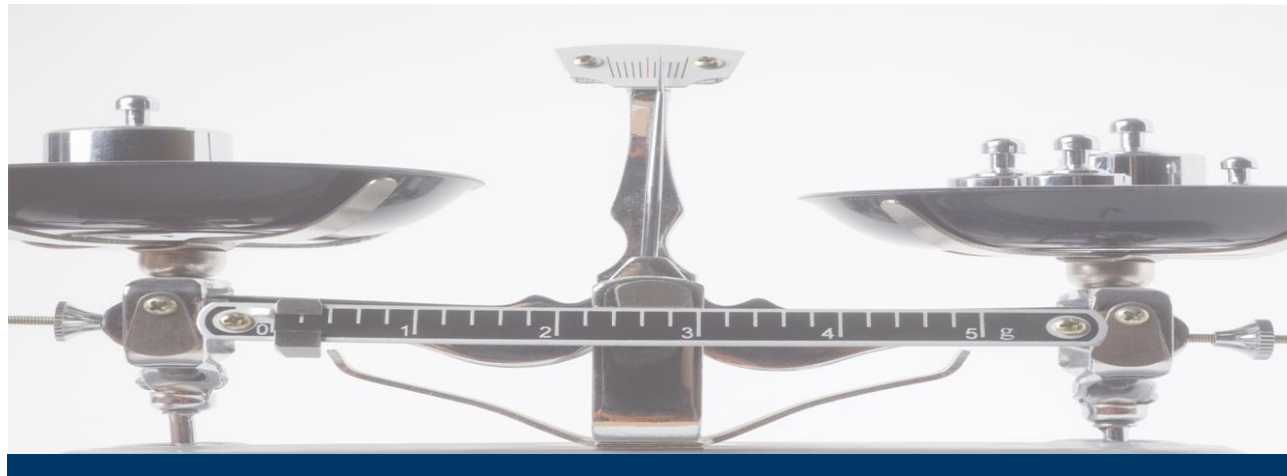


CS-SDF Agile Lifecycle Training

Product Owner



Enabling Practices & Tools

Course Etiquette

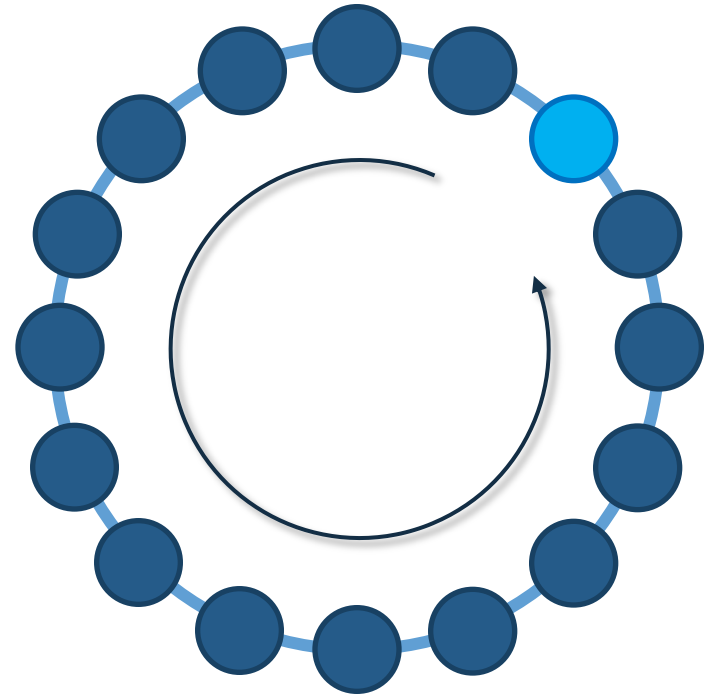
- Please turn your mobile phones off
- No laptops
- We start on time after breaks
- Please leave the seminar room clean and tidy
- Questions are encouraged at any time

Get to Know Each Other

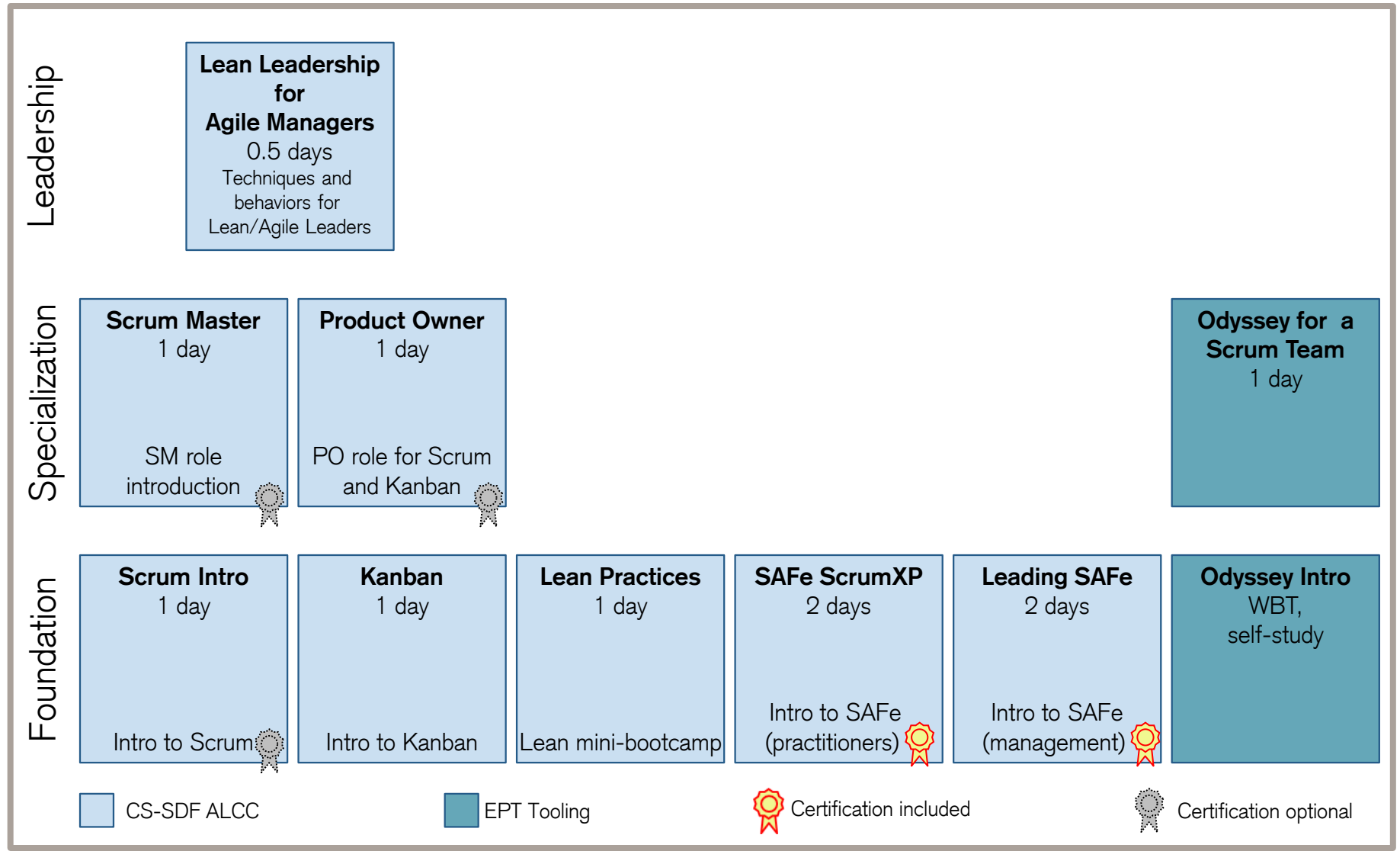
- Please introduce yourself
- What do you want to take with you after this course

Framework

- Your Name
- Your current Role
- What would you like to get out of this course?



Agile learning map



Training Objectives

As an outcome of this training, participants should:

- Understand the role of the Product Owner: Behaviors and Techniques
- Be able to manage requirements, plan and manage them, and create a Product Vision
- Understand the key differences between the Product Owner, Scrum Master, and Project Manager
- Have created a Backlog ready for execution



Agenda

1

Scrum Introduction

2

Role Product Owner

3

Vision

4

Roadmap and Product Backlog

5

Sprint Activities

6

Velocity

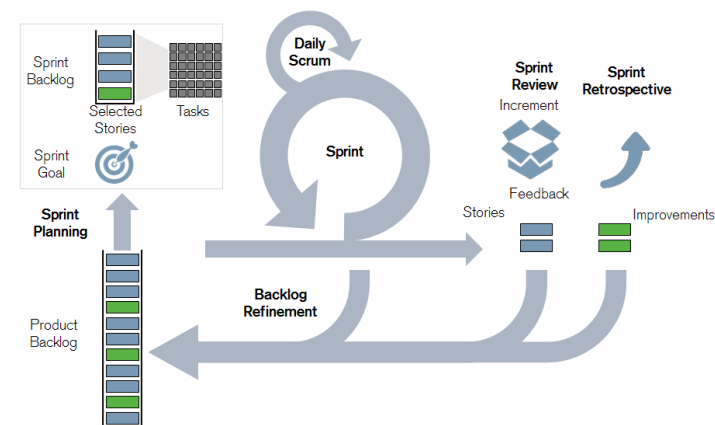
Scrum Introduction

Exercise



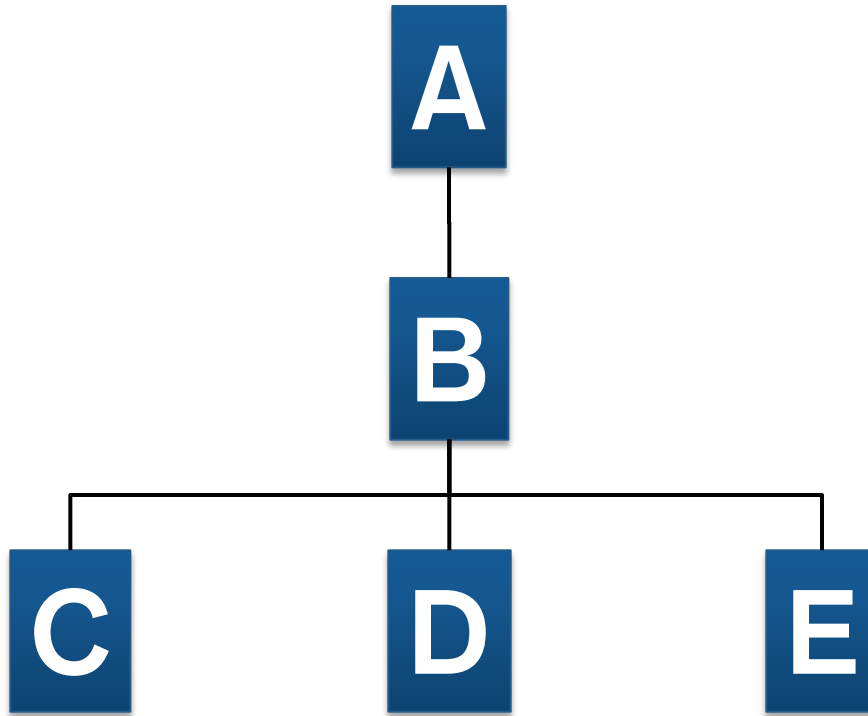
Recap

- An empirical process, for both Product Development and Process Improvement
- Time-boxed set of activities in “Sprint”
- Frequent team alignment
- Sprint work is reviewed and a shippable increment of Product shown to customer
- Built on core themes like
 - Value Delivery in small batches
 - Quality
 - Productivity
 - Change and Adaptability
 - Continuous Improvement



