PROFESSIONAL SCRUM PRODUCT OWNER LEVEL 2 (PSPO II) EXAM PREP

BOOTCAMP

Instructor: Barb Waters, MBA, PMP

Class will begin at 11:00 am Eastern Time

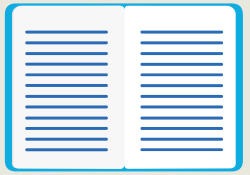
TARGET AUDIENCE

This Bootcamp is for:	This Bootcamp is:
• anyone who would like to **validate their advanced knowledge of Professional Scrum Product Ownership.**	• not for certified Product Owners unless you need a refresher
• anyone who would like to **expand their knowledge of the Scrum framework, and delivering valuable products**	• not aligned to the Project Management Institute's PMP or Agile certifications
• students who might be interested in pursuing the **PSPO II exam**. Also, students who would like to **combine PSPO I and PSPO II learning to pursue the PSPO II exam**.	• not limited to PSPO I and PSPO II candidates. Everyone is welcome!

PSPO II EXAM DETAILS

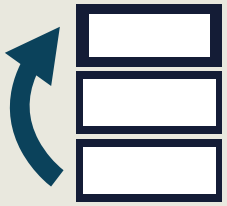
$250 per attempt

Multiple choice
Multiple answer, (partial credit)



Recommended reading
Scrum.org

60 minutes



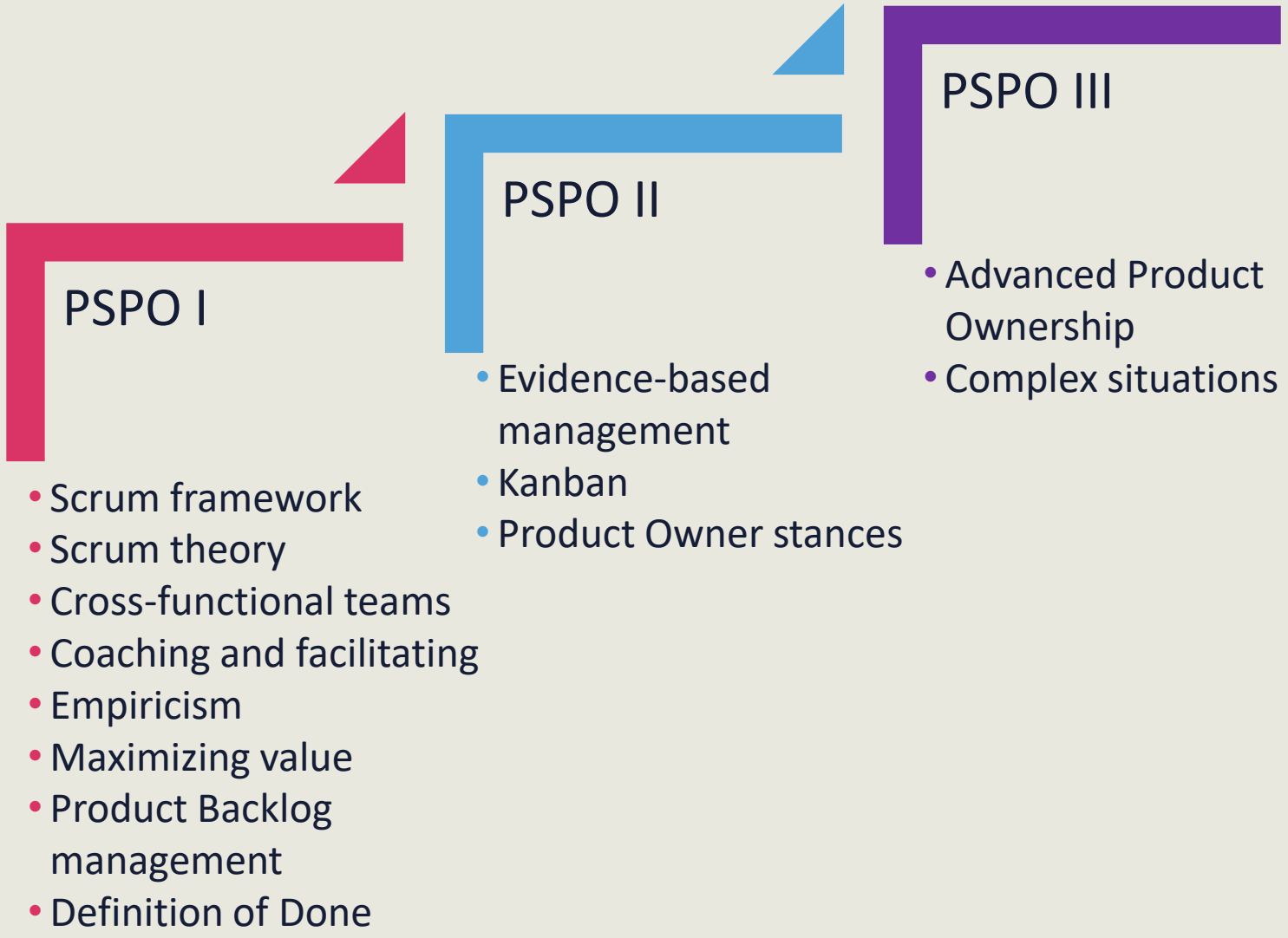
No prerequisites
PSPO I training
recommended

40 questions

Passing score is 85%

PROFESSIONAL SCRUM PRODUCT OWNER JOURNEY

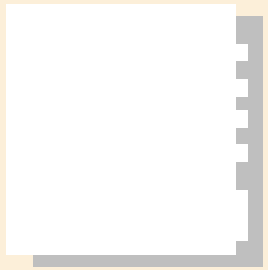
Progression of certifications



COMPARISON OF PSPO I AND PSPO II CURICULUM

TOPIC	PSPO I	PSPO II
Scrum framework	✓	
Scrum theory and principles	✓	
Cross-functional teams	✓	
Coaching and facilitation	✓	
Empiricism	✓	
Maximizing value	✓	
Product backlog management	✓	
Definition of Done (DoD)	✓	
Evidence-based management		✓
Kanban		✓
Product Owner stances		✓

INDEPENDENT STUDY



Articles

Videos

Podcasts

Practice

https://www.scrum.org/pathway/product-owner-learning-path

https://www.scrum.org/resources/suggested-reading-professional-scrum-product-owner-II

SCRUM GUIDE

BY KEN SCHWABER & JEFF SUTHERLAND

Ken Schwaber & Jeff Sutherland

The Scrum Guide

The Definitive Guide to Scrum: The Rules of the Game

November 2020

https://www.scrumguides.org/

THE SCRUM TEAM

Includes:

- Developers
- Scrum Master
- Product Owner

No Scrum Team member is in charge.

The team is self-managing.

The team ensures that their tasks for each Sprint are completed.

The typical size for a Scrum Team is "10 or fewer"

Developers

Scrum Master

Product Owner

EXAM TIP

THERE ARE NO OTHER JOB TITLES

The Scrum Team roles do not flex into other job titles. It is the other way around. For example, a Project Manager may serve on a Scrum Team, but they will need to adjust the way they work to align with Scrum.

