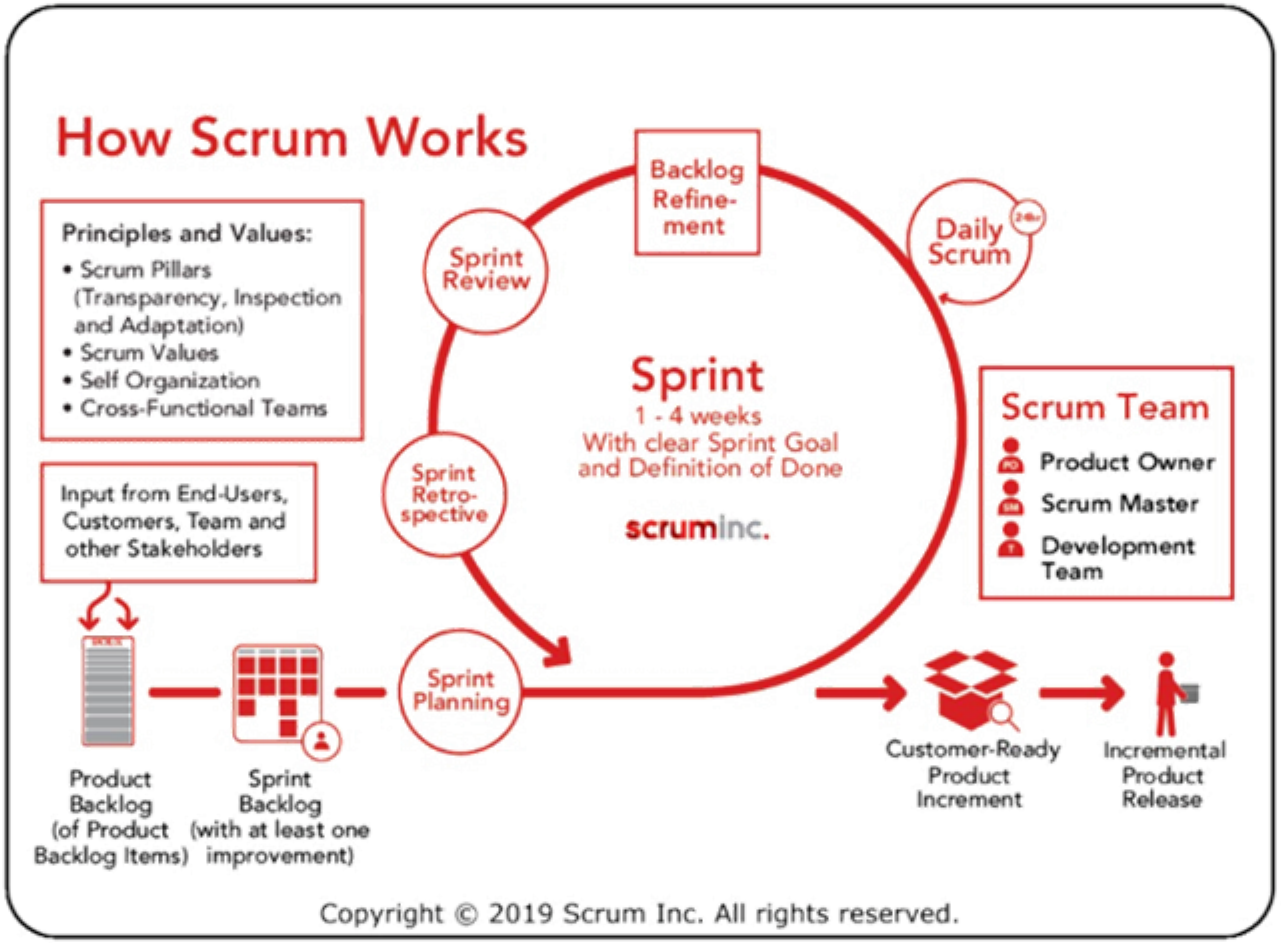
Faculty of SET / School of Computer Science

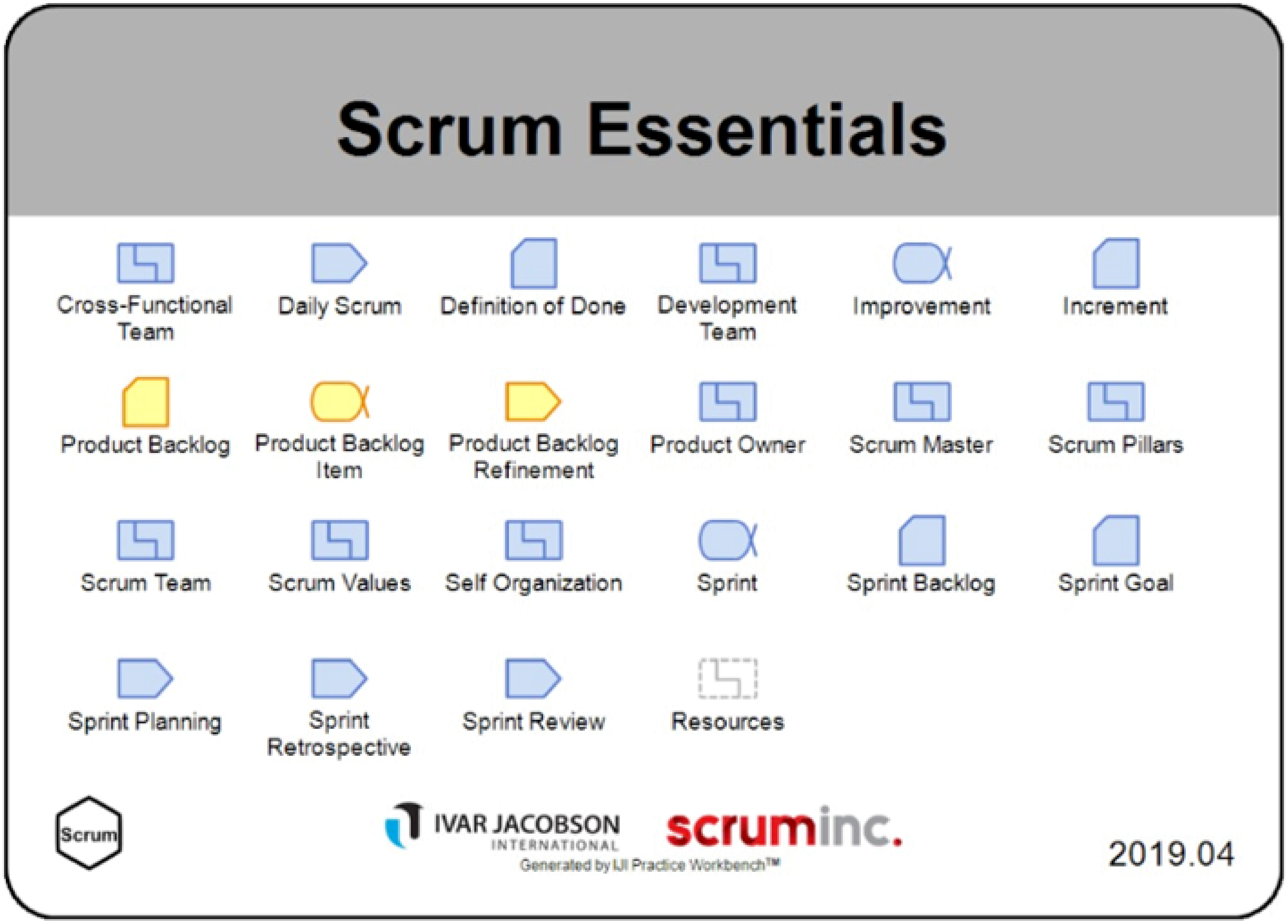
Software Engineering & Project
Lecture 2: Scrum I

Housekeeping

- Project signups are open until Friday 11:59
  - Please be considerate and ask to join groups
  - Look on the Discussions forums first

- Quiz 1 is published first thing Friday morning
  - Usually due on the Wednesday of the next week


How Scrum Works
Principles and Values:
• Scrum Pillars (Transparency, Inspection and Adaptation)
• Scrum Values
• Self Organization
• Cross-Functional Teams
Input from End-Users, Customers, Team and other Stakeholders
Backlog Refine-ment
Sprint Review
Daily Scrum
Sprint
1 - 4 weeks
With clear Sprint Goal and Definition of Done
scruminc.
Sprint Retro-spective
Scrum Team
Product Owner
Scrum Master
Development Team
Sprint Planning
Product Backlog (of Product Backlog Items)
Sprint Backlog (with at least one improvement)
Customer-Ready Product Increment
Incremental Product Release
Copyright © 2019 Scrum Inc. All rights reserved.