Role Product Owner

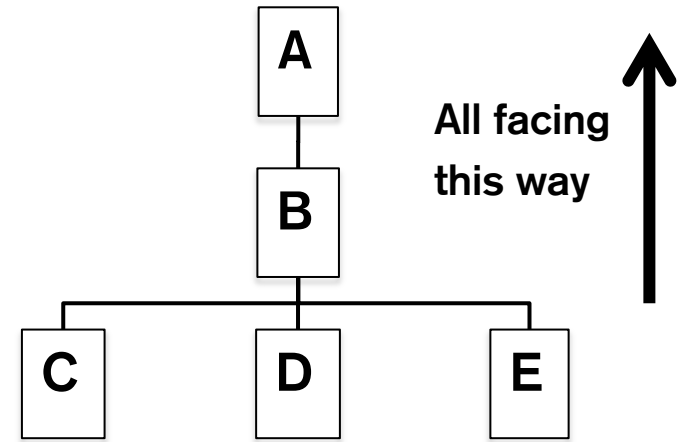
Exercise



The ABCDE Game

The ABCDE Game: Rules

- No talking, or signs
- Your team is executing a project
- C, D and E report to B
- B reports to A
- C, D and E may communicate with only B. Not communicate with each other
- Communication is via Bio Mail (the 6th person), using Post-It notes
- Indicate the recipient and the sender of the message, by their letters
- You have **15 min total**. Upon completion, A stands up and announces “Done”
- We will check the status at every **5 min interval**
- Instruction sheets are given out facedown, don’t turn them over until instructed



Overview



Responsible for the Vision, the Product and the Product Backlog....

... ensures the Value of the work the team performs ...

... provides the right information to the team: explains the “WHY”, does focus on the “WHAT”, but leaves the “HOW” to the team...

... accepts or rejects demonstrated work results and gives immediate feedback.

To be and not to be

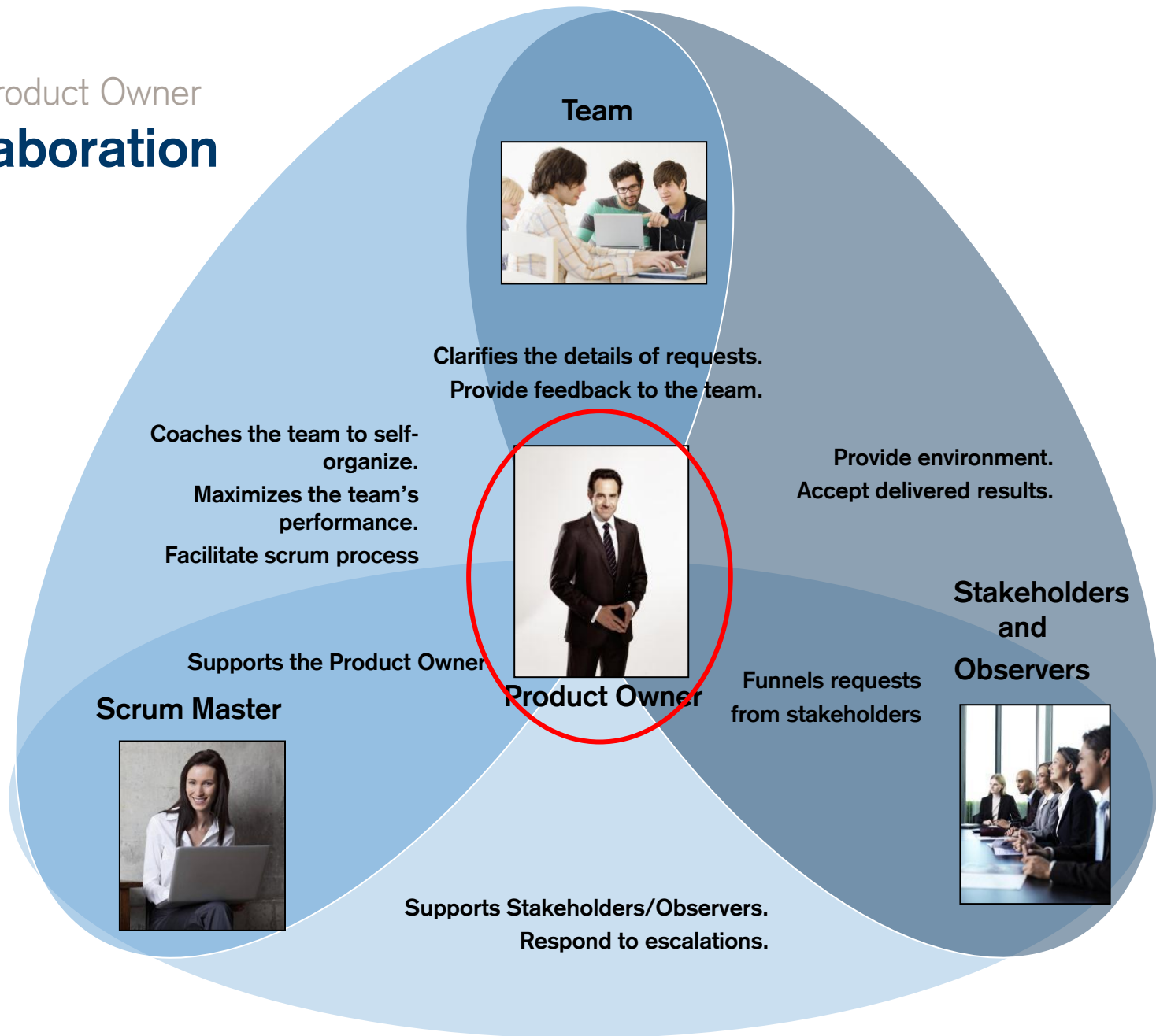
The Product Owner is

- A Visionary and a Doer
- A Leader and a Team Player
- A Communicator and Negotiator
- Empowered and Committed
- Available and Qualified

Rather than a

- Controller
- Task dispatcher
- Organizer
- Supervisor
- Commander
- «Line Manager»

Role Product Owner Collaboration



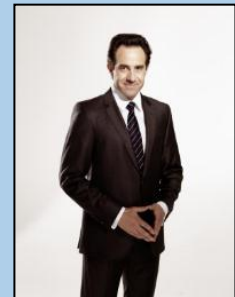
Working with the Team

- Communicates the Vision of the Product
- Provides the Product Backlog
- Proposes the Sprint backlog and goal
- Provides any information that the team needs
- Doesn't disturb the team during an Sprint

Team



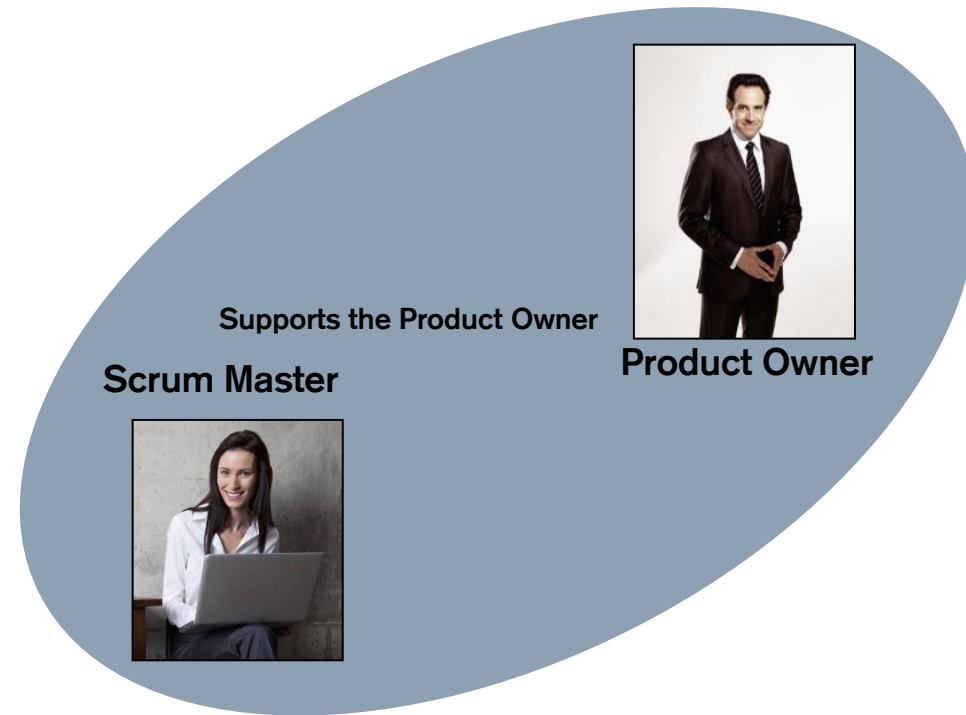
**Clarifies the details of requests.
Provide feedback to the team.**



Product Owner

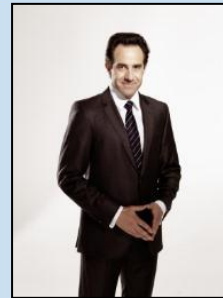
Working with the Scrum Master

- Respects and accepts the Scrum Master
- Supports in resolving impediments
- Collaborates with the Scrum Master
- Product Owner is responsible for the «What», Scrum Master for facilitating the «How»



Working with Stakeholders and Observers

- Alignment of Vision, Roadmap with stakeholders
- Management of Product Backlog by gathering requirements from all the stakeholders
- Prioritizes the requirements linearly to maximize ROI
- Ensures all input is organized and discussed
- Keeps stakeholders informed about the status of the current release/product



Product Owner

Funnels requests
from stakeholders

**Stakeholders
and
Observers**



Role Product Owner

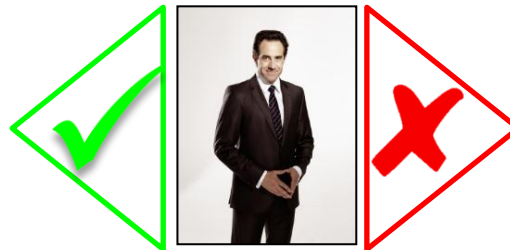
Exercise



Product Owner

Who can be a Product Owner

- Customer
- Business Project Leader
- Requirements Engineer
- Architect
- Business Analyst



Product Owner

Who shouldn't be a Product Owner

- Line Manager
- Scrum Master
- (Program Manager)
- (Project Manager)

