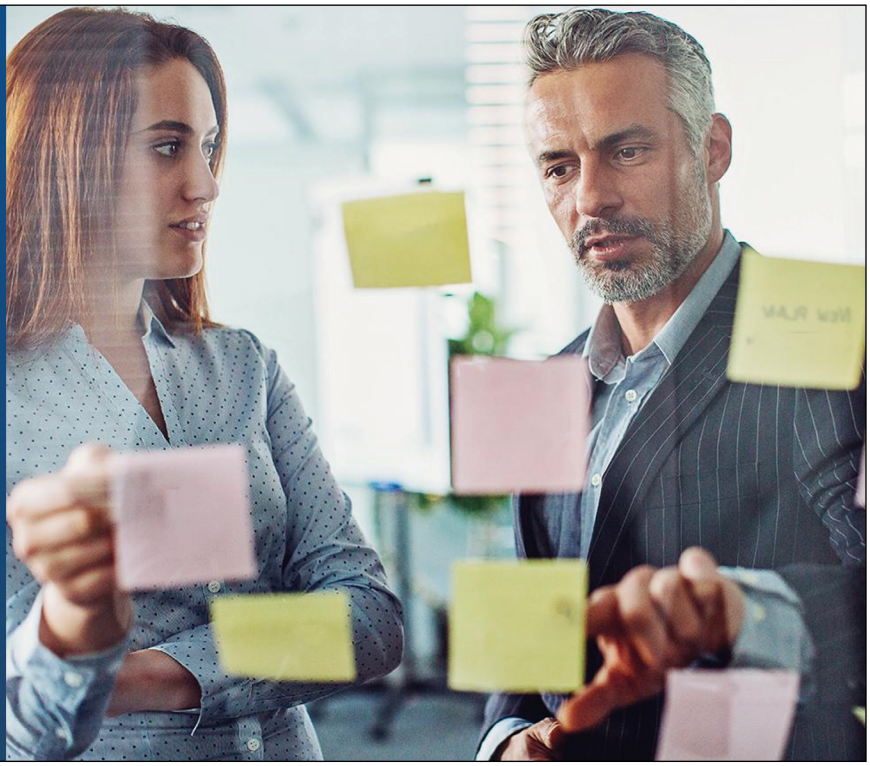
Executing Iterations

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

POPM SAFe 6

SCALED AGILE®

Lesson Topics

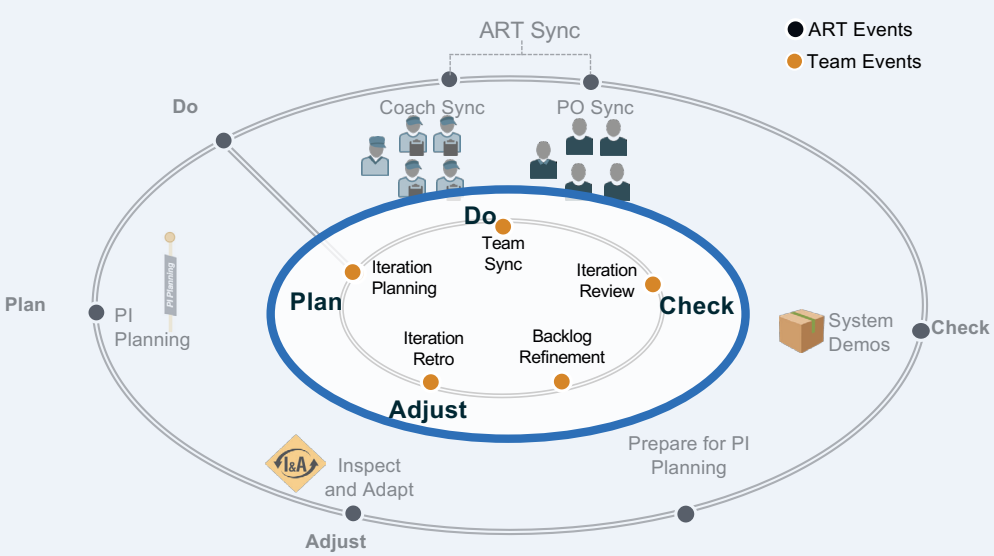


Learning objectives

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

- Create Stories
- Demonstrate how to plan an Iteration
- Outline how to manage flow with the Team Kanban
- Summarize how to continuously refine the Team Backlog
- Plan how to participate in the Iteration Review and Iteration Retrospective
- Explain how to support DevOps and Release on Demand

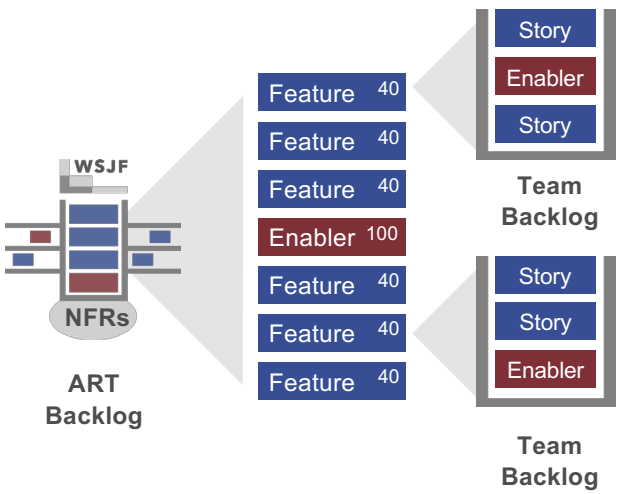
Executing Iterations



4.1 Stories and Story maps

Features are split into Stories

- Features are implemented through one or more Stories
- Features that represent a workflow are visualized with Story maps



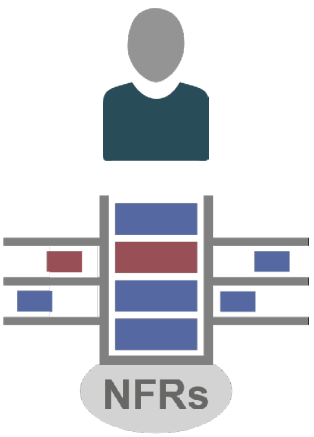
What are Stories?

Stories are short descriptions of a small piece of desired functionality sized so that they can be completed in a single Iteration.

- User Stories express desired end-user functionality written in the user's language
- Enabler Stories support exploration, architecture, infrastructure, and compliance
- Stories are created during PI Planning as the teams collaborate with POs and Product Management

The Team Backlog is composed of Stories

- Contains all the work for the team
- Created by the PO and the team
- Prioritized by the PO
- Contains User and Enabler Stories
  - User Stories provide Customers with value
  - Enabler Stories build the infrastructure and architecture that makes User Stories possible
- Stories for near-term Iterations are more detailed than Stories for later Iterations
- NFRs are backlog constraints



Write User Stories using a standard format

'User voice' format fosters customer-centric design:

As a (user role), I want (activity) so that (business value).

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

As a Fleet Manager, **I want** a notification before a van needs service **so that** I can balance service requests.

As a Fleet Manager, **I want** detailed service histories **so that** I can identify and track safety recalls and confirm repairs.

SCALED AGILE © Scaled Agile. Inc.

4-9

Writing good Stories: The Three Cs

Card

Written on a card or in a digital tool and can be annotated with notes

As a spouse, I want a clean garage so that I can park my car and not trip on my way to the door.

Conversation

The details are in a conversation with the PO

What about the bikes?

Oh yeah, we should hang the bikes.

Confirmation

Acceptance criteria confirm the Story correctness

- **Tools have been put away**
- **Items on the floor have been returned to the proper shelf**
- **Bikes have been hung**

Reference: Jefferies, "Essential XP: Card, Conversation, Confirmation"

SCALED AGILE © Scaled Agile. Inc.

