PROFESSIONAL SCRUM PRODUCT OWNER LEVEL 1 (PSPO I) EXAM PREP

BOOTCAMP SESSION 1

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 demonstrate a **fundamental level of Scrum mastery**	• not for certified Product Owners unless you need a refresher
• anyone who would like to **maximize the value delivered with a product**	• not aligned to the Project Management Institute's PMP or Agile certifications
• students who might be interested in pursuing the **PSPO I exam**. Also, students who would like a **foundational class before pursuing the PSPO II**.	• not limited to PSPO I and PSPO II candidates. Everyone is welcome!

COMPARISON OF PSM I AND PSPO I CURRICULUM

ABOUT KEN SCHWABER AND SCRUM.ORG

Timeline

- **Early 1990s:** Co-developed the Scrum Framework with Jeff Sutherland
- **2001:** Signed the Agile Manifesto
- **2002:** Co-founded Scrum Alliance
- **2009:** Founded Scrum.org, co-authored the Scrum Guide with Jeff Sutherland

Purpose of Scrum.org

- Formalize the Scrum body of knowledge
- Improve the quality and consistency of training

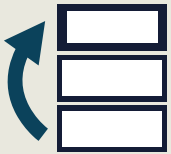
PSPO I EXAM DETAILS

$200 per attempt

Multiple choice
Multiple answer, True/False

Recommended reading
Scrum.org

60 minutes



No prerequisites

80 questions

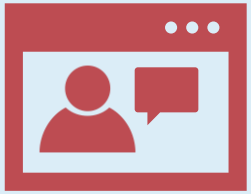
Passing score is 85%

PROFESSIONAL SCRUM PRODUCT OWNER JOURNEY

Progression of certifications

INDEPENDENT STUDY

Articles



Videos

Podcasts

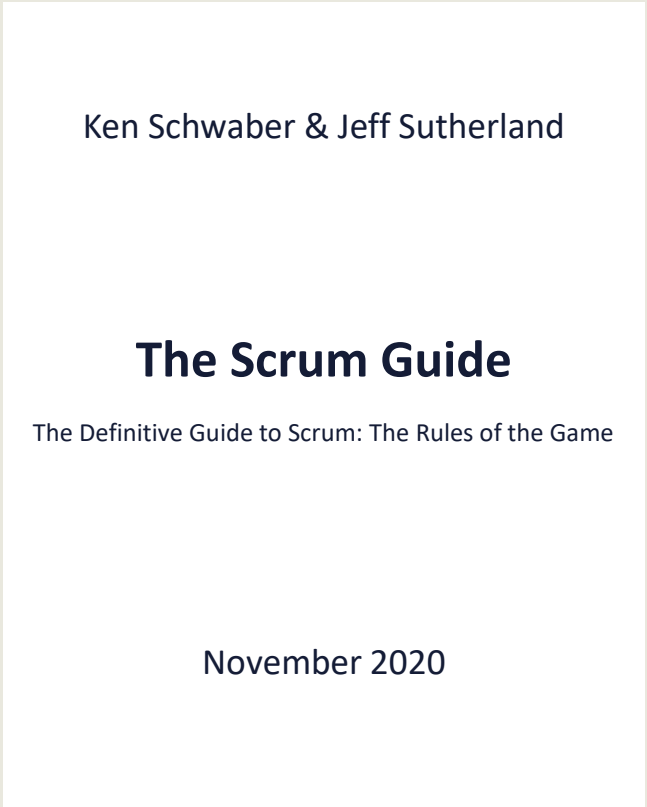
Practice

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

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

SCRUM GUIDE

BY KEN SCHWABER & JEFF SUTHERLAND



https://www.scrumguides.org/

THE AGILE MANIFESTO

In 2001, seventeen software developers met at a resort in Snowbird, Utah to discuss existing software development methods, among others Jeff Sutherland, Ken Schwaber, Jim Highsmith, Alistair Cockburn, and Bob Martin. Together they published the *Manifesto for Agile Software Development*.

The Four Values of the Agile Manifesto

We are uncovering better ways of developing software by doing it and helping others to do it. Through this work we have come to value:

1. **Individuals and interactions** over processes and tools
2. **Working software** over comprehensive documentation
3. **Customer collaboration** over contract negotiation
4. **Responding to change** over following a plan

There is value in all of these, but we value the items in bold more.

THE 12 CLARIFYING PRINCIPLES

- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software
- Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- Business people and developers must work together daily throughout the project.
- Build projects around motivated individuals. Give them the environment and support they need and trust them to get the job done.
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- Working software is the primary measure of progress.
- Agile processes promote sustainable development. The sponsors, developer, and users should be able to maintain a constant pace indefinitely.
- Continuous attention to technical excellence and good design enhances agility.
- Simplicity – the art of maximizing the amount of work not done – is essential.
- The best architectures, requirements, and designs emerge from self-organizing teams.
- At regular intervals, the team reflects on how to become more effective., then tunes and adjust its behavior accordingly.

AGILE METHODOLOGIES

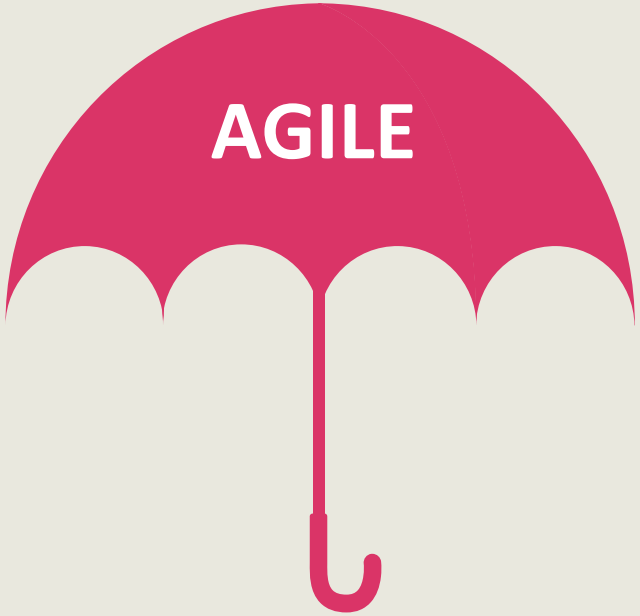
There are over a dozen agile methodologies

No single right way

Can be tailored once a team is experienced

Most common

- Scrum (really a framework)
- Extreme Programming (XP)
- Lean product development
- Kanban
- Disciplined Agile
- Feature-driven development (FDD)
- Dynamic Systems Development Method (DSDM)
- Crystal



SCRUM IS A FRAMEWORK FOR

Creating Complex Products
in
Complex Environments
in
Any Industry

Generate value through adaptive solutions to complex problems.

SCRUM

- Framework rather than a methodology
- Scrum is one of many Agile approaches
- Can be applied to any industry
- Employs various techniques
- High-performing cross functional teams
- Iterative, incremental approach
- Iterations are known as “Sprints”

The term “Scrum” comes from rugby.

A scrum (short for scrummage) is a method of restarting play. The players pack closely together with their heads down and attempt to gain possession of the ball.