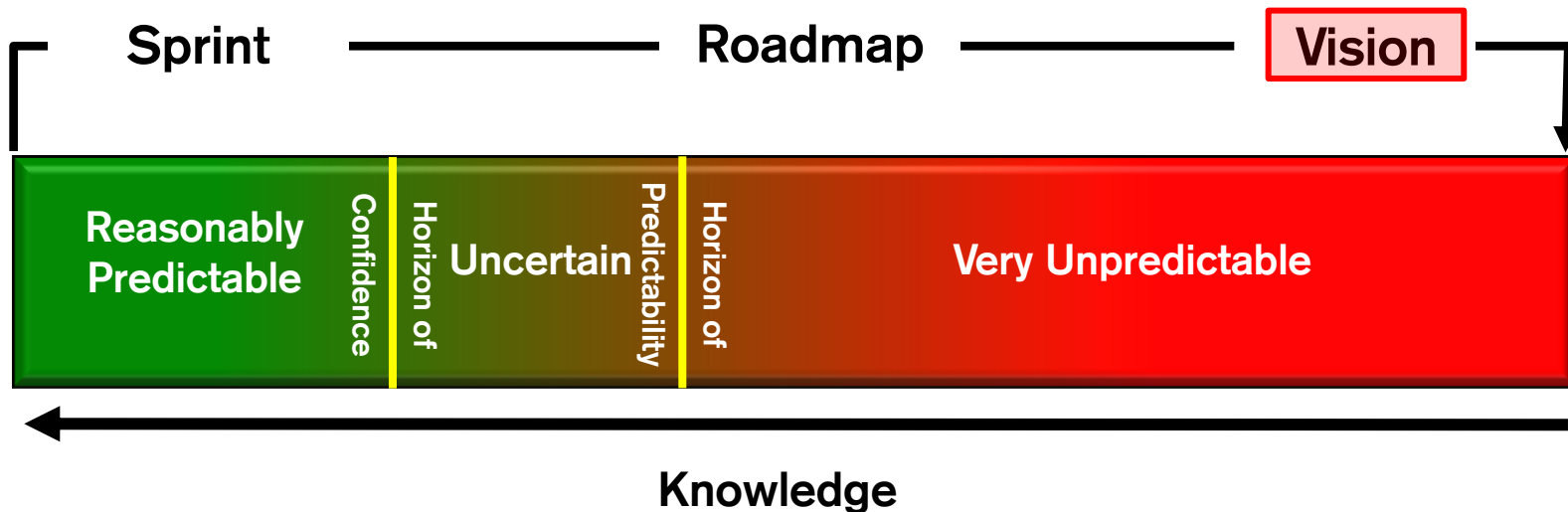
Product Owner Qualities

- **C**ommitted – to the success of the Product
- **R**esponsible – to the stakeholders, development team and for the success/failure of the Product
- **A**uthoritative – to be able to prioritize the backlog and take hard decisions
- **C**ollaborative – with the stakeholders and team to align towards Product Vision, Mission and Roadmap
- **K**nowledgeable – on the domain, market and Product

Vision

The Vision is a

- picture of the result in the future
- inspiration for everyone in the program or project
- framework for the strategic planning
- answer to the question “Where do we want to go”



The Vision is unpredictable

Cost, effort and
size

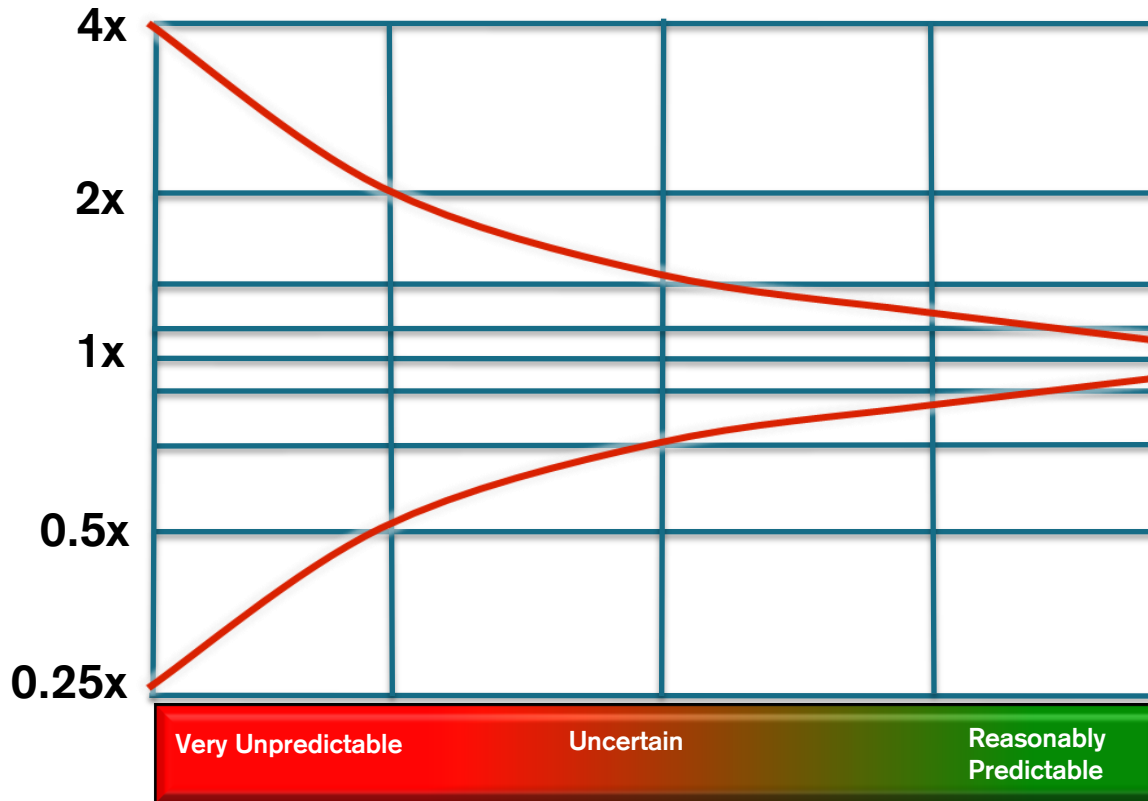
+



Variance



-



Knowledge →

Creating a Vision with the Vision Board Technique

An effective Vision:

- Motivates the team and keeps them focused
- States the objectives and key features of the product being developed
- Provides a purpose to the team

VISION STATEMENT: **FOR** <target customer> **WHO** <statement of need>, **THE** <product name> is a <product category> **THAT** <product key benefit, compelling reason to buy>. **UNLIKE** <primary competitive alternative>, **OUR PRODUCT** <final statement of primary differentiation>.



Source: Roman Pichler: <http://www.romanpichler.com/blog/product-vision/the-product-vision-board/>

Creating a Vision with the Vision Box Technique

- Select a Product Name
- Create a graphic that will represent the Product
- Find bullet points that would sell the product



Product Owner Exercise

Simulation



Product Owner Simulation

Introduction of the Simulation

■ Objective

- Have an executable Product Backlog at the end of the Training

■ Approach

- Self-organize into groups of 4 to 5
- Select a Product for the Simulation Exercise
 - From your working environment at CS
 - Mobile Banking App
- Create a Vision
- Write Features (High-Level Stories)
- Estimation of Features
- Plan Release and create a Roadmap
- Write User Stories (Low-Level Stories)
- Estimation of User Stories
- Plan Sprint
- Commitment

Product - Mobile Banking App

For iPhone: Banking to Go

The Mobile Banking App from Credit Suisse



Make payments now
from your mobile phone.



Vision

Exercise

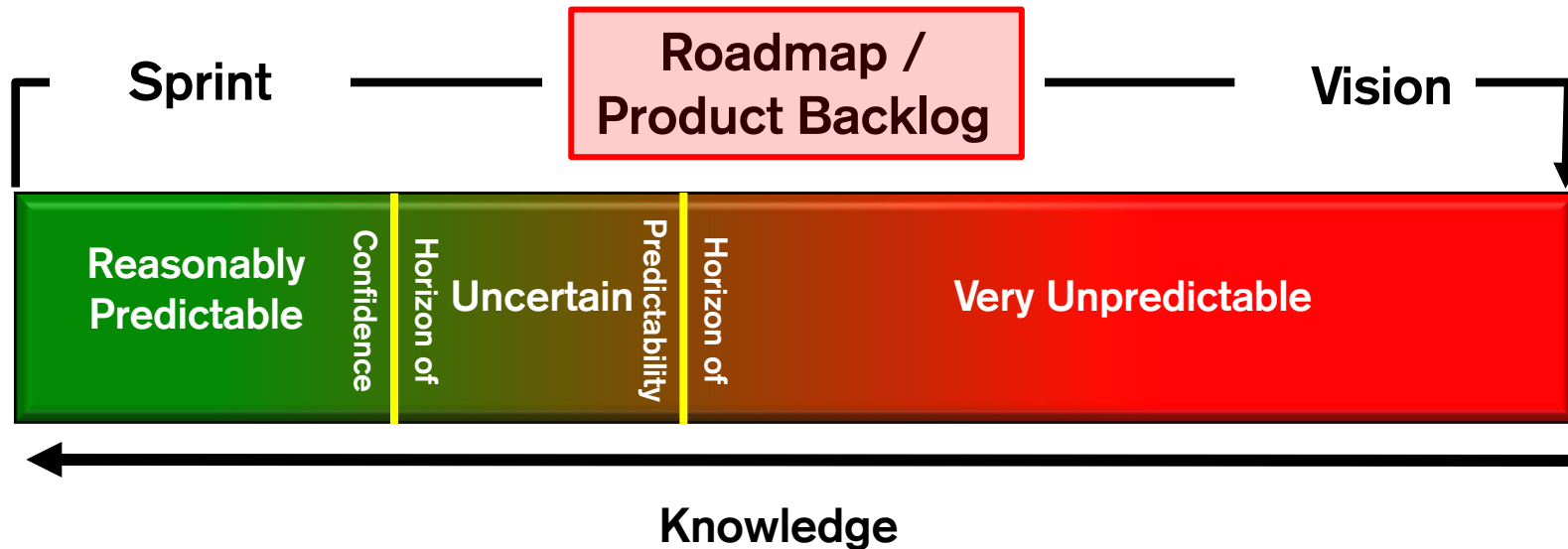


Create a Vision

Roadmap and Product Backlog

The Bridge between the Vision and the Sprint

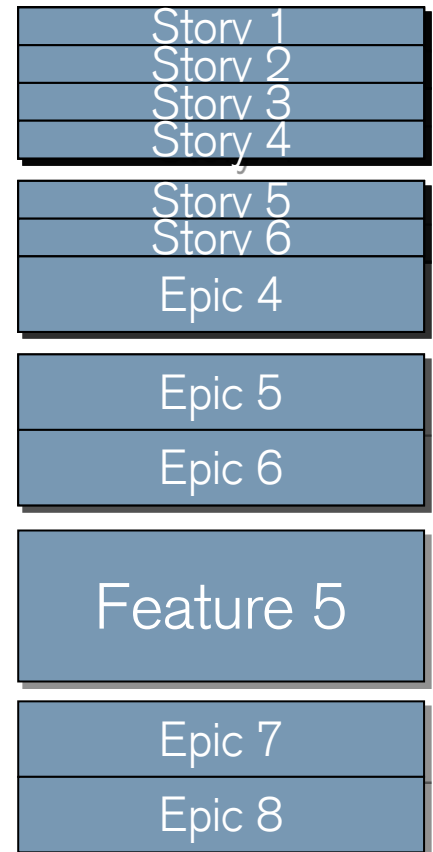
- Guides the way to realize the Vision
- Enables the Product Owner and the Stakeholders to plan Product Releases and supports the budgeting process
- Enhances predictability
- Facilitates discussions around the scope