THE KANBAN GUIDE FOR SCRUM TEAMS

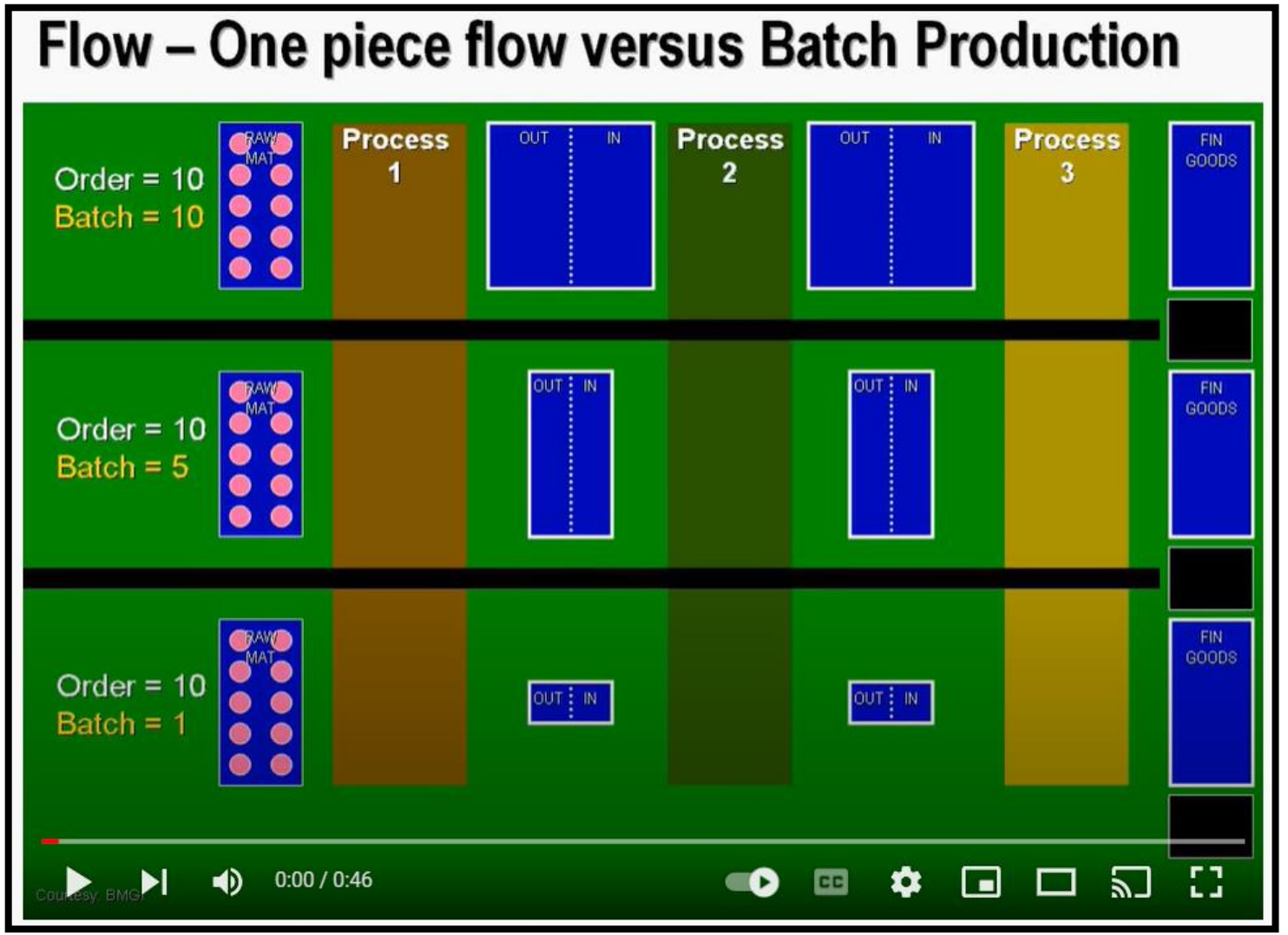
January 2021

Scrum.org
Developed and sustained by Scrum.org.,
Daniel Vacanti, and Yuval Yeret

THE KANBAN GUIDE FOR SCRUM TEAMS

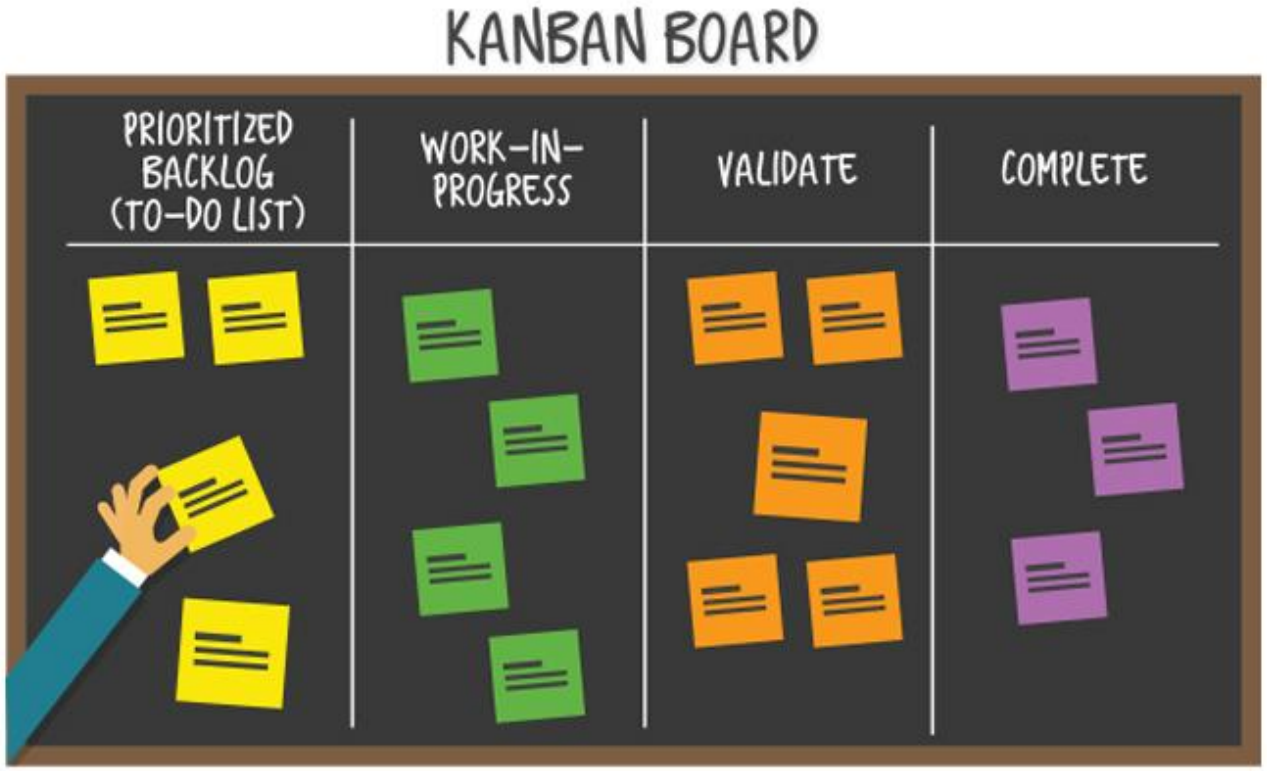
"This guide does not replace or discount any part of the Scrum Guide. It is designed to enhance and expand the practices of Scrum."

FLOW AND EMPIRICISM



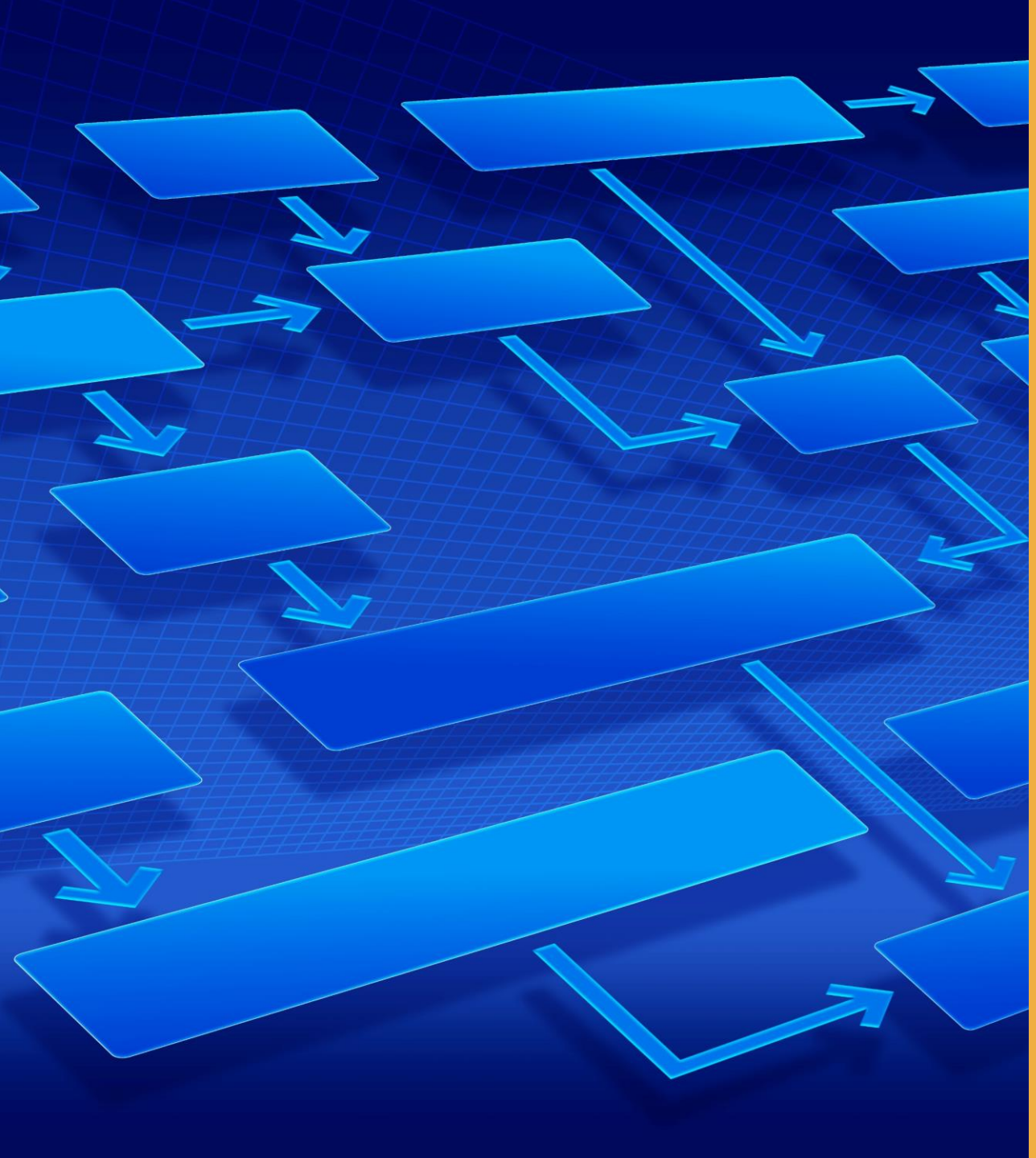
FOUR BASIC METRICS OF FLOW

1) **Work in Progress (WIP):** The number of work items started but not finished.
2) **Cycle Time:** The amount of elapsed time between when a work item starts and when a work item finishes.
3) **Work Item Age:** The amount of time between when a work item started and the current time. This applies only to items that are still in progress.
4) **Throughput:** The number of work items finished per unit of time.



FOUR PRACTICES TO OPTIMIZE FLOW

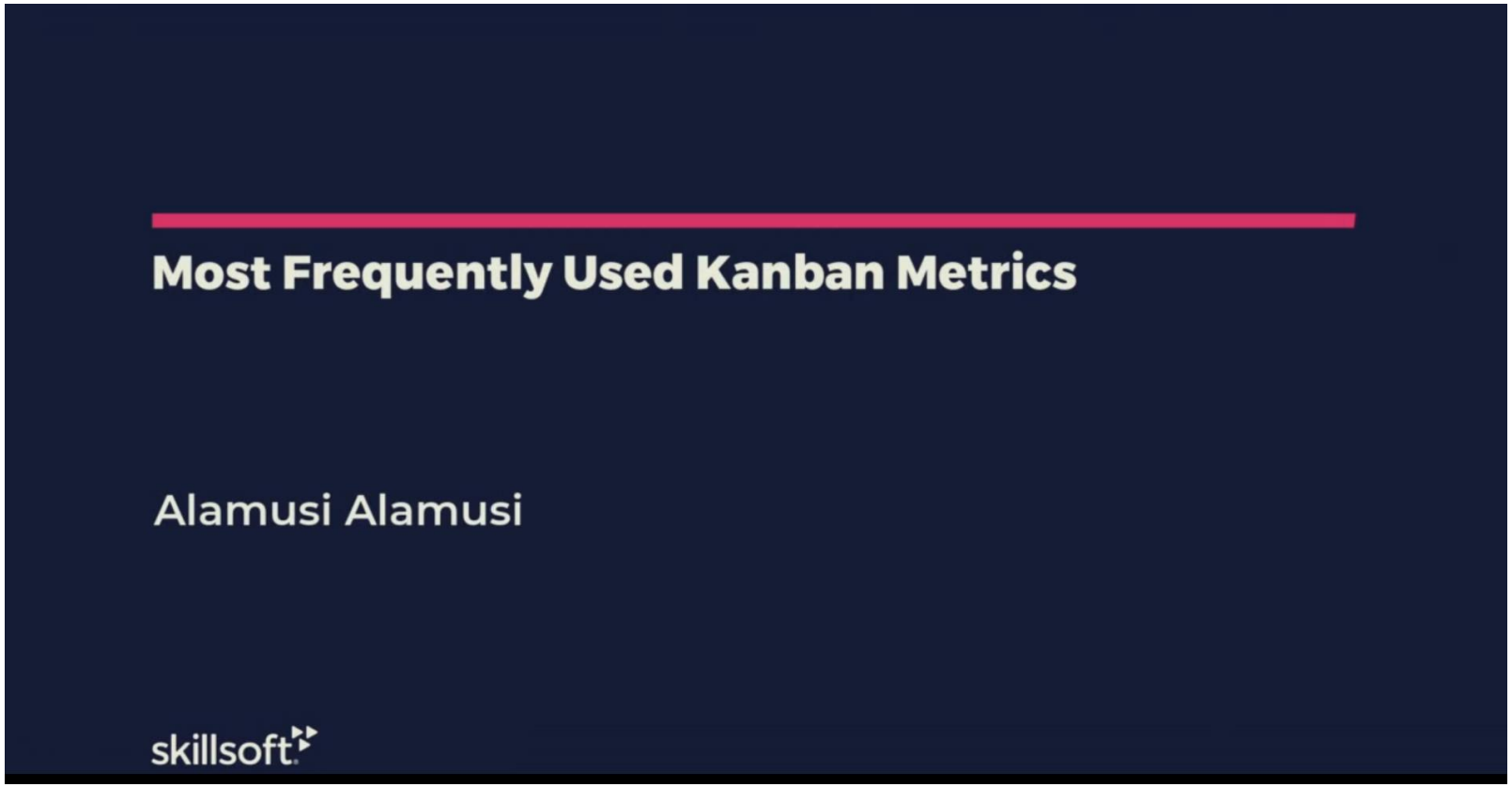
1) Visualization of the Workflow
2) Limiting Work in Progress (WIP)
3) Active management of work items in progress
4) Inspecting and adapting the team's Definition of Workflow



