



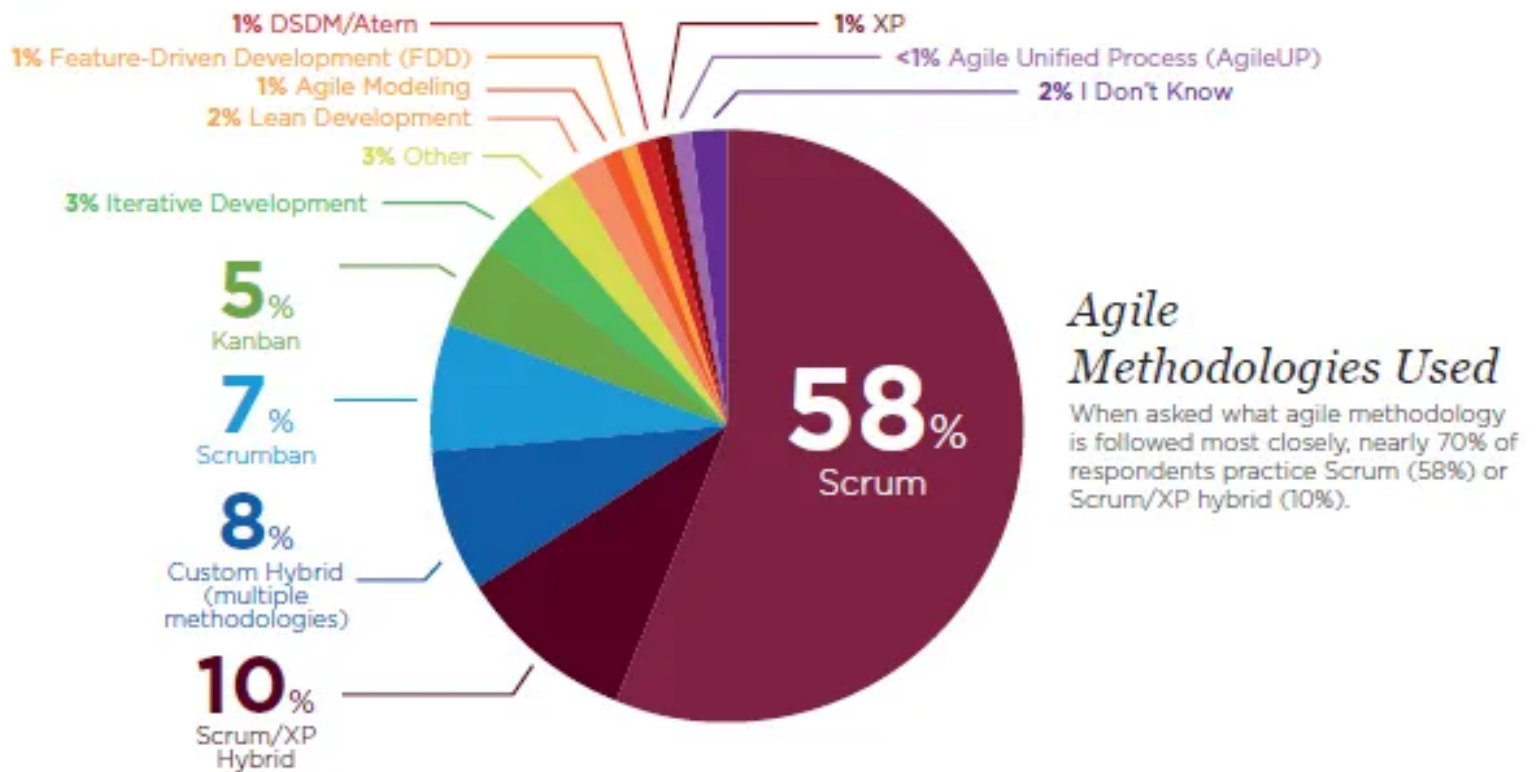
Agile scrum isn't a methodology



Here we continue our series of publications on approaches to software development.

If you missed the beginning, start with our ultimate Agile [guide](#). Once you've gained an understanding of this software development philosophy, come back for an in-depth look at one of its frameworks.

