PROFESSIONAL SCRUM MASTER LEVEL 1 (PSM 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 Scrum Masters unless you need a refresher
• anyone who would like to **add Agile to their existing project management knowledge**	• not aligned to the Project Management Institute's PMP or Agile certifications
• students who might be interested in pursuing the **PSM I exam**. Also, students who would like a **foundational class before pursuing the PSM II**.	• not limited to PSM I and PSM II candidates. Everyone is welcome!

COMPARISON OF PSM I AND PSM II CURICULUM

TOPIC	PSM I	PSM II
Scrum framework	✓	✓
Scrum theory and principles	✓	✓
Cross-functional teams	✓	✓
Coaching and facilitation	✓	✓
Done and not done		✓
Maximizing value		✓
Product backlog management		✓
Scaling fundamentals		✓

ABOUT KEN SCHWABER AND SCRUM.ORG

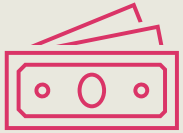
Timeline

- **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

PSM I EXAM DETAILS



$150 per attempt

Multiple choice
Multiple answer True/False

Recommended reading
Scrum.org

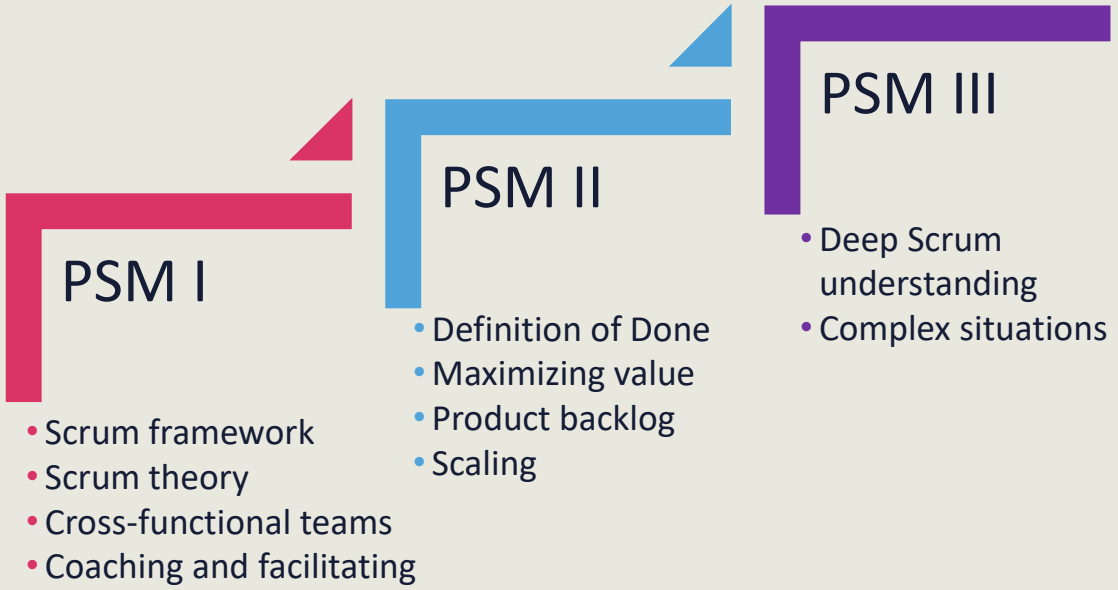
60 minutes

No prerequisites

80 questions

PROFESSIONAL SCRUM MASTER JOURNEY

Progression of certifications



INDEPENDENT STUDY

Articles

Videos

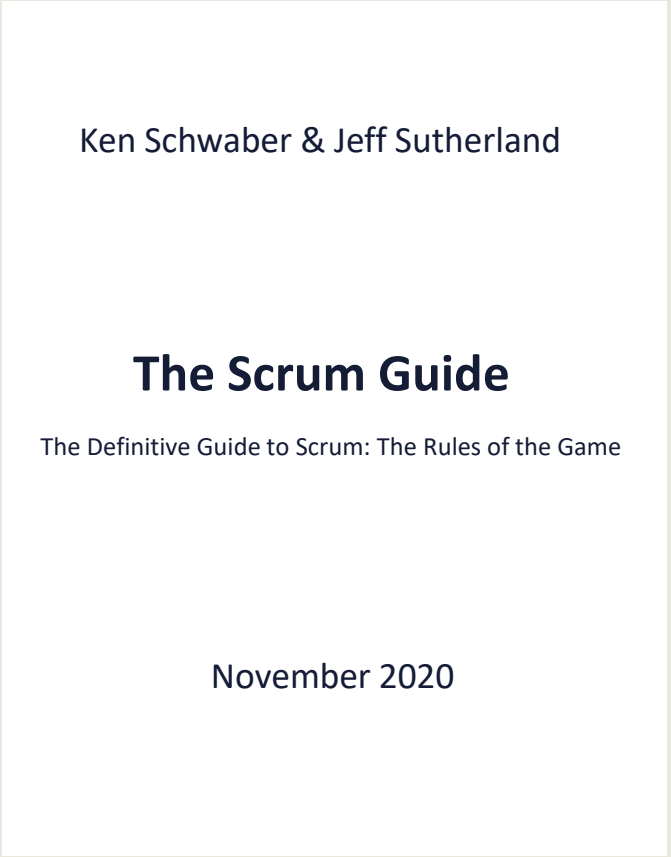
Podcasts

Practice

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

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 red 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

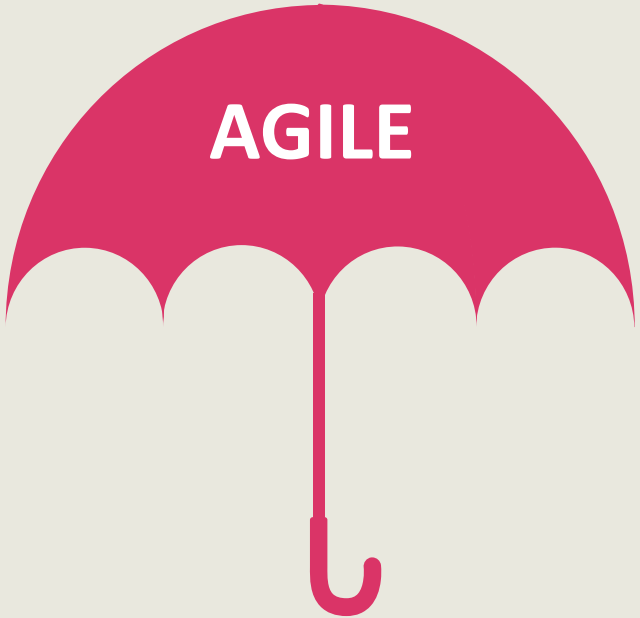
There are over a dozen agile methodologies