Scrum Essentials

Cross-Functional Team
Daily Scrum
Definition of Done
Development Team
Improvement
Increment
Product Backlog
Product Backlog Item
Product Backlog Refinement
Product Owner
Scrum Master
Scrum Pillars
Scrum Team
Scrum Values
Self Organization
Sprint
Sprint Backlog
Sprint Goal
Sprint Planning
Sprint Retrospective
Sprint Review
Resources

Scrum
IVAR JACOBSON INTERNATIONAL
scruminc.
Generated by IJI Practice Workbench™

2019.04

Lecture Content

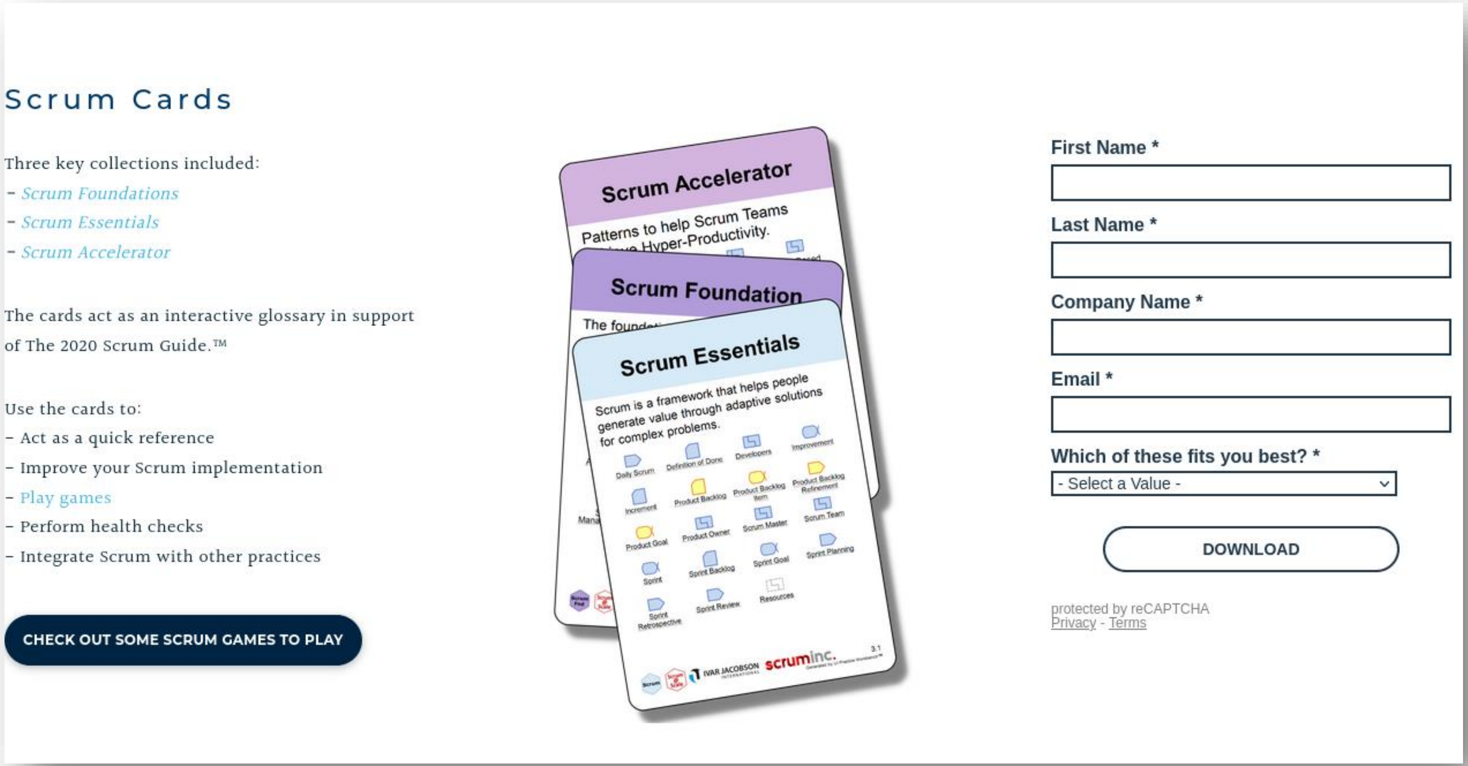
- Principles and Values of Scrum
- Scrum Roles
- Product and Sprint Backlogs
- User Stories and Tasks
- Task Board
- Time Estimation Technique
- Velocity
- Sprint Planning

History of Scrum

Jeff Sutherland
Scrum: How to do twice as much in half the time

https://youtu.be/s4thQcgLCqk

Scrum Essential Cards



Scrum Cards

Three key collections included:
- *Scrum Foundations*
- *Scrum Essentials*
- *Scrum Accelerator*

The cards act as an interactive glossary in support of The 2020 Scrum Guide.™

Use the cards to:
- Act as a quick reference
- Improve your Scrum implementation
- Play games
- Perform health checks
- Integrate Scrum with other practices

CHECK OUT SOME SCRUM GAMES TO PLAY

First Name *

Last Name *

Company Name *

Email *

Which of these fits you best? *

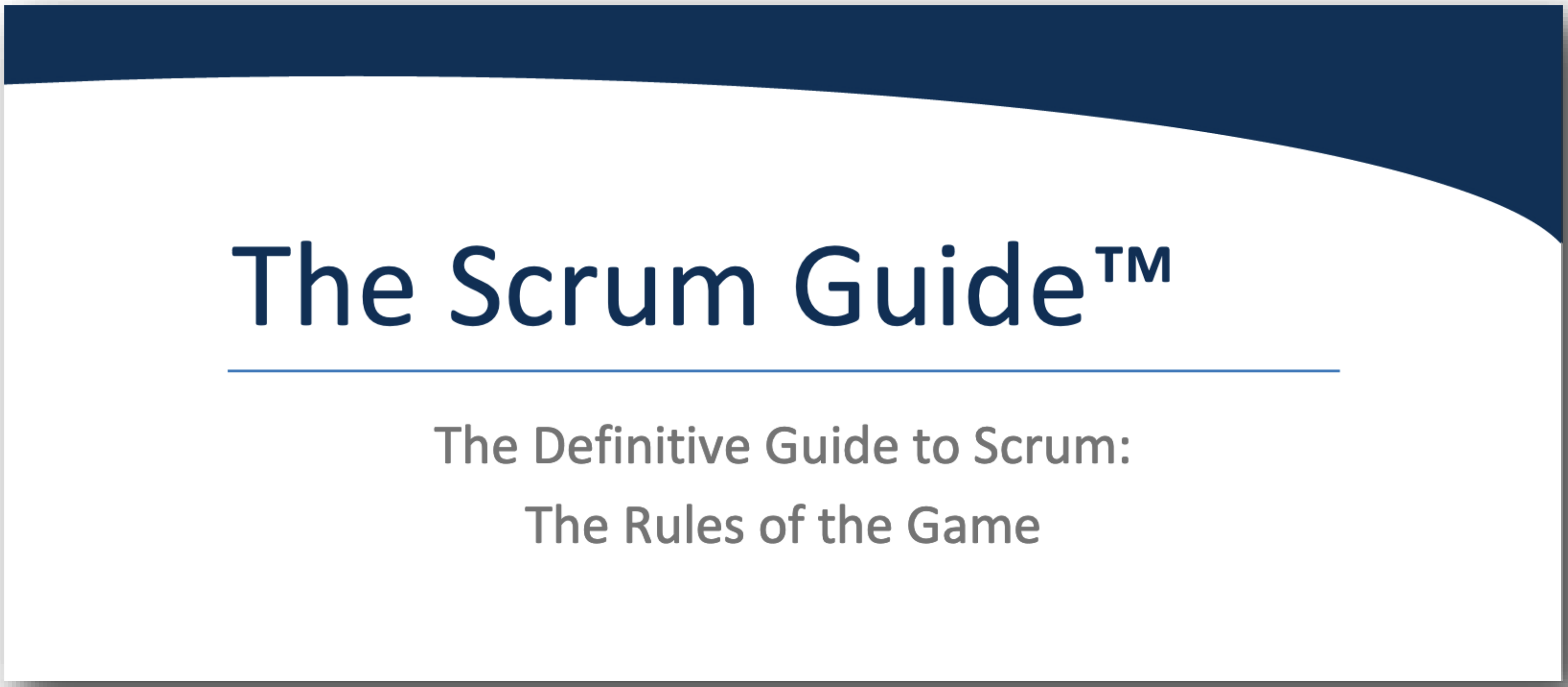
- Select a Value -

DOWNLOAD

protected by reCAPTCHA
Privacy - Terms

Download the cards from:
https://www.ivarjacobson.com/free-agile-coaching-cards

Scrum Guide



The Scrum Guide™

The Definitive Guide to Scrum:

The Rules of the Game

Download the Scrum guide (19 pages for English version) from:
https://www.scrumguides.org/

Principles and Values

- Scrum is based on **empiricism**

Principles and Values

- Scrum is based on **empiricism**

> *"Empiricism is a theory that states that* ***knowledge comes only or primarily from sensory experience****. […] Empiricism emphasizes the* ***role of empirical evidence in the formation of ideas****, rather than innate ideas or traditions."*
> https://en.wikipedia.orgwiki/Empiricism

Principles and Values

- Scrum is based on **empiricism**

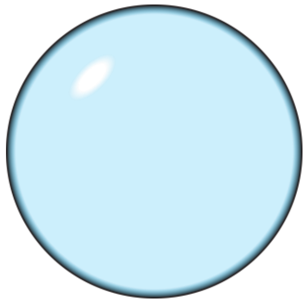
→ **Knowledge** comes from **experience**

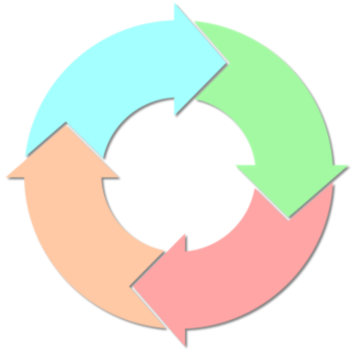
→ **Decisions** are made based on empirically derived **knowledge**

(not predefined plans/processes)

Three Pillars of Scrum

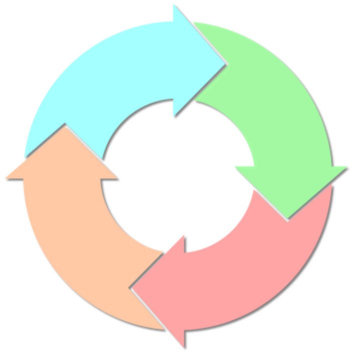
- **Transparency:** Shared understanding of process, common definition of "done"
- **Inspection:** Frequently inspect artifacts (backlogs, increments) during progress towards sprint and release goals
- **Adaptation:** If inspection reveals issues, adjustments must be made as soon as possible





Three Pillars of Scrum

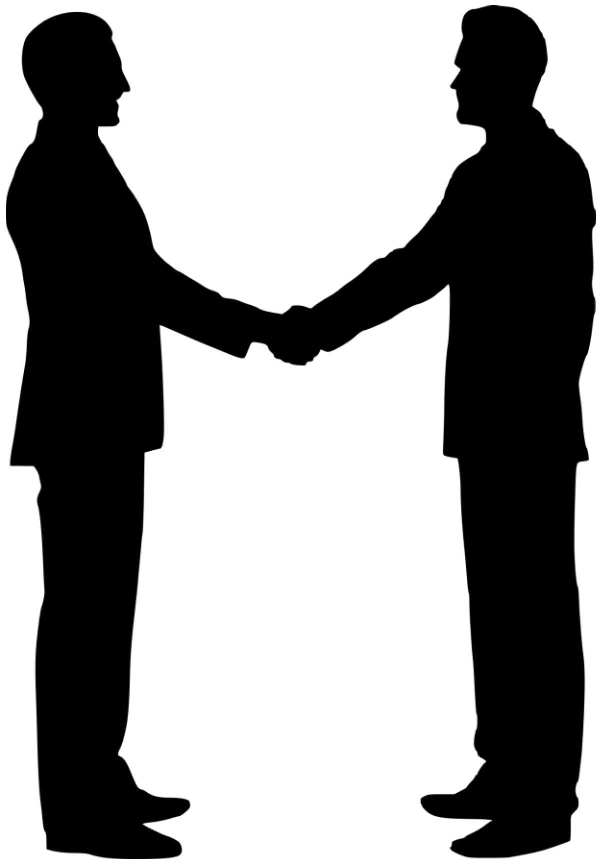
- **Inspection** and **Adaptation** mainly during:
  - Sprint Planning
  - Daily Scrum
  - Sprint Review
  - Sprint Retrospective



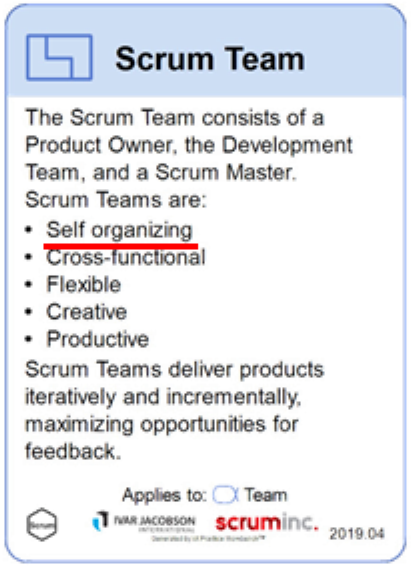
Three Pillars of Scrum

- **Inspection** and **Adaptation** mainly during:
  - Sprint Planning
  - Daily Scrum
  - Sprint Review
  - Sprint Retrospective

= Trust



Scrum Roles



Scrum Team

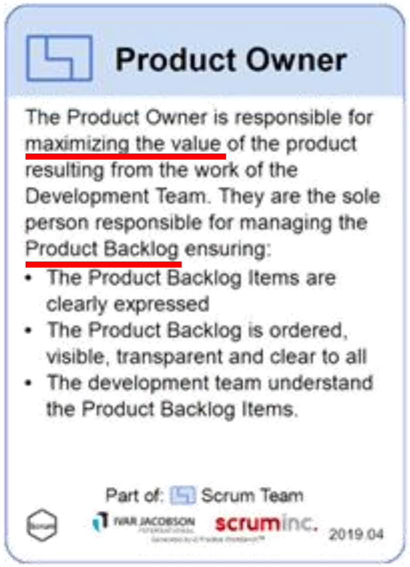
The Scrum Team consists of a Product Owner, the Development Team, and a Scrum Master.
Scrum Teams are:
- Self organizing
- Cross-functional
- Flexible
- Creative
- Productive

Scrum Teams deliver products iteratively and incrementally, maximizing opportunities for feedback.

Applies to: Team

IVAR JACOBSON scruminc. 2019.04

ST



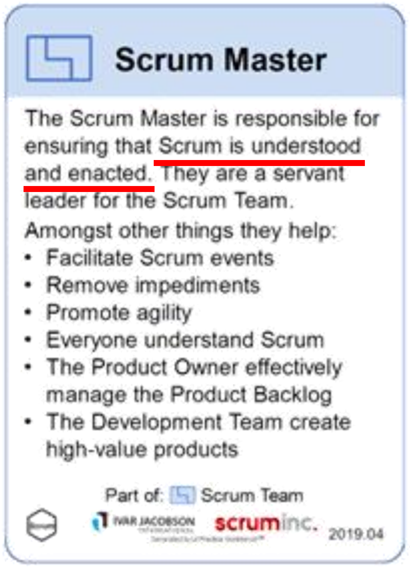
Product Owner

The Product Owner is responsible for maximizing the value of the product resulting from the work of the Development Team. They are the sole person responsible for managing the Product Backlog ensuring:
- The Product Backlog Items are clearly expressed
- The Product Backlog is ordered, visible, transparent and clear to all
- The development team understand the Product Backlog Items.