AGILE METHODS AND PRACTICES



What is Scrum?

“**Scrum** is an agile framework for managing knowledge work, with an emphasis on software development”, [as Wikipedia says](#). Early Wiki also said that Scrum is a methodology. But now we see that Scrum guys have changed it.

Imagine you need to build a house. It should have walls, windows, doors and a roof. You need a kitchen, a bedroom and a bathroom to feel comfortable.

However, nobody can tell you what colour to paint it or what brand of furniture to choose from. Similarly, Scrum is a framework – a set of recommendations.

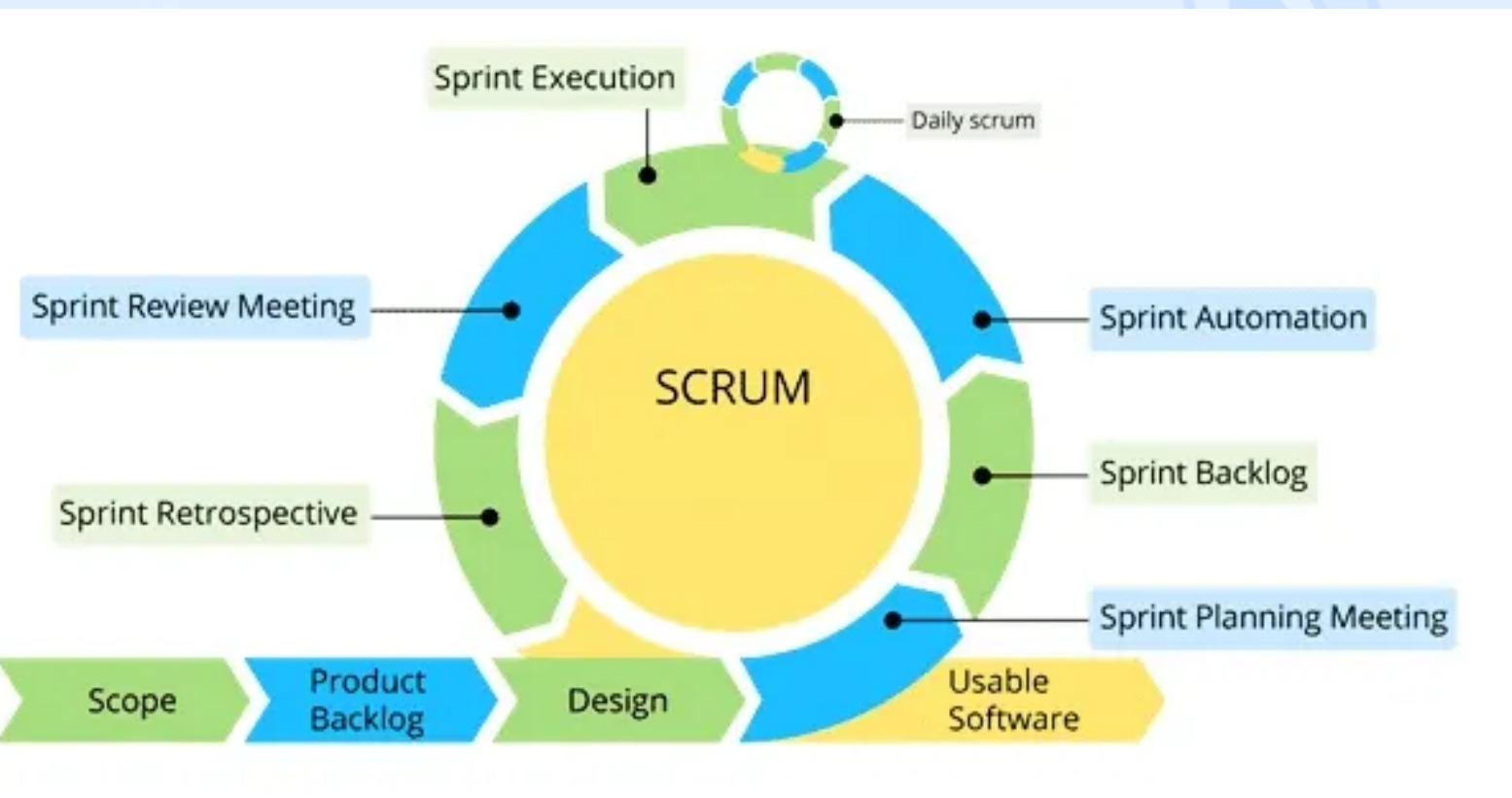
It guides the development process by helping to:

- define product value and set priorities,

- visualise all the requirements,
- clarify what's been done and what is in progress,
- simplify communication,
- demonstrate what results should be achieved at the end of each iteration.
- organize effective teamwork in the complex domains

What it does not provide is a list of To-Do instructions.

But it provides you with all the necessities to organise your project and give it a boost with your own tools and techniques, making for effective teamwork and a great end product.



Basic principles

Several popular frameworks exist within Agile: Kanban, Scrum, Extreme Programming (XP), etc. Let us see what makes Scrum different from other **approaches** to effective IT project management.

1. Values

Scrum aims to instil certain values in teams: **Courage, Focus, Commitment, Respect, Openness.** This framework advocates the inculcating of personal responsibility in each developer and the

encouragement of co-operation and a friendly working environment.

- Collegial thinking helps to control all the critical points of the complicated software development process and to contemplate lots of blockers. Team synergy leads to a significant increase in efficiency. And mutual responsibility of the Scrum team motivates them to achieve valuable results for business in spite of all difficulties.



2. Roles

In our Agile guide, we gave a profound explanation of the three vital roles in Scrum: **team**, **Product Owner** (=Product manager) and **Scrum Master**.

Basically, the Product Owner (PO) and the Scrum master are managers.

The Product Owner

The PO is a sculptor for the product, he (or she) presents the idea to the team, helps them set priorities and is responsible for the results.

The Scrum master

The Scrum Master (SM) is an independent person who helps development teams to work effectively. They facilitate meetings and make sure that all Scrum procedures are being respected.

- The SM sticks to the concept of “servant leadership”. This means that a Scrum master facilitates processes but does not actually impose any authority, instead of helping the team to reach their full potential.

SCRUM ROLES



PRODUCT OWNER

Represents the client and the business in general for the product on which they're working.



SCRUM MASTER

Responsible for ensuring the team has everything they need to deliver value.



DEVELOPMENT TEAM

A group of cross-functional team members all focused on the delivery of working software.

3. Meetings

Sprint Planning meetings, Daily Stand-ups, Grooming, Sprint review and demo, Scrum Retrospective and others referred to as “events” are an indispensable part of Scrum. Some experts tend to believe that without daily meetings, the approach no longer qualifies as Scrum at all. Read more about the events [here](#).

4. Scrum rituals

Planning Poker is one of Scrum's rituals for organising collaboration on the project. It's conducted during the Sprint Planning meeting. As you can see, Planning Poker is not mentioned in the [Scrum Guide](#), but for many teams is a regular and required ritual. This fact again confirms that Scrum is a framework, offering teams the ability to find and use the most effective tools to organise their work.

Planning Poker

- **Its goal:** to reach an agreement, mainly in assessing the complexity of the work ahead and understand what can be done during the next sprint.
- **Its form:** something like a game, where expert opinions are collected and all the arguments expressed without stress.



To organise Scrum Poker, one needs a list of features to discuss, a special set of measures for estimation (the most popular are the Fibonacci numbers: 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144,..) and a Scrum master to guide the process. Each of the stories is read aloud and assessed simultaneously by

each member of the team (except for the Scrum master and PO).

The highest and lowest scores must then be explained (proved) by their owners. After several sets, all the team members agree on the estimation and continue with the next feature. Follow this [link](#) to discover all the mechanics of this game procedure.

As a result, group decisions are done quickly.

5. Artefacts

Artefacts in Scrum can be understood as a result of the teamwork that helps to complete the project successfully. It's up to each team whether to use all of them or to implement some/mix with other framework elements.

The artefacts of Scrum

These are the artefacts and the tools used to acquire them according to the framework:

Product vision

This artefact represents a long-term goal of the project/product. It should be short, clear and memorable so that every member of the team knows it by heart.