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
- Feature-driven development (FDD)
- Dynamic Systems Development Method (DSDM)
- Crystal



CREATING COMPLEX PRODUCTS IN COMPLEX ENVIRONMENTS IN ANY INDUSTRY

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”

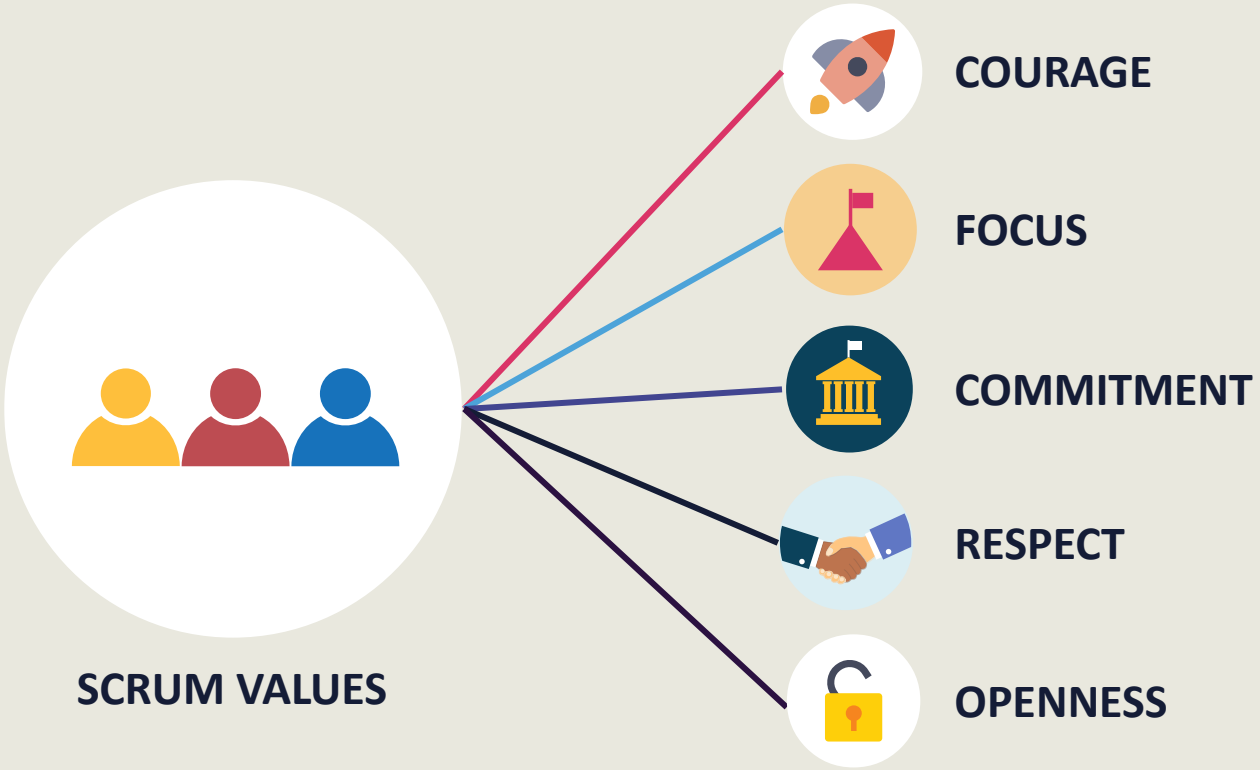
The term “Scrum” comes from rugby.

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

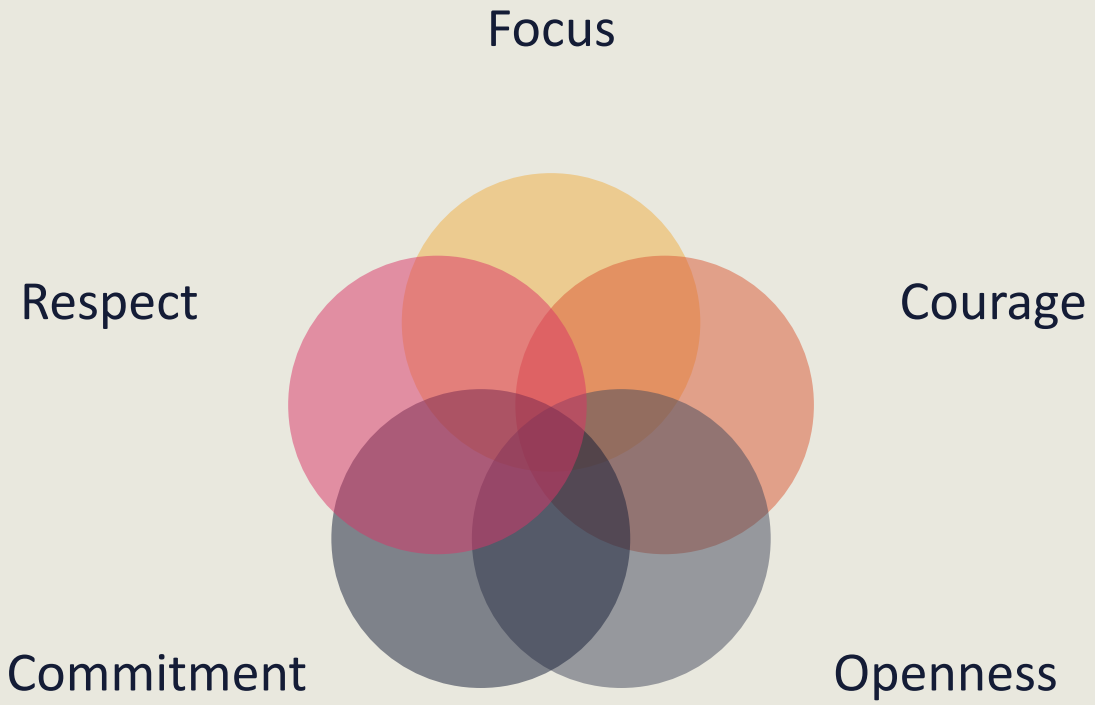
SCRUM.ORG



VALUES ARE CONNECTED

A team member begins to gossip and spread rumors about another colleague.