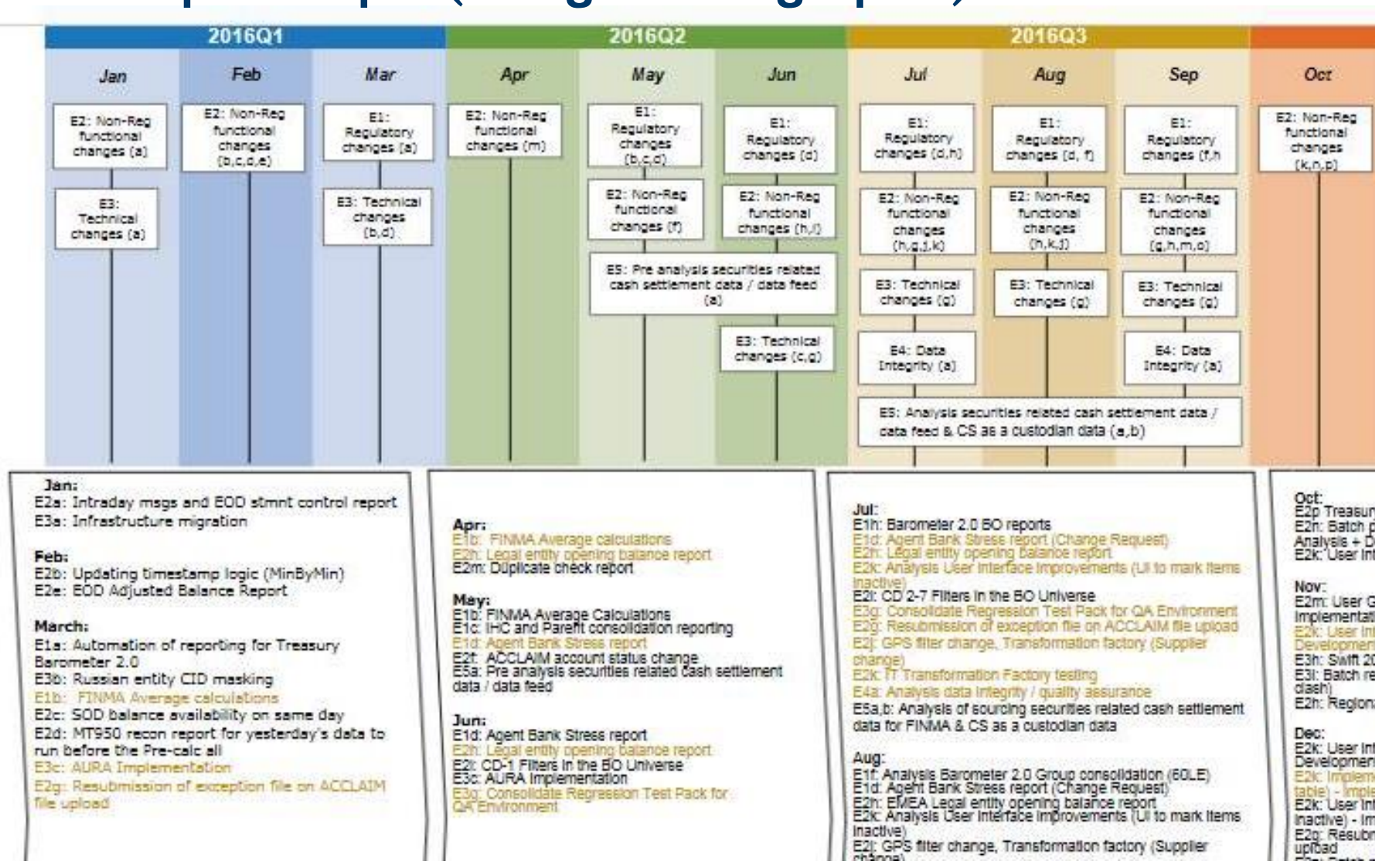
The Product Owner owns the Product Backlog

- Prioritizes Stories in the Backlog
- Challenges the quality of the Product Backlog, so the team is able to work on the User Stories
- Writes well-formulated Stories
- Defines the Acceptance Criteria
- Right level of detailing required

Product Backlog



Roadmap example (using Backlog Epics)



Stories vs. Requirements

Product Backlog Items

Represent an idea

Promise a common conversation

Card - Conversation - Confirmation

Requirements Specification

Document a physical need

Specify the necessary attributes of a system

Are incrementally developed

ID: G14589_458

As a [Some role]

I want to [Something]

so that [Some benefit]

REQ: YX 5869

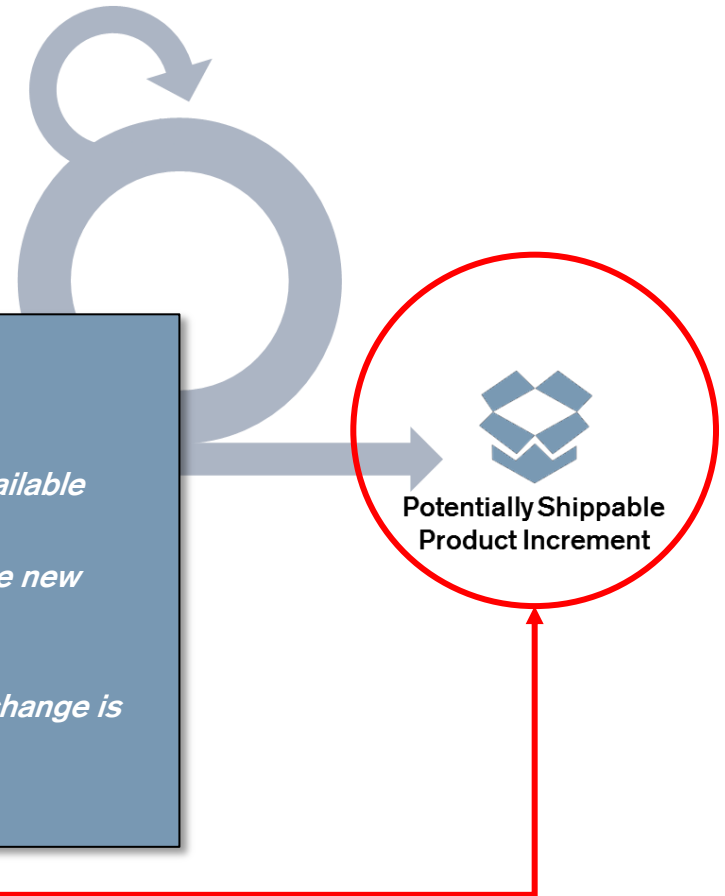
Acceptance Criteria (Set of high-level conditions)

Story

*As a risk analyst,
I want to be able to rearrange the grids in
my dashboard
so that I can view the most relevant
information first.*

Acceptance Criteria

- *A mechanism to order (rearrange) grids on the dashboard is available*
- *After confirming the changes to the dashboard arrangement, the new changes persist and are visible*
- *After rejecting the changes to the dashboard arrangement, no change is applied and the dashboard arrangement remains as earlier*



Non Functional Requirements

Product Backlog Item

*As a risk analyst,
I **want to** be able to rearrange the grids in
my dashboard
so that I can view the most relevant
information first.*

Acceptance Criteria

- ...

Non-Functional Requirements

*This functionality should be available only to
to users marked as advanced users*

INVEST – How to write “good” Product Backlog Items

Independent

- Avoid dependencies on other stories, sufficiently
- Scope the story across the architecture as a whole, not by chunks

Negotiable

- Stories need not be pledged to a particular Sprint
- Use story points and business value as a guideline to negotiate

Valuable

- Shows a working value to customers, in a demo
- Reassures each story is building the product, incrementally

Estimable

- Sufficient detail should be ready, to estimate the work required
- Stories should be small enough to be estimated

Sized right

- Stories should be sized down enough, to be implementable in the appropriate time-box
- Can be large-sized to start with (epic level), but sized-down later, before putting it in an Sprint

Testable

- Acceptance criteria should be ready and available
- Testable using the available and agreed test environment and, ideally, automated to repeat it on a frequent basis

Exercise



Write Features (High-Level Stories) for the Backlog

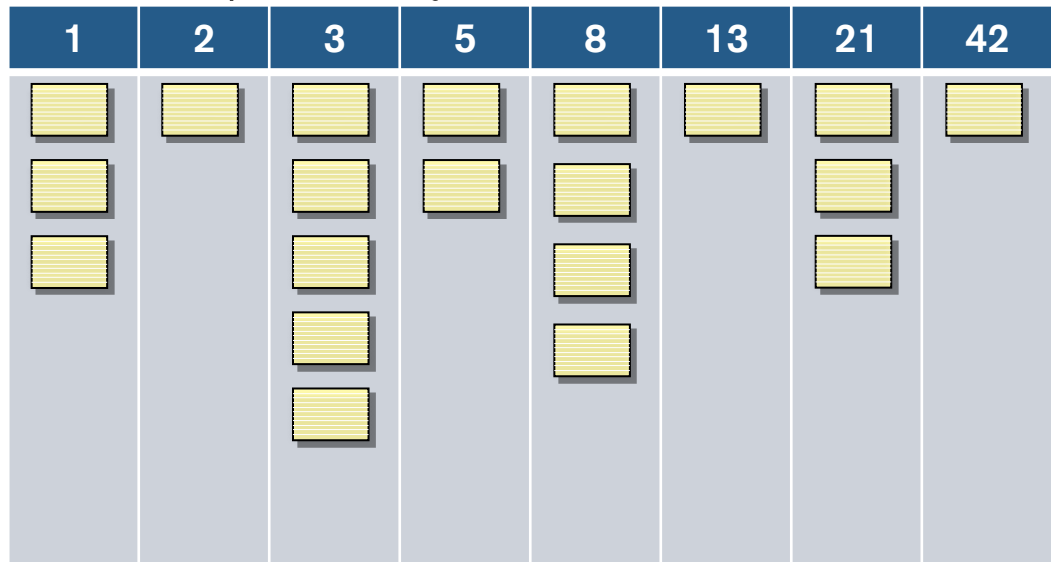
Instructions:

- Split in teams of 4-5 people
- Brainstorm a number of stories to that meet your Product Vision
- Recall how Stories are written
- Remember to include Acceptance Criteria
- 20 minutes time

*As a [Some role]
I want to [Something]
so that [Some benefit]*

Approaches to Estimation

- Estimates are made by a Group not an Individual
- Use a consistent relative scale, usually Story Points
 - Singular number that represents how big a work item is
 - Not connected to any specific unit of measure
 - 8-points stories should be, **on average**, 4x longer than 2-points Stories
- Be sure to always anchor new items to previously sized items