DEFINITION OF WORKFLOW

The **Scrum Team** members' explicit understanding of what their policies are for following the Kanban practices. This shared understanding improves transparency and enables self-management.

KANBAN METRICS

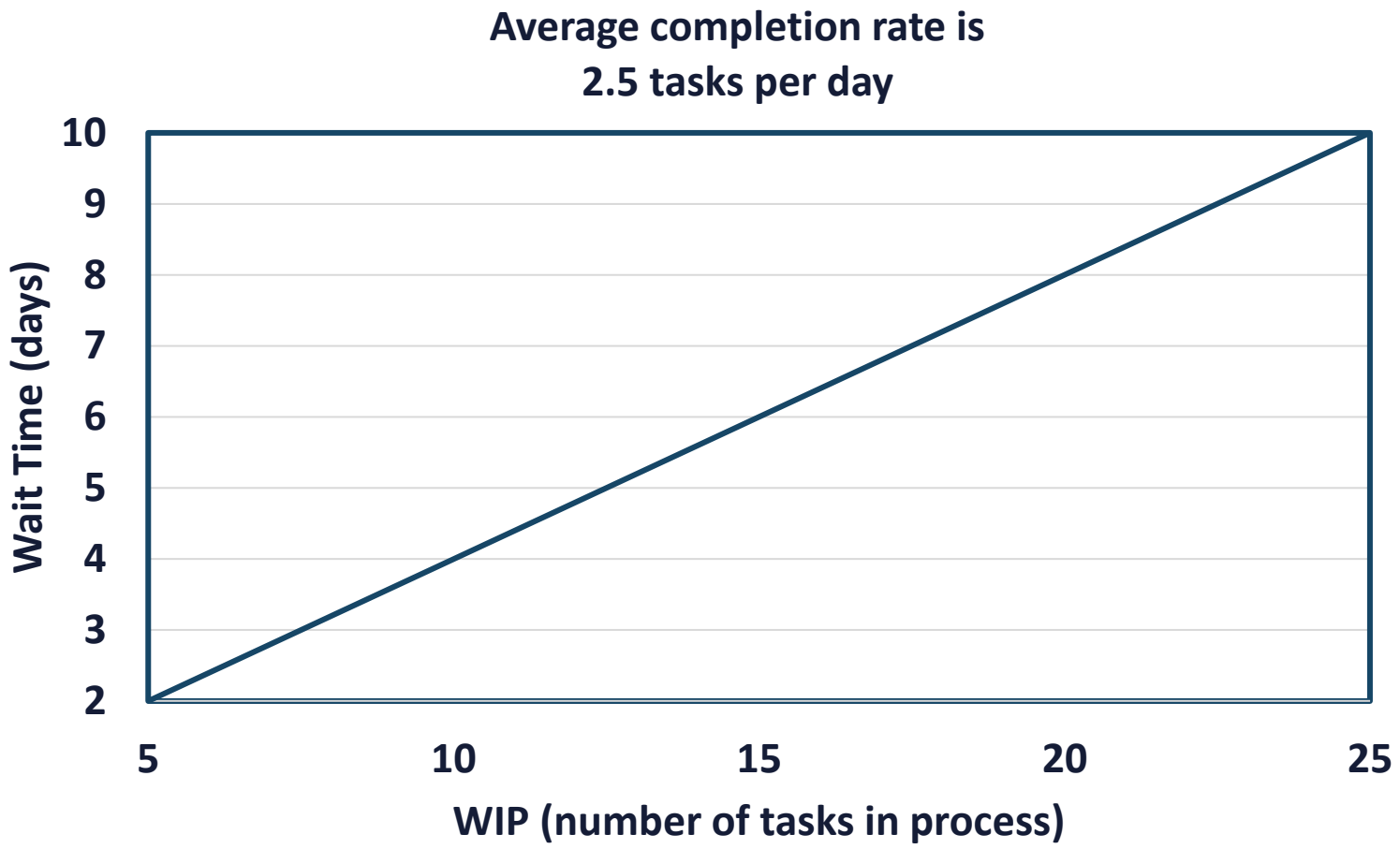


LITTLE'S LAW

- Inventory (WIP)
- Throughput
- Wait time

By changing one, we can change the others.

LITTLE'S LAW


Average completion rate is
2.5 tasks per day
Wait Time (days)
10
9
8
7
6
5
4
3
2
5
10
15
20
25
WIP (number of tasks in process)

LITTLE'S LAW

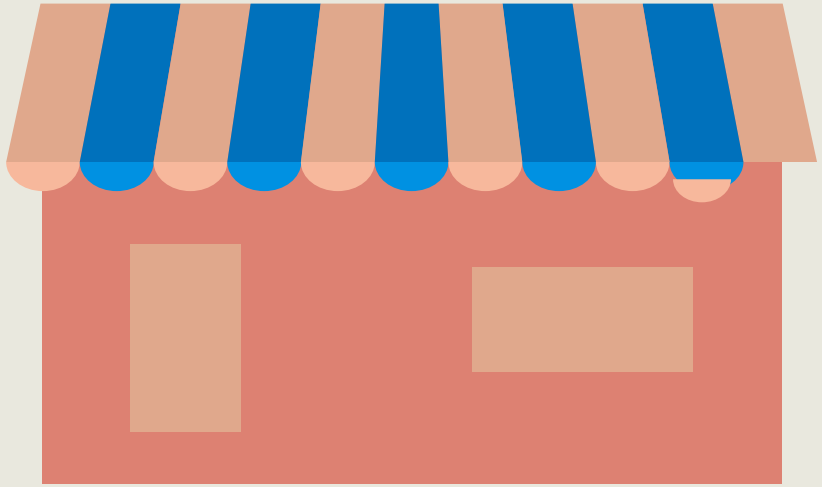
Formula	Example
$\frac{\text{Inventory (WIP)}}{\text{Throughput (Speed)}} = \text{Wait Time}$	$\frac{\text{10 tasks in progress}}{\text{2.5 tasks per day}} = \text{4 days lead time}$

LITTLE'S LAW

$$\frac{\text{Inventory (WIP)}}{\text{Throughput (Speed)}} = \text{Wait Time}$$

Reduce the Queue (WIP)
10 customers in line/2 customers per minute = 5 minutes
5 customers in line/2 customers per minute = 2.5 minutes

Increase the Throughput (Speed)
10 customers in line/2 customers per minute = 5 minutes
10 customers in line/5 customers per minute = 2 minutes