Which Scrum value is impacted?



THE SCRUM TEAM

Includes:

Developers

Scrum Master

Product Owner

Developers

Scrum Master

Product Owner

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

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 such as “QA” or “Tester”
- Every necessary skillset is represented

SCRUM MASTER

Servant leader to Developers

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

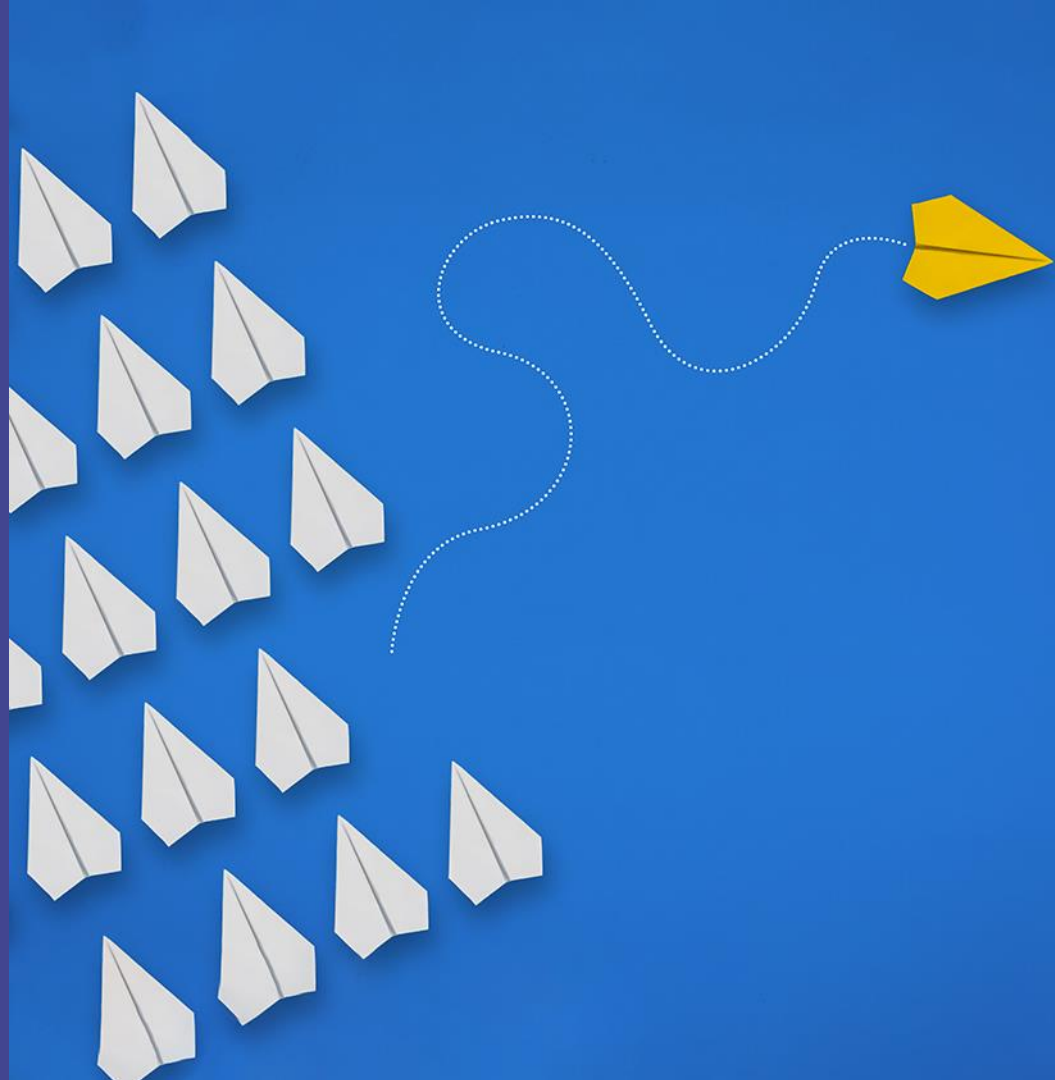
Coaches other team members

Assists Product Owner with managing backlog

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				✓

PROBLEM WITH DUAL ROLES

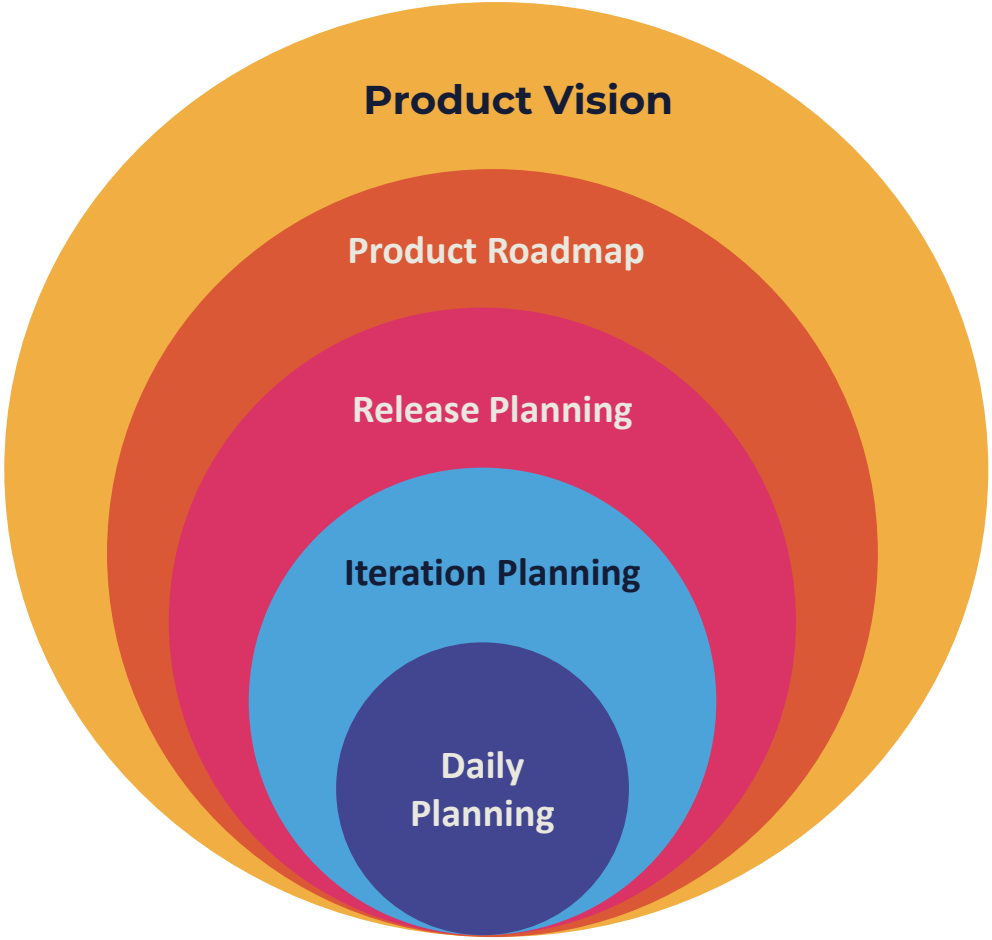
- Product owner promotes value delivery and new functionality
- Developers sets realistic expectations and perform the work
- Scrum Master protects the team and promotes governance