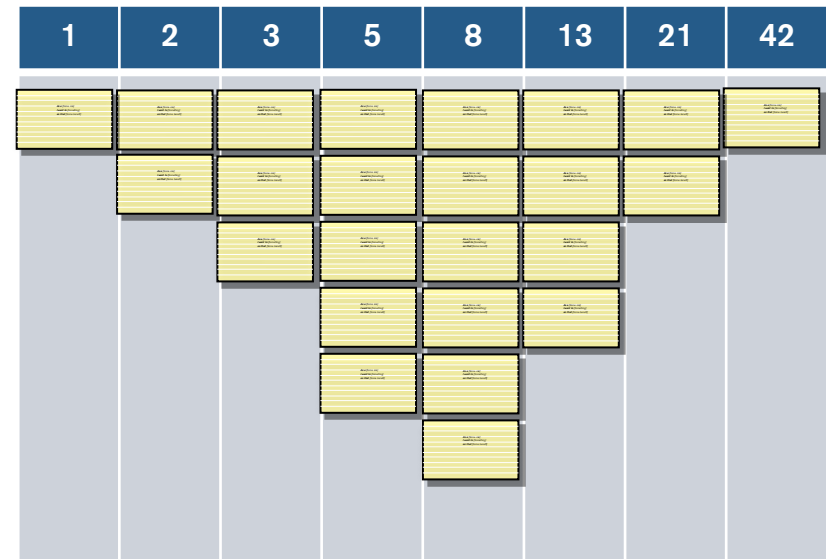
Exercise



Estimation of Features for the Backlog (Roadmap)

Instructions:

- Split in teams of 4-5 people
- Relative Sizing of Features (High-Level Stories) using the Grouping Estimation Technique
- 20 minutes time



Items Ordering Parameters

- Ordering should be based on different parameters
 - Return / Value of the work item (including risks reduction and learning)
 - Cost of implementation

$$ROI = \frac{Value}{Cost}$$

- Cost of implementation is estimated relatively using planning poker or grouping estimation technique
- Return / Value of the work item can be:
 - Monetary
 - Relative
 - MoSCoW
 - Kano
 - Hundred dollars

Relative Value Estimation – MoSCoW

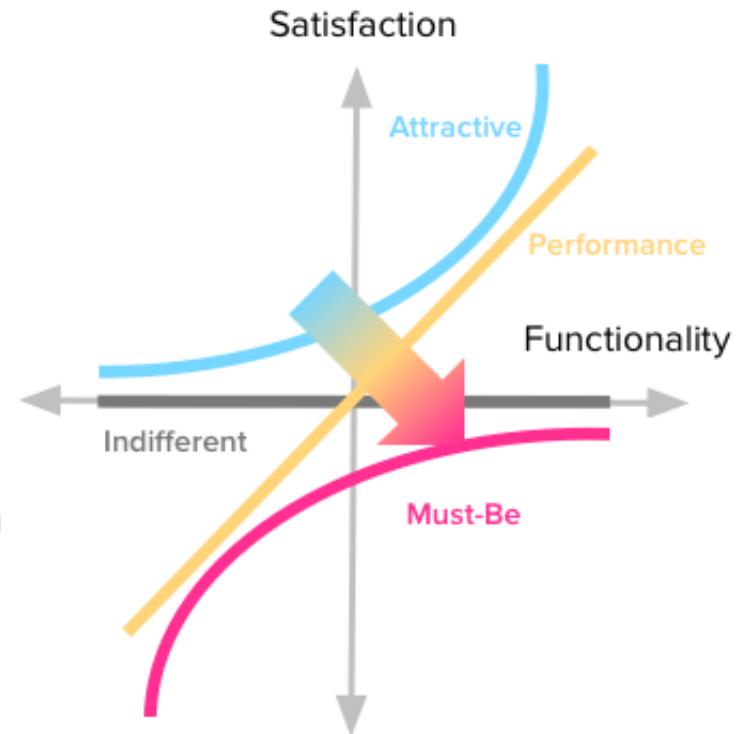
Using usual words to define the most valuable items

Feature	Must	Should	Could	Won't	Consensus
Feature 1	6	1			M
Feature 2	1	5	1		S
Feature 3	2	2	2	1	?
Feature 4			1	6	W
Feature 5	7				M

Relative Value Estimation – Kano Model (1)

Functionalities are classified in 4 categories

- Performance
 - The more the better
 - Example: Number of reports automated*
- Must-Be
 - If it is missing, the product is not valuable
 - Example: Risk grid by division*
- Attractive
 - Not expected but causing a positive reaction
 - Example: Ability to see reports on mobile*
- Indifferent
 - Presence or absence doesn't make a difference
 - Example: Ability to export grid to PDF*



Relative Value Estimation – Kano Model (2)

Evaluate each functionality on 2 dimensions

- Functional
 - How do you feel if you had / did not have this feature?
- Dysfunctional
 - How do you feel if you did not have this feature?

2 new categories emerge

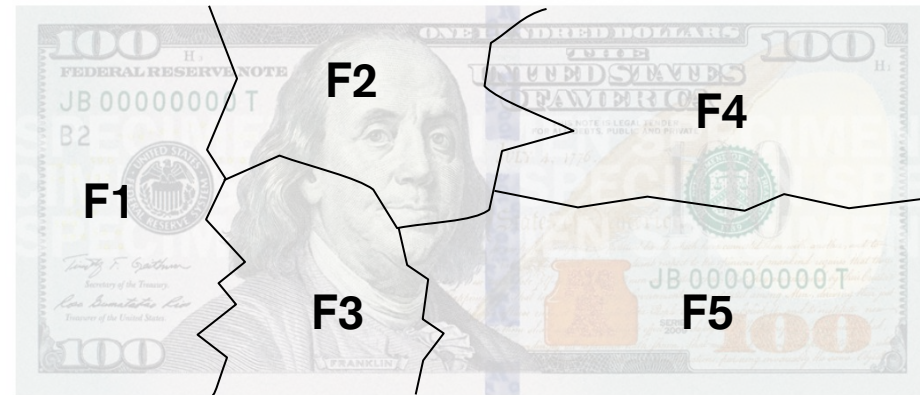
- Reverse
 - What is proposed is the opposite of what is wanted
- Question
 - The questions or the feature is not understood

		Dysfunctional (feature absent)				
		Like it	Expect it	Don't Care	Live With	Dislike
Functional (feature present)	Like it	Q	A	A	A	P
	Expect it	R	I	I	I	M
	Don't Care	R	I	I	I	M
	Live With	R	I	I	I	M
	Dislike	R	R	R	R	Q

Relative Value Estimation – Hundred Dollars

This technique is used to estimate the business value relatively

- Assign one hundred dollars to your entire functionality set
- Estimate and agree on how much value each functionality is providing out of the hundred dollars



Ordering Considerations

Different approaches to order the Product Backlog

- The Product Backlog Items are ordered, not bucketed
- The ordering is reviewed regularly
- The process based in value and cost, and is transparent
- Dates are calculated based on velocity and ordering, not priority

Product Backlog

1	Story 1
2	Story 2
3	Story 3
4	Story 4
5	Story 5
6	Story 6
7	Epic 4
8	Epic 5
9	Epic 6
10	Feature 5
11	Epic 7
12	Epic 8

Exercise

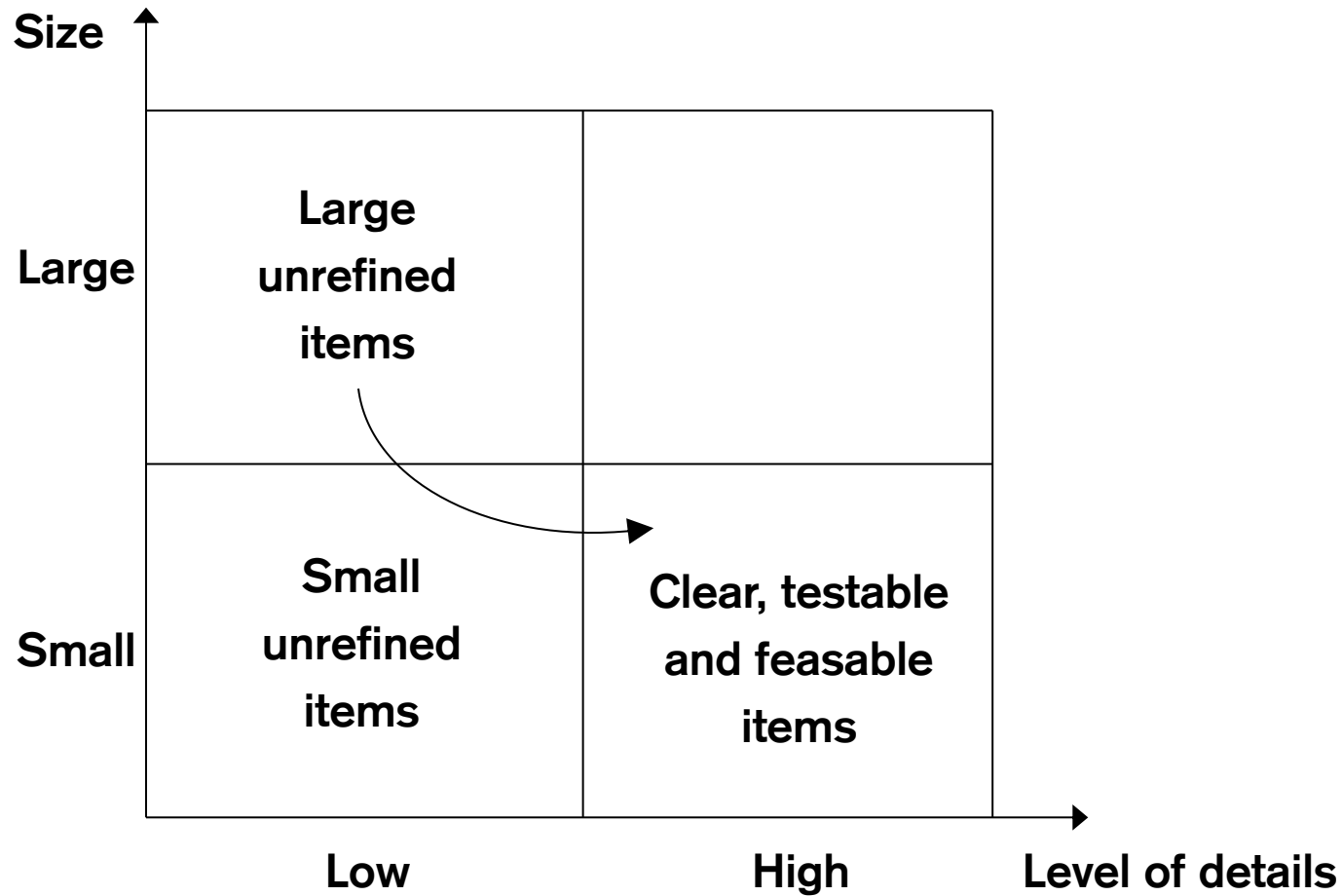


Order Items and Plan Release (Roadmap)

Instructions:

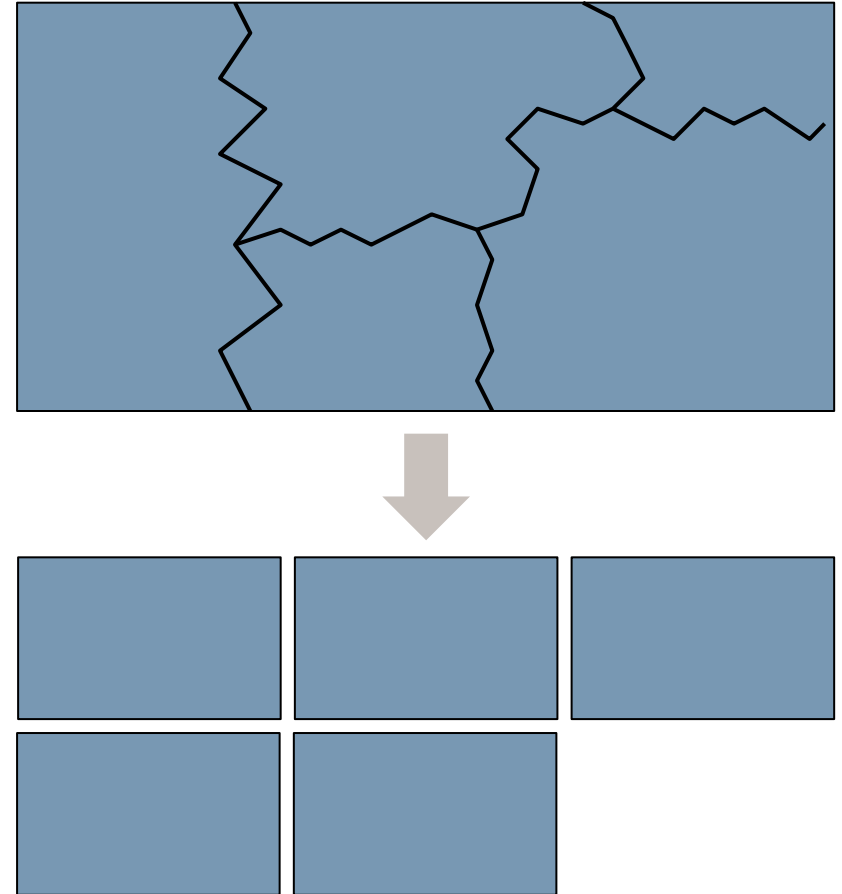
- Split in teams of 4-5 people
- Prioritize Features
- Agree on a size for a release (What is the capacity of a release)
- Create Release Plan (Roadmap)
- 20 minutes time

Story Decomposition



Splitting User Stories

- Separate by business functionality
- Split considering priorities
- Story thinning (reduce content)
- Split based on natural data boundaries
- Split using operational boundaries
- Split according to architectural interfaces
- Split according to acceptance criteria (formulate acceptance criteria as own user story)

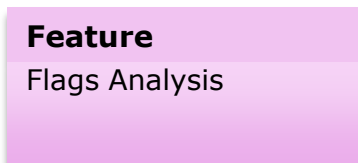
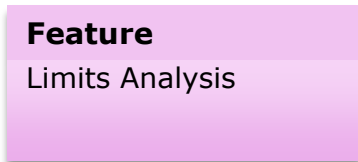


Story Breakdown with Story Mapping



In scope for this release

Out of scope for this release



Exercise



Write User Stories

Write User Stories (Low-Level Stories) for Sprint Backlog

Instructions:

- Split in teams of 4-5 people
- Take Features from Release 1 and create the User Stories as a group
- Recall how User Stories are written
- Stories are executable for a team in a 2 weeks Sprint
- 20 minutes time

***As a** [Some role]
I want to [Something]
so that [Some benefit]*

Sprint Activities

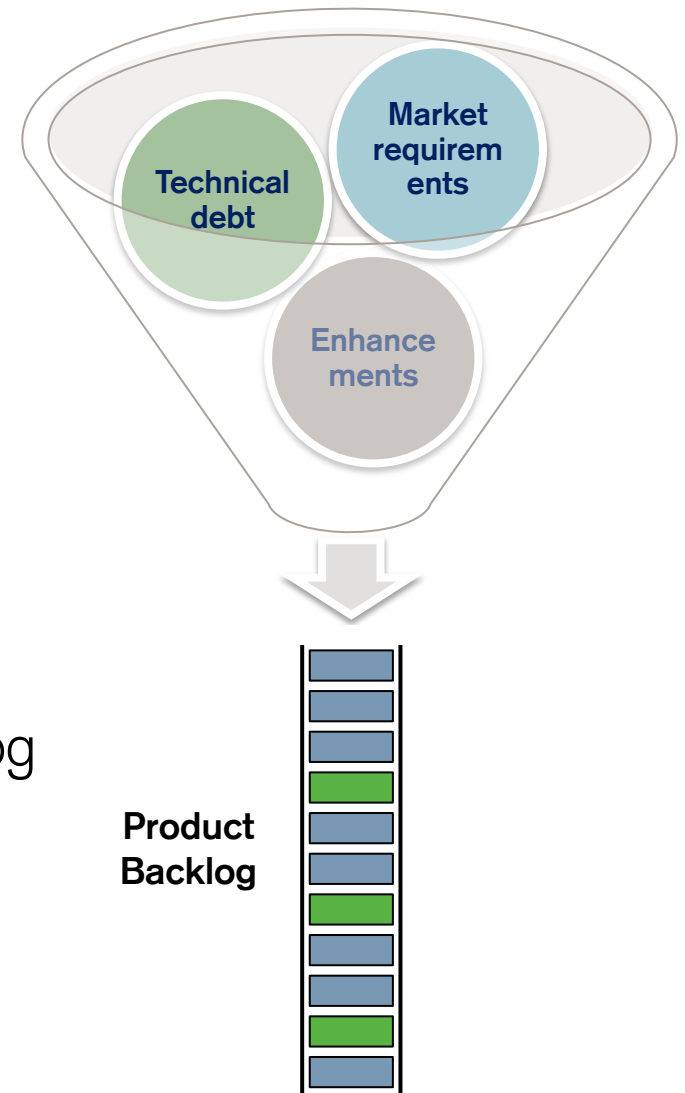
Ongoing activities during Sprint

- Clarify details on Sprint Stories
- Stays up to date on Team progress
- Provide feedback on already completed Stories
- Work with Customers & Stakeholders on what has been achieved and next set of priorities
- Keep refining the Product Backlog
- Keep priorities up-to-date



Backlog Refinement

- Backlog refinement is a continuous process
- Inputs comes from
 - Market changes
 - Competitor features
 - Customers
 - Stakeholders
 - Development team
- PO needs to make sure that sufficient backlog is refined for the next 2-3 sprints
- Balance between too many and too granular items vs. too many unrefined items



Backlog Refinement Meeting

- The Product Owner discusses the rationale behind prioritization of Epics & user stories
- Explains user stories, come to common understanding and “condition of satisfaction” with the team
- Makes a trade-of between focusing the discussion on the highest value stories and minimizing context switch
- Product Owner is not sizing the User stories – the Team does



Updating Story Map and Backlog

Currently planned for

Sprint 1

Sprint 2

Sprint 3

Feature

Risk Dashboard

Epic

Analysis by
Business Org

Epic

Analysis by VaR

Epic

Analysis by Net
Sensitivities

Story

Select Business
Org

Story

Chart by
Business Org

Story

View Risks in
Grid

Story

Chart by Risk
Type

Story

Sortable
Columns

Story

Chart by VaR
Components

Story

Show
Sensitivities by
Risk Type

In scope for this release

Out of scope for this release

Feature

Limits Analysis

Feature

Flags Analysis

Sprint Planning

- The Product Owner proposes the goals for the sprint and identifies stories supporting these goals
- Respects team's velocity when defining the scope
- Helps the team to verify the understanding of the stories planned for the sprint
- Is available for clarification and/or re-negotiation during story breakdown into tasks



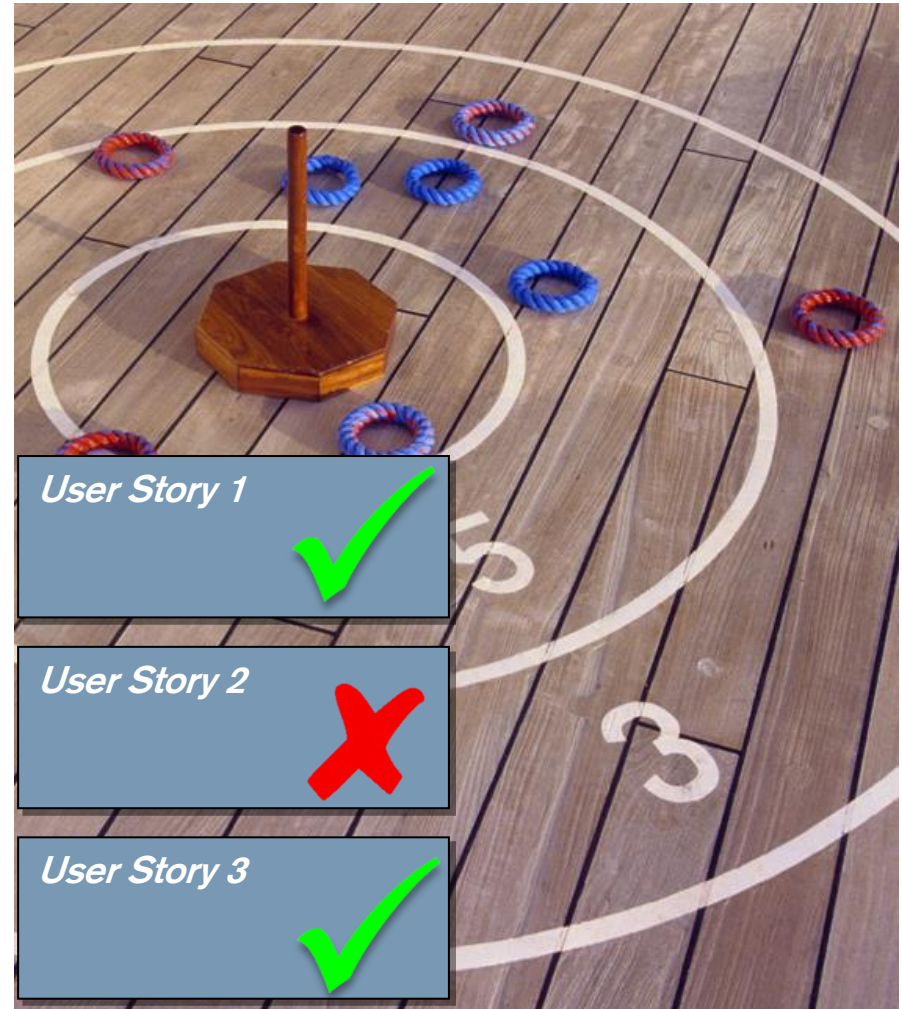
Daily Scrum

- The Product Owner attends as often as possible, goal is always
- Helps resolving issues, provides clarifications
- Understands which stories might be worth looking at
- Might channel information into and from the team