4-10

INVEST in a good Story

- Write Stories that can be developed separately
- Write Stories in which scope can be negotiated
- Write Stories that are valuable to the Customer
- Write Stories that can be estimated
- Write Stories that can fit into an Iteration
- Write Stories that are testable

I	Independent
N	Negotiable
V	Valuable
E	Estimable
S	Small
T	Testable

Reference: Wake, "INVEST in Good Stories, and SMART Tasks"

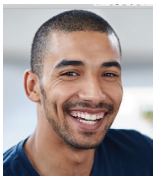
Stories strive to convey an amount of detail that's just right

Insufficient detail	Just right	Overly constrained
As a Fleet Manager, **I want** to search for vans **so that** I can find the van I want.	**As a** Fleet Manager, **I want** to search my fleet **so that** I can find the vans that need a safety recall.	**As a** Fleet Manager, **I want** to search for a van by its Vehicle Identification Number, locator, or driver **so that** I can find the van I want.

Relating Features and Stories to personas improves design

Feature: Safety recall management
Fleet Managers seek to maintain the safety of their vehicles by ensuring that all safety updates are applied to their vans.

Benefits:
- Increased driver safety
- Reduced liability
- Increased compliance



Mike the Fleet Manager
Age: 36
Location: Reno, Nevada, US
Manages: 50 vans, 80 part-time and full-time drivers

Story:
As a Fleet Manager,
I want to search my fleet
so that I can find the vans that need maintenance. Vans that are overdue or need a safety recall are highlighted.

Story:
As a Fleet Manager,
I want to review safety recalls
so that I can prioritize the maintenance schedules of my fleet.

Ten patterns for splitting Features into Stories

Business Feature

Feature: Navigate vehicle for delivery request
Hypothesis: Vehicle can drive delivery route autonomously
Acceptance criteria:
- Pick-up and drop-off are fixed
- Assume well-marked streets with US-only traffic indicators

User Stories

As a vehicle owner, **I want** the vehicle to determine the speed limit and set the speed to that limit **so that** traffic laws are obeyed during delivery.

Enabler Stories

Characterize the camera's ability to read international speed signs.

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

Reference: Lawrence and Green, "The Humanizing Work Guide to Splitting User Stories"

Activity: Split Features into Stories

Duration 10 min

- **Step 1:** With your group, select one of the Features you previously defined
- **Step 2:** Split the Feature into Stories using the User Story format:

  As a (user role), I want (activity) so that (business value)

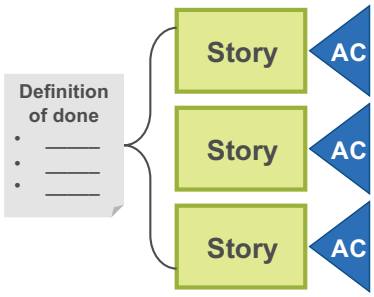
SCALED AGILE © Scaled Agile. Inc. 4-15

When is a Story complete?

A Story is complete when it satisfies the definition of done (DoD). The definition of done requires that the Story:

- Satisfies the acceptance criteria
- Is accepted by the Product Owner

The definition of done applies to all Stories in the Team Backlog.



Acceptance criteria are unique to each Story.

SCALED AGILE © Scaled Agile. Inc. 4-16

Stories have acceptance criteria

Acceptance criteria:

- Provide the details of the Story from a testing point of view
- Are created by the PO and the team as Stories are refined

As a Fleet Manager,
I want a notification before a van needs service
so that I can balance service requests.

Refined

Story:
As a Fleet Manager,
I want a notification before a van needs service
so that I can balance service requests.

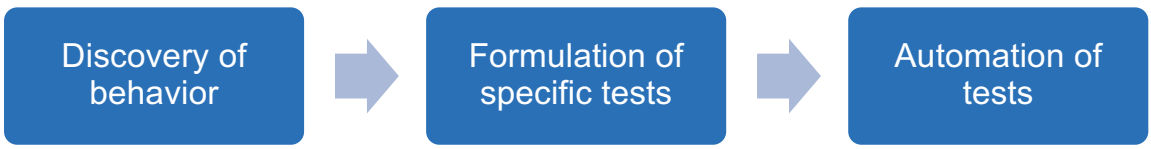
Acceptance criteria:

- The notification contains the van ID defined by the Fleet Manager
- The notification is delivered on a schedule determined by the Fleet Manager

4-17

Write acceptance criteria using behavior-driven development (BDD)

- Behavior is often first described in general terms, which can be ambiguous
- Specific examples of behavior provide a better understanding
- Specific examples can directly become tests or can lead to specific behaviors, which then are transformed into tests



4-18

Acceptance criteria are testable with ‘Given-When-Then’ syntax

- **Example 1: “**System delivers scheduled maintenance notifications.”
  In ‘Given-When-Then’ syntax:

> **Given** a van associated with a maintenance schedule
> **When** the van is due for a maintenance activity
> **Then** a notification is sent to the designated user

- **Example 2: “**System delivers scheduled oil change notifications.”
  In ‘Given-When-Then’ syntax:

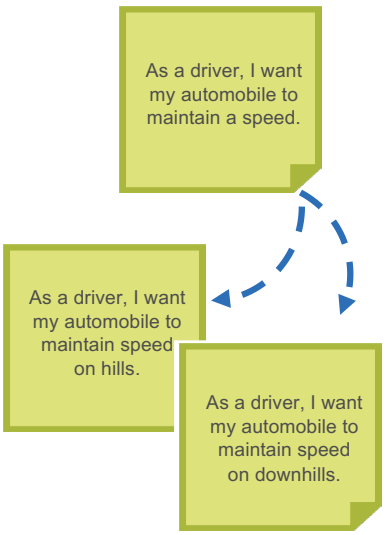
> **Given** a van and an oil maintenance schedule
> **When** the van is due for an oil change in the next month
> **Then** a text message is sent to the Fleet Manager

Activity: Write acceptance criteria

- **Step 1:** With your group, choose three Stories created in the previous activity
- **Step 2:** Write acceptance criteria in the ‘Given-When-Then’ format for each of the Stories
- **Step 3:** Make sure the acceptance criteria are testable
- **Step 4:** Be prepared to share with the class

Split Stories that are too big to fit into an Iteration

Stories are split using the same techniques as splitting Features



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

Reference: Lawrence and Green, "The Humanizing Work Guide to Splitting User Stories"

4-21

Activity: Split Stories

Prepare 10 min | Share 10 min

- **Step 1:** With your group, choose a Story from the ones you created that might not fit into one Iteration
- **Step 2:** Split the Story using the techniques from this lesson
- **Step 3:** Be prepared to share with the class

Hint: How can you ensure that split Stories provide end-user value?

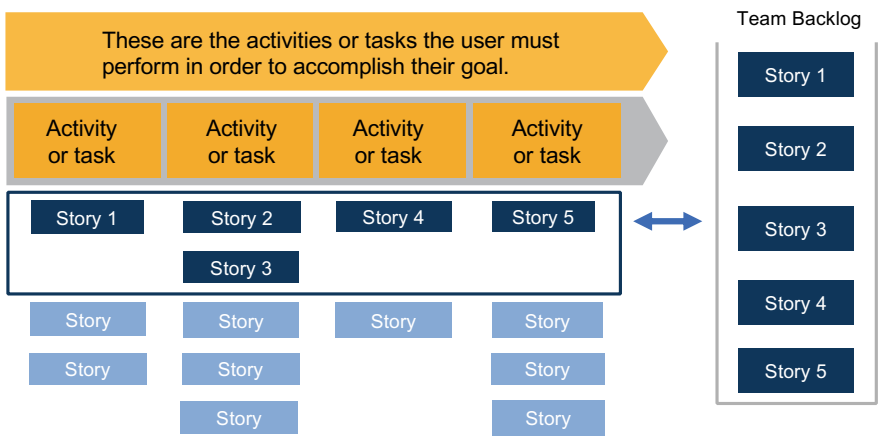
4-22

Use Story maps to capture workflows

- A Story map is a Design Thinking tool that captures user workflow and the Stories that support the workflow
- Story maps help teams:
  - Design workflows
  - Manage the improvement of the product over time by showing how successive Stories can improve the Solution
  - Validate that the Stories in the backlog support all the steps a user needs to accomplish the objective

Integrating user experience and interface design

- Story maps support user experience and interface design in creating design prototypes
- Design prototypes provide fast feedback and help further refine Features and Stories