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 PRODUCT VISION

- Interview stakeholders
- Focus on how a product adds value
- Motivates Developers

PRODUCT VISION

Why you're building a product

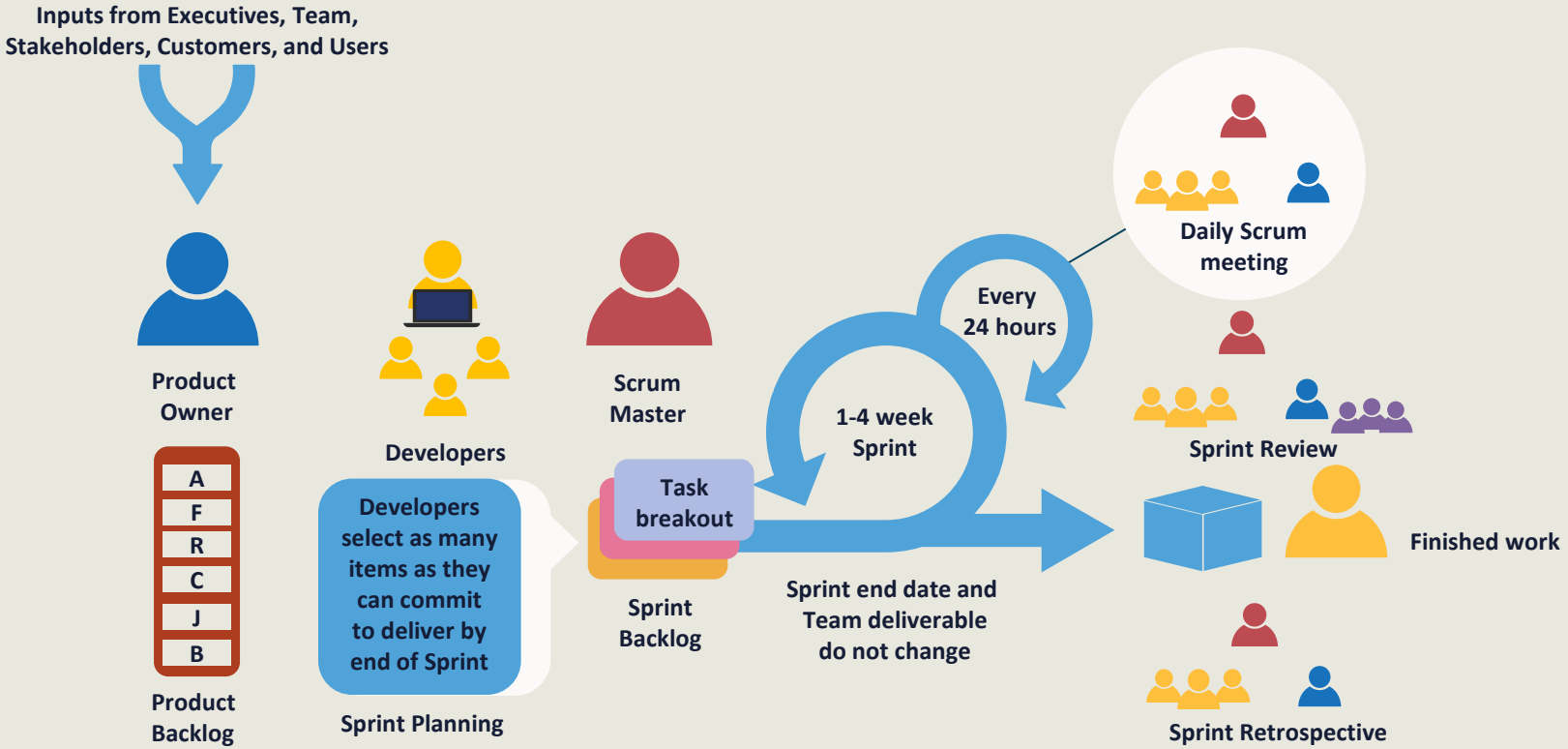
Benefits of product

Who you're building it for

Why you are positioned to develop it

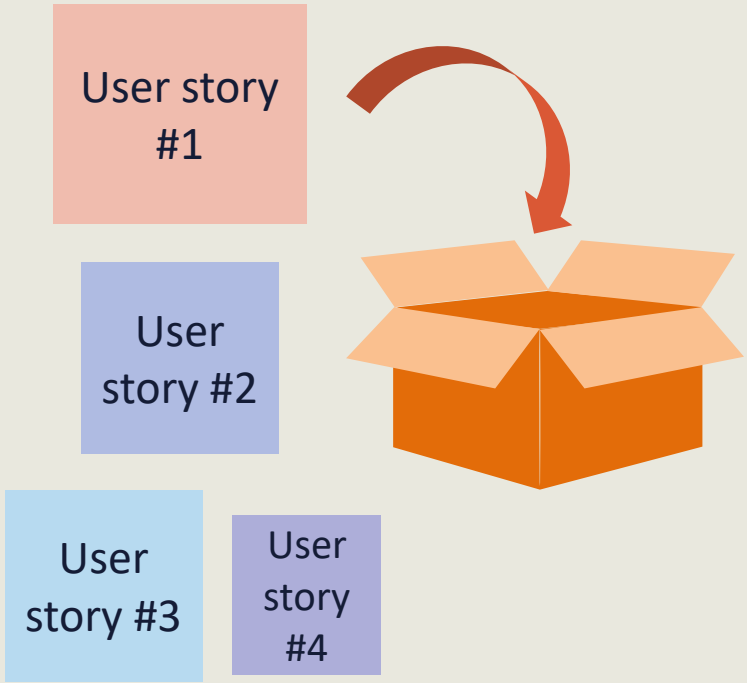
Since scope is evolving it is important to share an understanding of what is being created