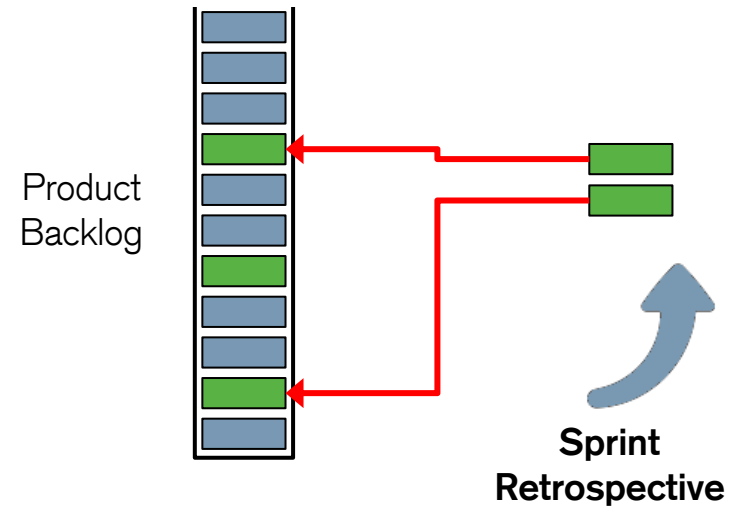
Sprint Review

- The Product Owner provides feedback on the achieved work
- Accepts the Story if the expected value for customer is delivered
- Rejects the Story if the expected value for customer is not delivered or incomplete
- Brings real users to the demos if possible



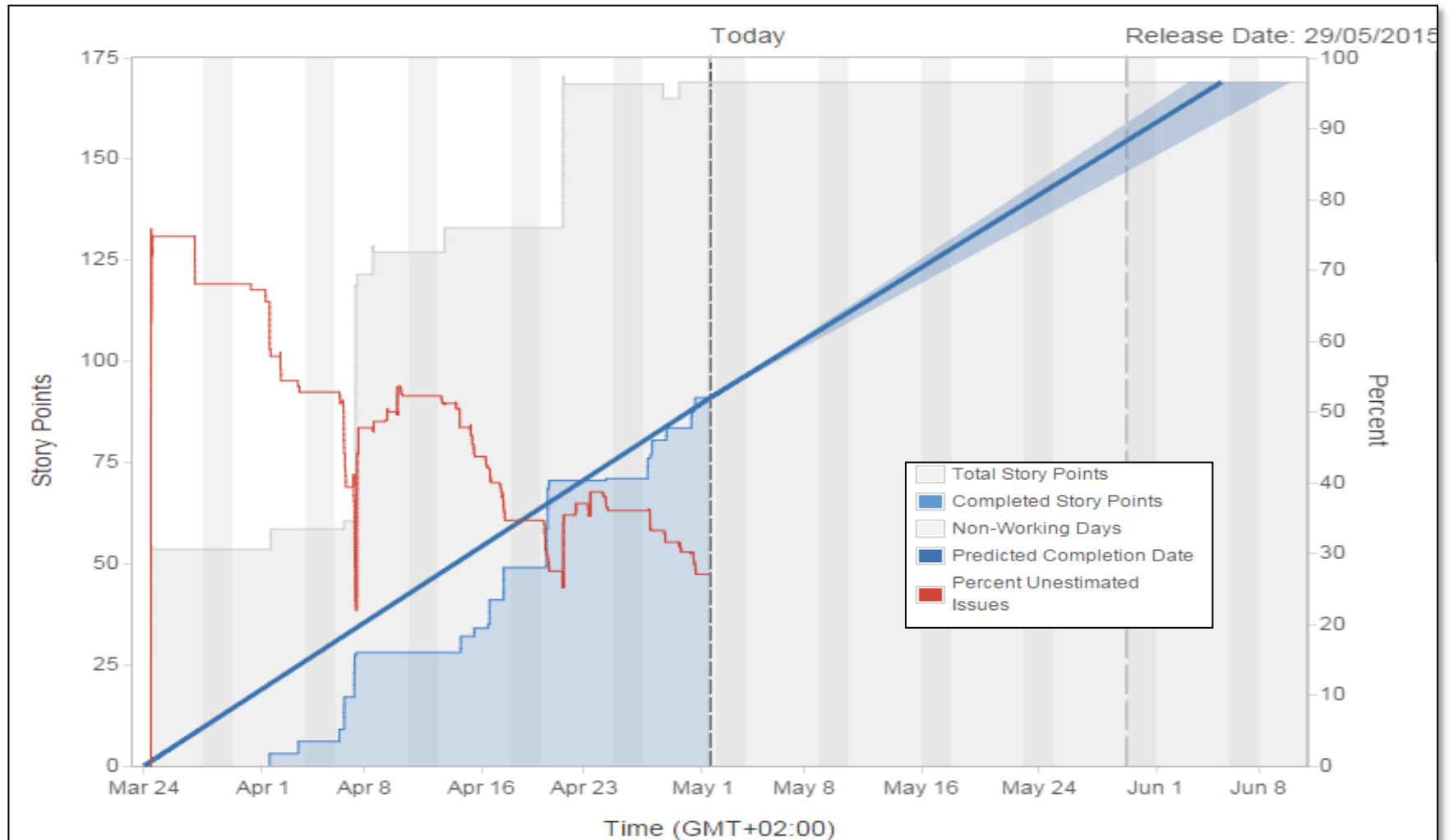
Sprint Retrospective

- The Product Owner respects the impediments of the team
- Gives the team the time and priority needed to improve
- Helps the team by escalating impediments
- Understands the value of an improving team to the product



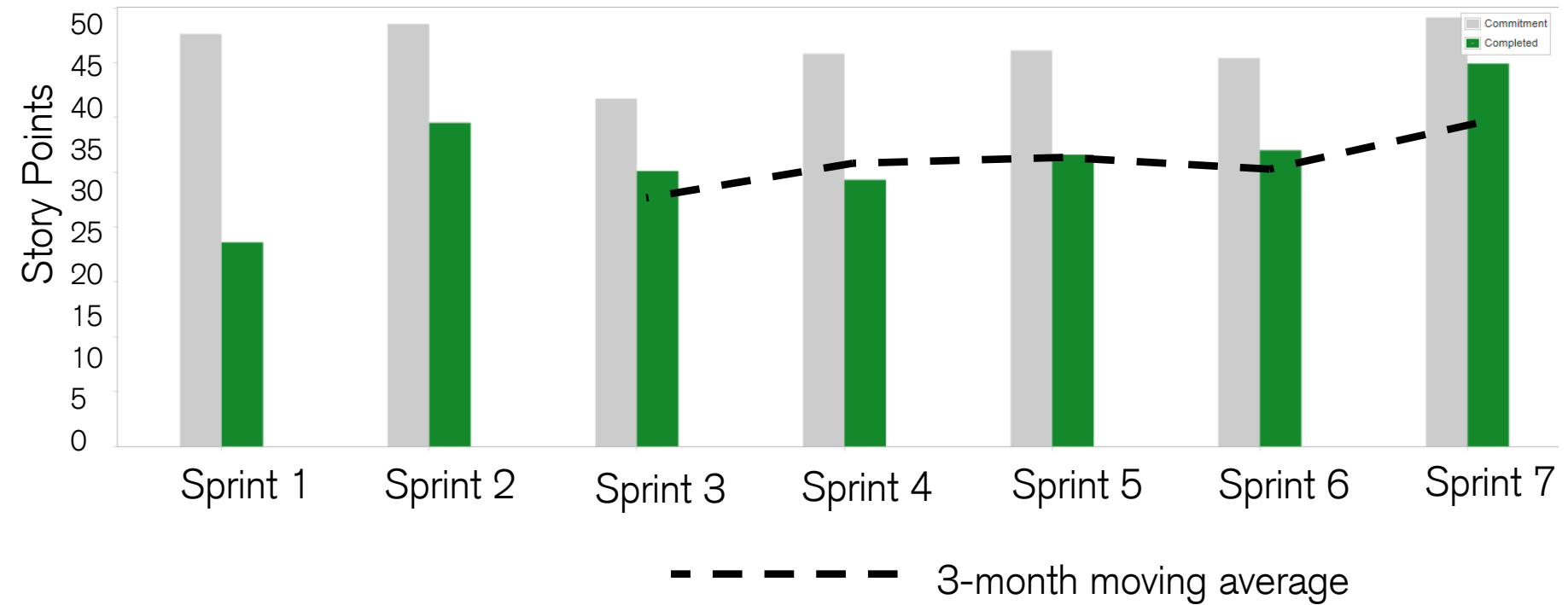
Progress Monitoring

Release Burn-up Chart (JIRA Version Report)



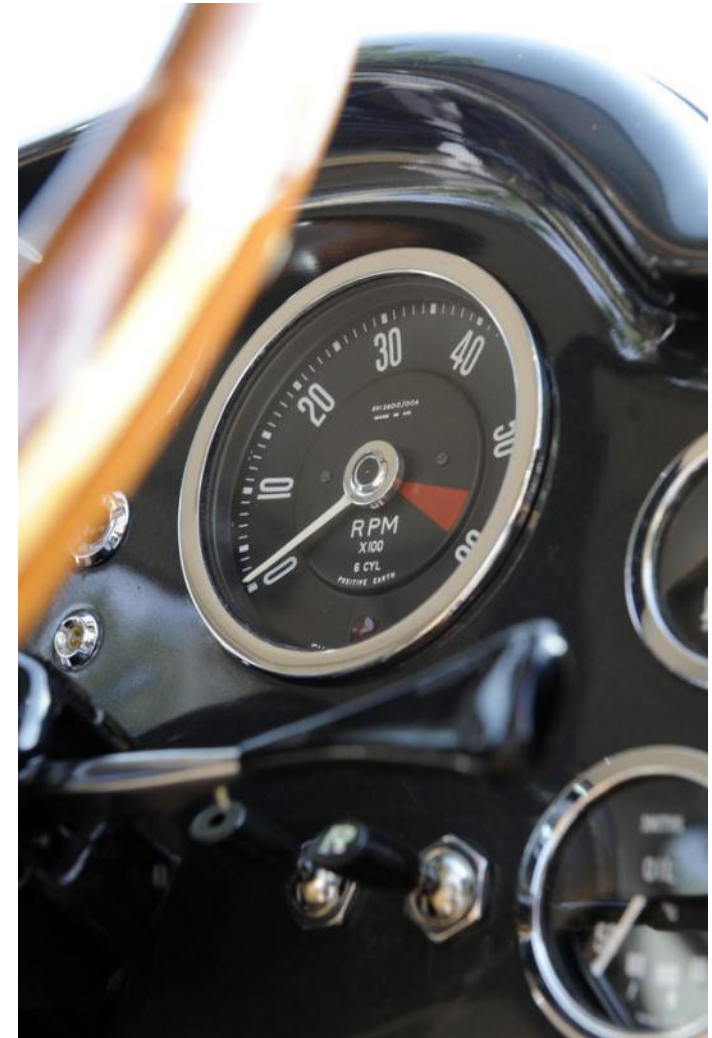
Progress Monitoring

Velocity (JIRA)



