

Lesson 3

Planning the Iteration

1. Introducing the Scaled Agile Framework
2. Building an Agile Team
- 3. Planning the Iteration**
4. Executing the Iteration
5. Executing the PI

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

Learning objectives

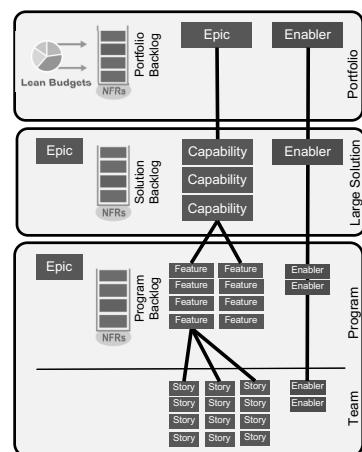
- 3.1 Prepare the Backlog
- 3.2 Plan the Iteration

3.1 Prepare the Backlog

Define Solution Features for the Program Backlog

Features are services that fulfill user needs.

- ▶ “Feature” is an industry-standard term familiar to marketing and Product Management
- ▶ Expressed as a phrase, “value” is expressed in terms of benefits
- ▶ Identified, prioritized, estimated, and maintained in the Program Backlog



Features have benefits and acceptance criteria

- ▶ Benefit hypothesis justifies Feature implementation cost and provides business perspective when making scope decisions
- ▶ Business benefits impact economic prioritization of the Feature
- ▶ Acceptance criteria is typically defined during Program Backlog refinement
- ▶ Reflect functional and nonfunctional requirements

SSO example:

Benefit hypothesis

Multifactor authentication

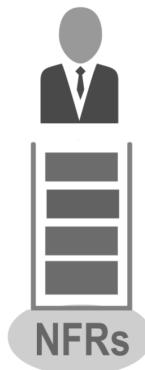
Enhance user security via both password and a device.

Acceptance criteria

1. USB tokens as a first layer
2. Password authentication as a second layer
3. Multiple tokens on a single device
4. User activity log reflecting both authentication factors

The Team Backlog

- ▶ Contains all the work the team needs to work on
- ▶ Created by the Product Owner and the team
- ▶ Prioritized by the Product Owner
- ▶ Contains User and Enabler Stories
 - User stories provide Customers with value
 - Enabler Stories build the infrastructure and architectures that makes user stories possible
- ▶ Stories in the backlog are prioritized
- ▶ Stories for the next Iteration are more detailed than Stories for later Iterations
- ▶ Nonfunctional requirements are a constraint on the backlog



NFRs

User stories

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

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

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

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

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

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

(Roles can be people, devices, or systems)

User Story guidelines — The 3 Cs

Card

They are written on a card or in the tool and may annotate 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 Product Owner.

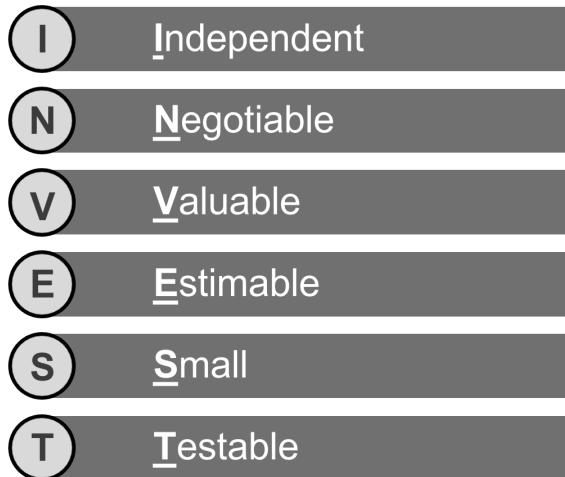


Confirmation

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

INVEST in a good Story



Enabler Stories

Enabler Stories build the groundwork for future user stories.

- ▶ Three types of Enabler Stories:
 1. Infrastructure – Build development and testing frameworks that enable a faster and more efficient development process
 2. Architecture – Build the Architectural Runway, which enables smoother and faster development
 3. Exploration – Build understanding of what is needed by the customer, to understand prospective solutions and evaluate alternatives

Splitting Features and Stories

Techniques for splitting Features and Stories to fit within their boundaries (PI and Iteration, respectively).

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

Exercise: Break Features into Stories

Work with your team to break Features from the Program Backlog into Stories that are small enough to fit into an Iteration.