SCRUM FRAMEWORK



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



USER STORIES

Short, simple descriptions of a feature

Told from the user's perspective

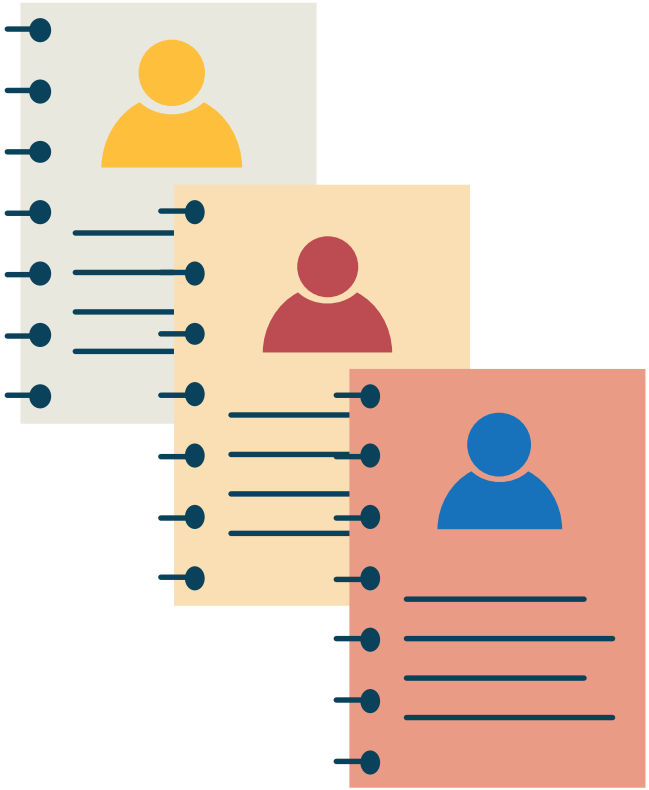
When large or complex, can be called "epics"

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."



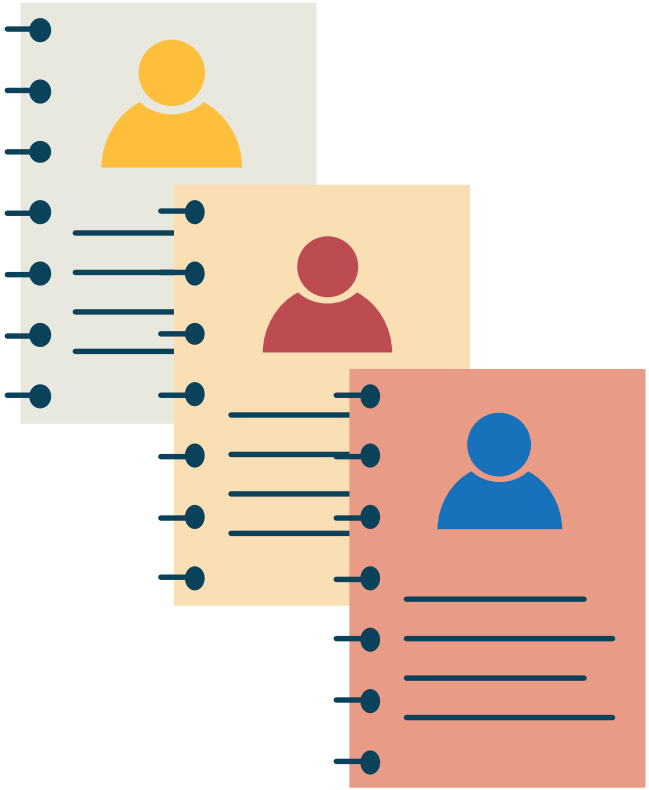
FORMATTING USER STORIES

Keep them simple

- **Gather feedback**
- **Experiment**
- **Use storyboards**
- **Use annotations**
- **Provide explanations**

INVEST criteria

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



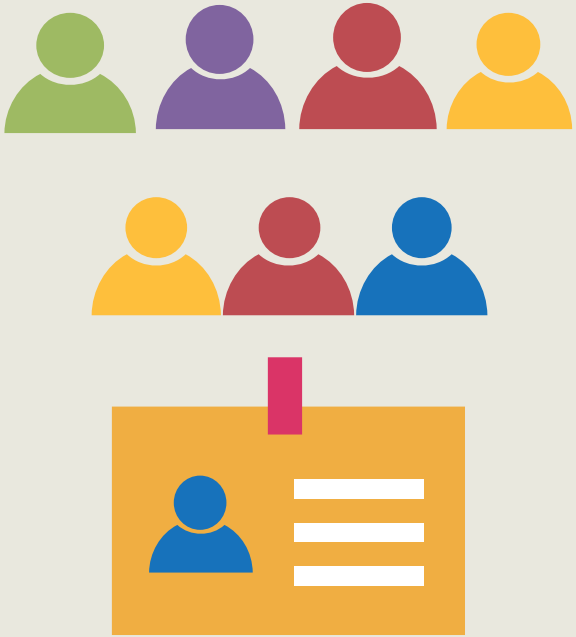
PERSONAS: BEST PRACTICES

- Based on reality
- Specific
- Have defined goals
- Include negative personas
- The end user you are *not* designing for
  - Example: the user who always cancels after the free trial
- Ideally 3 or less, or the product may become confusing

PERSONAS

- Quick reminder of stakeholder needs
- Composite
- Represents majority of actual users
- Not a replacement for stakeholders
- Focus on value and priorities

Mario is an employee of ABC Company. He must use his ID badge to access his work computer. For security reasons, the computer automatically logs users out after 5 minutes of inactivity. Mario would like to remain logged in while he is sitting at his desk.