Part of: Scrum Team

IVAR JACOBSON scruminc. 2019.04

PO



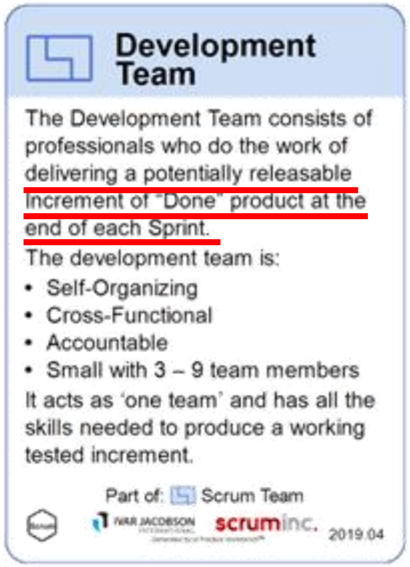
Scrum Master

The Scrum Master is responsible for ensuring that Scrum is understood and enacted. They are a servant leader for the Scrum Team.
Amongst other things they help:
- Facilitate Scrum events
- Remove impediments
- Promote agility
- Everyone understand Scrum
- The Product Owner effectively manage the Product Backlog
- The Development Team create high-value products

Part of: Scrum Team

IVAR JACOBSON scruminc. 2019.04

SM



Development Team

The Development Team consists of professionals who do the work of delivering a potentially releasable Increment of "Done" product at the end of each Sprint.
The development team is:
- Self-Organizing
- Cross-Functional
- Accountable
- Small with 3 – 9 team members

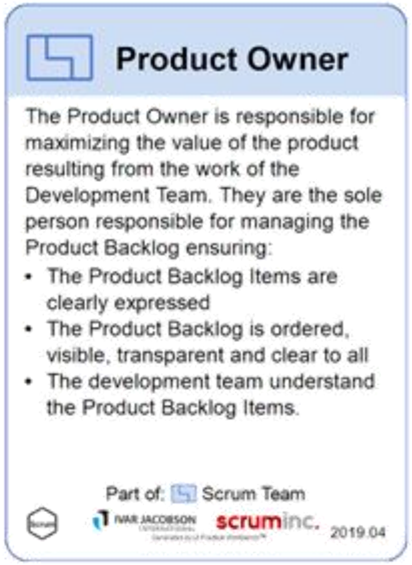
It acts as 'one team' and has all the skills needed to produce a working tested increment.

Part of: Scrum Team

IVAR JACOBSON scruminc. 2019.04

DT

Product Owner (PO)


Product Owner

The Product Owner is responsible for maximizing the value of the product resulting from the work of the Development Team. They are the sole person responsible for managing the Product Backlog ensuring:

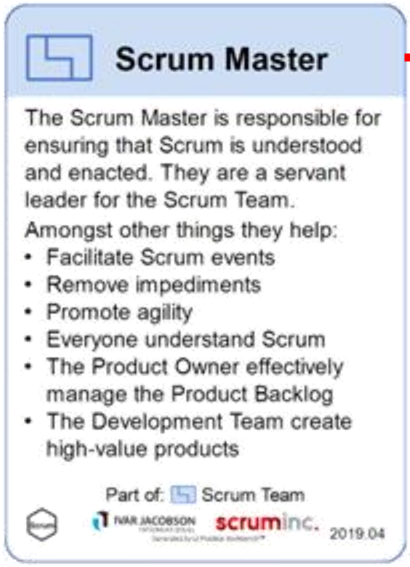
- The Product Backlog Items are clearly expressed
- The Product Backlog is ordered, visible, transparent and clear to all
- The development team understand the Product Backlog Items.

Part of: Scrum Team

IVAR JACOBSON scruminc. 2019.04

- Holds the **vision** of the product
- Represents the **business/customers**
- Owns the **product backlog**
- **Prioritizes** product backlog items (e.g., user stories)
- Creates **acceptance criteria** for user stories (i.e., tasks)
- Is available to **answer** team members' **questions**

Scrum Master (SM)



Scrum Master

The Scrum Master is responsible for ensuring that Scrum is understood and enacted. They are a servant leader for the Scrum Team.

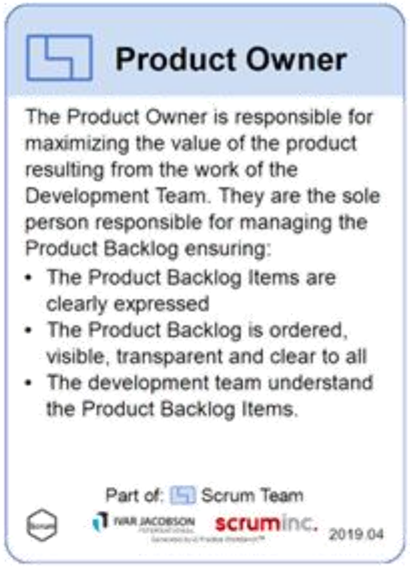
Amongst other things they help:

- Facilitate Scrum events
- Remove impediments
- Promote agility
- Everyone understand Scrum
- The Product Owner effectively manage the Product Backlog
- The Development Team create high-value products

Part of: Scrum Team

2019.04

Services



Product Owner

The Product Owner is responsible for maximizing the value of the product resulting from the work of the Development Team. They are the sole person responsible for managing the Product Backlog ensuring:

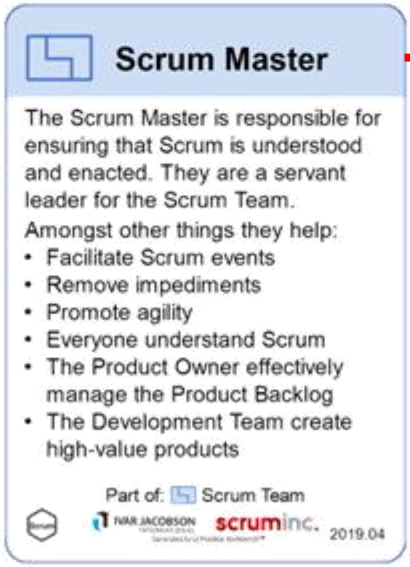
- The Product Backlog Items are clearly expressed
- The Product Backlog is ordered, visible, transparent and clear to all
- The development team understand the Product Backlog Items.

Part of: Scrum Team

2019.04

- Ensuring that **goals, scope, and product domain** are understood by everyone on the Scrum team as well as possible
- Helping to effectively **manage product backlog** to maximize value
- Facilitating **scrum events**

Scrum Master (SM)



Scrum Master

The Scrum Master is responsible for ensuring that Scrum is understood and enacted. They are a servant leader for the Scrum Team.

Amongst other things they help:

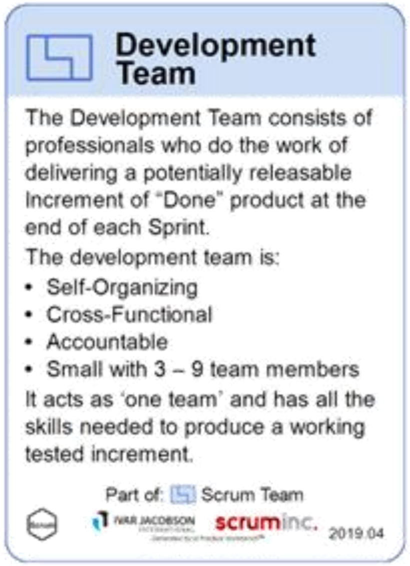
- Facilitate Scrum events
- Remove impediments
- Promote agility
- Everyone understand Scrum
- The Product Owner effectively manage the Product Backlog
- The Development Team create high-value products

Part of: Scrum Team

IVAR JACOBSON scruminc. 2019.04

Services

- Coaching in **self-organization**
- Removing **impediments**
- Facilitating **Scrum events**



Development Team

The Development Team consists of professionals who do the work of delivering a potentially releasable Increment of "Done" product at the end of each Sprint.

The development team is:

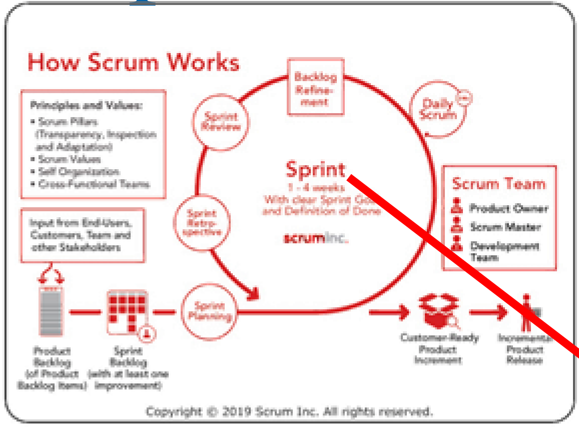
- Self-Organizing
- Cross-Functional
- Accountable
- Small with 3 – 9 team members

It acts as 'one team' and has all the skills needed to produce a working tested increment.

Part of: Scrum Team

IVAR JACOBSON scruminc. 2019.04

Sprints

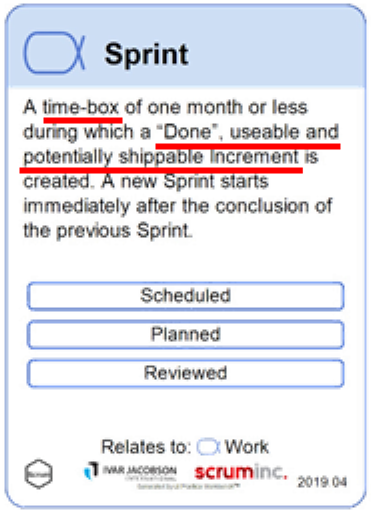

How Scrum Works
Principles and Values:
• Scrum Pillars (Transparency, Inspection and Adaptation)
• Scrum Values
• Self Organization
• Cross-Functional Teams
Input from End-Users, Customers, Team and other Stakeholders
Backlog Refinement
Daily Scrum
Sprint Review
Sprint Retrospective
Sprint Planning
Sprint
1 - 4 weeks
With clear Sprint Goal and Definition of Done
scruminc.
Scrum Team
Product Owner
Scrum Master
Development Team
Product Backlog (of Product Backlog Items)
Sprint Backlog (with at least one improvement)
Customer-Ready Product Increment
Incremental Product Release
Copyright © 2019 Scrum Inc. All rights reserved.