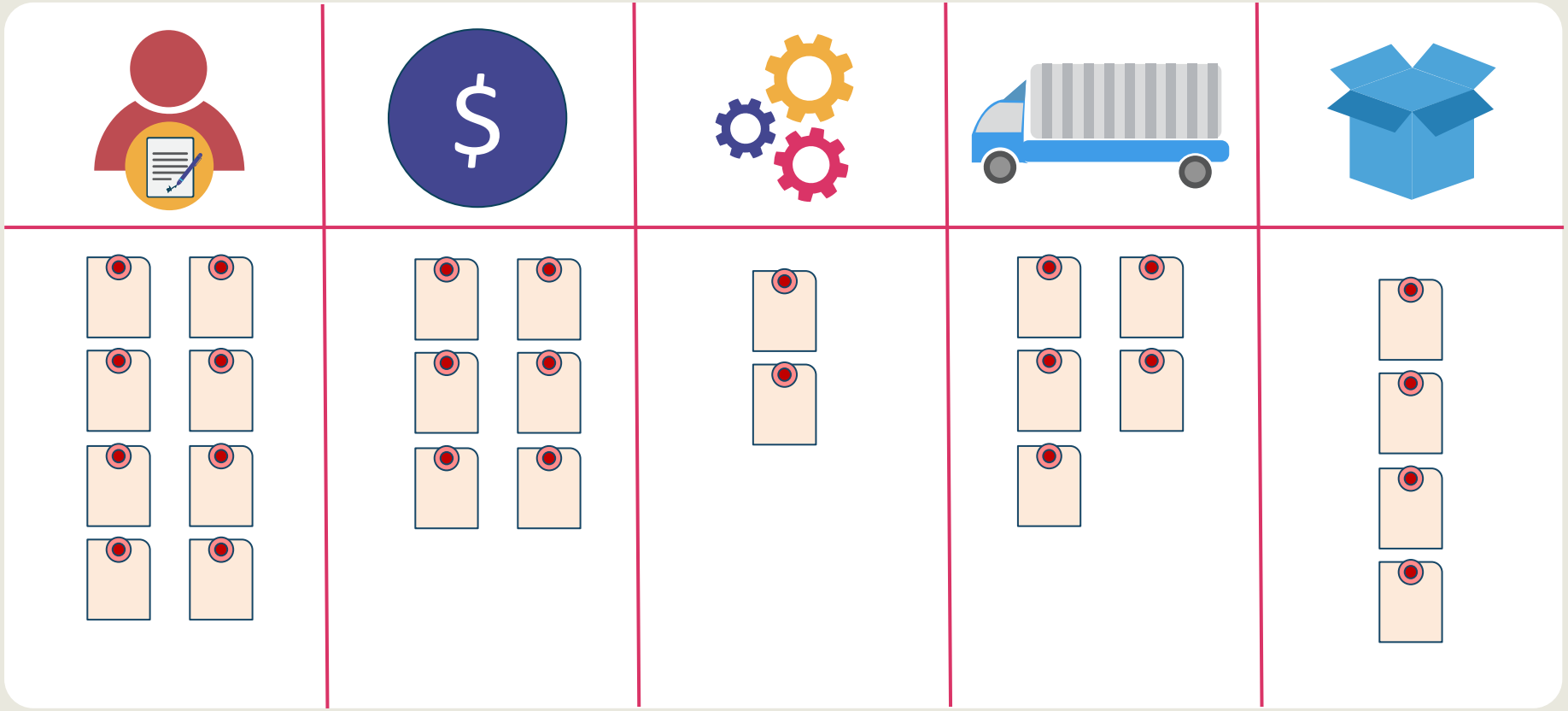


IMPROVING WAIT TIME

Reduce the queue (WIP)	Increase the throughput (Speed)
10 customers in line/2 customers per minute = 5 minutes	10 customers in line/2 customers per minute = 5 minutes
5 customers in line/2 customers per minute = 2.5 minutes	10 customers in line/5 customers per minute = 2 minutes

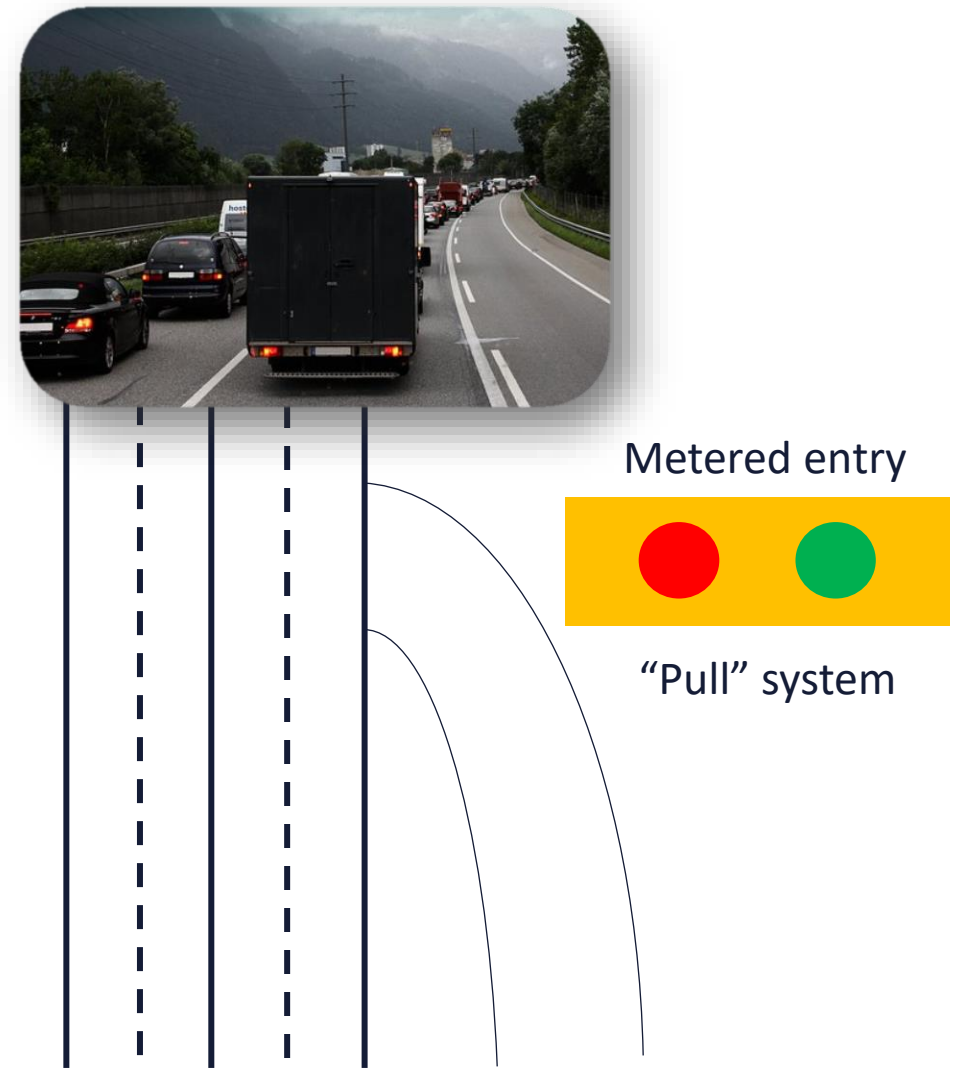
KANBAN BOARD

- Visualization of the Workflow
- Defined points at which the Scrum team considers the work to have started and to have finished
- Policies for limiting Work in Progress (WIP)



WHY LIMIT WORK IN PROGRESS (WIP)?

- Reduce Inventory
- Reduce bottlenecks
- Improve rate of throughput
- Control workloads of team members
- Goals:
  - Consistently sized tasks
  - Couple of days duration each
  - Assign to skills
  - Reduce idleness
  - Protect quality of work

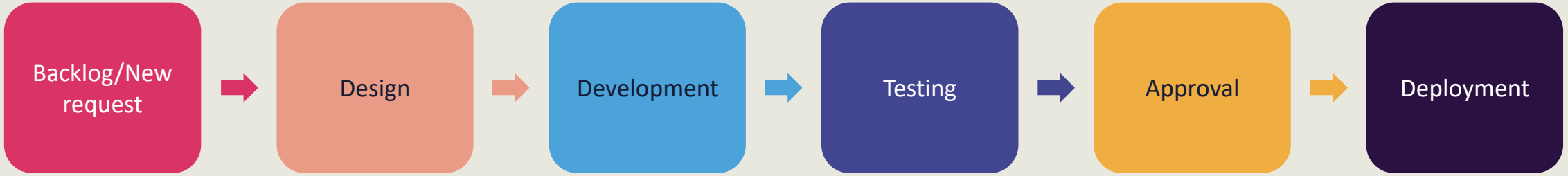


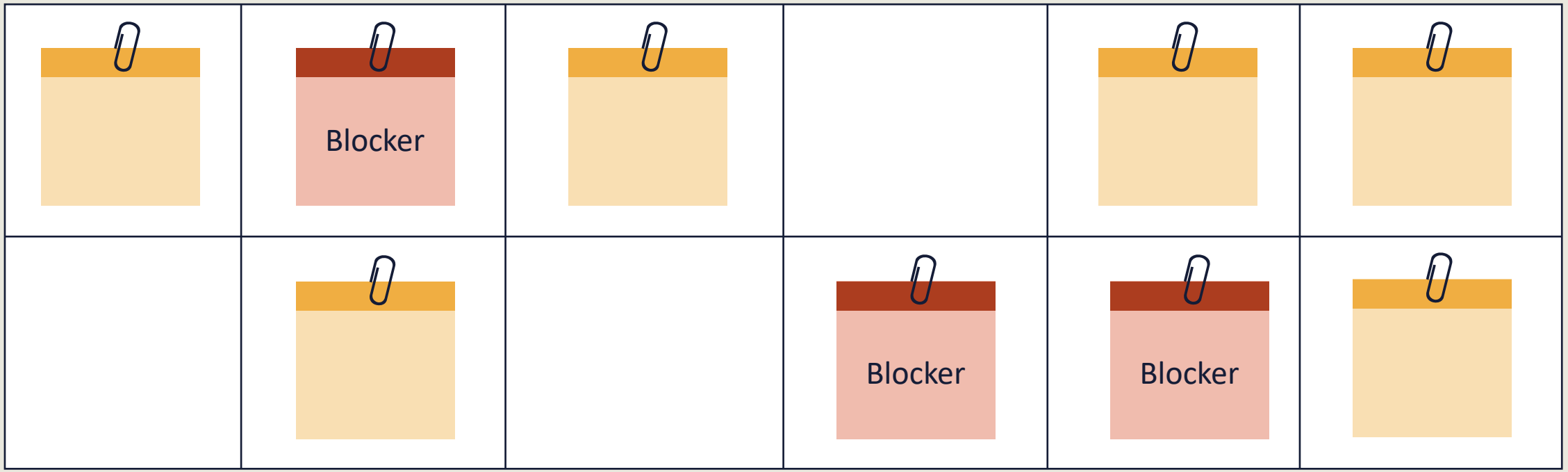
SERVICE LEVEL EXPECTATION (SLE)

- How long it should take a given item to flow from start to finish within the Scrum Team's Workflow
- Two parts
  - Range of elapsed time
  - Probability associated with that period
  - Example: 90% of items should be finished in 8 days or less

BLOCKERS AND WORK ITEM AGING

- Visualize blockers
- Determine Work Item Aging - how long ago each item was pulled into Work in Progress (WIP)





WASTE (MUDA)