Activity: Develop a Story map

Prepare 10 min

Share 10 min

- **Step 1**: With your group, create a Story map for the Feature. Think about:
  - What are the main tasks?
  - What is the minimum number of Stories required to complete the Feature?
- **Step 2:** Be prepared to share with the class

Delivery order insurance

Benefit hypothesis:

Allows user to purchase extra insurance for an existing delivery order

Acceptance criteria:

- Access insurance catalog from account profile
- Display available insurance options
- Integrate with shopping cart
- Process payment

4-25

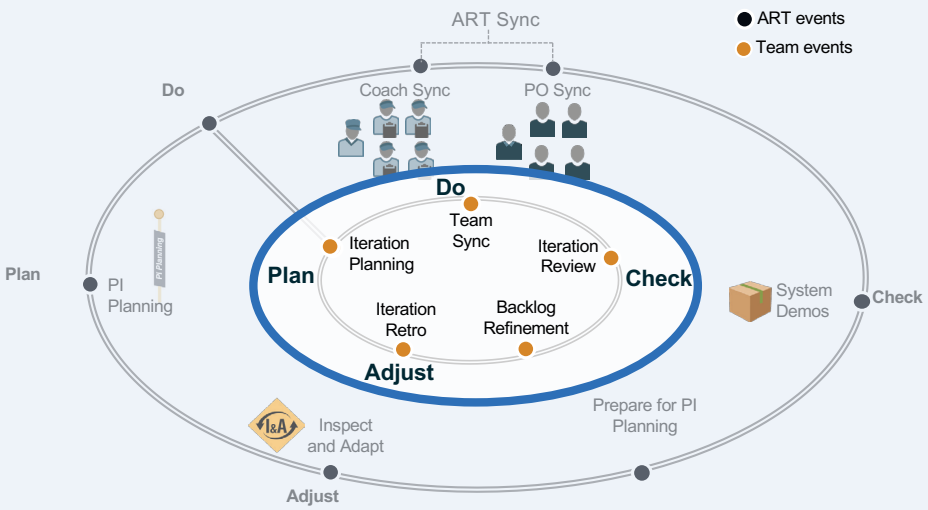
4.2 Iteration Planning

4-26

Iterations are the basic building blocks of Agile development

Product Owner

Product Owners participate in all team events.



Iterations are time-boxed events

Here is an example of events in an Iteration that starts on a Wednesday.

Monday	Tuesday	Wednesday	Thursday	Friday
		Iteration Planning	Team Sync	Team Sync
Team Sync	Team Sync and Backlog Refinement	Team Sync	Team Sync	Team Sync
Team Sync and Iteration Review	Iteration Retrospective			

Functionality is demonstrated throughout and can be released at any time as market needs warrant.

Video: The Product Owner and Iteration Planning

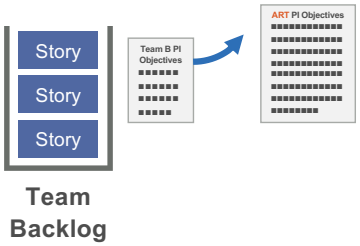
Duration 5 min

https://bit.ly/Video-POIterationPlanning

Product Owners lead Iteration Planning

Iteration Planning refines the Iteration plans created during PI Planning.

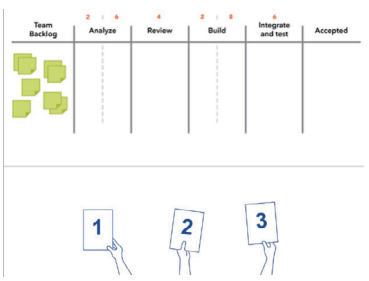
Iteration Planning Preparation → **Iteration Planning** → **Iteration Planning Outputs**



Team Backlog

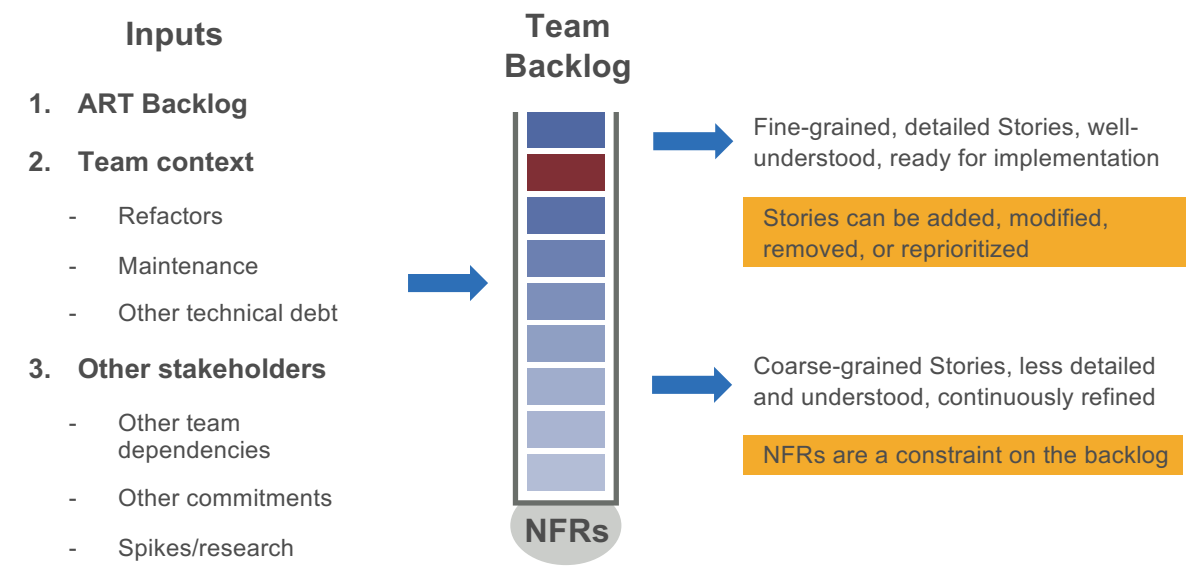
Prioritized backlog and team PI Objectives

Confirm objectives, estimate and detail Stories, develop goals



Committed goals, committed backlog

The Product Owner ensures that the Team Backlog captures all the work



4-31

Sequencing Stories

- Primary economic prioritization happens in the ART Backlog, where Agile Teams sequence work for efficient execution of business priorities
- The Product Owner and the team sequence work based on:
  - Story priorities inherited from ART Backlog priorities
  - Events, Milestones, releases, and other commitments made during PI Planning
  - Dependencies with other teams
  - Local priorities
  - Capacity allocations for defects, maintenance, and refactors
- Initial sequencing happens during PI Planning
- Adjustments happen at Iteration boundaries

4-32

Iteration Planning agenda

Iteration Planning

- Timebox is four hours or less
- This meeting is by and for the team
- SMEs may attend as required

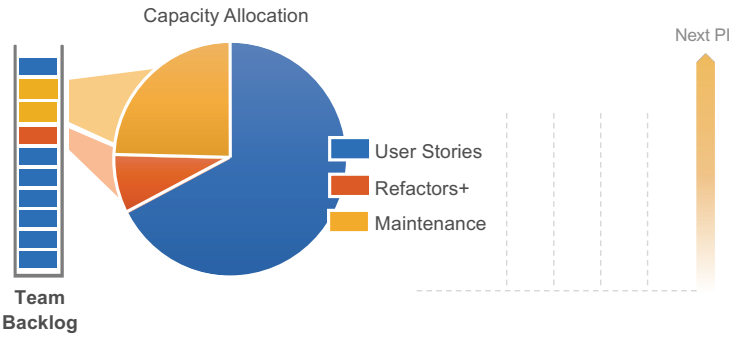
1. Establishing capacity
2. Story analysis and estimating
3. Detailing Stories
4. Developing Iteration Goals
5. Committing to Iteration Goals

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 Product Owner, in collaboration with the team, selects the highest priority backlog items for each slice of the capacity allocation to implement in an Iteration

Capacity allocation for a healthy balance

It's important to balance internal work, like maintenance and refactors, with new User Stories. Capacity allocation allows the PO and team to establish an amount of capacity for each type of work and pull from a prioritized list of each work item. Then they can prioritize those work items against each other.