THREE PILLARS OF EMPIRICISM

- **Transparency**
  - Discuss product requirements
  - Establish shared product vision
  - Create a Definition of Done
- **Inspection**
  - Assess productivity during Daily Scrum
  - Burn-down chart
  - Demonstrate product increment during Sprint Review
  - Objective assessment based on Acceptance Criteria and Definition of Done
- **Adaptation**
  - Welcome change
  - React quickly to variance in order to meet Sprint goal
  - Sprint Retrospective promotes continuous improvement

SCRUM.ORG

SCRUM VALUES

COURAGE

FOCUS

COMMITMENT

RESPECT

OPENNESS

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

PRODUCT OWNER

- Manages and prioritizes the product backlog
- Collects requirements from stakeholders
- Serves as voice of the stakeholders (liaison)
- Develops product vision and ensures transparency
- Communicates the product goal
- Ensures value delivery
- Controls the budget

Product Owner

Stakeholders

KEY CONCEPT

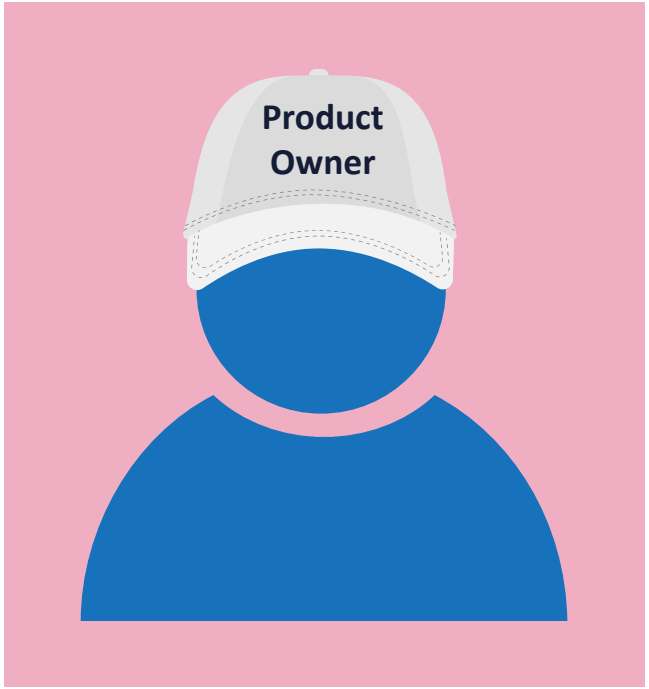
Product Owner

The Product Owner is a single person, not a committee.

- Ensures rapid decision-making
- Simplifies communication
- One point of contact on the Product Backlog
- One person is accountable for value delivery

A PRODUCT OWNER DELEGATE OR PROXY

- The Product Owner may delegate certain responsibilities to others who are not already on the Scrum Team.
  - Product Backlog management
  - Communicating the Product Goal
  - Ordering the Product Backlog
  - Making the Product Backlog transparent
- The Product Owner does not typically delegate attendance to Scrum events.
- Product Owner **remains accountable** for all of these items.



DEVELOPERS

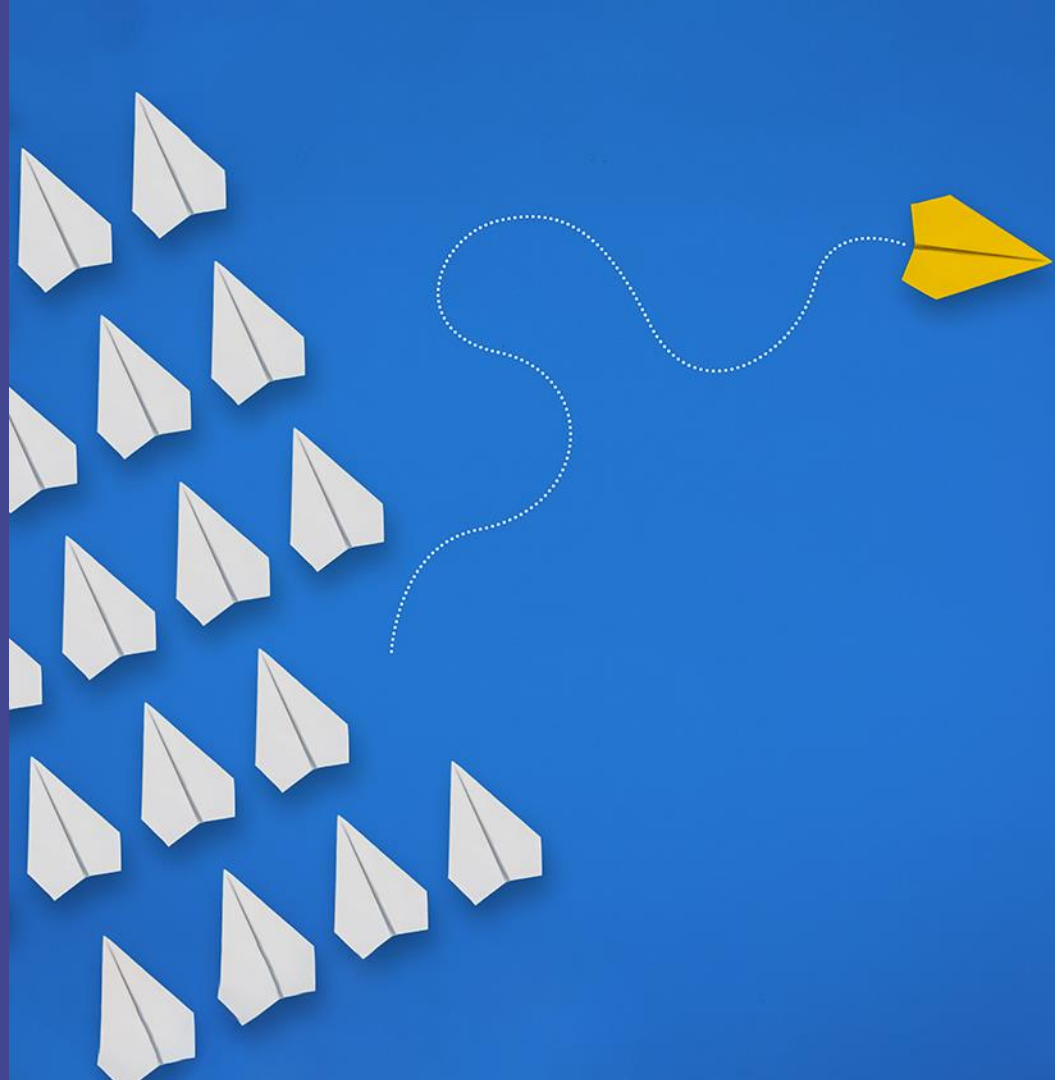
- Also known as the Development Team
- Self-organized
- Builds the product increments during each Sprint
- Estimates the work
- Decides what can be done during each Sprint
- Cross-functional
- Includes all skillsets and responsibilities, including “QA” or “Tester”
- Every necessary skillset is represented

SCRUM MASTER

- Servant leader to the Scrum Team
- Ensures adherence to Scrum framework and roles
- Facilitates meetings
- Removes impediments (roadblocks, blockers)
- Serves as a buffer to prevent interruptions
- Provides essential resources
  - Tools
  - Resources
  - Rewards
  - Encouragement
- Coaches other team members
- Assists Product Owner with managing backlog with the goal of maximizing value
- Serves as Scrum “ambassador” to the organization

SCRUMBUT

We use Scrum, but…



WHAT ABOUT THE PROJECT MANAGER?

Activity	Scrum Master	Product Owner	Developers	Project Manager
Integration		✓		✓
Scope		Product Backlog	Sprint Backlog	
Time	Meeting Time Blocks	Releases	Sprint	
Cost		Manages Budget	Estimates Tasks	
Quality	Definition of Done and Sprint Review	Definition of Done and Sprint Review	Definition of Done, QA, and Sprint Review	✓
Risk	Requirements	Requirements	Requirements	✓
Resources				✓

The Scrum Team should have all of the skills and competencies needed to deliver the product increment.