Any activity or process that doesn't add value to a product but does add cost

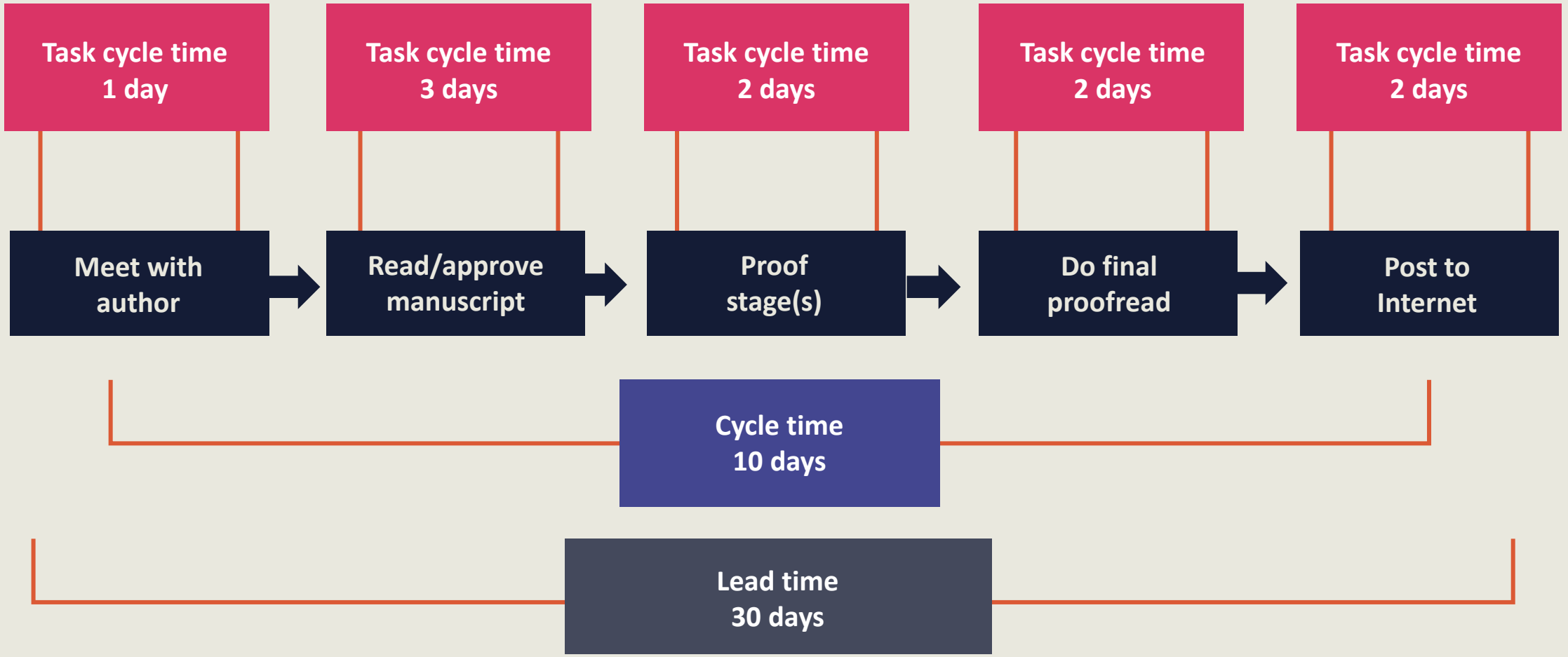
IDENTIFYING WASTE, OR MUDA

Eight Wastes

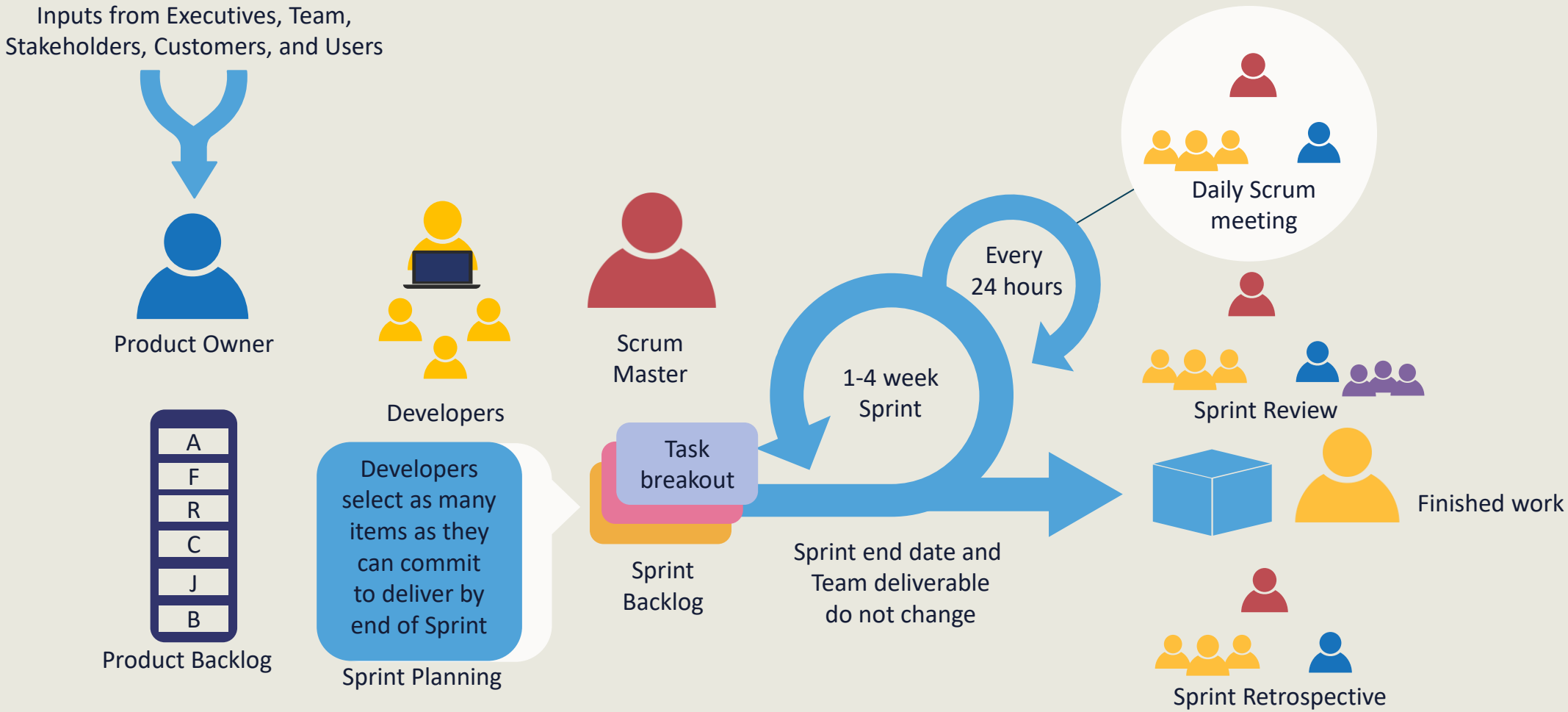
What is waste?

- Does NOT add value
- DOES add cost

VALUE STREAM MAPPING



KANBAN WITH SCRUM



"When Kanban practices are applied to Scrum, they provide a focus on improving the flow through the feedback loop."
– The Kanban Guide for Scrum Teams

THE EVIDENCE-BASED MANAGEMENT GUIDE

A Formal Approach to Measuring and Maximizing Value

Scrum.org

The Evidence-Based Management Guide

Measuring Value to Enable Improvement and Agility

September 2020

EVIDENCE-BASED MANAGEMENT

Four Key Value Areas (KVAs):

1) Current Value (CV)

What value are we delivering to the customer today?

2) Time to Market (T2M)

How quickly can we deliver a new product, service, or feature?

3) Ability to Innovate (A2I)

Are we able to deliver a new feature to better meet customer needs?

4) Unrealized Value (UV)

What additional value could we deliver if we met all of the customer needs now?

KEY VALUE MEASURES (KVMs)

- Types of measures:
  - Determine the current state
  - Identify the future state
  - Measure progress
- Not prescribed. You may select your own metrics.
- Supports empiricism.

CURRENT VALUE (CV)

KEY VALUE MEASURES (KVMs)

Customer Usage Index

- Usage by feature, to determine if the product is useful

Revenue per Employee

- Gross revenue divided by # of employees

Product Cost Ratio

- Total costs divided by the system being measured

Employee Satisfaction

- Surveys and feedback

Customer Satisfaction

- Customer engagement and happiness with the product

UNREALIZED VALUE (UV)

KEY VALUE MEASURES (KVMs)

Market Share

- Percent of the market not owned by the product

Customer or User Satisfaction Gap

- Difference between desired and actual experience

Desired Customer Experience or Satisfaction

- Measurement of the experience the customer would like to have

TIME TO MARKET (T2M)

KEY VALUE MEASURES (KVMs)

This section can get technical. If you are not comfortable with the technical descriptions, simply think of T2M as the turnaround time needed to create products and features, fix them if they are broken, and collect customer feedback.

Faster is better.

TIME TO MARKET (T2M)

KEY VALUE MEASURES (KVMs)

Build and Integration Frequency
- The number of integrated and tested builds per time period.

Release Frequency
- Number of releases per time period

Release Stabilization Period
- Time spent correcting problems between when we *think* it is releasable and when it is actually released.

Mean Time to Repair
- The average amount of time between error detection and resolution

Customer Cycle Time
- Amount of time from when work starts on a release until the actual release.

Lead Time
- Amount of time from when an idea is proposed until the customer can benefit.

Lead Time for Changes
- Amount of time to go from code-committed to code running in production

Deployment Frequency
- Number of times the organization released a new version of the product to customers/users

Time to Restore Service
- The amount of time between the start of a service outage and the restoration of full availability

Time-to-Learn
- The total time needed to design, build, and deliver and idea, and learn from its usage

Time to Remove Impediment
- Average amount of time from when an impediment is raised until it is resolved.

Time to Pivot
- Elapsed time between when an organization receives feedback or new information and when it responds.

ABILITY TO INNOVATE (A2I)

KEY VALUE MEASURES (KVMs)

Innovation Rate

- The percentage of effort or cost spent on new product capabilities, divided by total effort or cost.

Defect Trends

- Measurement of change in defects since last measurement.

On-Product Index

- The percentage of time teams spend working on product and value

Installed Version Index

- The number of versions of a product that are currently being supported.

Technical Debt

- The extra development work that arises when quick fixes result in more work later.

Production Incident Count

- The number of times in a given period that the Developers are interrupted to fix a problem in an existing product.

Active Product (Code) Branches

- The number of different versions of a product or service.

Time Spent Merging Code Between Branches

- Amount of time spent applying changes across different versions of a product or service.

Time Spent Context Switching

- Time spent switching between tasks, and time lost due to interruptions

Change Failure Rate

- Percentage of released product changes that result in degraded service and require support (patch, hotfix, rollback)

WHICH KEY VALUE MEASURES (KVMS) CAN HELP WITH YOUR PRICING STRATEGY?