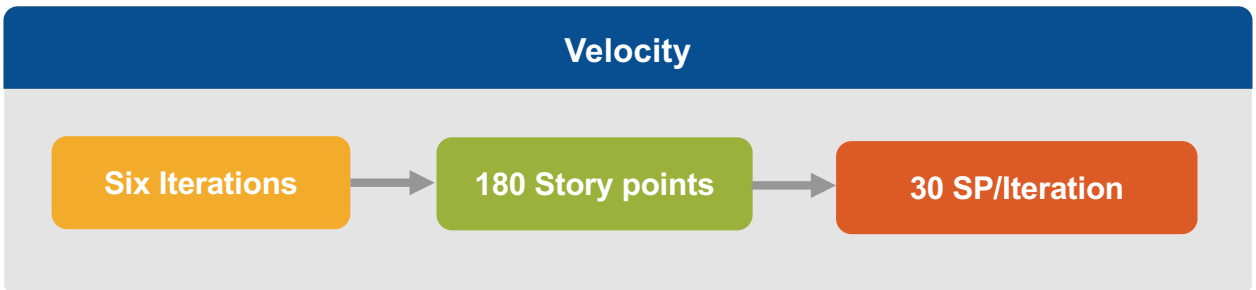


Capacity allocation

- Helps alleviate velocity degradation due to technical debt
- Keeps existing Customers happy with bug fixes and enhancements
- Can change at Iteration or PI boundaries

Using velocity to establish capacity

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



Establishing capacity before historical data exists

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

Example: A seven-person team composed of three developers, two testers, one PO, and one SM/TC, with no vacation, and a two-week Iteration

Note: Exclude PO and SM/TC from the calculation

Estimated capacity =
5 x 8 points = 40 points/Iteration

Estimate Stories with relative Story points

- A Story point is a number that represents:
  - **Volume** - How much there is
  - **Complexity** - How difficult it is
  - **Knowledge** - What we know
  - **Uncertainty** - What's not known
- Story points are relative, and are not connected to any specific unit of measure

?	1	2	3	5	8	13	20	40	100	∞

What size is it?

Guidance: An eight-point Story should take relatively four times longer than a two-point Story.

Apply estimating poker for fast relative estimating

- Estimating poker combines expert opinion, analogy, and disaggregation for quick but reliable estimates
- All members participate
- Increases accuracy by including all perspectives
- Builds understanding and creates shared agreement

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

Reference: *Agile Estimating and Planning* by Mike Cohn

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

4-39

Iteration Goals

Iteration Goals provide clarity, commitment, and management information. They serve three purposes:

Align team members to a common purpose

Align teams to common PI Objectives and manage dependencies

Provide transparency and management information

4-40

Iteration Goals: Examples

Software example

Iteration Goals

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

Business example

Iteration Goals

1. Roll out the incident report procedures
2. Have documentation in one folder for external audit
3. Obtain commitment to audit days from auditors and internal leaders

Commit to the Iteration Goals

Team commitments are not just to the work.
Teams are committed to other teams, the ART, and the stakeholders.

A team meets its commitments:

By doing everything they said they would do.
- or -
By immediately raising the concern if it isn't feasible to do so.

Commitment

Too rigid of a commitment can lead to burnout, inflexibility, and quality problems.

Adaptability

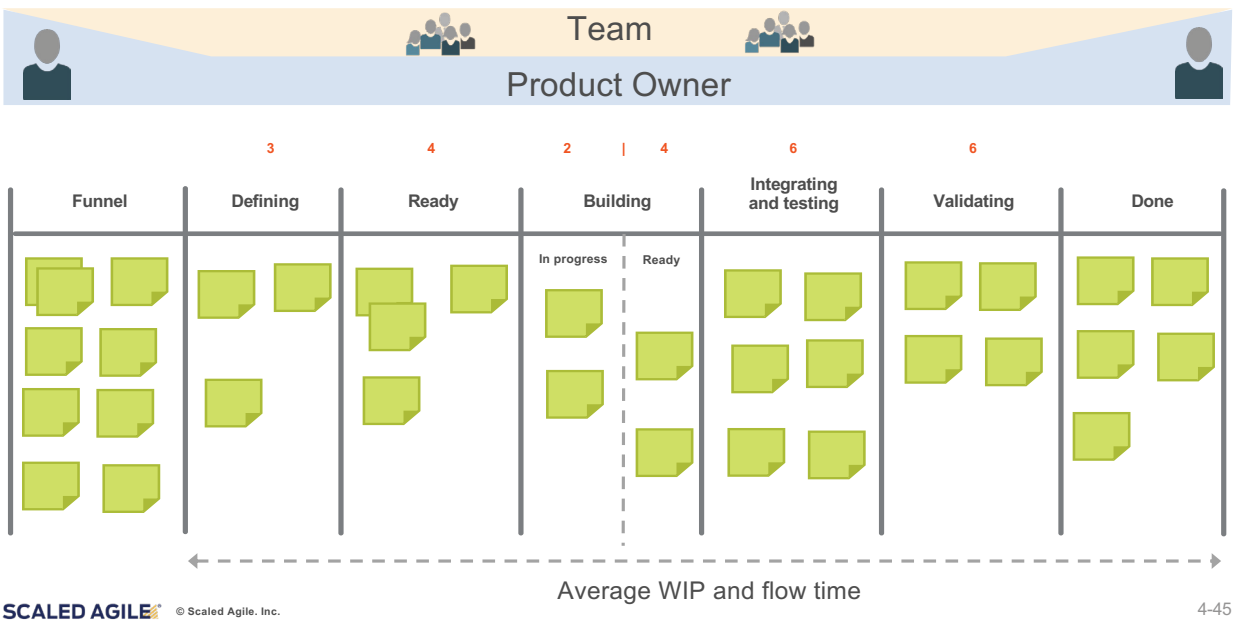
Too little commitment can lead to unpredictability and lack of focus on results.

Tips for effective Iteration Planning

Best approaches	Common anti-patterns
Review and reprioritize the Team Backlog ahead of Iteration Planning	Delving too deep into technical discussions
Clearly communicate Story details and priorities to ensure understanding and alignment	Prioritized Stories don't align to the business objectives or the team's PI Objectives
Maintain neutrality so as not to influence the team to over-commit	Bringing Stories that haven't been refined and prioritized
Prior to Iteration Planning, prepare some preliminary Iteration Goals based on the team's progress in the PI, so far	PO directs the team on how the work should be done
Apply capacity allocation to the Team Backlog to make prioritizing unlike work easier	The team under-commits due to fear of failure
	No time is reserved for support activities

4.3 The Team Kanban

The Team Kanban promotes collaboration and facilitates flow



Video: The Product Owner and the Team Sync

Duration 5 min

https://bit.ly/Video-POTeamSync

POs and the Team Sync

POs:

- Should attend the Team Sync as members of the Agile Team
- Should listen carefully to any impediments that they can resolve immediately during the meet-after
- Should be ready to clarify Story intent and acceptance criteria
- Can interfere with the Team Sync unintentionally, so don't feel bad if your SM/TC provides helpful advice
- Should be attentive for opportunities to release value or engage stakeholders based on the team's progress

Discussion: The PO's role in the Team Sync

Duration

In your work as a PO for TTC, you often attend trade shows and industry conferences to support your sales and marketing team, identify industry trends, and assess competitive offerings. You know that you will be gone for two weeks attending a trade show and visiting a few key Customers.

- **Step 1:** Individually, think about how you and your team might handle your absence in the Team Sync for two weeks
- **Step 2:** Discuss your ideas with your group

4.4 Backlog refinement

Video: The Product Owner and Backlog Refinement

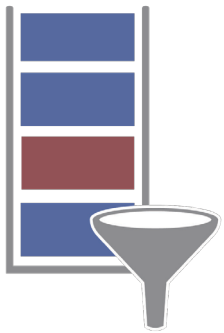
Duration 5 min

https://bit.ly/Video-POBacklogRefinement

Backlog refinement

Backlog refinement is a preview and elaboration of upcoming Stories.

