

# Scrum Cheat Sheet by agile42

## Product Owner

### Owns the Product Backlog

The Product Owner represents the interests of everyone with a stake in the project (Stakeholder) and he is responsible for the final product.

- elicit product requirements
- manage the Product Backlog
- manage the release plan
- manage the Return on Investment

## Scrum Master

### Owns the Scrum process

The Scrum Master is responsible for the Scrum process. He ensures everybody plays by the rules. He also removes impediments for the Team. The Scrum Master is not part of the Team.

- manage the Scrum process
- remove impediments
- facilitate communication

## Development Team

### Owns the software

The team figures out how to turn the Product Backlog into an increment of functionality within a Sprint. Each team member is jointly responsible for the success of each iteration and of the project as a whole.

- software quality
- technical implementation of User Stories
- delivery of a “potentially shippable” product increment at every Sprint

## Requirements

Make **SMART** Requirements: Simple, Measurable, Achievable, Realistic, Traceable.

## User Stories

**INVEST** in User Stores: Independent, Negotiable, Valuable, Estimable, Small, Testable.

## Tasks

Make sure a Task is **TECH**. Time boxed, Everybody (can pick it up), Complete and Human-readable.

## Sprint Planning

### Commit the deliverable(s) to the PO

Two part meeting. First, the PO presents the User Stories. Second, when the Team thinks they have enough Stories to start the Sprint, they begin breaking it down in Tasks to fill the Sprint Backlog (normally 3 to 4 days of work, than inspect & adapt).

Timebox: 4 hours

Owner: Product Owner

Participants: Team, Scrum Master

## Daily Scrum

### Inspect and Adapt the progress

In this standup meeting the Team daily inspects their progress in relation to the Planning by using the Burndown Chart, and makes adaptation as necessary.

Timebox: 15-20 minutes

Owner: Scrum Master

Participants: Team, all interested parties may silently attend.

## Sprint Review

### Demonstrate the achievements

The team demonstrate the PO the result - on the developed product - of the Sprint. The PO can accept or reject features depending on the agreed acceptance criteria.

Timebox: 4 hours

Owner: Team

Participants: Scrum Master, Product Owner, optionally the PO can invite other Stakeholders

## Retrospective

### Maintain the good, get rid of the bad

At the end of a Sprint, the Team evaluates the finished Sprint. They capture positive ways as a best practice, identify challenges and develop strategies for improvements.

Timebox: 3 hours

Owner: Scrum Master

Participants: Team, (Product Owner)

## Product Backlog

### Dynamic prioritized list of requirements

The requirements for the product are listed in the Product Backlog. It is an always changing, dynamically prioritized list of requirements ordered by Business Value. Requirements are broken down into User Stories by the PO.

*Prioritize the requirements by playing the Business Value game.*

*Buy these at [www.agile42.com](http://www.agile42.com)*

## Burndown Chart

### Displays the remaining work

The Burndown chart shows the amount of work remaining per Sprint. It is a very useful way of visualizing the correlation between work remaining at any point in time and the progress of the Team(s).

*Use a tool such as Agilo to automatically create the Burndown Chart.*

*Learn more at [www.agile42.com](http://www.agile42.com)*

## Sprint Backlog

### List of Tasks to fulfill the Sprint Goal

The Sprint Backlog contains all the committed User Stories for the current Sprint broken down into Tasks by the Team. All items on the Sprint Backlog should be developed, tested, documented and integrated in order to fulfill the Sprint Goal.

*Estimate Story complexity by playing Planning Poker.*

*Buy these at [www.agile42.com](http://www.agile42.com)*

## Potentially Shippable Product

Scrum requires at the end of each Sprint that the product is potential shippable to the customer. That means the increment is:

- thoroughly tested and stable
- well-structured
- well-written code
- user operation of the functionality is documented