During a sprint:

- Usually no changes to sprint goal and definition of done
- Scope (details of user stories) may be renegotiated between PO and DT
- Only PO can cancel a sprint (e.g., if sprint goal becomes obsolete)

Set of User Stories

Non-functional requirements



Sprint

A time-box of one month or less during which a "Done", useable and potentially shippable Increment is created. A new Sprint starts immediately after the conclusion of the previous Sprint.

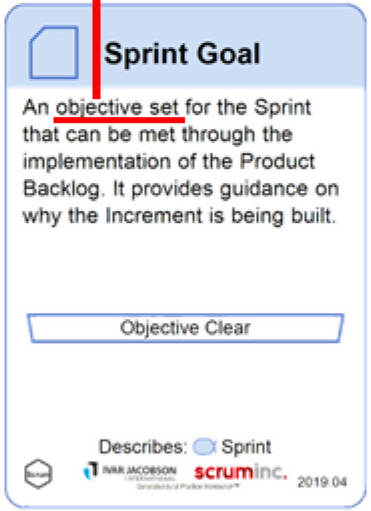
Scheduled

Planned

Reviewed

Relates to: Work

IVAR JACOBSON scruminc. 2019.04



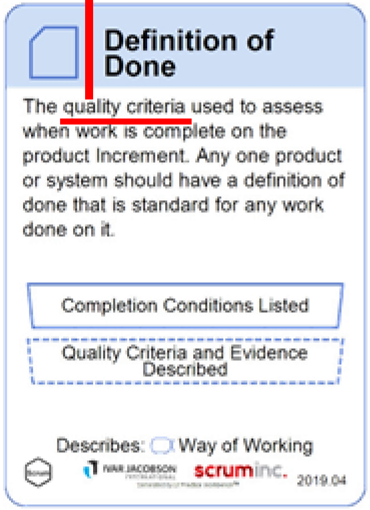
Sprint Goal

An objective set for the Sprint that can be met through the implementation of the Product Backlog. It provides guidance on why the Increment is being built.

Objective Clear

Describes: Sprint

IVAR JACOBSON scruminc. 2019.04



Definition of Done

The quality criteria used to assess when work is complete on the product Increment. Any one product or system should have a definition of done that is standard for any work done on it.

Completion Conditions Listed

Quality Criteria and Evidence Described

Describes: Way of Working

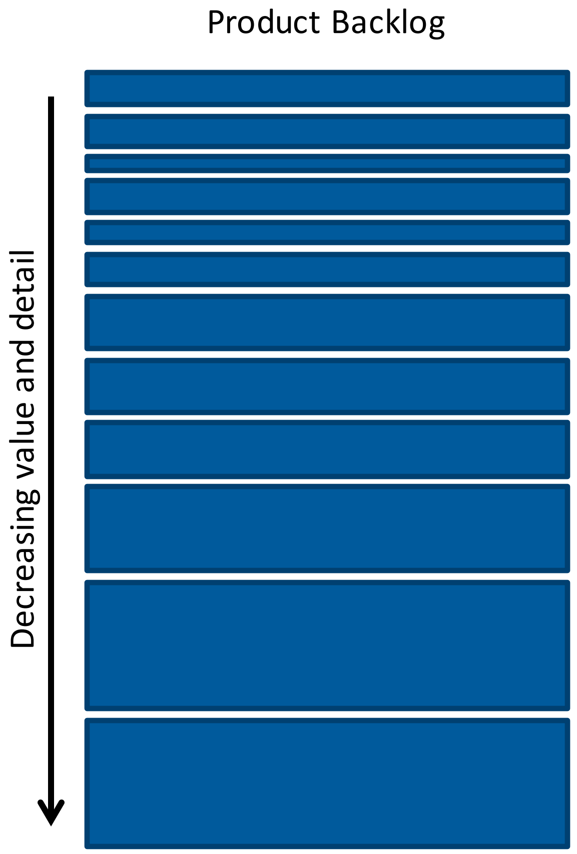
IVAR JACOBSON scruminc. 2019.04

Sprint Planning

1. PO: **What** can be done in the upcoming sprint?
2. DT: **How** will the chosen work get done?

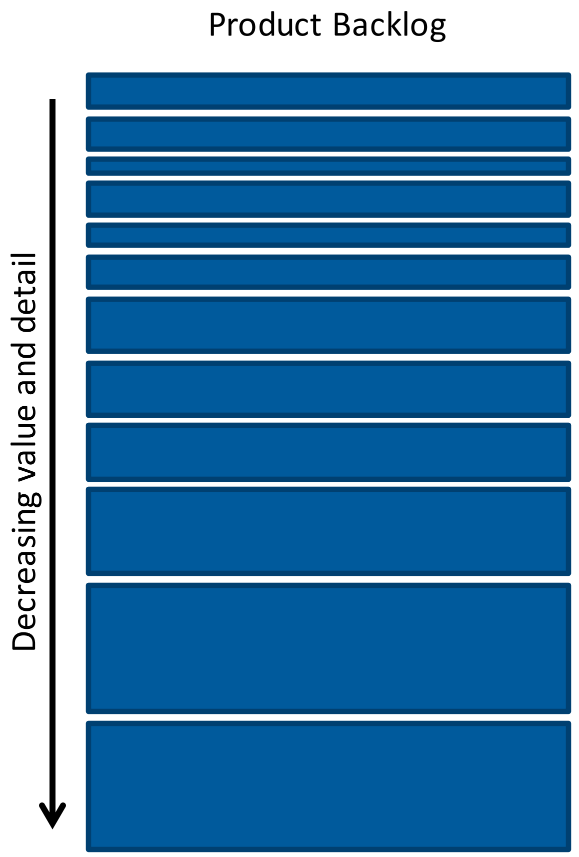
- PO proposes **set of user stories** for sprint, which are then collaboratively discussed
- **DT makes final decision**
- SM ensures that meeting takes place, attendees know purpose of meeting, and meeting stays within timebox
- **Inputs:** Product backlog, current increment, definition of done, past performance and projected capacity of DT
- **Output:** Sprint goal, sprint backlog (user stories and tasks)

Product Backlog



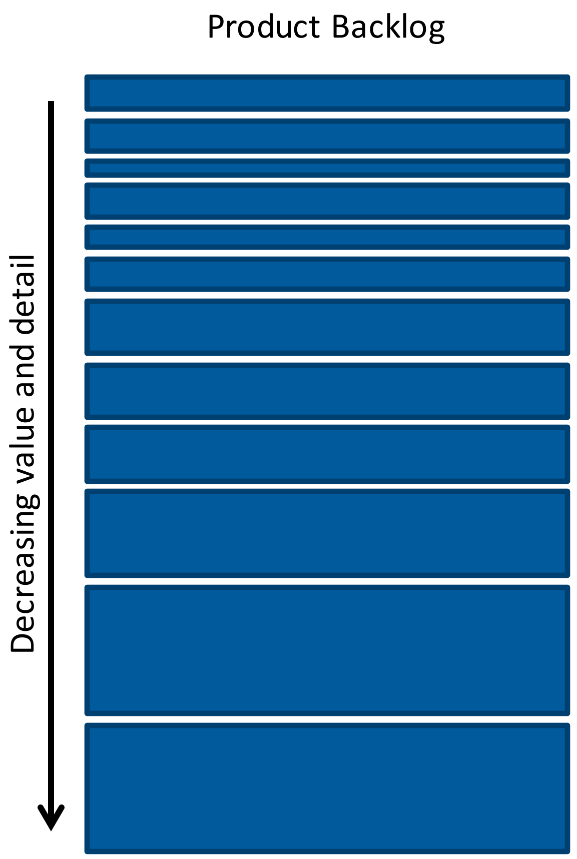
- Ordered and emerging list of **user needs** plus **everything else** to fulfil the product vision
- Product Backlog Items:
  - **User Stories**
  - Bug fixes
  - Refactorings (reduction of technical debt)
  - Here: GitHub issues (see tool lectures)
- PO is ultimately responsible for content and state of Product Backlog, but everyone can contribute