PROBLEM WITH DUAL ROLES

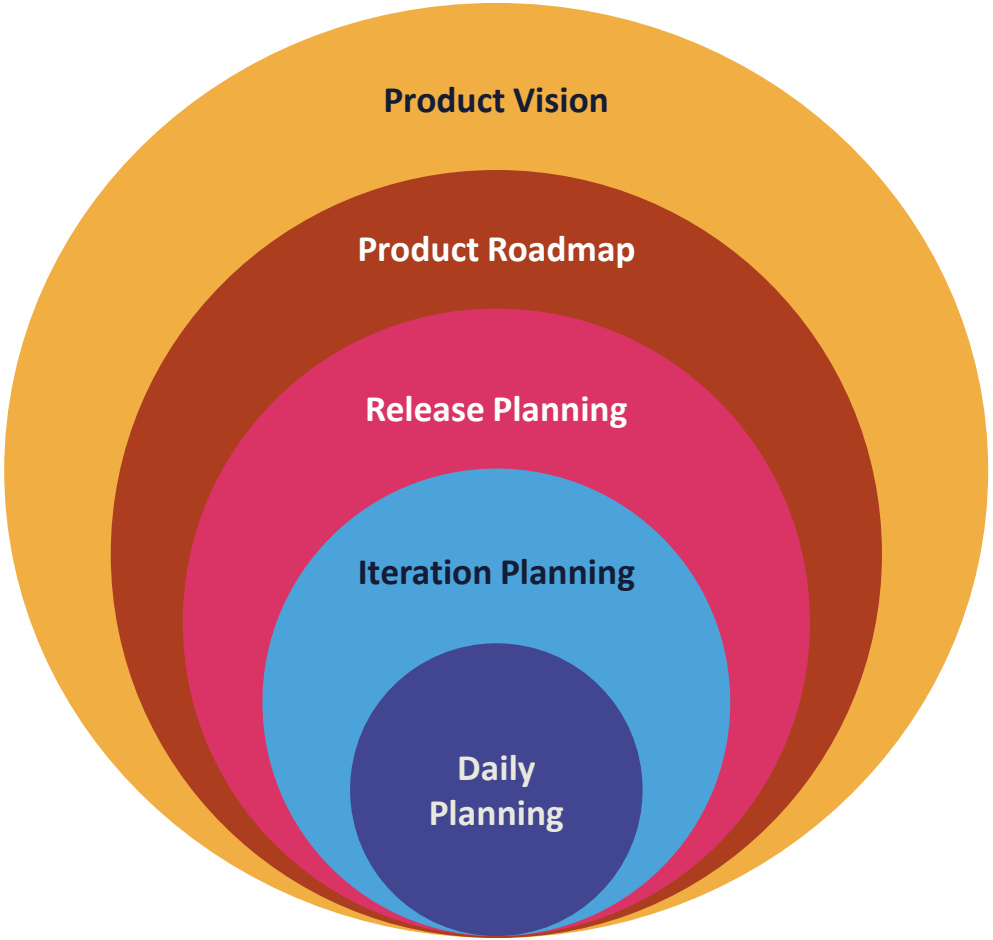
- Product owner promotes value delivery and new functionality
- Developers sets realistic expectations and perform the work
- Scrum Master protects the team and promotes governance

Can someone serve as both the Scrum Master and Product Owner?

What if one of these roles is not represented?

EXECUTIVE TEAM INVOLVEMENT

- Scrum Master is the liaison between the executive team and the Developers
- Product Owner is the contact for the product backlog
- Developers defer to Scrum Master and Product Owner, depending on the request



LEVEL OF DETAIL

CREATING THE SHARED PRODUCT VISION

- Interview stakeholders
- Focus on how a product adds value
- What problem will the product solve?
- Motivates Developers
- Helps to establish a measurable Product Goal
- As Product Goals are achieved, the team moves toward the Product Vision
- Gives purpose to individual Sprints

PRODUCT VISION

Why you're building a product

Benefits of product

Who you're building it for

Why you are positioned to develop it

Since scope is evolving it is important to share an understanding of what is being created

USER STORIES

Short, simple descriptions of a feature

Told from the user's perspective

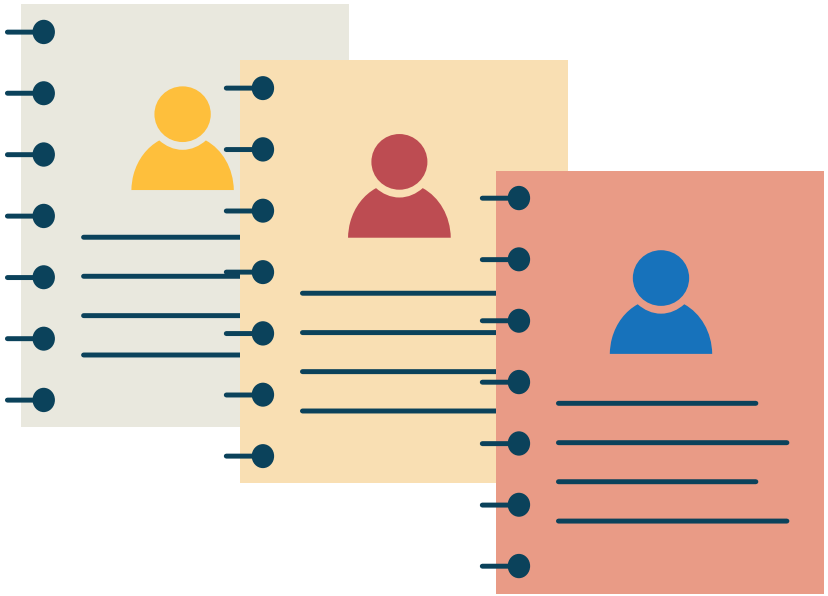
When large or complex, can be called "epics"

Sentence structure:

"As a ***role***, I want ***functionality***, so that ***business benefit***."

Example:

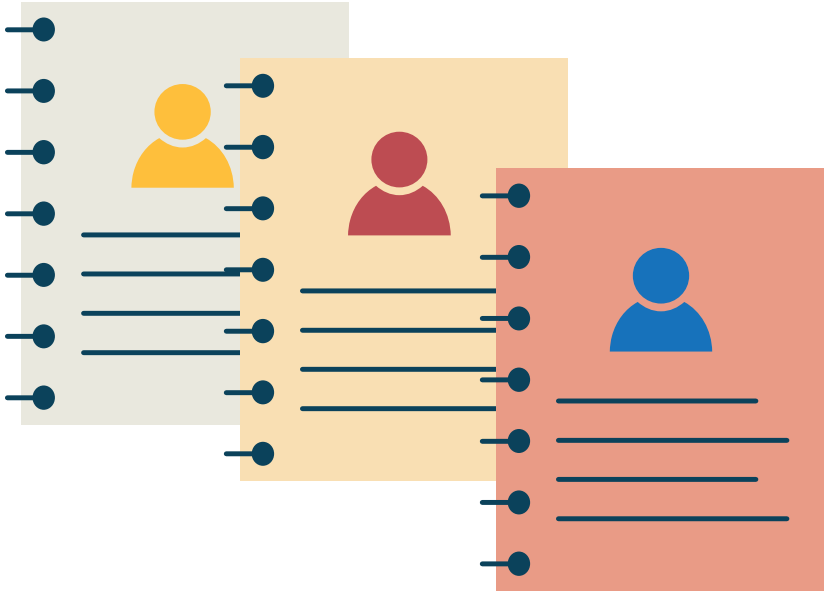
"As a customer, I want my credit card information to be stored, so that I save time when checking out."