SPIDR APPROACH TO SPLITTING STORIES

Spike

Paths

Interfaces

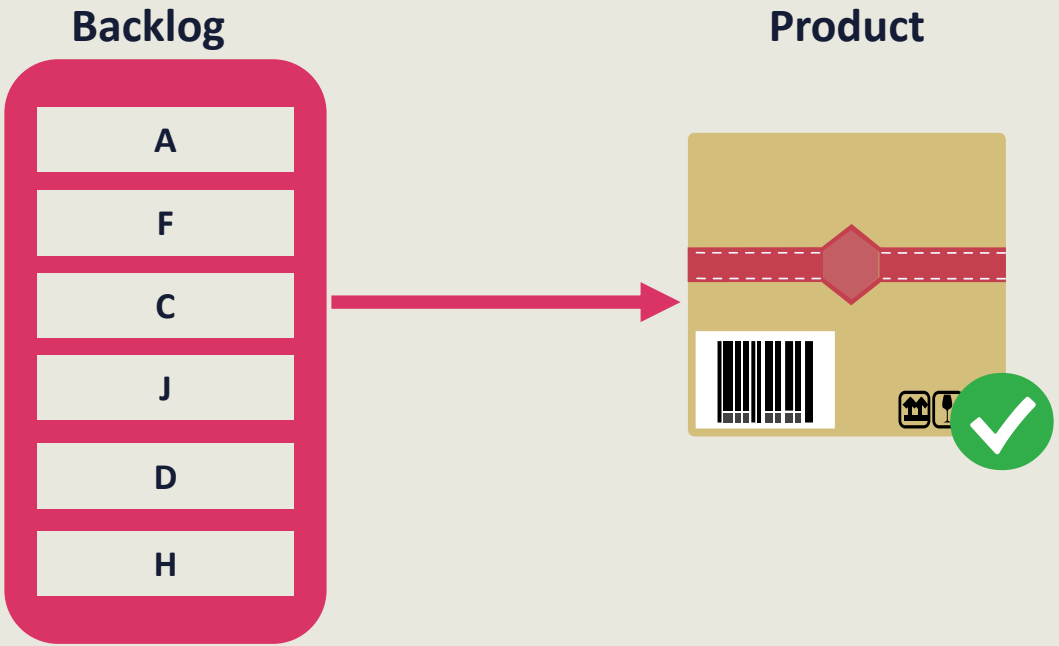
Data

Rules

"Five Simple but Powerful Ways to Split User Stories" mountaingoatsoftware.com. Accessed April 2020.
https://www.mountaingoatsoftware.com/blog/five-simple-but-powerful-ways-to-split-user-stories

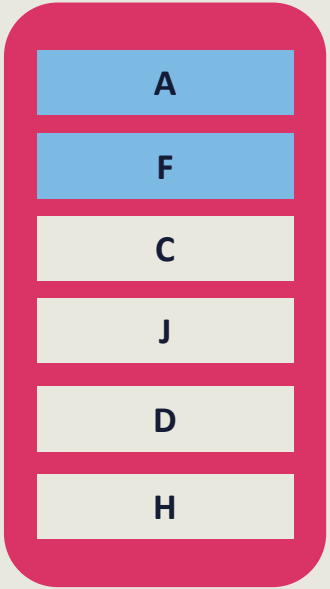
PRODUCT BACKLOG

- Prioritized list of everything that is needed in the product
- All work should be included
  - Bug fixes
  - Security features
  - Changes
- Single source of product requirements
- Always changing
- Items are added, dropped, and reprioritized based on value
- The product is built incrementally based on work selected from the backlog



PRODUCT INCREMENT

- The result of the latest sprint
- Demo during sprint review
- Must meet the “definition of done” established during planning
- Incomplete work is not demonstrated