Product Backlog Refinement

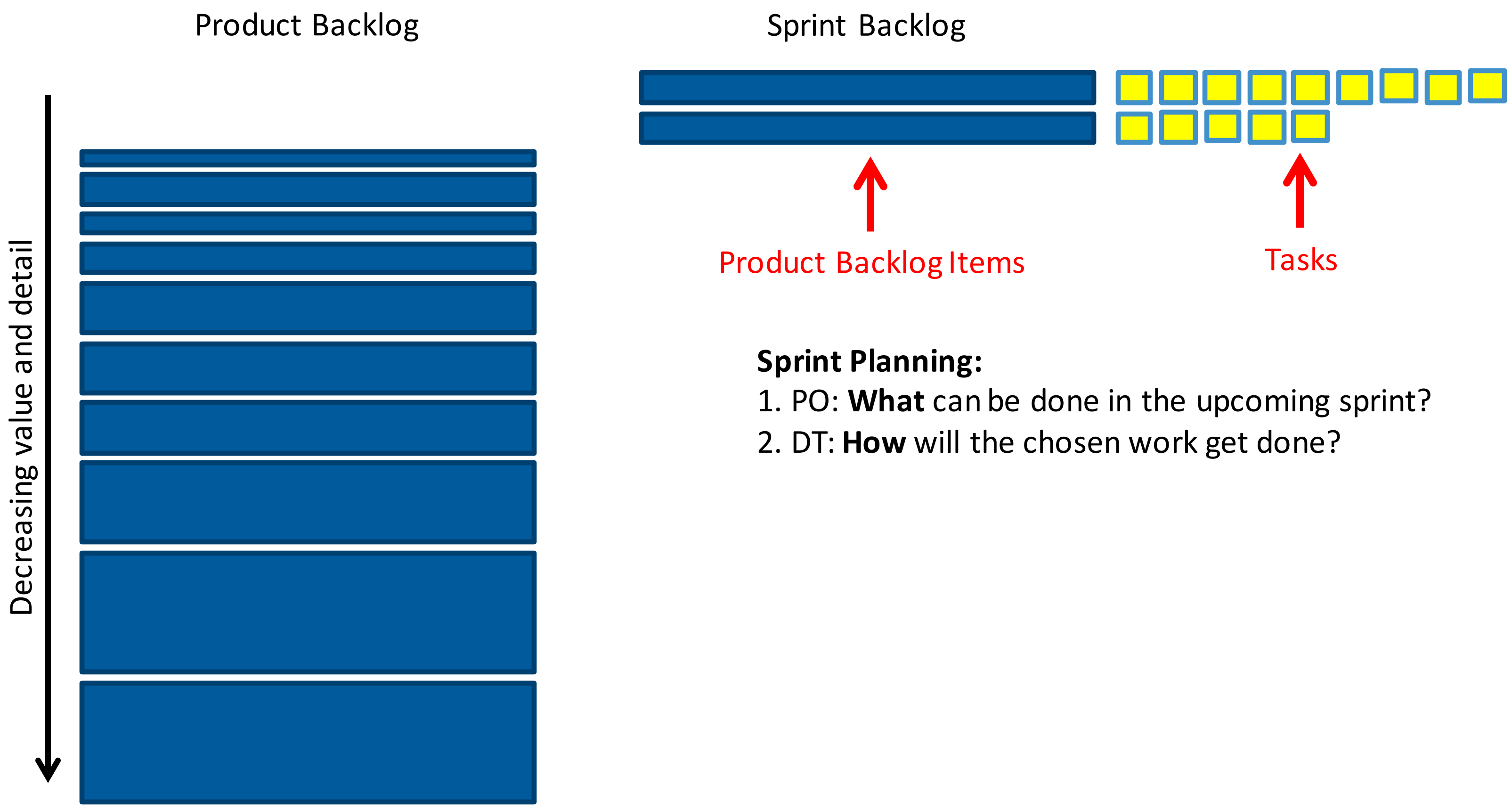


- **Continuous activity**
- DT **refines estimates/required tasks** for user stories, helps PO maintain the product backlog (max. 10% of work time during sprint)
- PO continuously **adds/removes/clarifies** items and ensures value (order)

Sprint Backlog



Sprint Backlog



Sprint Planning:
1. PO: **What** can be done in the upcoming sprint?
2. DT: **How** will the chosen work get done?

Sprint Backlog

- 3 minute exercise:

Oh no! You've run out of tasks assigned to you this sprint.

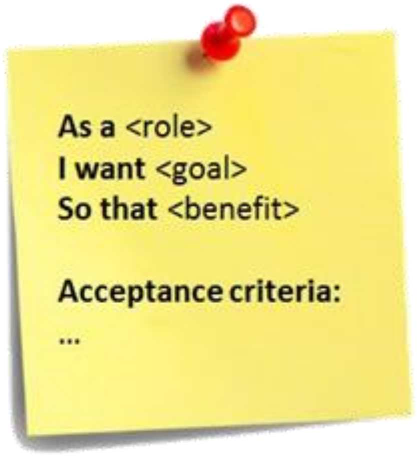
You know you generally don't add tasks to an in-progress sprint.

What can you do?

- 1 Minute each: Think, Discuss, Share

Product Backlog Items / User Stories

- In most cases, the product backlog items will be user stories
- User stories are **tickets to conversations**, not complete requirements
- User stories can be split or merged
  (e.g., during sprint planning or product backlog refinement)
- After **shared understanding** of story is reached, decide on **acceptance criteria**:
  - *"How will we know when the system does what it is supposed to do?"*
  - Generate list of pass/fail tests written in plain English
  - Ideally, tests can be automated before implementation
  - Possible template:
    Scenario name:
    Given <precondition(s)>
    When <some user action(s)>
    Then <expected result>



As a <role>
I want <goal>
So that <benefit>

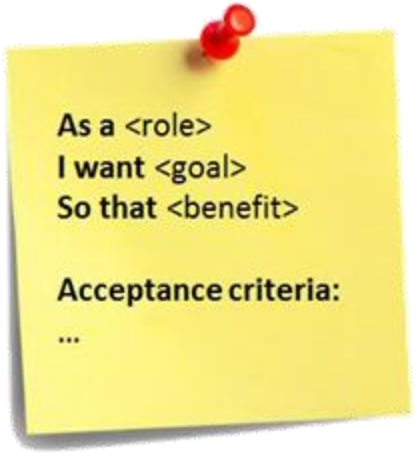
Acceptance criteria:
...

Example

As an online shopping customer

I want a way to collect items I want to buy all at once

So that I only have to complete one transaction


As a <role>
I want <goal>
So that <benefit>

Acceptance criteria:

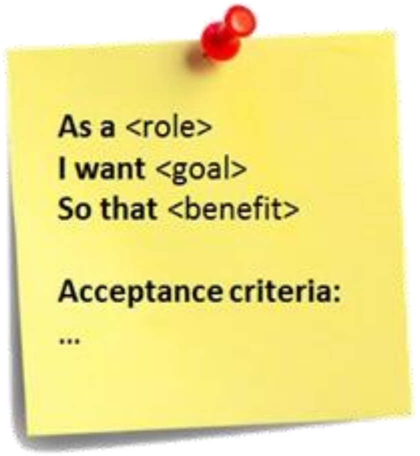
...

Exemplary acceptance criteria:

- Checking Out:
  Given a customer has added multiple items to the shopping cart
  When they click the check-out button on the user interface
  Then they are asked to complete the transaction for all the items in the cart together.
- Checking Out Nothing:
  Given a customer that has not added any items to their shopping cart
  When they click the check-out button on the user interface
  Then an error message is displayed notifying them there is nothing in the cart.