- Helps the team think about new Stories prior to Iteration Planning
- Provides enough time to identify and resolve dependencies and issues that could impact the next Iteration
- The team can improve Stories, add acceptance criteria, and identify missing information
- Most of the focus is on the next Iteration, but it allows time to discuss future Iterations and even Features for the next PI



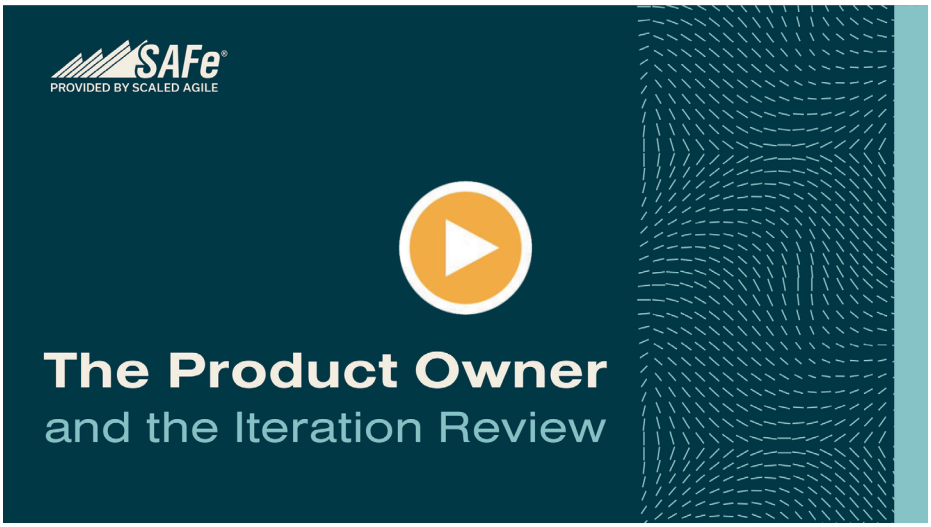
Tips for effective backlog refinement

Best approaches	Common anti-patterns
Revisit Stories as often as needed to finalize and commit to them in Iteration Planning	Arriving to the Iteration with Stories that are not ready
Maintain the right level of a deep Team Backlog versus ready Team Backlog for two Iterations	Completing team backlog refinement on an inconsistent basis
Make sure all team members participate	Team sees Stories for the first time during Iteration or PI Planning
Invite the right subject matter experts	Feature estimations impact Story estimation
Hold the event on a regular cadence	

4.5 Iteration Review and Iteration Retrospective

Video: The Product Owner and the Iteration Review

Duration: 5 min

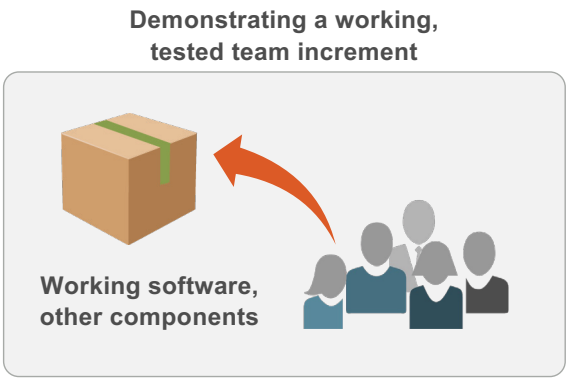


https://bit.ly/Video-POIterationReview

The Iteration Review

- The Iteration Review provides the true measure of progress by showing working software functionality, hardware components, and so on
- Preparation for the review starts with planning
- Teams demonstrate every Story, spike*, refactor, and NFR
- Attendees are the team and its stakeholders

*A spike is a research Story, considered an exploration-style Enabler



Iteration Review guidelines

- **Timebox** - One to two hours
- **Preparation** - Review preparation should be limited to one or two hours; minimize presentation and 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, and NFR
3. Discuss Stories not completed and why
4. Identify risks or impediments
5. Revise Team Backlog and team PI Objectives as needed

Scalable definition of done




Team Increment	System Increment	Solution Increment	Release
• Stories satisfy acceptance criteria • Acceptance tests passed (automated where practical) • Unit tests • Cumulative unit tests passed • Assets are under version control • Engineering standards followed • NFRs met • No must-fix defects • Stories accepted by PO	• Stories completed by all teams in the ART and integrated • Completed features meet acceptance criteria • NFRs met • No must-fix defects • Verification and validation of key scenarios • Included in build definition and deployment process • Increment demonstrated; feedback achieved • Accepted by Product Management	• Capabilities completed by all ARTs meet acceptance criteria • Deployed/installed in the staging environment • NFRs met • System end-to-end integration verification and validation done • No must-fix defects • Included in build definition and deployment/transition process • Documentation updated • Solution demonstrated; feedback achieved • Accepted by Solution Management	• All capabilities done and meet acceptance criteria • End-to-end integration and solutions V&V done • Regression testing done • NFRs met • No must-fix defects • Release documentation complete • All standards met • Approved by Solution and Release Management

4-57

What to do when a Story isn't done

- **Split it?** "Now that I see the Story, I've realized that I only really need part of it. The rest is a new Story that we can work on later."
- **Continue it?** "I still need this Story, and it's still my top priority. Can we finish this Story in the next iteration?"
- **Delay it?** "This Story is important to me, but we've discovered it's huge. I'd prefer that we focus on other Stories with better ROI."
- **Abandon it?** "If the Story is this difficult to develop, it's not worth it for me anymore. The Story is too expensive to develop relative to the value."

Note: If a team frequently ends Iterations with incomplete Stories, consider imposing stricter WIP limits on the Team Kanban.

SCALED AGILE © Scaled Agile, Inc.

4-58

Measure progress (examples)

Measurement	Related Flow Metric
# Stories committed this Iteration	Flow Velocity
# Stories accepted this Iteration	Flow Velocity
# Stories carried over to next Iteration	Flow Velocity
# Deployments	Flow Velocity
# Work items by type committed	Flow Distribution
# Work items by type accepted	Flow Distribution
Average time to complete a Story	Flow Time

Quality and test automation
Test coverage %
Test automation %
Tests executed
Failed tests
New tests created
New tests automated
Mean Time to Recover from failure (MTTR)

Relentless improvement

Agile Teams continuously adapt to new circumstances and improve the methods of value delivery. To support relentless improvement:

- Understand where you are
- Foster a Continuous Learning Culture
- Use retrospectives as summary points but not as limitations
- Actively engage with the SMs/TCs to drive improvement on the ART

Video: The Product Owner and Iteration Retrospective

Duration 5 min

https://bit.ly/Video-POIterationRetro

Iteration Retrospective

- **Timebox:** 30 to 60 minutes
- **Purpose:** Pick one or two items that can be done better for next Iteration
- **Outcome:** 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 Metrics.

Part 2: Qualitative

1. What went well?
2. What didn't go well?
3. What can we do better next time?

Improving everywhere

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

Examples
Move from manual to automated testing
Communicate with remote teams, subject matter experts, and so on
Consider the team's skill set
Prepare and run the demo
Include nonfunctional requirements testing
Provide more efficient and disciplined design sessions