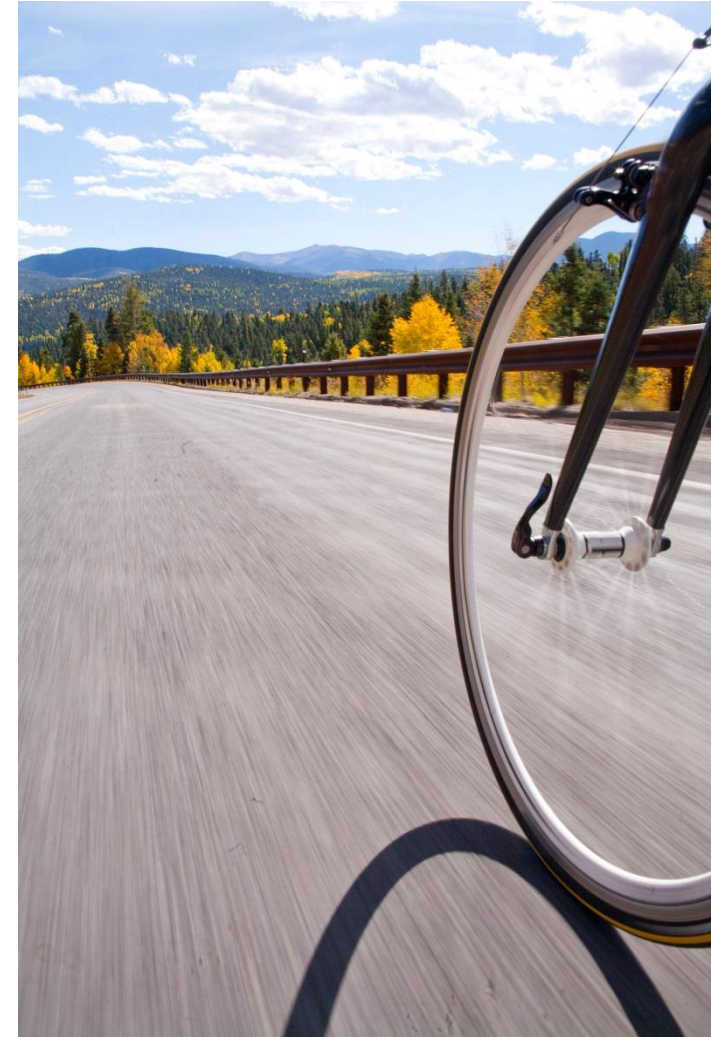
Why Is Velocity Important?

- Velocity allows us to make commitments
 - Timelines
 - Roadmaps
 - Dependencies
 - Releases
 - Partners



How do we use Velocity?

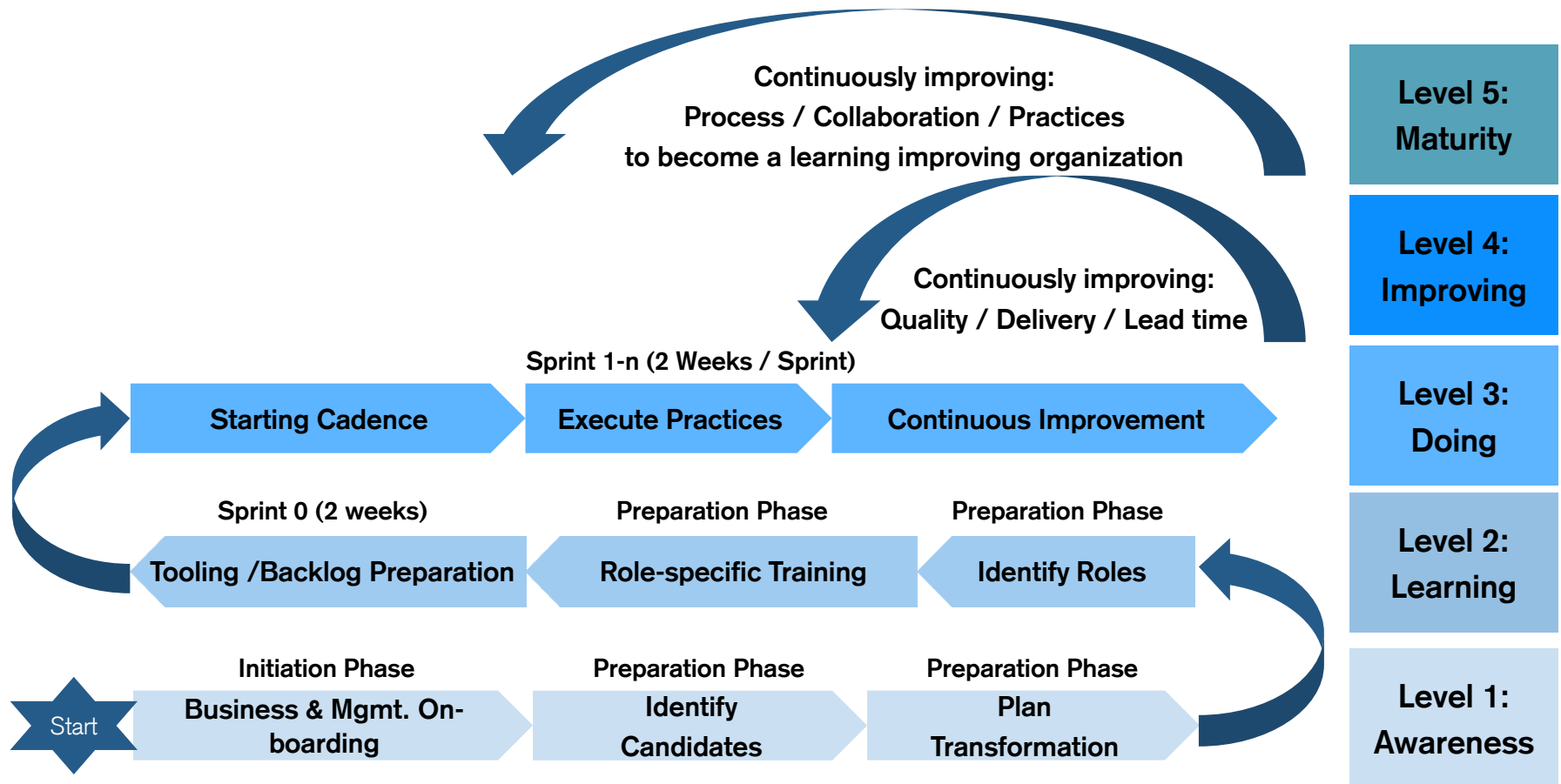
- It is based on our past performance
- It is self-adjusting
- Velocity helps to identify how much work we can plan to accomplish within a time-box
 - Match story points for Sprint backlog to the velocity
 - Split stories if necessary



Q&A

Agile Implementation

Agile Transformation Roadmap



Learning

Learn more about Agile

Agile Library Home

Created and last modified by Carlo Crinzi just a moment ago



Agile Learning Map



Agile Events



Agile Quick Start



Agile in Practice

Available on:

<https://mylink.csintra.net/AgileL>

Course Retrospective



- What went well in the training course?
 - *Content and delivery*
 - *Relevance to work and examples*
 - *Clarifications and Q&A*
- What could be improved?
- Any suggestions?

Further Information on CS-SDF



For general questions, please email the EPT Communication Team

communications.ept@credit-suisse.com

CS-SDF and EPT Intranet containing overviews

<http://cs-sdf.csintra.net/>

CS-SDF Practice Library

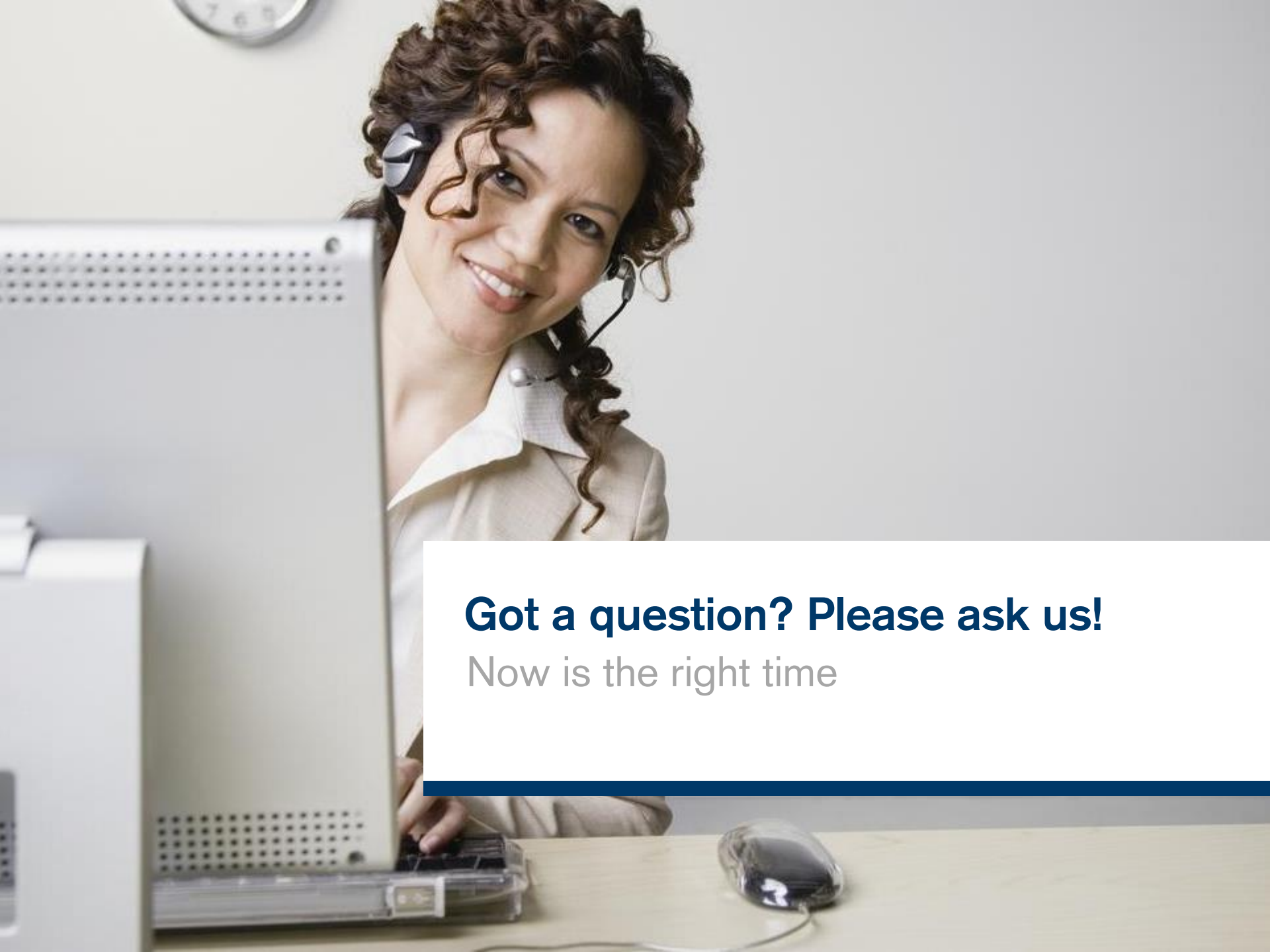
<https://practices.csintra.net/>

Odysse in BaSZ

<https://odyssey.apps.csintra.net/odysseus/>

Odyssey in DvSZ

<https://odyssey.rowini.net/odysseus/>



Got a question? Please ask us!

Now is the right time