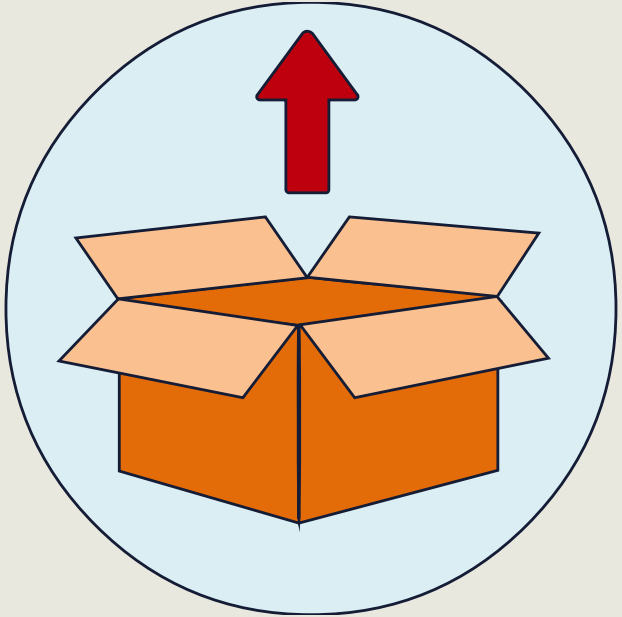


SMALL RELEASES

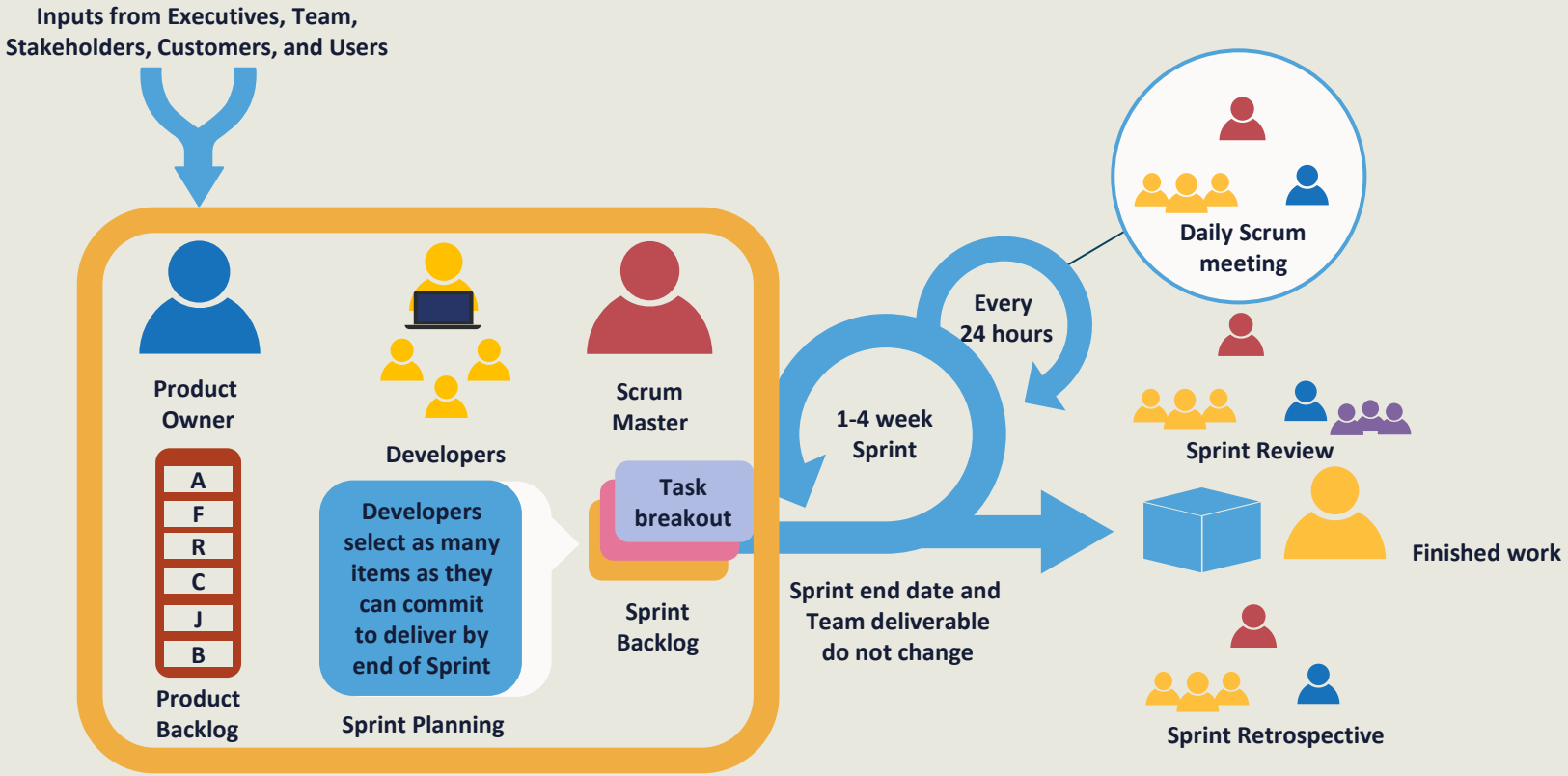
Demonstrates progress

Increases visibility to the customer

Smaller increments means rapid deployments



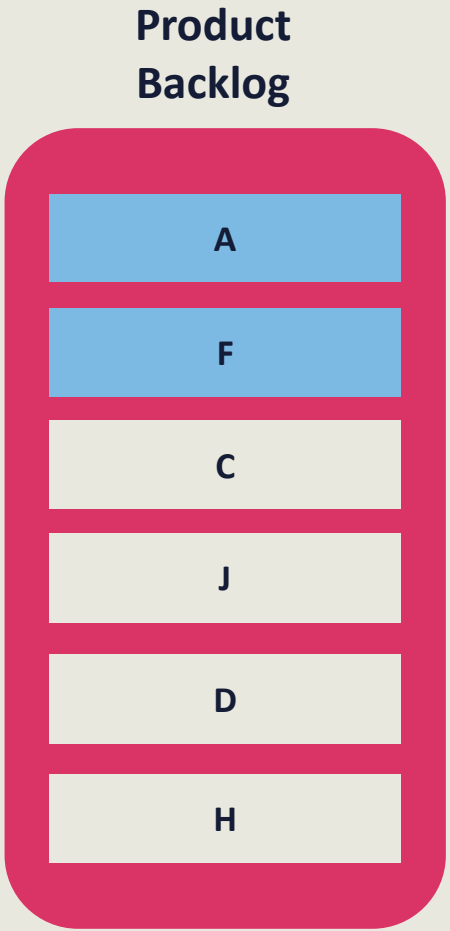
SPRINT PLANNING



SPRINT PLANNING

- Includes all Scrum Team members
- Product Owner presents the updated backlog
- Developers estimate the work
- Work is selected from the product backlog to create the Sprint backlog.
- Developers commit to a set of deliverables for the Sprint
- Establish a “Definition of Done”

- Typically 2 hours per week of Sprint



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

SPRINT GOAL

- Start with high level goal for functionality needed
- Select product backlog items that align with the 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

Sprint Goal

“Why”

Product Backlog Items

“What”

Action Plan

“How”

WHAT IS IN THE SPRINT BACKLOG?

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

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

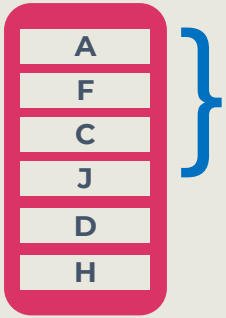
- Excessive planning
- Rework
- Anti-agile

DECOMPOSING THE WORK

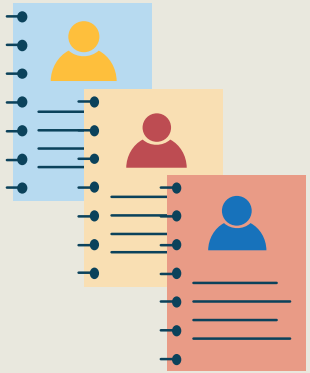
- Decompose the work for no more than 2 Sprints
- The Developers will decide how to approach the work
- User stories with uncertainty may not be decomposed into tasks right away

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.