4.6 DevOps and Release on Demand

Video: What is DevOps

Duration 2 min

https://bit.ly/Video-WhatisDevOps

SCALED AGILE

4-65

Video: Continuous Delivery Pipeline

Duration 5 min

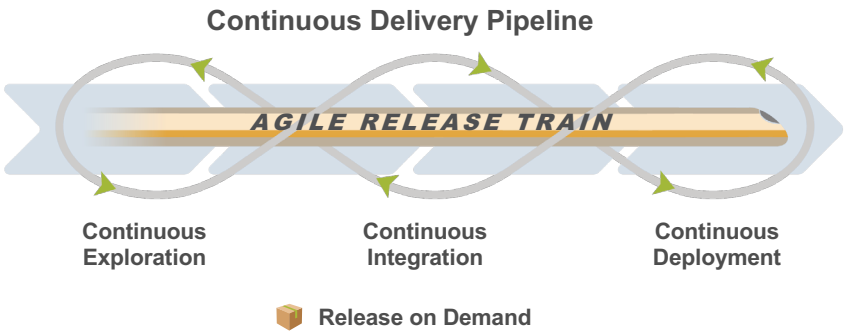
https://bit.ly/Video-CDP

SCALED AGILE

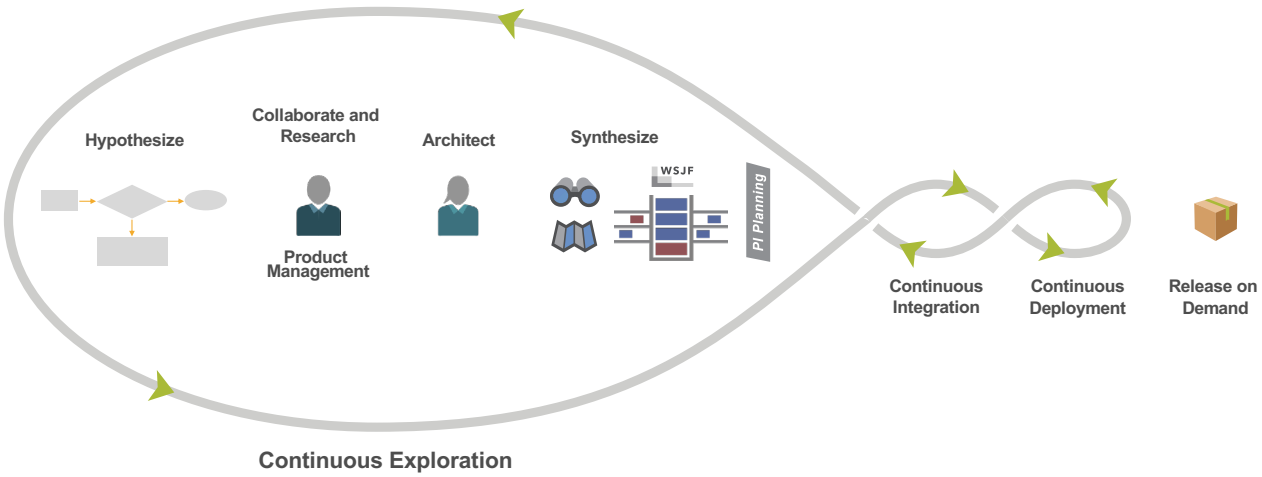
4-66

Building the Continuous Delivery Pipeline with DevOps

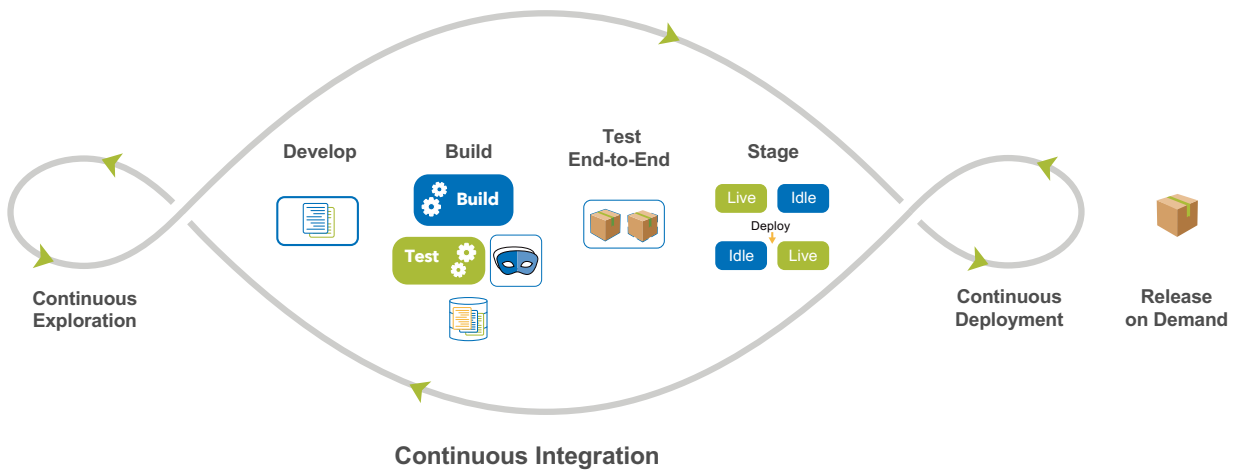
- The Continuous Delivery Pipeline (CDP) represents the workflows, activities, and automation needed to deliver new functionality more frequently
- Each ART builds and maintains, or shares, a pipeline
- Organizations map their current pipeline into this new structure, remove delays, and improve the efficiency of each step



Continuous Exploration (CE) – Understand Customer needs

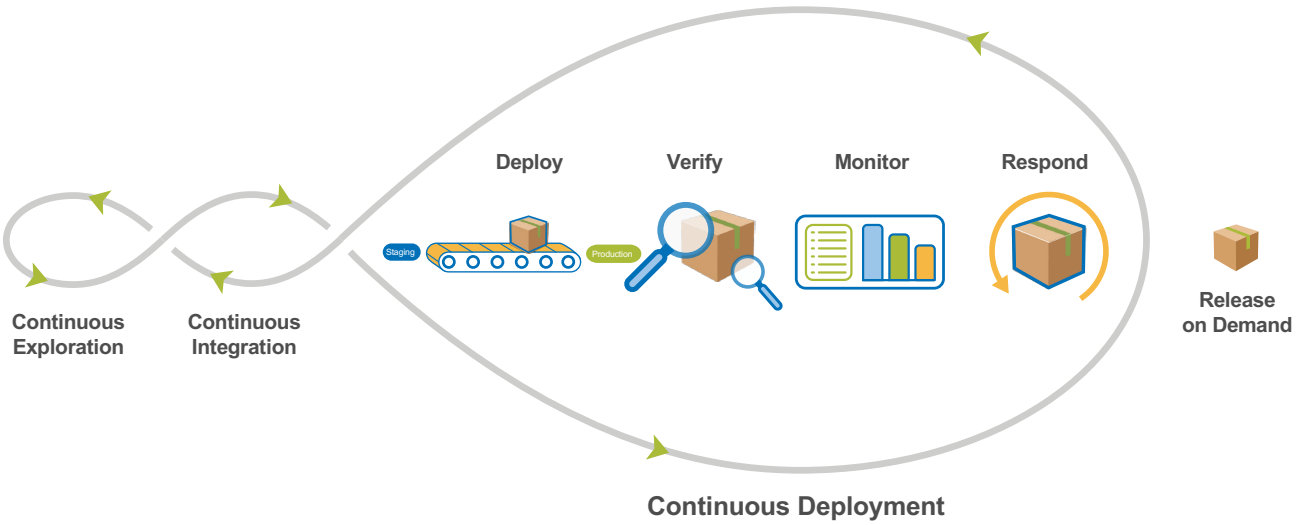


Continuous Integration (CI) – A critical technical practice of the ART



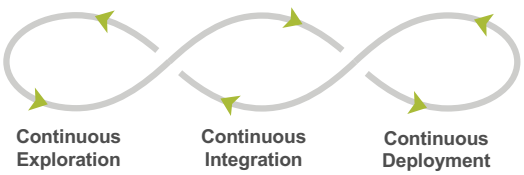
SCALED AGILE © Scaled Agile, Inc. 4-69

Continuous Deployment (CD) – Getting to production early

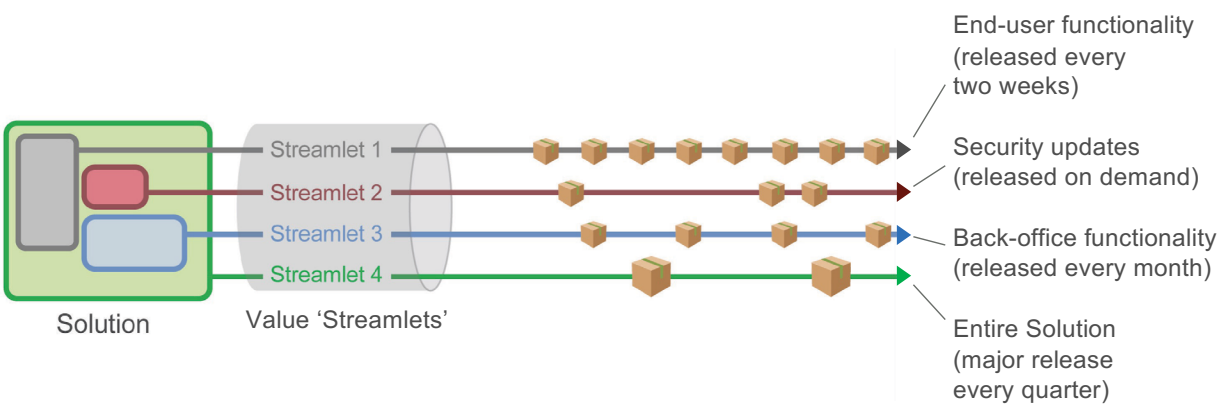


SCALED AGILE © Scaled Agile, Inc. 4-70

Release on Demand – Making value available when it's needed



Decouple release elements from the total Solution



Discussion: Decouple release elements

Duration 10 min

The TTC Van Maintenance Advisor is a complex Solution. It includes components that operate in the van, a web application, and a smartphone application.