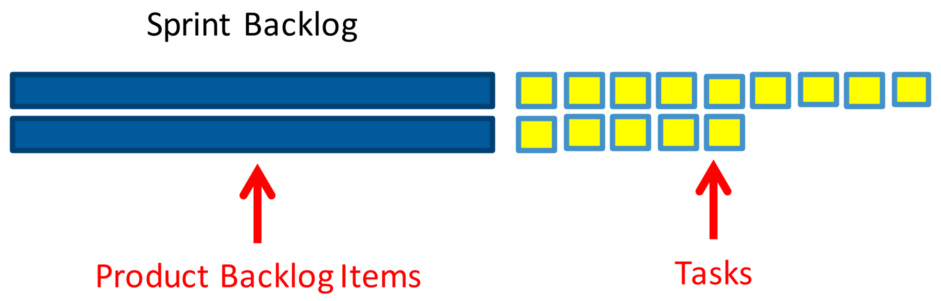
Quality: SMART User Stories

- **S**pecific and **M**easurable:
  - Acceptance criteria should be testable (see template on previous slides)
  - Counterexample: “The usability should be good.”
- **A**chievable:
  - If not deliverable in one sprint, split the story
- **R**elevant:
  - Adds “business value”
- **T**imeboxed:
  - Discuss with DT and PO if story exceeds estimation


As a <role>
I want <goal>
So that <benefit>

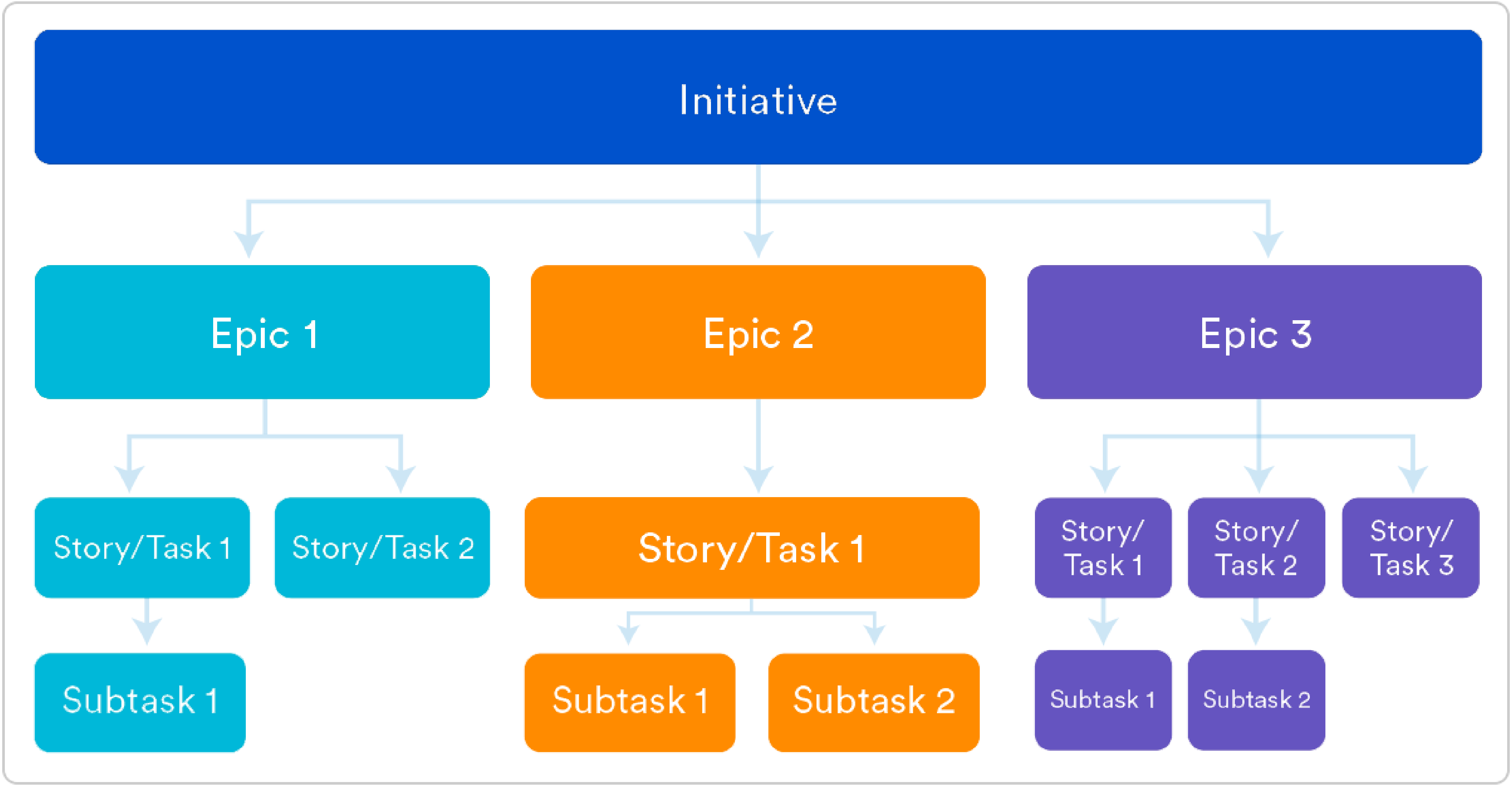
Acceptance criteria:
...

Tasks



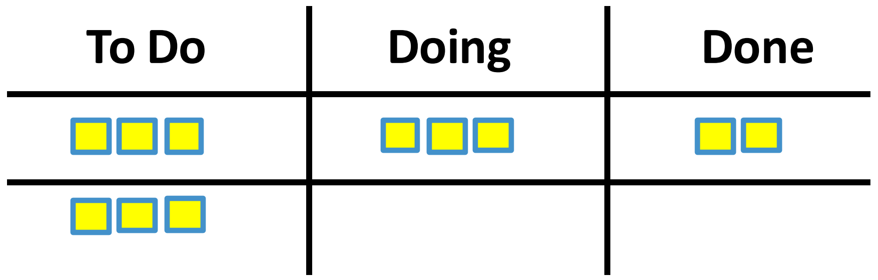
- A task is a **single unit of work** related to the implementation of a user story (or another product backlog item)
- Usually, the unit of work is carried out by **one team member** alone
- Besides **technical aspects** such as setting up a database or implementing a certain feature or test, tasks can also cover **non-technical aspects** such as doing research, meeting with stakeholders, etc.

Tasks

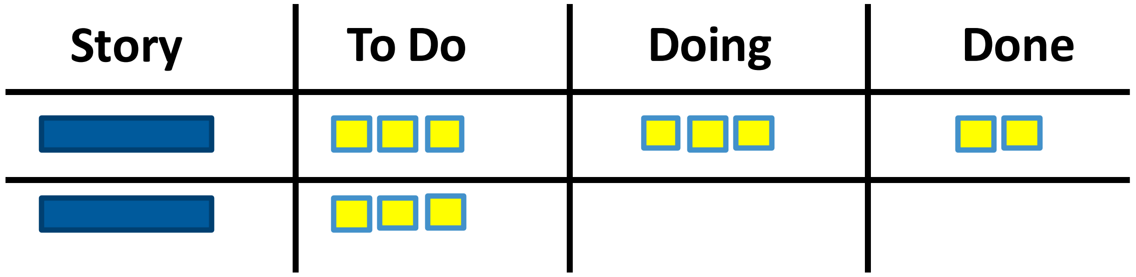


https://www.atlassian.com/agile/project-management/user-stories

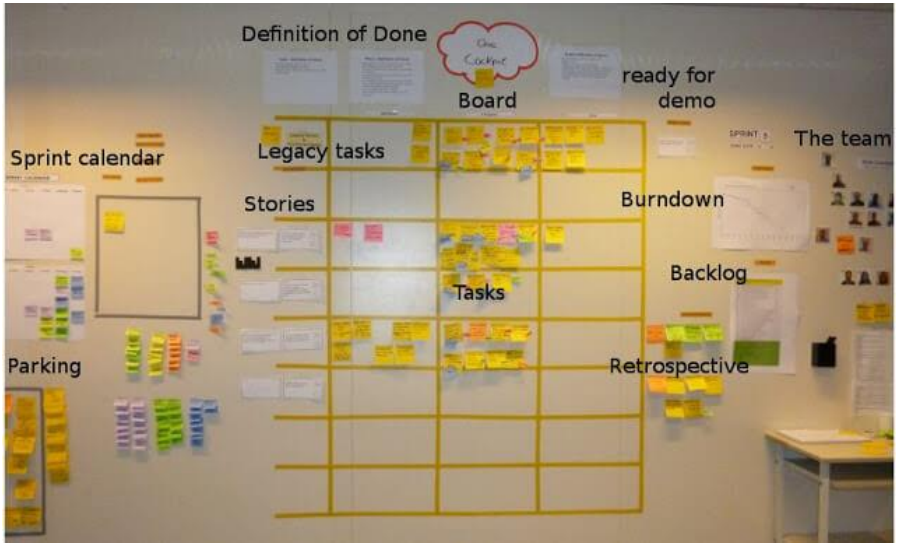
Task Board (or Scrum Board)



Task Board (or Scrum Board)



Task Board (or Scrum Board)