Product vision can contain the following lines:

- The product name, category and features;
- For whom you're developing it;
- Why they need it;
- Reasons to buy it;
- How your product is different from others on the market;
- Problems it solves for different kind of stakeholders (also for business owners);
- Success criteria.

The statement can contain all or some of this information:

For (target customer)

Who (statement of need or opportunity)

The (product name) **is a** (product category)

That (key benefit, reason to buy)

Unlike (primary competitive alternative)

Our product (statement of primary differentiation)

For example, the product vision for our StarBuffet project looked like this:

“For an Australian self-catering service who wish to increase the efficiency and transparency of their workflow, the StarBuffet app will decrease labour costs and increase profit by controlling at-station food availability automatically. Unlike its competitors, this app also manages order completion timing and sends notifications to customers when food is ready”.

Sprint goal

A sprint goal is an objective to be met within the Sprint. It includes the implementation of Sprint Backlog items and is the result of PO-team negotiations.

- It must be measurable and specific. Keeps the team motivated and focused.
- An example goal could be to integrate international payment systems (Paypal, Payoneer, cards) into an app.

As part of the planning of the new sprint, the Development Team gives its commitment to achieving the Sprint Goal.

Product Backlog

Product Backlog is a dynamic list of all the required features and functions compound by the product owner. It sets the priorities for the team and points out what items are still undone.

Sprint Backlog

A Sprint Backlog comprises Product Backlog items selected for the Sprint and the plan for Increment delivery. It shows how likely the successful realisation of the Sprint Goals.



Definition of done

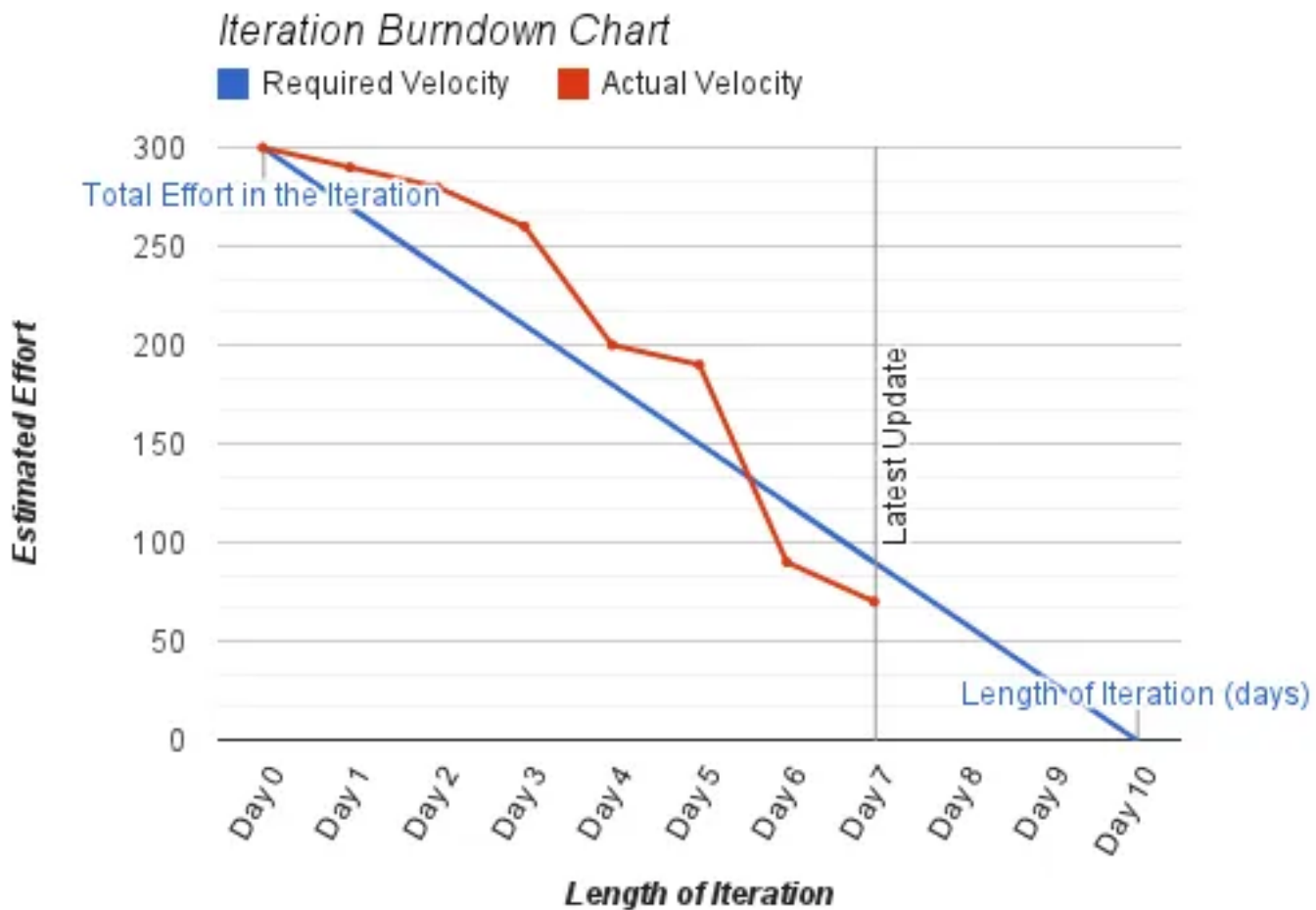
Every Product Backlog item has acceptance criteria. They define what allows the work to be declared done.