AGILE ESTIMATING TECHNIQUES

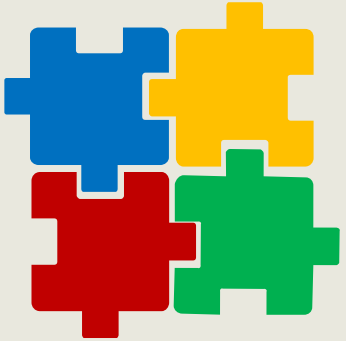
Ideal days

- **Alternative to story points**
- **Units of time rather than arbitrary measure**
- **How long to build, test, and release functionality with ideal conditions**

FACTORS IN ESTIMATING

Volume of work
How much effort?

Uncertainty
How risky is the work?



Complexity
How complex is the work?

AGILE GAMES

- Collaboration activities
- Brainstorming activities
- Retrospectives
- Drawing
- Storytelling

- Foster collaboration, communication, innovation
- Used to teach, demonstrate, improve
- Help model complex processes
- Facilitate issue resolution
- Drive good behaviors
- Overcome destructive behaviors

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

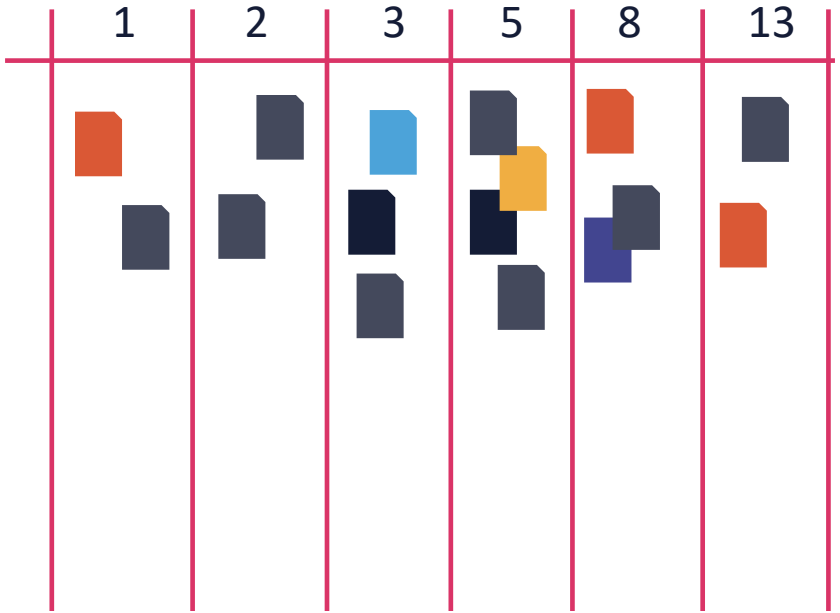
Reference for future estimates

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

RELATIVE SIZING

Quick and easy technique

Absolute value not considered

T-shirt sizing

Similar to Fibonacci

- Sizes instead of numbers

TEAM VELOCITY

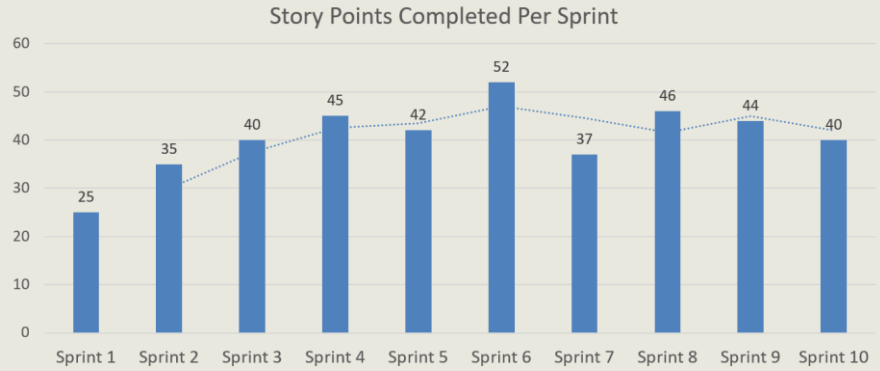
Velocity:

- 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

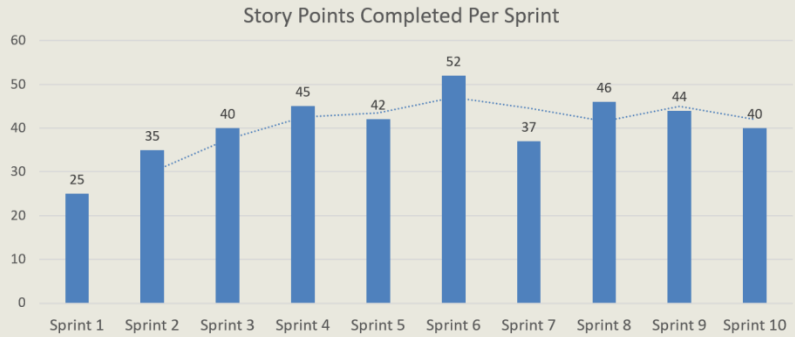
If first time:

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



VELOCITY VS CAPACITY

Velocity is based on story points achieved historically



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

SPRINT ZERO

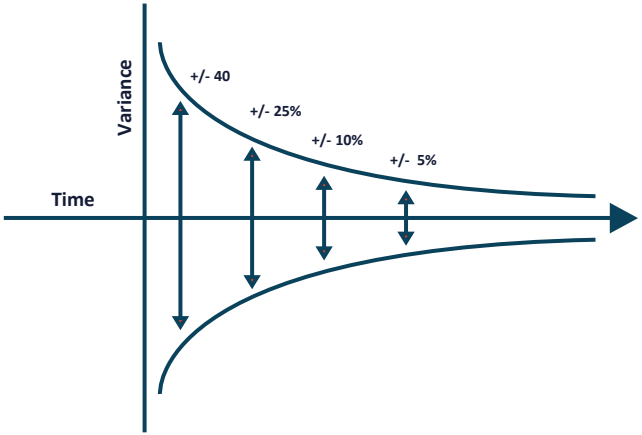
Some suggest that it…

- 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, potentially releasable product increment.



Barry Boehm's Estimate Convergence Graph

SCHEDULE BUFFERS

Not recommended

- **May appear that team is padding the schedule**
- **If used, buffer the number of sprints needed rather than individual task estimates**

Exceptions

- **When customer insists on a firm estimate**
- **Must account for contingencies**
- **When team velocity is undetermined**
- **When there are regular interruptions**

Estimate in ranges

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