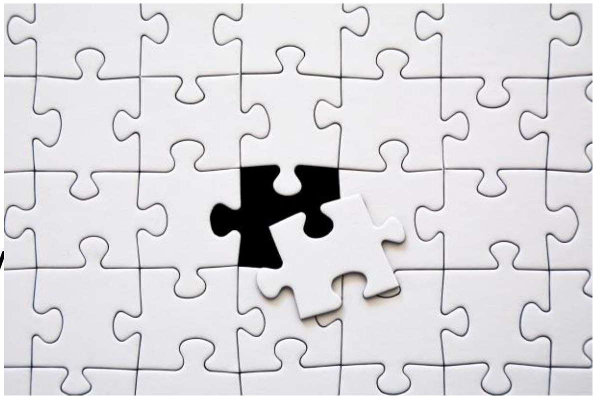


https://www.eylean.com/blog/2015/07/top-5-most-interesing-scrum-boards/

https://medium.com/@sashabondareva/scrum-task-board-offline-or-online-b341719fa472

Definition of Done (DOD)

- First version should be developed in **kickoff meeting**
- Should be **shared** by the whole Scrum team (store, e.g., in GitHub Wiki)
- Question: *Is a feature ready to ship?*
- DoD can include:
  - Details on code and design review
  - Static analysis results to control code quality
  - Passing of unit or performance tests
  - Anything else the Scrum team agrees on
- DoD != Acceptance criteria
- Acceptance criteria focus on user, DoD is rather technical/closer to the implementation



Estimation: User Story Points

- **Goal:** Make schedules/sprint planning more predictable
- **Estimating absolute time** (e.g. hours, days) required to implement a certain user story is **very hard**
- **Comparing** two stories/tasks and deciding which one takes longer **is easier**
- Story points are **relative time estimates**
- Possible **strategies**:
  - Use 1 for straightforward stories, 2 for medium stories, and 3 for very complex stories
  - Force team to use **Fibonacci numbers** (and a placeholder) for estimation: 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, ?
    Motivation: Bigger stories are harder to estimate
  - T-Shirt Sizes: S, M, L, XL
    Motivation: Abstract, less likely to be directly compared to actual time

Estimation: Task Points

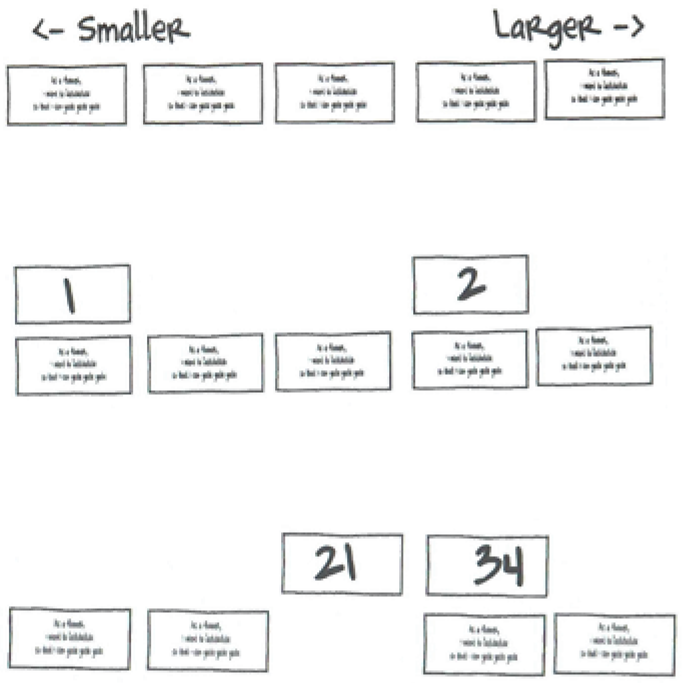
- Tasks can be estimated similar to user stories using **task points**
- Some teams prefer to just use the **task count**, assuming they are more or less of same size
- **Alternative:** Use Fibonacci for user stories, simple 1-3 scale for tasks

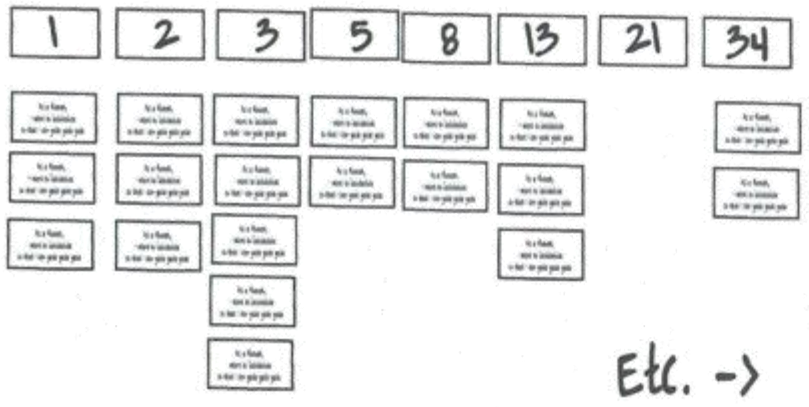
The Estimation Game

1. **Order user stories from small to large**
   - One team member after the other either adds a new story to an ordered list or moves one story within the list
   - Stop when list does not change anymore
   - (potential infinite loop)

2. **Assign point estimates**
   - Starting with smallest Fibonacci number, place number above the first story (from left to right) of that size
   - Alternatively, numbers or user stories can be moved
   - Stop when numbers don’t change anymore

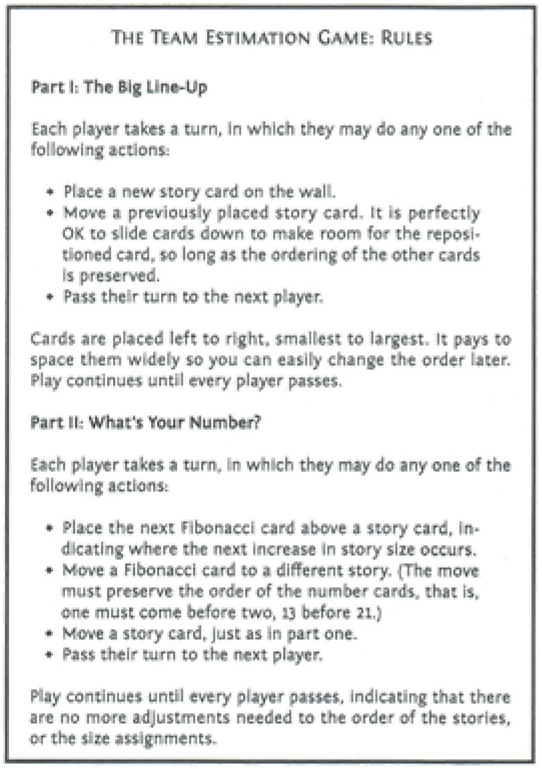
The Estimation Game





Sims and Johnson – The Elements of Scrum

The Estimation Game



THE TEAM ESTIMATION GAME: RULES

Part I: The Big Line-Up

Each player takes a turn, in which they may do any one of the following actions:

- Place a new story card on the wall.
- Move a previously placed story card. It is perfectly OK to slide cards down to make room for the repositioned card, so long as the ordering of the other cards is preserved.
- Pass their turn to the next player.

Cards are placed left to right, smallest to largest. It pays to space them widely so you can easily change the order later. Play continues until every player passes.

Part II: What's Your Number?

Each player takes a turn, in which they may do any one of the following actions:

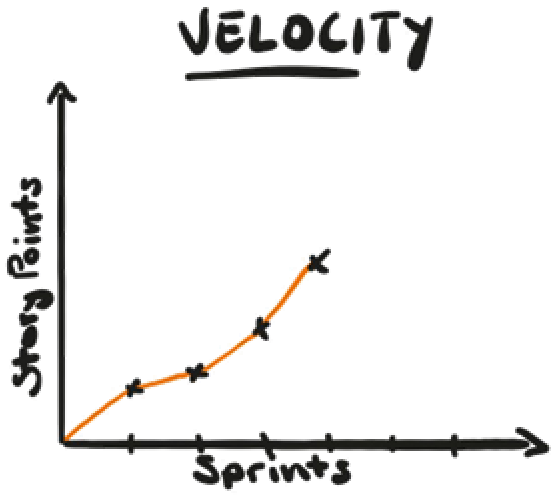
- Place the next Fibonacci card above a story card, indicating where the next increase in story size occurs.
- Move a Fibonacci card to a different story. (The move must preserve the order of the number cards, that is, one must come before two, 13 before 21.)
- Move a story card, just as in part one.
- Pass their turn to the next player.

Play continues until every player passes, indicating that there are no more adjustments needed to the order of the stories, or the size assignments.

Sims and Johnson – The Elements of Scrum

Velocity

- Average number of done* story points per sprint
- Supports sprint planning



https://www.agile-academy.com/