FORMATTING USER STORIES

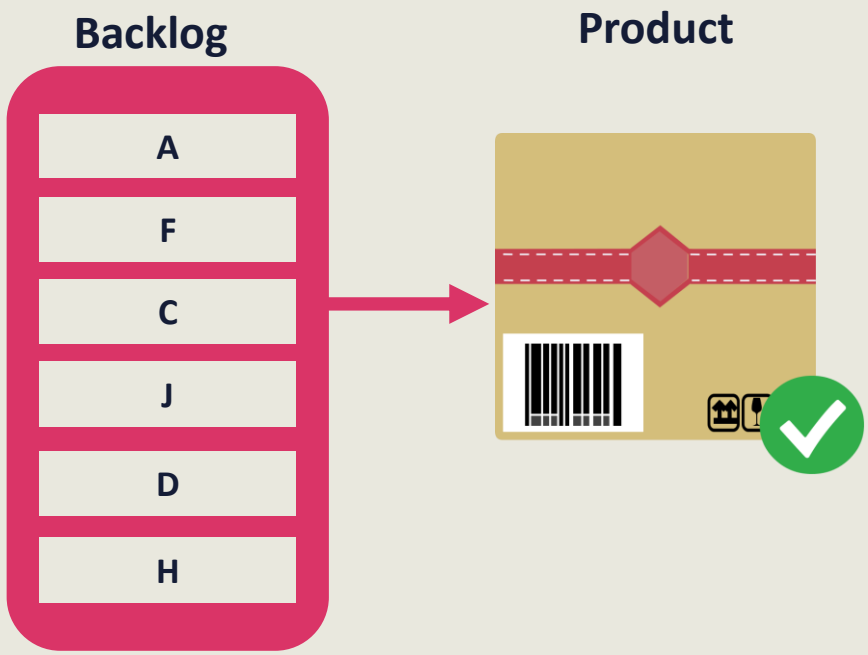
INVEST criteria

- Independent
- Negotiable
- Valuable
- Estimable
- Small
- Testable



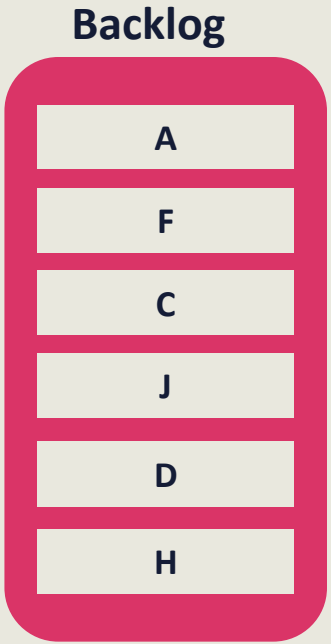
PRODUCT BACKLOG

- Items are ordered based primarily on **value**
- Priority, dependencies, and risk are also factors
- **All work** should be included
  - Bug fixes
  - Security features
  - Changes
- **Single source** of product requirements
  - Functional and non-functional
- Always changing – items can be added, dropped, or reprioritized as needed
- Can be **updated any time** – there is no Scrum "event" for backlog updates

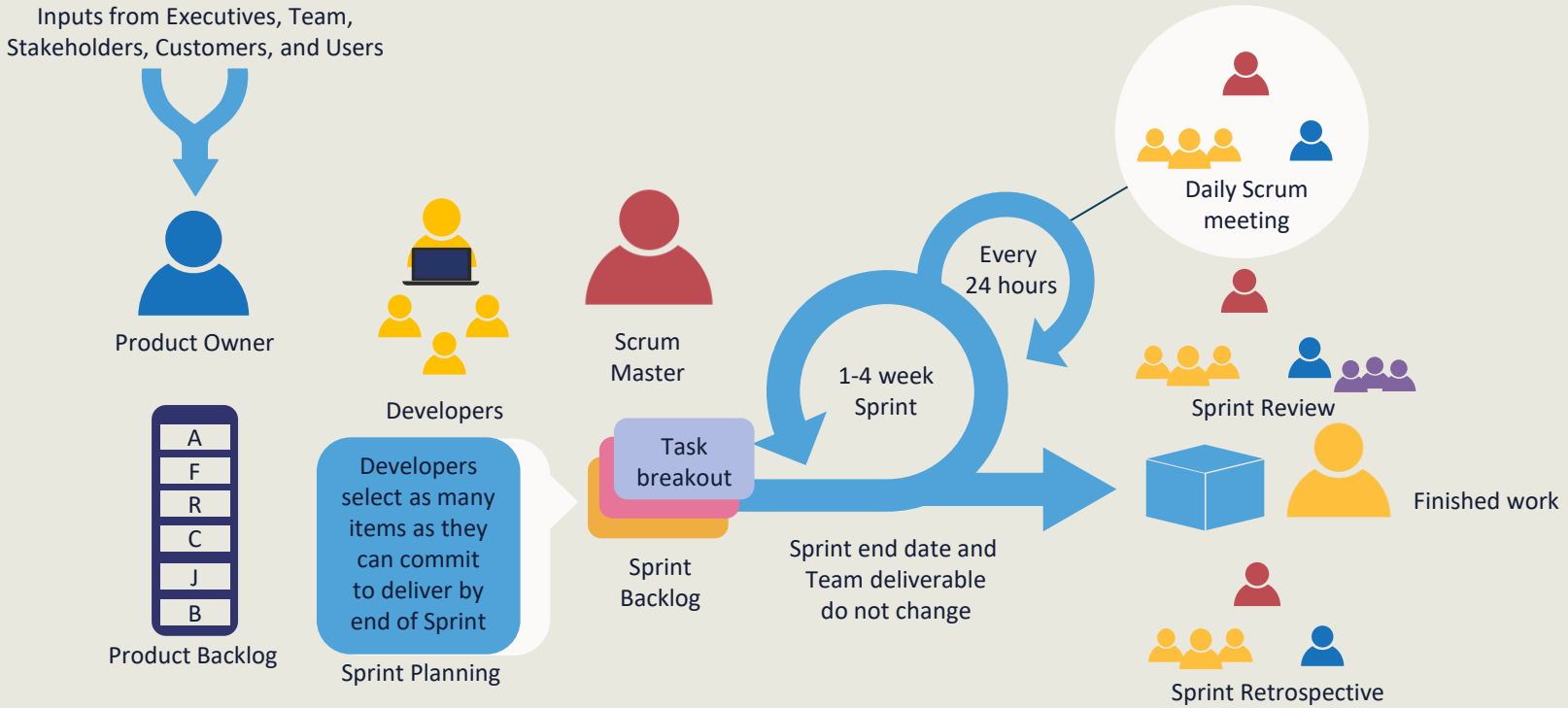


PRODUCT BACKLOG IS THE PLAN

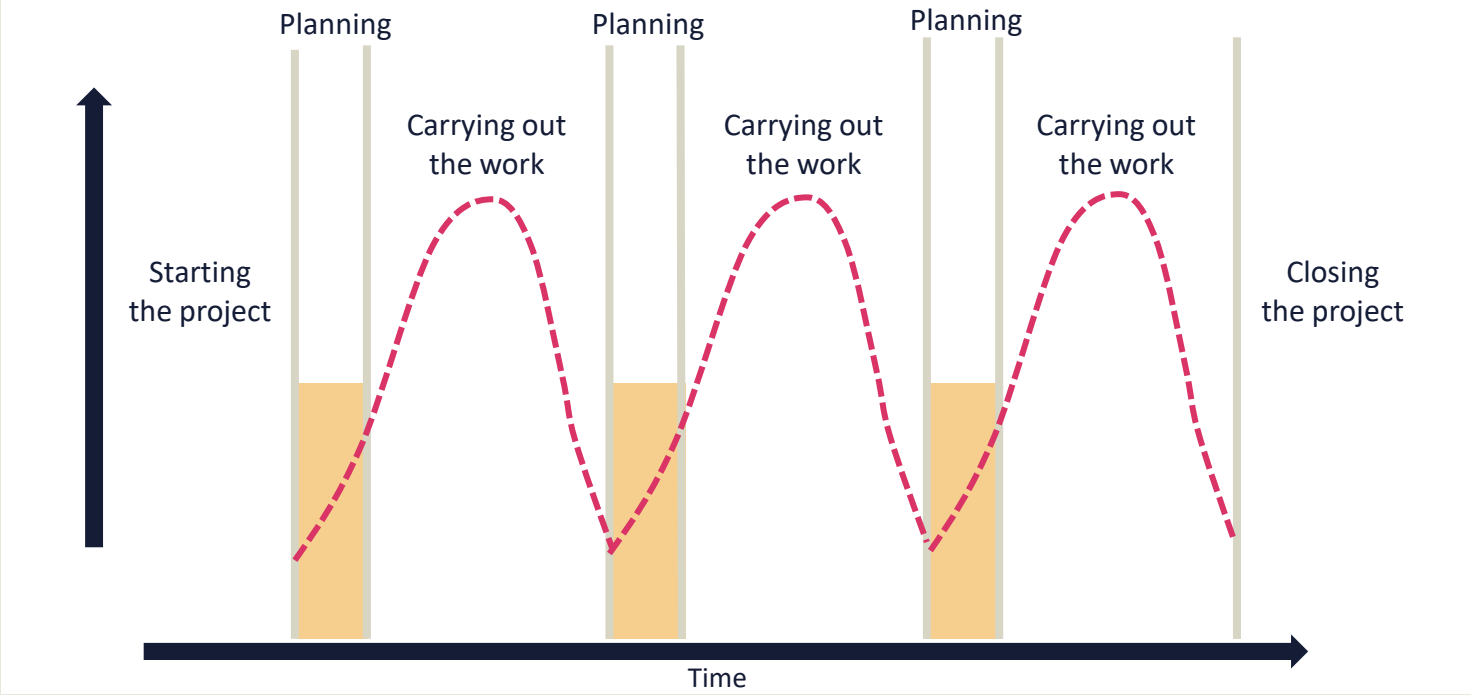
- Unlike a traditional project, the scope can change and evolve
- The product is built incrementally based on work selected from the backlog
- Prioritized list of everything that is needed in the product
- The Product Backlog **IS** the plan. There is no separate planning document.