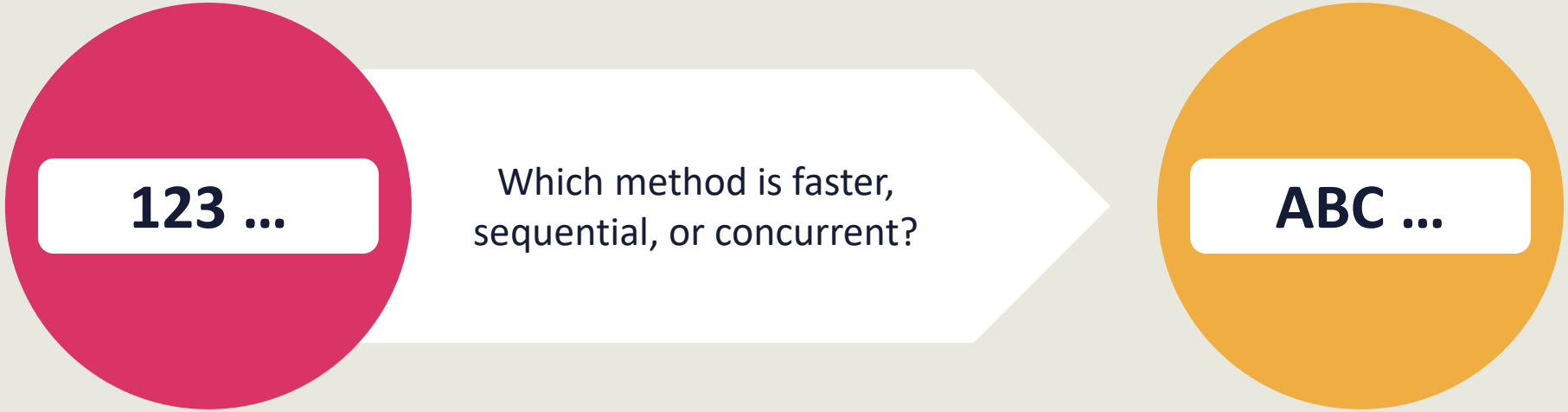
- Competitor’s pricing
- Market share
- Customer satisfaction gap, as it relates to pricing

TECHNICAL DEBT

- Impacts Value
- More time and money to be spent on maintenance
- Less time and money spent on new functionality

Ward Cunningham coined the term "Technical Debt" and likened it to a bank loan.

TASK SWITCHING



INPUTS TO MAXIMIZE VALUE

Product Owner must constantly assess the project and product environment to maximize value. Inputs can include:

- Customer feedback
- Product vision
- Competitive research
- Forecasting and feasibility
- Current state of the marketplace
- Any other inputs that the Product Owner finds relevant

GOAL SETTING

Four Levels:

- **Strategic Goal**

High level - states “what” is needed, but how is not clear.

- **Intermediate Goal**

Milestones and objectives that show progress toward the strategic goal

- **Current State**

Where we are now compared to the strategic goal

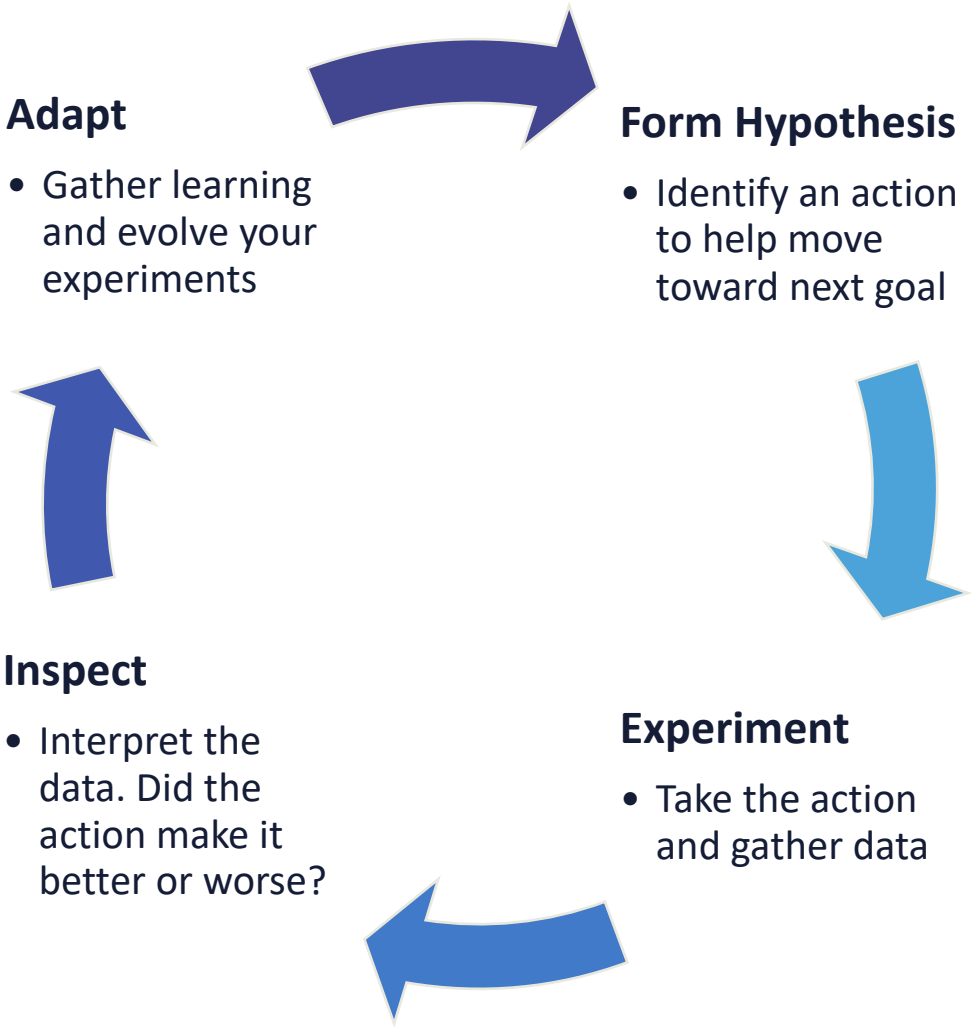
- **Starting State**

Gap between where were at the start and the strategic goal

PROJECT BENEFITS

Benefits Analysis Challenge: it can be easy to measure activities and outputs, but it can be difficult to translate and track to outcomes and value.

EXPERIMENT LOOP



Small Steps to Achieving Goals

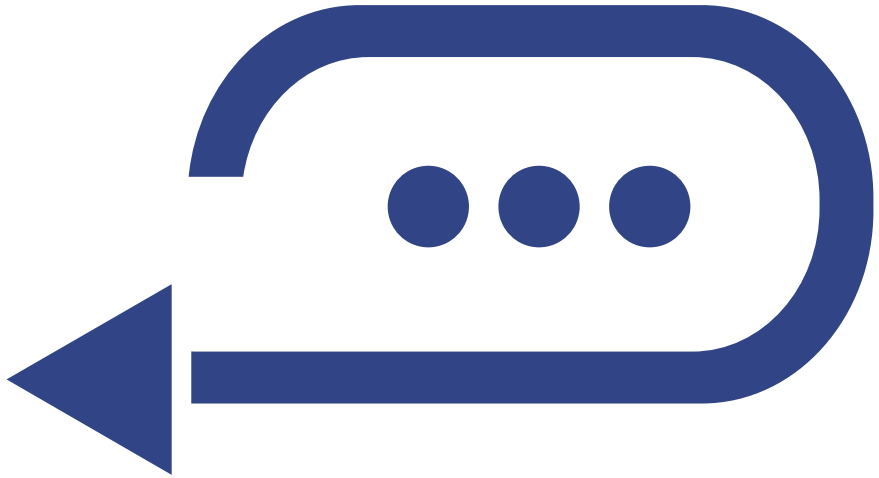
Move from the current state to the strategic goal incrementally

CONTROLLED EXPERIMENT

- Change one variable at a time
- If several variables are changed at the same time, we will not know which variable was responsible for results

FEEDBACK LOOPS AND VALUE

- Release frequently
- Validate assumptions
- Incorporate feedback
- Reprioritize as needed



YOU HAVE SEVERAL RISKY NEW PROJECTS. HOW SHOULD YOU ALLOCATE THE BUDGET?

- Provide a small budget to each project, along with a short goal.
- Measure the results, and repeat.
- Experiment loop will inform next steps.

WHY WOULD MANAGEMENT DECIDE TO STOP INVESTING IN A PRODUCT THAT IS STILL GENERATING REVENUE?

- Must consider product life cycles
- Maintaining a product requires time and money
- Current Value (CV) is only one part of Evidence-based Management
- Product may be approaching the end of its life cycle
- Perhaps focusing on new product innovation is the better decision

LONG TERM OPERATIONAL COSTS

- Can stifle innovation
- Products require maintenance
- Too many products can burden Developers
- Must balance innovation and ability to focus

KEY PERFORMANCE INDICATORS (KPIs)