- All the criteria can be collected in one document (the Definition of Done) or be applied to different layers as their own artefacts.

For example, the team can set up a specific DoD for User Stories (to understand that User Story was completed), DoD for Sprint to know when Increment ready to ship, and/or a DoD for the overall product to be sure it's ready for production release.

Burndown chart

Burndown charts are graphs that give an overview of project progress and predict will Sprint goals be achieved or not. As tasks are completed, the graphs “burn down” to zero.



Product Increment

An increment describes how many Product Backlog items have been completed during the current and previous sprints.

When an increment is ready to ship it must be marked as “Done”. For this to happen it should meet these two criteria:

- meet the Scrum Team’s Definition of “Done”;
- be usable no matter whether the Product Owner decides to actually release it or not.

User Stories

User stories are short descriptions of certain features from the perspective of a user. Find out [here](#) how to write great user stories and how the teams use it to develop products that customers really want.

Writing a user story

- 1 Define your **end user**
- 2 Specify what **they want**
- 3 Describe **the benefit**
- 4 Add **acceptance criteria**

Copyright © 2019 Knowledge Train Limited

Where Scrum comes from

Scrum appeared at the same time [the Agile manifesto](#) was written – in 2001.

In 2002, Ken Schwaber founded The [Scrum Alliance](#), offering certified ScrumMaster programs. Having completed them, anyone could become a Scrum master with the job of introducing the Scrum framework to IT companies and helping them to maintain it.

The first guide on Scrum was published in 2010. Ever since Scrum has been gaining widespread popularity not only in IT but also in business management.

Scrum and Agile relate as a philosophy and a practical means of bringing it to life – with roles, instruments, artefacts and guidelines.



**CERTIFIED
SCRUMMASTER®**



**CERTIFIED SCRUM
PRODUCT OWNER®**

Why adopt Scrum?

Scrum is the most popular Agile framework in IT project management today. There are reasons for this:

- **Better quality products**

A people-orientated development process with a focus on quick delivery of valuable business results organised in short iterations does not permit the wasting of time on remaking the whole project after release.

- **Increased productivity**

Only a limited number of features are worked on at any given time. This allows teams to focus more on quality and deliver outstanding products.

- **Flexibility, ability to meet the market changes and improved stakeholder satisfaction**

In Agile, the development process is much more transparent for clients thanks to regular demos and intermediate requirements correction. This guarantees that the final product will be what the client expected.

Stay tuned for more posts about Agile frameworks, the next one is already in the queue. In the meantime, check out the other posts on our blog and a couple of books, which we recommend:

- Learning Agile: Understanding Scrum, XP, Lean, and Kanban 1st Edition by [Andrew Stellman](#);
- [Agile Estimating and Planning](#) by Mike Cohn.