- ▶ Break a Feature into 5 to 10 Stories
- ▶ Make sure the Stories retain a business value
- ▶ Try to create Stories that are less than a week in size
- ▶ Identify spikes as needed
- ▶ You should have at least 5 Stories from your Features



Acceptance criteria

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

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

Acceptance criteria:

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

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

Acceptance criteria :

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

Exercise: Write acceptance criteria

- ▶ Write acceptance criteria for the Stories you have identified
- ▶ Make sure the acceptance criteria is testable
- ▶ The Product Owner needs to approve the acceptance criteria



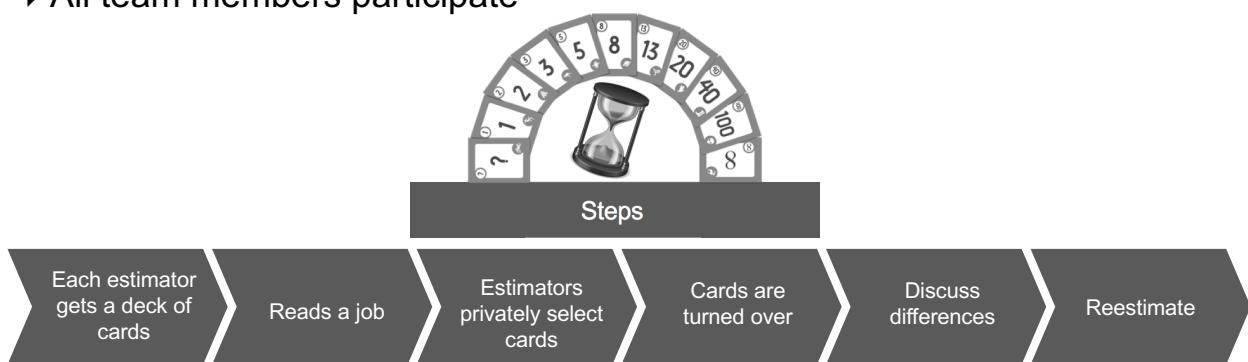
Estimate Stories with relative Story points

- ▶ 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
- ▶ Compare with other Stories (an 8-point Story should take four times longer than a 2-point Story)



Apply Estimating Poker for fast, relative estimating

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



Estimation is a whole-team exercise

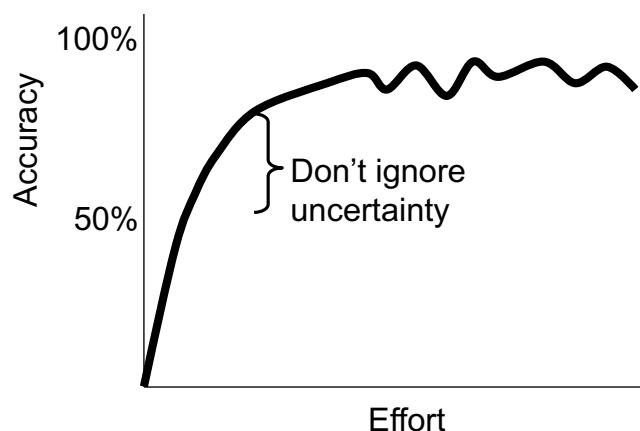
- ▶ Increases accuracy by including all perspectives
- ▶ Builds understanding
- ▶ Creates shared commitment

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



How much time to spend estimating

A little effort helps a lot. A lot of effort only helps a little.



Exercise: Estimate Stories

- ▶ Estimate the Stories you have identified
- ▶ Use Estimating Poker together as a group
 - The Scrum Master will facilitate the activity
- ▶ The Product Owner doesn't vote but can ask or answer questions
- ▶ You should have at least three Stories estimated in Story points



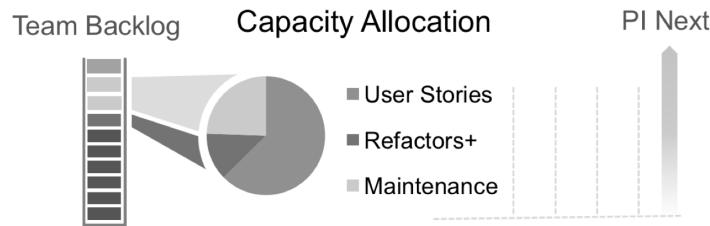
Sequencing Stories

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

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

Capacity allocation for a healthy balance

- ▶ By having capacity allocation defined, the Product Owner doesn't need to prioritize unlike things against each other
- ▶ Once the capacity allocation is set, the PO and team can prioritize like things against each other



Notes:

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

3.2 Plan the Iteration

Plan and commit

Purpose	Define and commit to what will be built in the Iteration	
Process	<ul style="list-style-type: none">▶ The Product Owner defines <i>what</i>▶ The team defines <i>how</i> and <i>how much</i>▶ Four hours max	
Result	Iteration Goals and backlog of the team's commitment	
Reciprocal commitment	<ul style="list-style-type: none">▶ Team commits to delivering specific value▶ Business commits to leaving priorities unchanged during the Iteration	

Iteration Planning flow

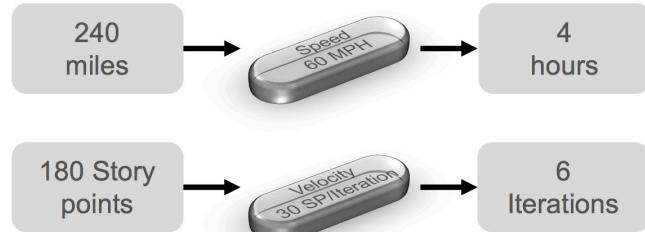
- 1 The team establishes its velocity 
 - 2 The team clarifies and estimates the Stories 
 - 3 The team optionally breaks Stories into tasks 
 - 4 The process continues while there is more capacity 
 - 5 The team synthesizes Iteration Goals 
 - 6 Everyone commits 
- ▶ Timebox: 4 hours
 - ▶ This meeting is by and for the team
 - ▶ SMEs may attend as required

Using size to estimate duration

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



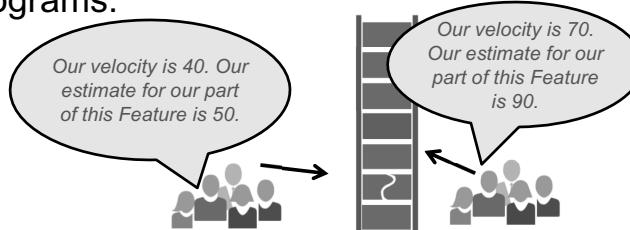
Examples



Establishing velocity before historical data exists

Normalized Story point estimating provides the economic basis for estimating work within and across programs.

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



Example: Assuming a 6-person team composed of 3 developers, 2 testers, 1 PO, with no vacations, etc.

$$\text{Estimated velocity} = 5 * 8 \text{ pts} = 40 \text{ pts/Iteration}$$

Exercise: Calculate your initial velocity

Use the slide on the previous page to calculate your team's starting velocity.

- ▶ Calculate your estimated velocity for the next two-week Iteration, which starts after PI Planning
- ▶ Use your real availability
- ▶ Each team should have their estimated velocity for two Iterations



Iteration Goals

Iteration Goals provide clarity, commitment, and management information.

They serve three purposes:

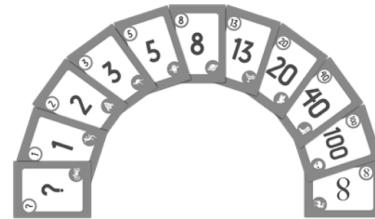
1. Align team members to a common purpose
2. Align Program Teams to common PI Objectives and manage dependencies
3. Provide continuous management information

Iteration Goals example

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

Story analysis and estimation

- ▶ The Product Owner presents Stories in order of priority
- ▶ Each Story
 - Is discussed and analyzed by the team
 - Has its acceptance criteria refined
 - Is estimated
- ▶ The process continues until the estimation of the Stories has reached the velocity of the team



Exercise: Plan your first Iteration

- -
 -
 -
- Use the team's initial velocity and the estimated Stories to allocate Stories to the first Iteration.
- ▶ Plan which Stories fit into the first Iteration
 - ▶ Estimate more Stories as needed to fill the first Iteration
 - ▶ Each team should have a list of Stories that they can deliver by the end of the first Iteration



Commit to the Iteration Goals

A Team meets its commitment:

By doing everything they said they would do.

- or -

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

Commitment

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



Adaptability

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

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

Iteration planning for Kanban teams

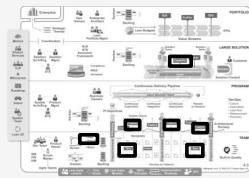
- ▶ Some teams have a more responsive nature to their work, such as maintenance teams, System Teams, DevOps
- ▶ These teams find less value in trying to plan the Iteration in detail
- ▶ Kanban teams still publish Iteration Goals, which consist of the known parts of their work
- ▶ They commit to the goals as well as to a cycle time SLA for incoming work based on their known historical data



Lesson summary

In this lesson, you:

- ▶ Prepared your backlog of stories through breaking down features
- ▶ Planned your iteration using story estimation



Suggested Scaled Agile Framework reading:

- “Stories” article
- “Features and Capabilities” article
- “Iteration Planning” article