- **Step 1:** Individually, consider if all the components should be released at the same time. Why or why not?
- **Step 2:** Discuss as a class.

Action Plan: Executing Iterations

Duration 5 min

On the Action Plan page in your workbook, answer the following:

- Think about two to three specific actions you can take to create and manage Stories with your team. Write a plan for each.
- Consider and plan one to two ways you could apply Story mapping.
- Identify one to three actions to improve your participation in Iteration events.

Action Plan

Executing Iterations

Lesson review

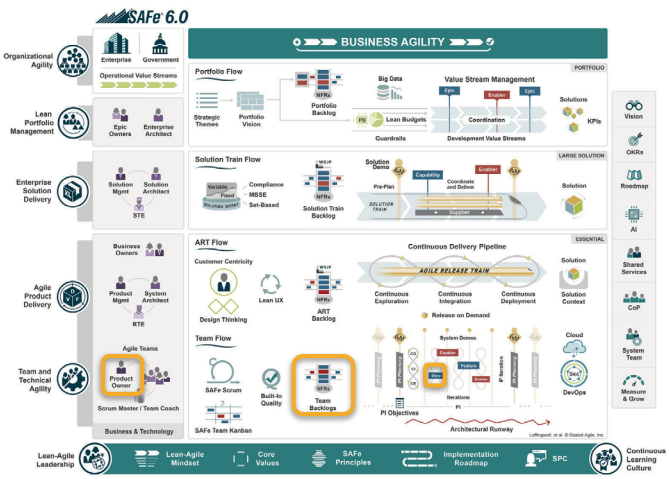
In this lesson, you:

- Created Stories
- Demonstrated how to plan an Iteration
- Outlined how to manage flow with the Team Kanban
- Summarized how to continuously refine the Team Backlog
- Planned how to participate in the Iteration Review and Iteration Retrospective
- Explained how to support DevOps and Release on Demand

Articles used in this lesson

Read these Framework articles to learn more about topics covered in this lesson

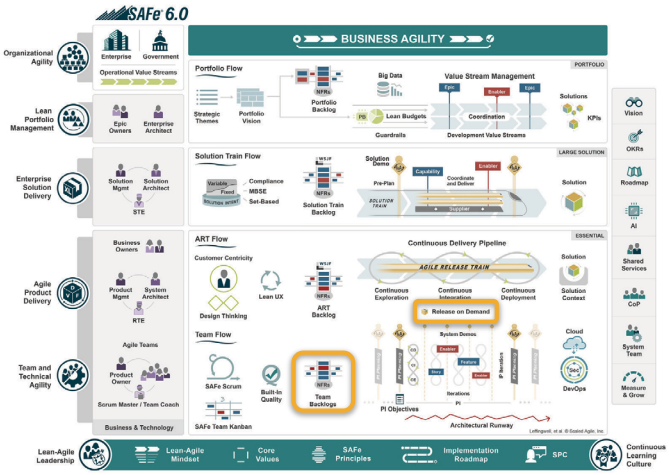
- "Story" https://www.scaledagileframework.com/story/
- "Team Backlog" https://www.scaledagileframework.com/team-backlog/
- "Product Owner" https://www.scaledagileframework.com/product-owner/



Articles used in this lesson

Read these Framework articles to learn more about topics covered in this lesson

- "Release on Demand" https://www.scaledagileframework.com/release-on-demand/
- "SAFe Team Kanban" https://scaledagileframework.com/safe-team-kanban/
- "Spikes" https://www.scaledagileframework.com/spikes/



Continue your SAFe journey with the following resources:

Use the *Story Storming & Refining* Collaborate template to support your team's Story creation and refinement activities. **https://bit.ly/Template-StoryRefining**	Use the *Story splitting on an Agile Team* Collaborate template to lead your team through splitting large Stories and refining smaller Stories. **https://bit.ly/Template-StorySplitting**
Use the *Develop a Story Map* Collaborate template to organize a sequence of Stories according to the tasks the team needs to complete in order to accomplish their goals. **https://bit.ly/Template-StoryMapping**	Watch and share this eight-minute playlist on *Stories*, including *Introduction to Stories* and *Writing Effective Stories*, to support your understanding of how to make Stories as effective as possible. **https://bit.ly/Playlist-Stories**

Continue your SAFe journey with the following resources:

Use the *Facilitator's Guide to SAFe - Backlog Refinement* document to successfully prepare for backlog refinement activities. https://bit.ly/Studio-FGBacklogRefinement	Use the *Facilitator's Guide to SAFe - Iteration Planning* document to successfully prepare for Iteration Planning activities. https://bit.ly/Studio-FGIterationPlanning
Use the *Facilitator's Guide to SAFe - Iteration Review* document to successfully prepare for Iteration Review and demo events. https://bit.ly/Studio-FGIterationReview	

4-79

References

Cohn, Mike. *Agile Estimating and Planning*. Pearson Education, Inc.: Upper Saddle River, 2006. 56-59.

Jeffries, Ron. “Essential XP: Card, Conversation, Confirmation.” Ron Jeffries. August 30, 2001. https://ronjeffries.com/xprog/articles/expcardconversationconfirmation.

Lawrence, Richard and David Green. “The Humanizing Work Guide to Splitting User Stories.” Humanizing Work. Updated August 15, 2022. https://www.humanizingwork.com/the-humanizing-work-guide-to-splitting-user-stories/.

Wake, Bill. “INVEST in Good Stories, and SMART Tasks.” XP123. August 17, 2003. https://xp123.com/articles/invest-in-good-stories-and-smart-tasks.

4-80

Lesson 4 notes

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

Lesson 5

Executing the PI

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

POPM SAFe 6

SCALED AGILE

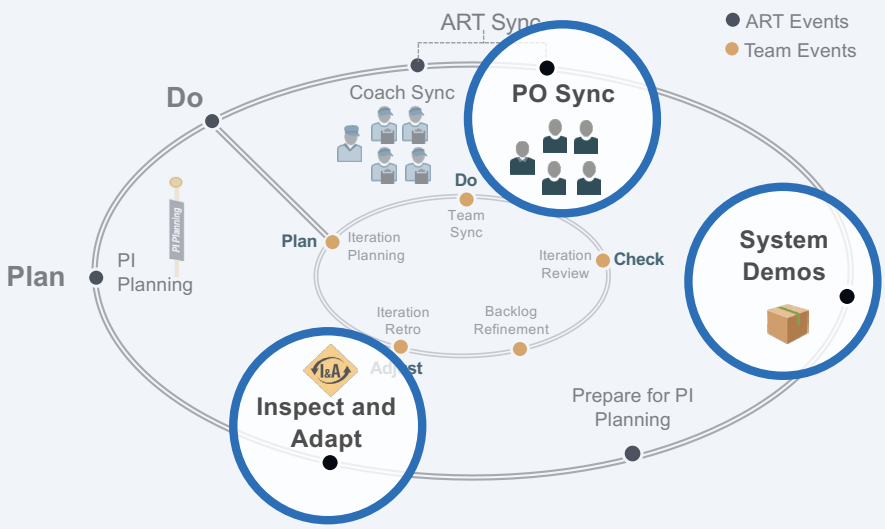
Lesson Topics

Learning objectives

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

- Define how to participate in the PO Sync
- Plan how to participate in the System Demo
- Explain how to innovate throughout the PI
- Summarize how to Inspect and Adapt

Executing the PI



5.1 The PO Sync

Video: ART Events: PO Sync

Duration 5 min

https://bit.ly/Video-ARTEventsPOSync

The PO Sync helps ARTs respond to change

The PO Sync:

- Provides visibility into how well the ART is progressing toward meeting the ART PI Objectives
- Provides an opportunity to assess scope adjustments
- May be used to prepare for the next PI, including sharing learnings from Continuous Exploration, ART backlog refinement, and WSJF (weighted shortest job first) prioritization
- Is facilitated by the RTE or Product Management
- Includes Product Managers, POs, stakeholders, and SMEs, as necessary
- Occurs weekly or more frequently and lasts 30–60 minutes long

POs communicate adjustments to their teams after the sync.