SCRUM FRAMEWORK



AGILE PROJECT LIFE CYCLE (TIME BOXING)



Risk is limited to one month or less. Feedback loop is shorter, progress is inspected frequently, and assumptions can be validated.

SMALL RELEASES

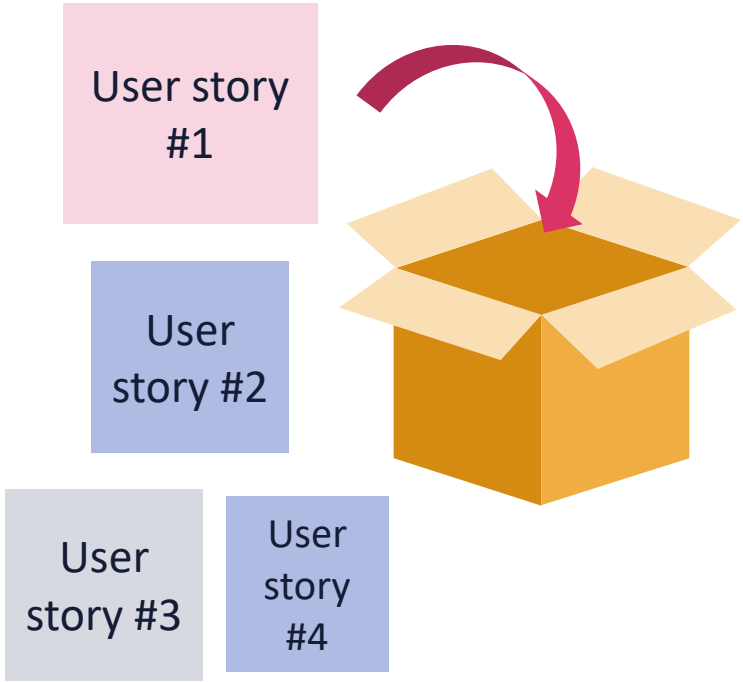
Demonstrates progress

Increases visibility to the customer

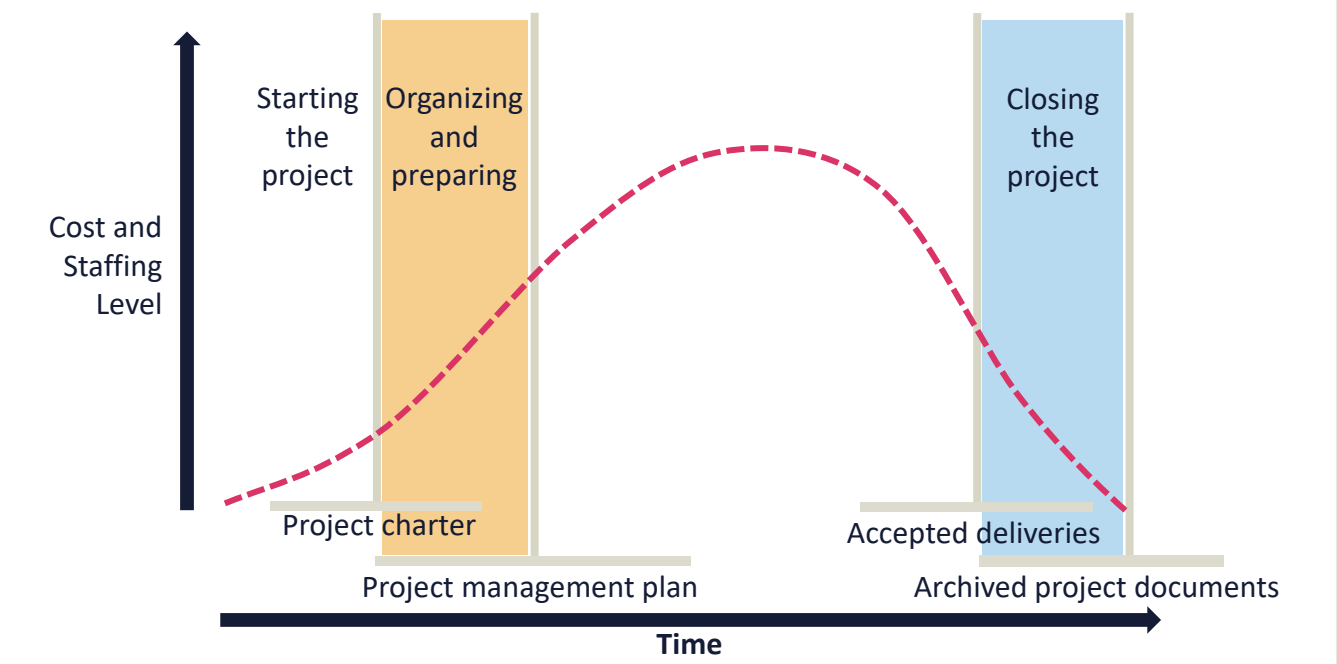
Smaller increments means rapid deployments

TIME BOXING

- Each Scrum event has maximum time allotted
  - Ex: 2-week Sprint
- User stories are estimated
  - Planned into the iteration
  - If it doesn't fit it has to wait
- Tool for completing work



TRADITIONAL PROJECT LIFE CYCLE



A long project horizon increases risk. The project team goes a long time without gathering feedback.

PRODUCT OWNER'S

#1 Responsibility

Maximize Value

Product Owner works with the rest of the Scrum Team to create value

DETERMINING VALUE

Is revenue the only way to determine value?

No.

- **Value can be tangible or intangible.**
- **Perception of value differs across stakeholders.**
- **Value can vary by product and/or by organization.**
- **Tool and techniques to measure value are not prescribed. Product Owner can choose.**

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
- Additional indicators of value:
  - Direct customer feedback
  - The extent to which the product is being used
  - How quickly and easily the product can be used

EVIDENCE-BASED MANAGEMENT

Four Key Areas (KVAs):

1) Current Value (CV)

What value are we delivering to the customer today?

2) Time to Market (T2M)

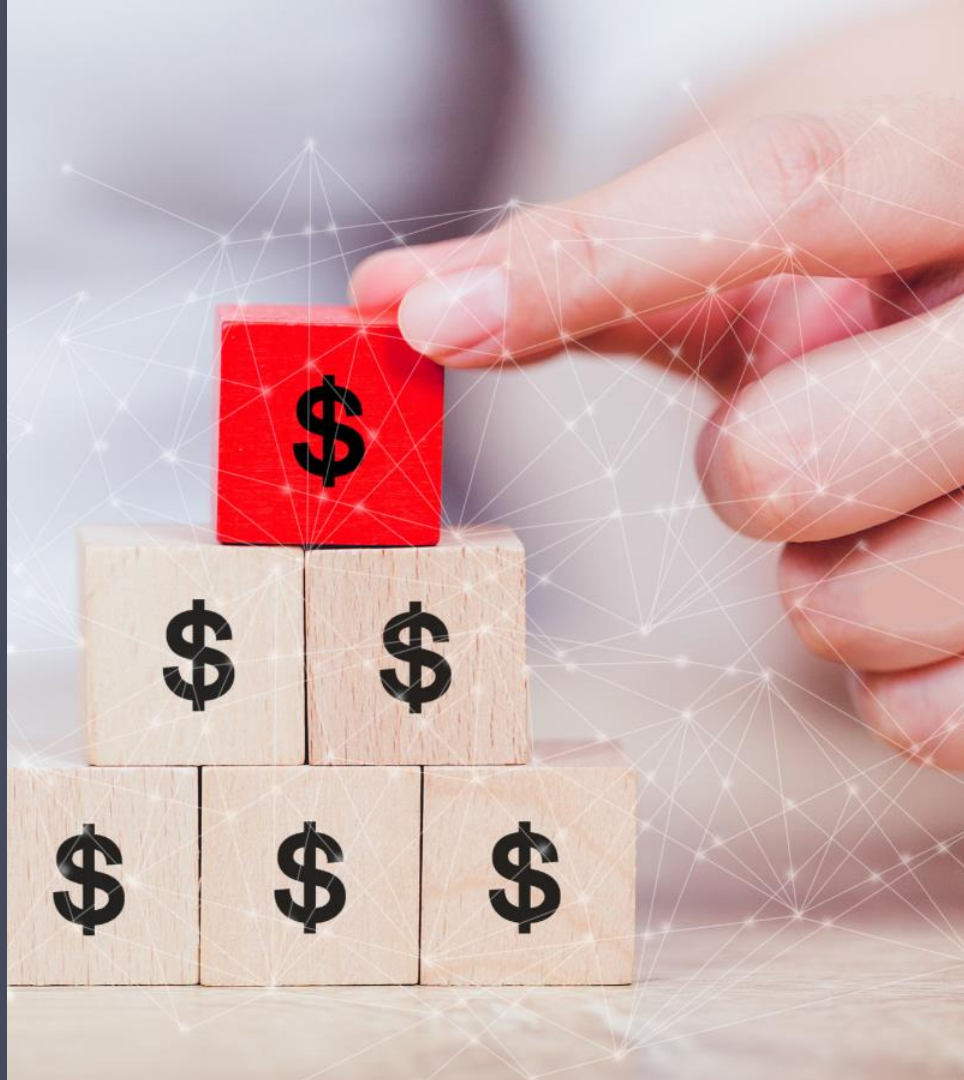
How quickly can we deliver a new product, service, or feature?