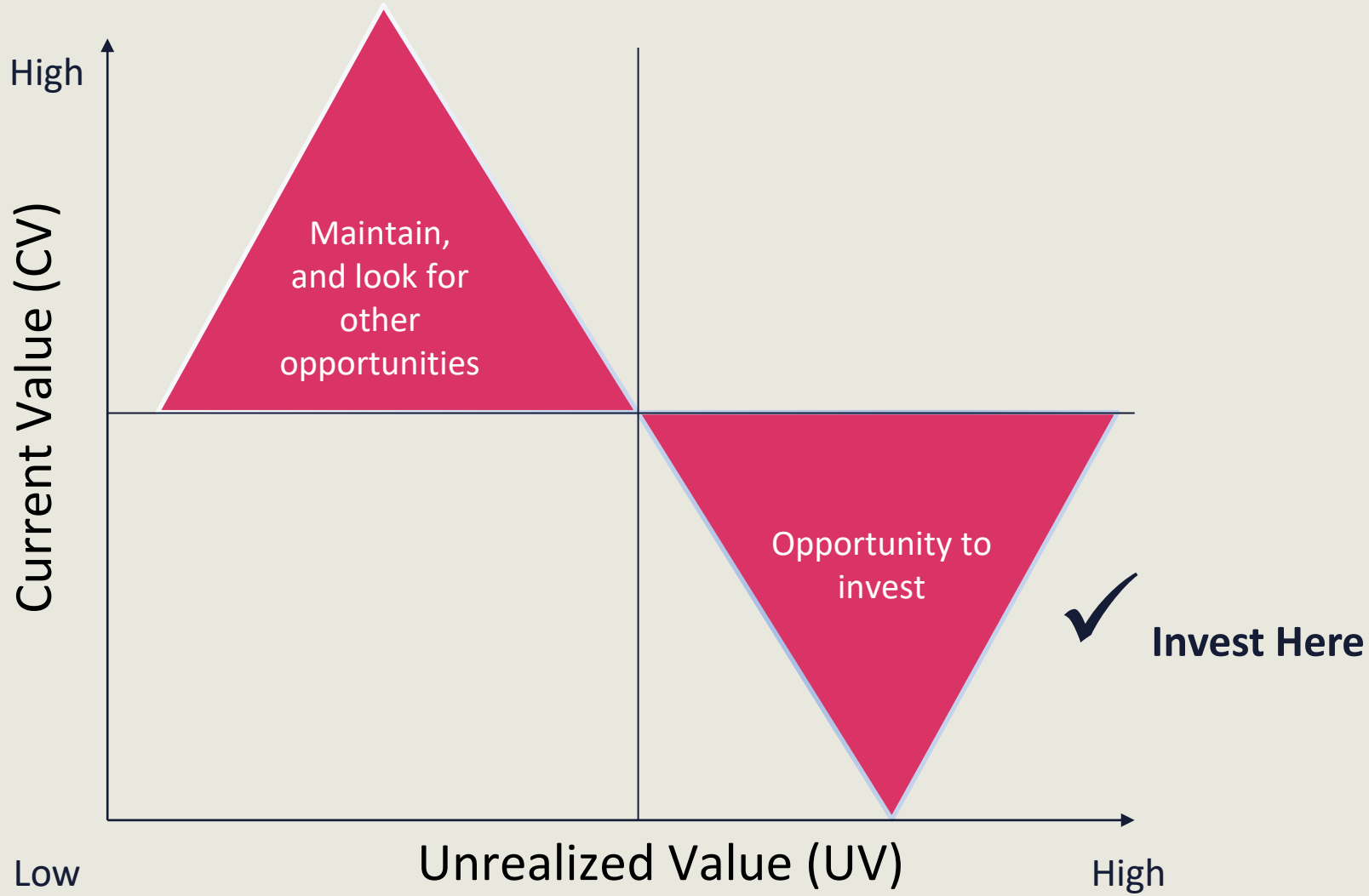
What has already happened

- Easy measure, hard to change
- How much product was purchased?
- What was our Net Promoter Score?

What can be done to achieve goals

- Tricky to measure, easy to change
- How can we speed up development?
- What skills and training does the team need?

CURRENT VALUE (CV) AND UNREALIZED VALUE (UV)

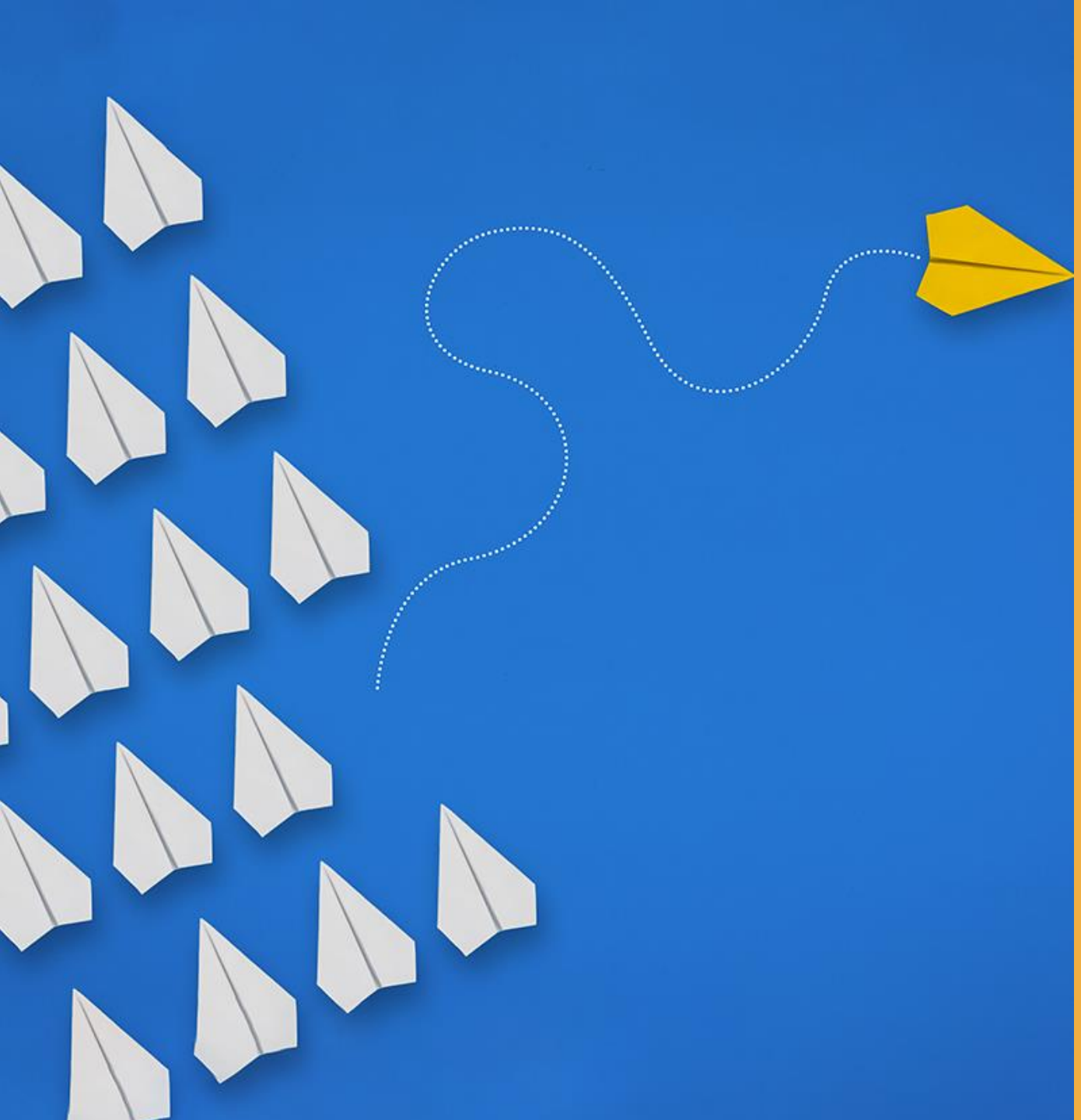


PRODUCT OWNER

- Develops product vision
- Serves as voice of the stakeholders (liaison)
- Collects requirements from stakeholders
- Determines value of features
- Prioritizes backlog items based on value
- Controls the budget
- Oversees return on investment
- Validates product quality

Product Owner

Stakeholders



“THAT ISN’T HOW IT WORKS IN OUR ORGANIZATION”

Stances (or common patterns) of Product Owners

MISUNDERSTOOD STANCES
OF THE PRODUCT OWNER

- The Clerk
- The Story Writer
- The Manager
- The Project Manager
- The Subject Matter Expert
- The Gatekeeper

https://www.scrum.org/resources/blog/stances-product-owner

THE CLERK

- “Sure, we can add that to the backlog.”
- Rarely says no to stakeholders
- Does not provide a clear product vision
- Backlog tends to be long and unfocused

THE STORY WRITER

- Well-organized Product Backlog
- Too much focus on the details
  - designs/sketches
  - functional and technical documentation
  - Task oriented
- Too much attention to short-term results
- Lack of strategy

THE MANAGER

- One-on-one conversations with each of the team members
- Concerned with team performance
- Focused on personal development of team members
- Distracted from maximizing product value

THE PROJECT MANAGER

- Output maximizer
- Focuses on graphs and schedules
- Emphasizes velocity
- Delivers as many features as possible
- Doesn’t recognize that output is independent from value

THE SUBJECT MATTER EXPERT

- Expert knowledge in their field
- Tends to get caught in the details
- Guards knowledge, rather than sharing it
- Blurs the line between Product Owner and Developer roles
- Stifles developer learning

THE GATEKEEPER

- Wants to sign off on everything
- Always in the middle
- Blocks communications between stakeholders and Developers
- Can create a bottleneck in the development process

PREFERRED STANCES
OF THE PRODUCT OWNER

The Visionary

The Collaborator

The Customer Representative

The Decision Maker

The Experimenter

The Influencer

https://www.scrum.org/resources/blog/stances-product-owner

THE VISIONARY

- Clearly communicates the product vision, strategy, and goals
- Believes in the product vision
- Thinks of what might be
- Not afraid of failure
- Focuses on the bigger picture

THE COLLABORATOR

- Strong team player
- Open and transparent
- Allows give and take
- Listens for understanding
- Follows through on commitments

THE CUSTOMER REPRESENTATIVE

- Understands customer needs, challenges, and perspectives
- Thoughtful about how the team’s work impacts users
- Knows the customer and users, actual and personas

THE DECISION MAKER

- Strong listener
- Uses data to guide decisions
  - Understands key performance measures
  - Maintains real-time knowledge of how the product is doing

Takes a “we” approach to work

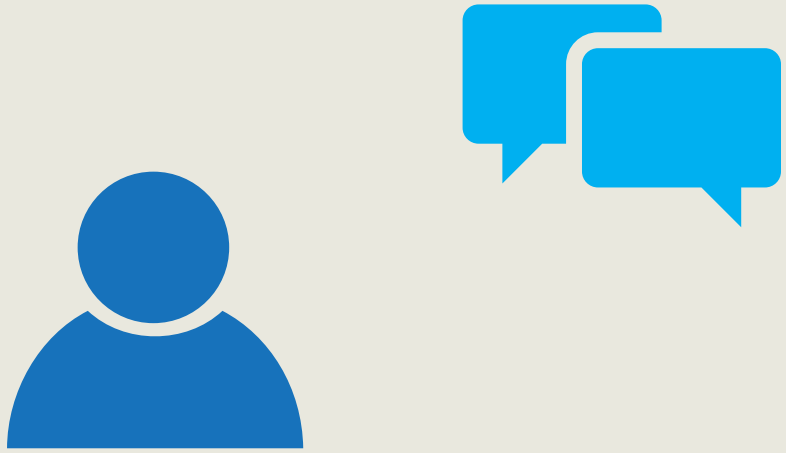
THE EXPERIMENTER

- Understands there is more unknown than known
- Optimistic about the future
- Not afraid of failure
- Never stops learning
- Willing to pursue multiple options