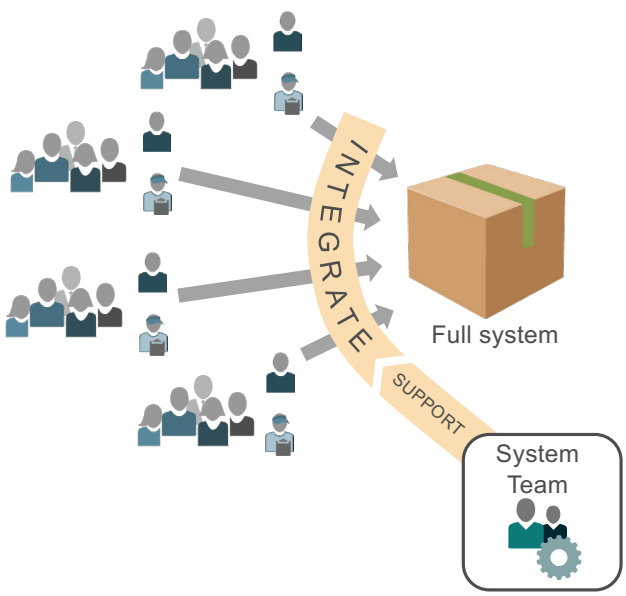
Discussion: Enabling alignment across the ART with sync events

- **Step 1:** Individually, think about the various sync events that are part of SAFe (Team Sync, Coach Sync, PO Sync, ART Sync, Architect Sync)
- **Step 2:** As a class, discuss the following:
  - How can you leverage these sync events as a PO or Product Manager to ensure alignment across the ART?
  - What issues and opportunities might you bring up?
  - What potential solutions might you offer to resolve the issues and follow through with opportunities?

5.2 The System Demo

Demo the full system increment every two weeks

- Features are functionally complete or toggled so as not to disrupt demonstrable functionality
- New Features work together and with existing functionality
- Demos occur after the Iteration Review
  - May lag by as much as one Iteration, maximum
- Demo from a staging environment that resembles production as much as possible



Recommended System Demo agenda

5 min:	Briefly review the business context and the PI Objectives.
5 min:	Briefly describe each new Feature before running the demo.
25 min:	Demo each Feature. Frame each Feature in the context of how a Customer or persona will gain benefit from the Feature or how the Feature will create business value.
15 min:	Identify current risks and impediments.
10 min:	Open discussion for questions and feedback, and summarized progress.

Apply the meet-after pattern to keep the System Demo focused.

SCALED AGILE

5-11

Tips for effective System Demos

Best approaches	Common anti-patterns
Begin to consider how and what to demo in Iteration Planning	A lot of time is spent preparing for the demo
Make sure the right participants are present	Demo does not showcase Customer value
Ensure that the team celebrates its accomplishments and that stakeholders acknowledge them	POs and Product Management see things for the first time in the System Demo
Make sure different team members have the opportunity to demo	System Demo is not done because “the Team Demo is enough”
Ensure that the team is ready for the System Demo and coordinates with the System Team	Team members are not invited to the System Demo to save time
	Demos are not interesting or relevant to ART level stakeholders
	Using test data

SCALED AGILE

5-12

5.3 The Innovation and Planning Iteration

SAFe Core Values

Alignment

- Communicate the Vision, mission, and strategy
- Connect strategy to execution
- Speak with a common language
- Constantly check for understanding
- Understand your Customer

Transparency

- Create a trust-based environment
- Communicate directly, openly, and honestly
- Turn mistakes into learning moments
- Visualize work
- Provide ready access to needed information

Respect for People

- Hold precious what it is to be human
- Value diversity of people and opinions
- Grow people through coaching and mentoring
- Embrace 'your Customer is whoever consumes your work'
- Build long-term partnerships based on mutual benefit

Relentless Improvement

- Create a constant sense of urgency
- Build a problem-solving culture
- Reflect and adapt frequently
- Let facts guide improvements
- Provide time and space for innovation

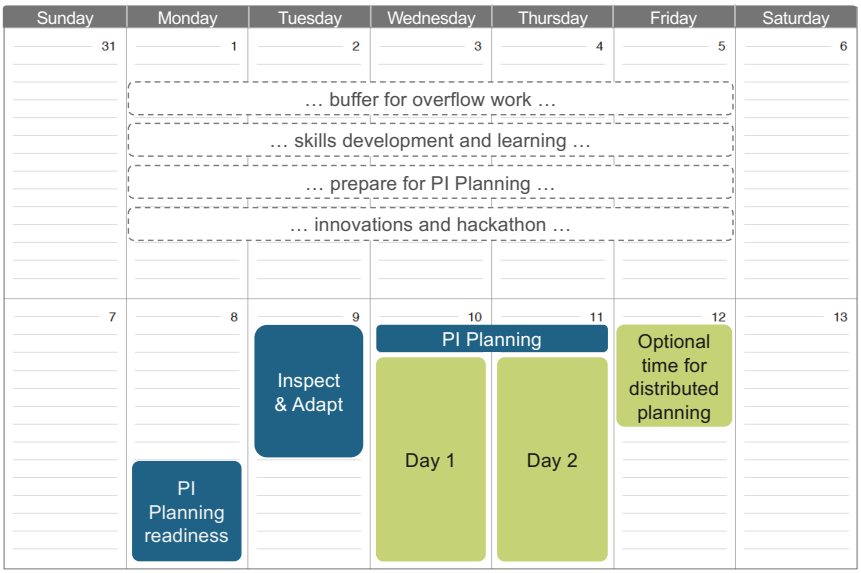
Innovation and Planning (IP) Iteration

PI Planning occurs on cadence within the IP Iteration, which facilitates reliability, PI readiness, planning, and innovation.

- **Innovation** - Opportunity for innovation, hackathons, and infrastructure improvements
- **Planning** - Provides for cadence-based planning
- The IP Iteration also provides an estimating guard band for cadence-based delivery

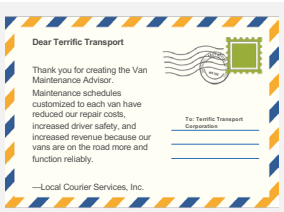
> "Provide sufficient capacity margin to enable cadence."
>
> —Donald G. Reinertsen, *The Principles of Product Development Flow*

The IP Iteration



Innovative ideas come from many sources

Internal sources



Agile Teams

- Aspirational, yet realistic and achievable
- Motivational enough to engage others on the journey
- **Result:** The teams start thinking about how to apply their strengths in order to get there

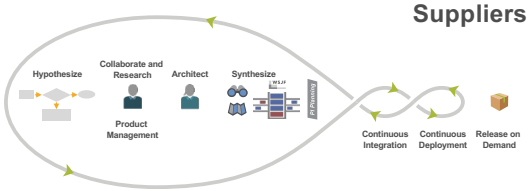
Reference: Heath and Heath, *Switch*

External sources

Competitors

Customer requests

Suppliers



SAFe has many ways to promote and capture innovation