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?

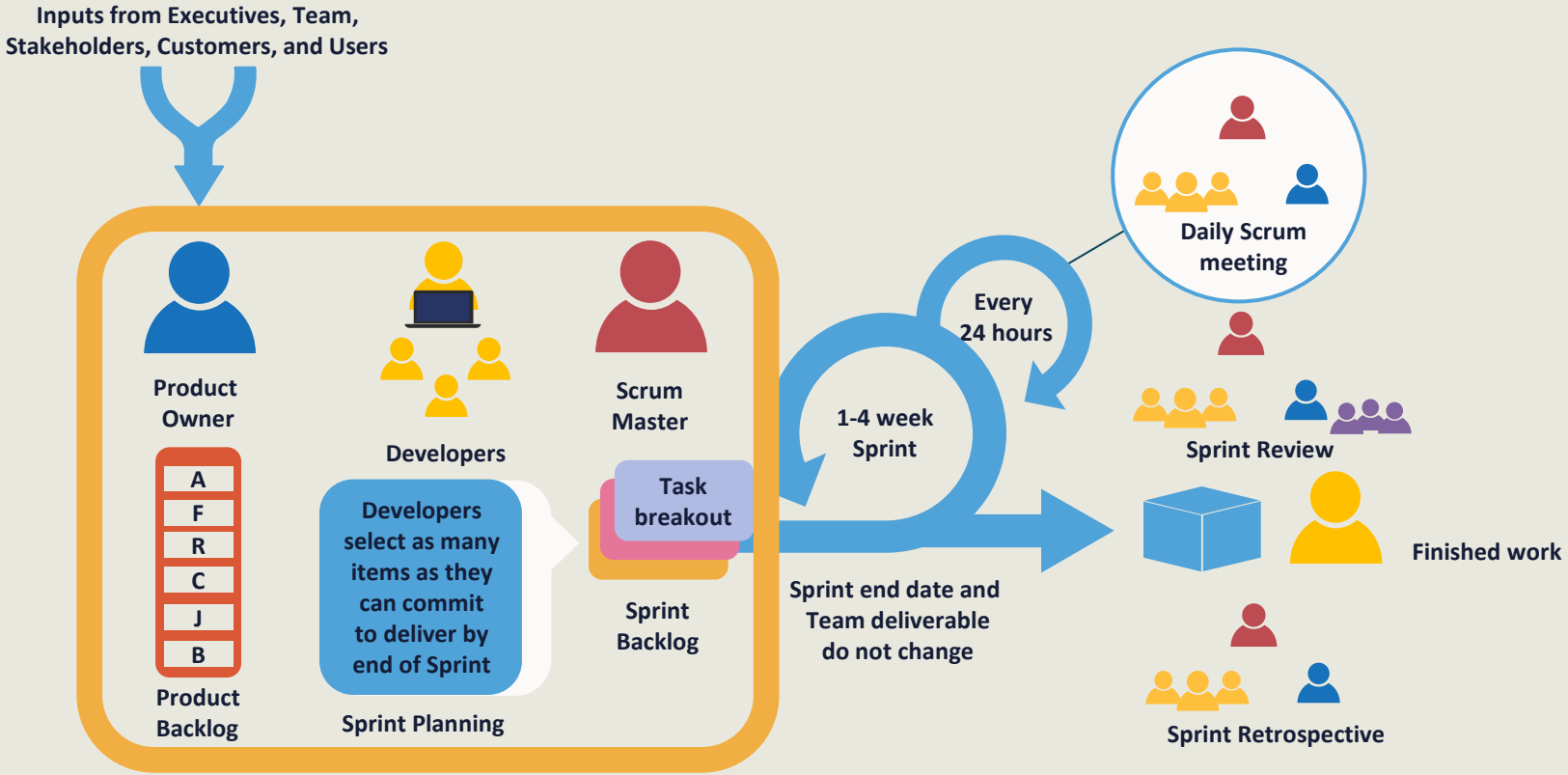


INPUTS TO MAXIMIZE VALUE

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

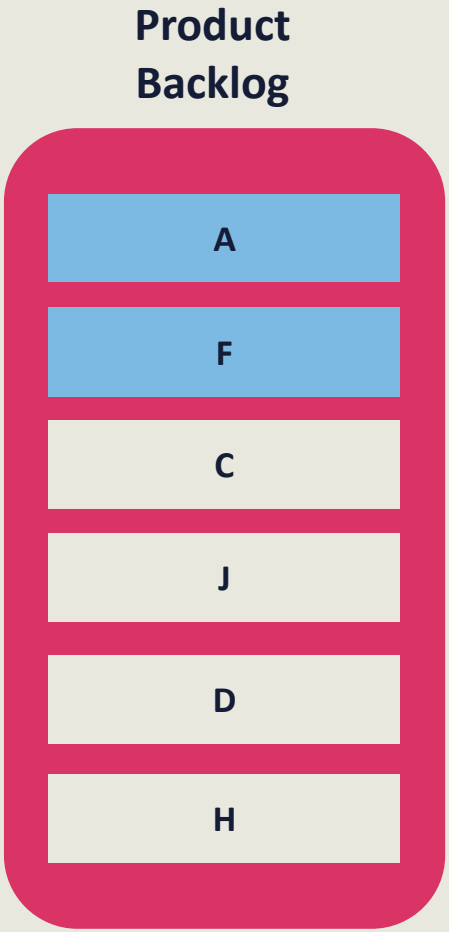
SPRINT PLANNING



SPRINT PLANNING

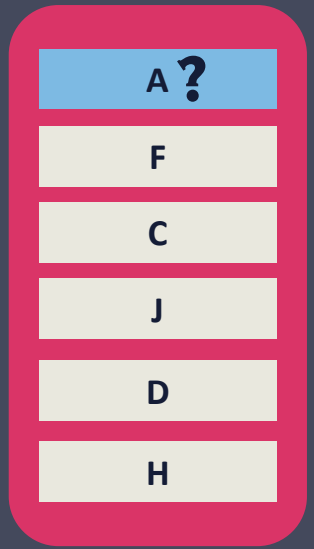
- Includes all Scrum Team members
- Product Owner presents the updated backlog and the Sprint Goal
- Developers estimate the work
- Product Owner works with Developers to understand:
  - level of effort
  - complexity
  - uncertainty
- Scrum Team selects from the top of the product backlog to create the Sprint backlog.
- Items should align to Product Goal and Product Vision
- Establish a “Definition of Done”

Typically 2 hours per week of Sprint



THE ACCEPTANCE CRITERIA FOR A PRODUCT BACKLOG ITEM ISN'T CLEAR.