THE INFLUENCER

- Motivates the team to work towards the same goals and objectives
- Open, honest, and transparent
- Builds credibility and trust
- Uses compassion to understand the needs of others
- Communicates often and brings others to the table

COLLABORATION LEADS TO VALUE



Product Owner

Understands effort vs. value

Developers

Keep stakeholders needs in mind

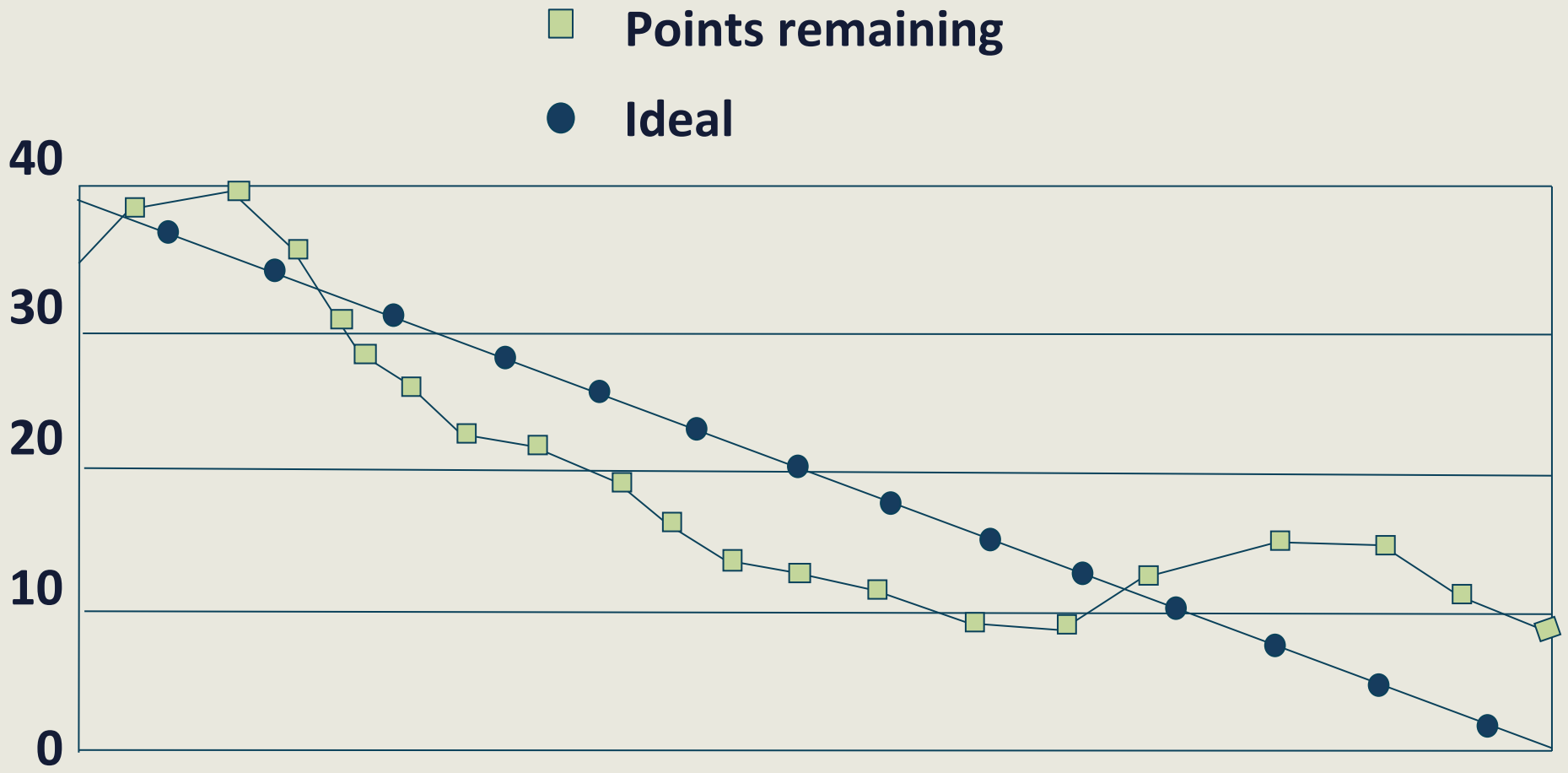
If the Product Owner is not available, the developers make the best decisions they can and touch base when possible.

VALUE METRICS

- Responsibility of the Product Owner
- Return on Investment (ROI)
- Total Cost of Ownership (TCO)
  - The sum of all costs across the product life cycle

    conceive + develop + operate + maintain
- Product Owner can choose any metric

BURN CHARTS



What can the burn chart tell the team about the amount of value being created?

IS IT DONE?

We have just a little more testing to do on this product increment.

Is it done?

Can it be released?

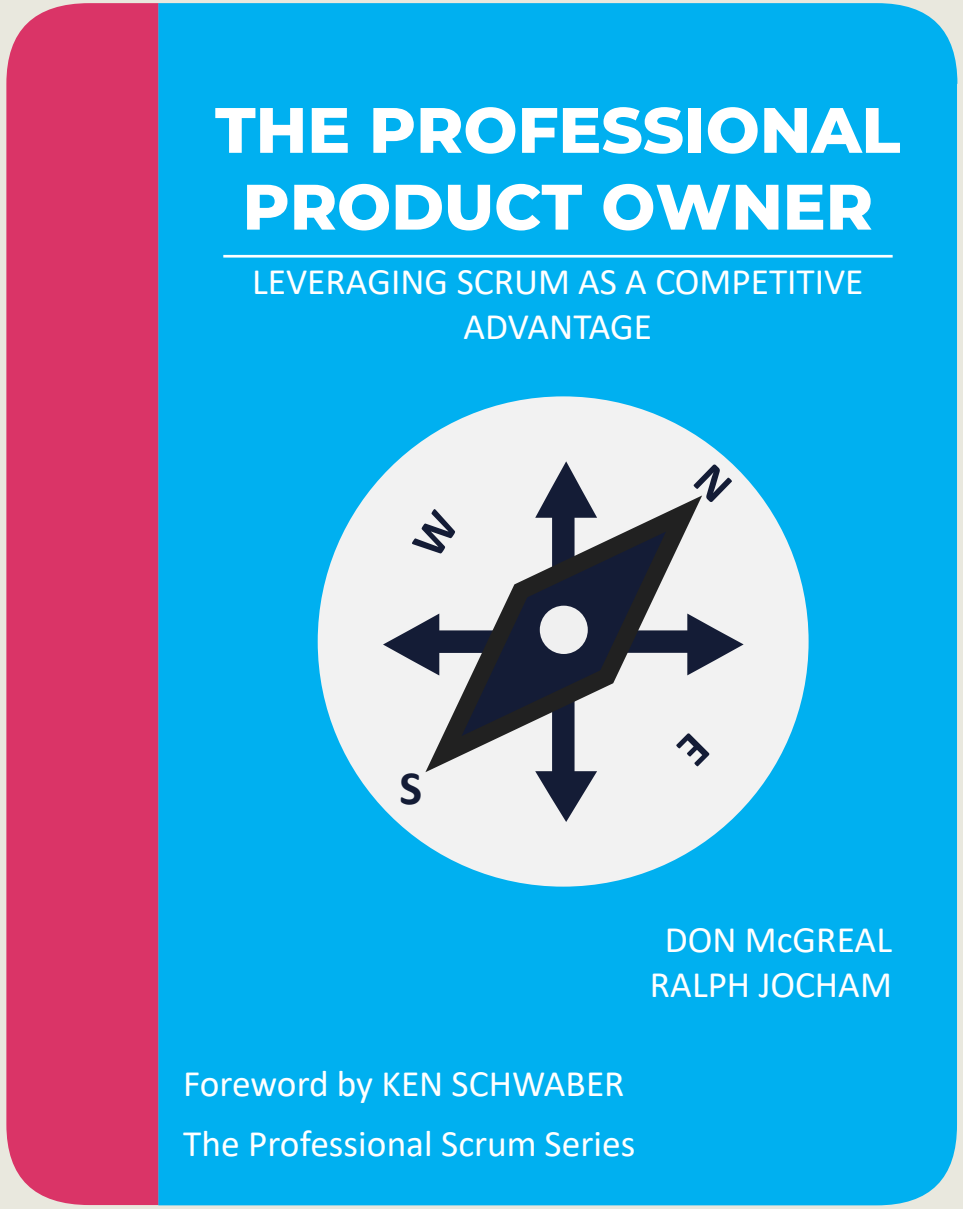
LEVEL OF EFFORT IS NOT AN INDICATOR OF VALUE

How can the team determine if value was created?

WHAT IF....

STAKEHOLDERS ARE NOT USING THE NEW FUNCTIONALITY?

- Did you understand the customers' needs?
- Are the features solving the problem or fulfilling a need as intended?
- Ask questions to clarify the requirements.
- Do not disable the functionality or make assumptions.

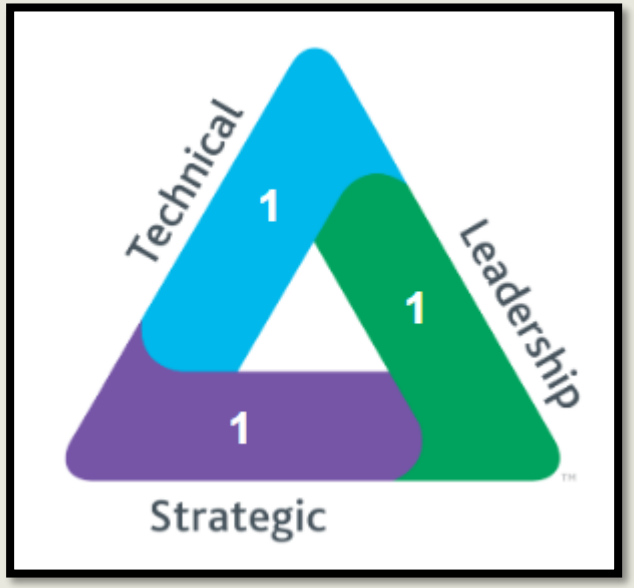


ADDITIONAL SUGGESTED READING

PMI RECERTIFICATION (PDUs)

If you have a PMI certification, you may use the PDU claim code below to claim three (3) Professional Development Units (PDUs) for this course.

PDU Claim Code 1008VXMILU



DAILY BOOTCAMP SURVEY

At the end of each Bootcamp session please let us know how we are doing. Your feedback helps us to offer the best possible Bootcamp experience.

Please share your thoughts