CAN WE PULL IT INTO SPRINT PLANNING? YES.

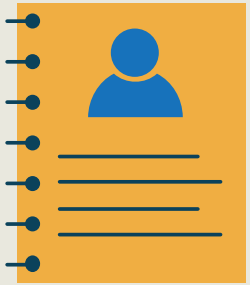


- The Product Owner is not required to create clear acceptance criteria before Sprint Planning
- The Product Backlog may be incomplete
- A Product Backlog item can be selected and discussed during Sprint Planning
- The Scrum Team can work together to examine user stories and define requirements
- Each Sprint can start without knowing the details of lower Product Backlog items.

DEFINITION OF DONE: IMPORTANCE TO THE PRODUCT OWNER

- Creates transparency within the Scrum Team regarding progress
- Establishes the expected quality of the increment so that it doesn't become a disagreement during the Sprint Review
- The product increment inspected at the Sprint Review must meet the Definition of Done
- Although the Product Owner is accountable for delivering value, the Definition of Done is created by the organization or the Scrum Team
- Can affect the product's total cost of ownership
  - (costs to conceive + develop + operate + maintain)

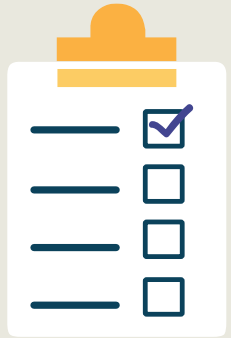
ACCEPTANCE CRITERIA AND DEFINITION OF DONE



Acceptance Criteria	Definition of Done
Specific (user story)	Applies to all work
Meet user needs	Team's shared understanding
Functional features for users	Includes non-functional and quality requirements

Acceptance Criteria and Definition of Done

- agreed to as a team
- updated as new learnings come to light
- testable
- clear
- concise



Development Organization may have standards for security and compliance

"Organizational DoD"

Team A Team B Team C

Developers may have standards for product requirements

Team A

DEFINITION OF DONE

Established by the **Development Organization and/**or by the **Developers**

"Organizational DoD"

And

"Product DoD"

NON-FUNCTIONAL REQUIREMENTS

Criteria that is related to the operation of a system. A property of the system as a whole.

- Non-functional requirements tend to be less visible
- Transparency must be more intentional
  - Add to the Product Backlog
  - Include in the Definition of Done

Response Time

Security

Capacity

Recoverability

Regulatory

Privacy

NON-FUNCTIONAL REQUIREMENTS

- Part of Definition of Done
- Often invisible, but critical features
- Specified by subject matter experts
- Customer may not think about these
- Examples:
  - Security
  - Regulatory
  - Scalability
  - Reliability

The site must be available 99.9% of the time and must be able to accommodate 1,000 simultaneous users.

ROLES DURING SPRINT PLANNING

What about upper management and stakeholders outside of the Scrum Team?

Product Owner

- Presents the updated product backlog
- Answer questions about the backlog items
- Provide clarification on user stories
- Reprioritize backlog as appropriate
- Assist with defining Done

Developers

- Ask clarifying questions about backlog items
- Select items from the backlog
- Estimate the work
- Negotiate with the Product Owner
- Commit to a set of deliverables
- Assist with defining Done

Scrum Master

- Maintain Scrum best practices
- Adhere to the meeting time block
- Understand capacity of the Developers
- Assist with defining Done

THERE IS NO SPRINT ZERO

Some suggest that it…

- allows a team to get a better idea of the work
- gives the team a chance to self-organize in order to perform better in the long run
- builds confidence within the team that they can handle the work
- provides clarity, which leads to the success of subsequent sprints
- opportunity to plan a framework for success

Scrum.org says…

- no to Sprint Zero or any specially named Sprints
- it is anti-Scrum because it creates no business value
- it sets a precedent that a sprint can have no potentially shippable product
- It is not favored by many founders of scrum
  - Alistair Cockburn – Agile Manifesto
  - Ken Schwaber – Scrum.org
  - Mike Cohn – Mountain Goat Software

Even during the first Sprint, the Developers should create at least one functional, valuable, and potentially releasable product increment.

THE FIRST SPRINT

To get started, Scrum requires very little:

- A Product Owner with enough ideas for a first Sprint
- Developers to implement the ideas
- A Scrum Master to guide the process

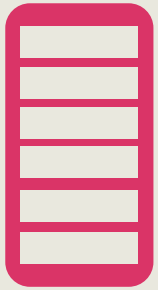
SPRINT GOAL

- Start with high level goal for functionality needed
- Select product backlog items that align with the overall Product Goal
- Some unrelated but priority backlog items may be included
- Developers will decide if the amount of work is realistic
- The goal is fixed, the Sprint backlog can evolve

WHAT IS IN THE SPRINT BACKLOG?

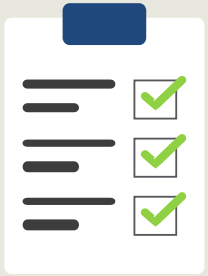
Sprint Goal

“Why”



Product Backlog Items

“What”



Action Plan

“How”

SPRINT BACKLOG UPDATES

The Sprint Backlog is a plan by and for the Developers. It is a highly visible, real-time picture of the work that the Developers plan to accomplish during the Sprint in order to achieve the Sprint Goal. Consequently, the Sprint Backlog is updated throughout the Sprint as more is learned. It should have enough detail that they can inspect their progress in the Daily Scrum.

-The Scrum Guide

HOW MUCH TIME SHOULD THE PRODUCT OWNER SPEND WORKING WITH DEVELOPERS?

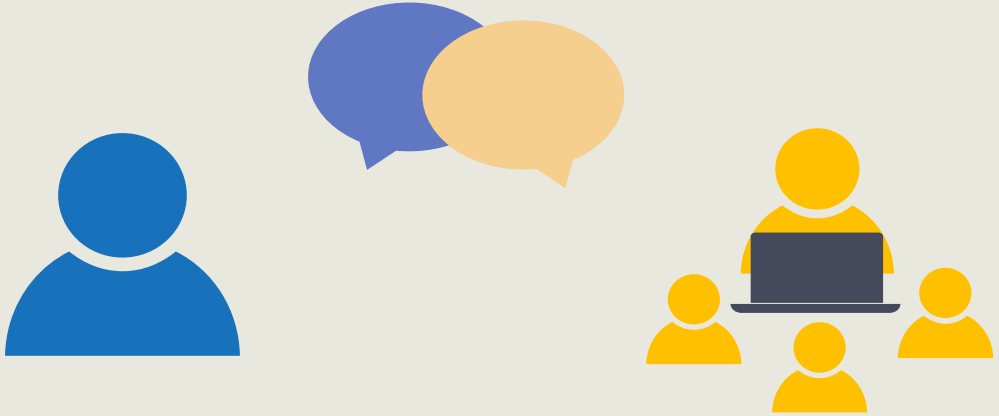
THERE IS NO PRESCRIBED AMOUNT.

- The Product Owner should be confident that the increment will meet the intended value
- If they are **working together too much** it could take away from Developer productivity
  - Product Owner may need to better understand stakeholder needs
  - Be aware of attempts to “manage” Developers
- If they are **working together too little** it could result in lack of transparency
  - The product increment may not reflect what the Product Owner thought was agreed to

Developers

Product Owner

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.

PROBLEMS WITH NOT ENOUGH DETAIL

Team is not familiar with the user stories

Multiple questions about the work reduce productivity

Delays while waiting for answers or clarification from the Product Owner/stakeholders

The story is vulnerable to inflation

PROBLEMS WITH TOO MUCH DETAIL

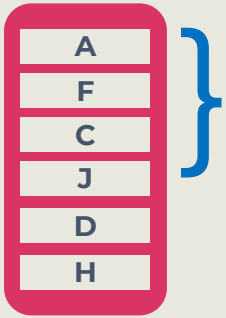
- Excessive planning
- Rework
- Anti-agile

DECOMPOSING THE WORK

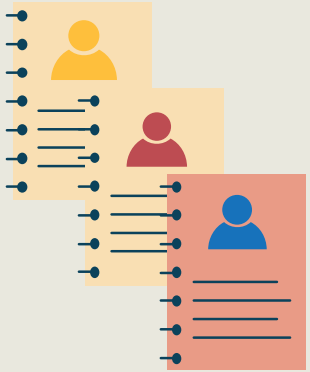
- Decompose the work for no more than 2 Sprints
- The Developers will decide how to approach the work
- User stories with uncertainty may not be decomposed into tasks right away

Prioritized Features
"User Story Backlog"

Product backlog



User stories



Tasks

One feature may equal one or more user stories.

AGILE ESTIMATING TECHNIQUES

Story points

- Relative estimation
- Arbitrary measure
- Usually used by Scrum teams
- Express effort required to implement a story
- 3 items taken into consideration: level of complexity, level of unknowns, effort to implement.

FACTORS IN ESTIMATING

Volume of work
How much effort?

Uncertainty
How risky is the work?

Complexity
How complex is the work?

T-SHIRT SIZING

Quick and easy technique

Absolute value not considered

Relative estimates

STORY POINTS

Relative sizing

- We aren’t good at absolute estimate
- We are better at relative estimates

Not tied to days, hours, or dates

- Removes pressure or emotion

Based on quantity of work, not speed Unique to a team

- Not comparable to the work of other teams
- Removes competition between teams

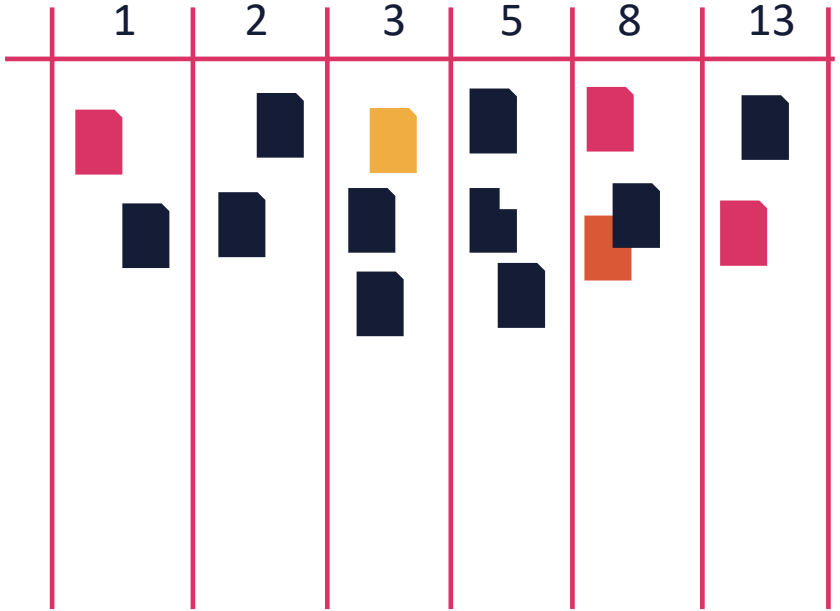
Reference for future estimates

Reserves and buffers are not necessary

While story points is the most commonly used metric, teams may choose any unit to represent work.

PLANNING POKER

- Uses Fibonacci sequence
- Each player receives a deck of cards
- Facilitator reads a user story
- On the count of 3, everyone shows their estimate
- Purpose is to build consensus
- Close to consensus, move on and round to higher number
- Scattered estimates, discuss and estimate again
- Estimates are approximates



AFFINITY ESTIMATES

Quick and easy method

Group User Stories

- Follows estimating
- Similar sizes are grouped together

Prevents story point inflation

TEAM VELOCITY

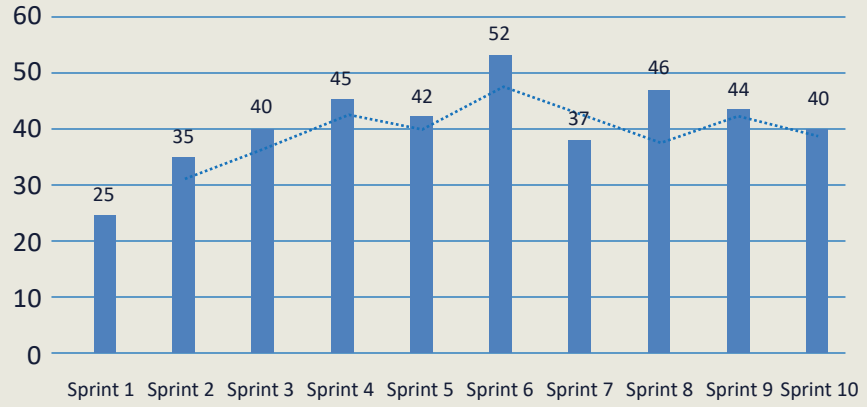
Velocity:

- Actual amount of development work completed per a certain amount of time or time-box
- Usually measured using a sprint as the time-box
- Used to estimate how quickly a certain amount of work can be completed
- Expressed as points (typically)
- Useful for forecasting

Use historical velocity data and take an average

If first time:

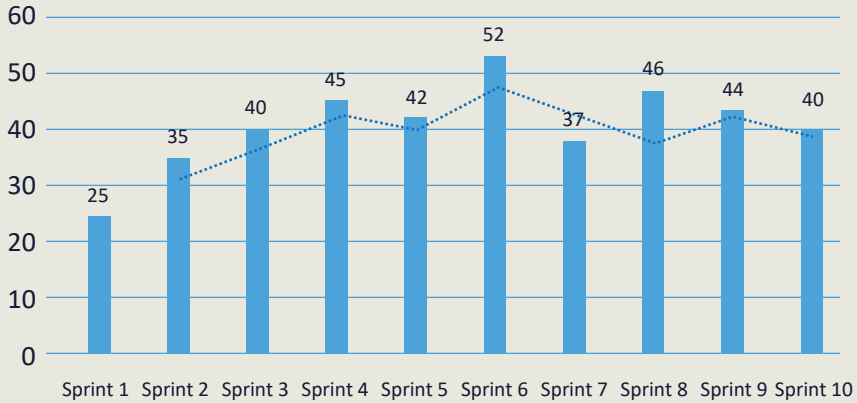
- Historical value from other projects
- Run a few iterations for a baseline



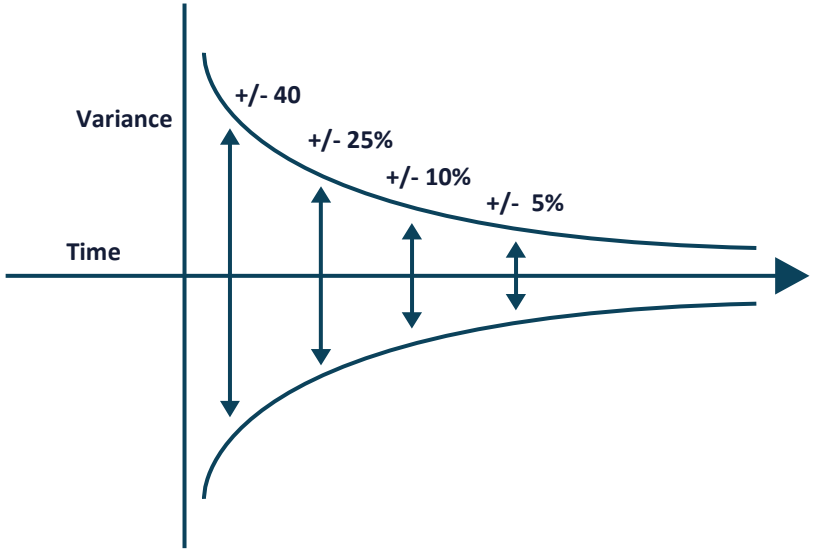
Velocity is <u>not</u> a measure of product value.

VELOCITY VS CAPACITY

Velocity is based on story points achieved historically



Capacity is based on team’s availability to do the work



Barry Boehm's Estimate Convergence Graph

CONE OF UNCERTAINTY

- The longer the project forecast, the less certain our scope will be.
- The uncertainty about scope reduces with each iteration.
- Can help the Product Owner estimate a range of the number of Sprints needed to complete the remaining work.

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