



**Ways of work**

# **What is Agile?**

Please do not copy without permission. © ALX 2024.

# Agile and data science

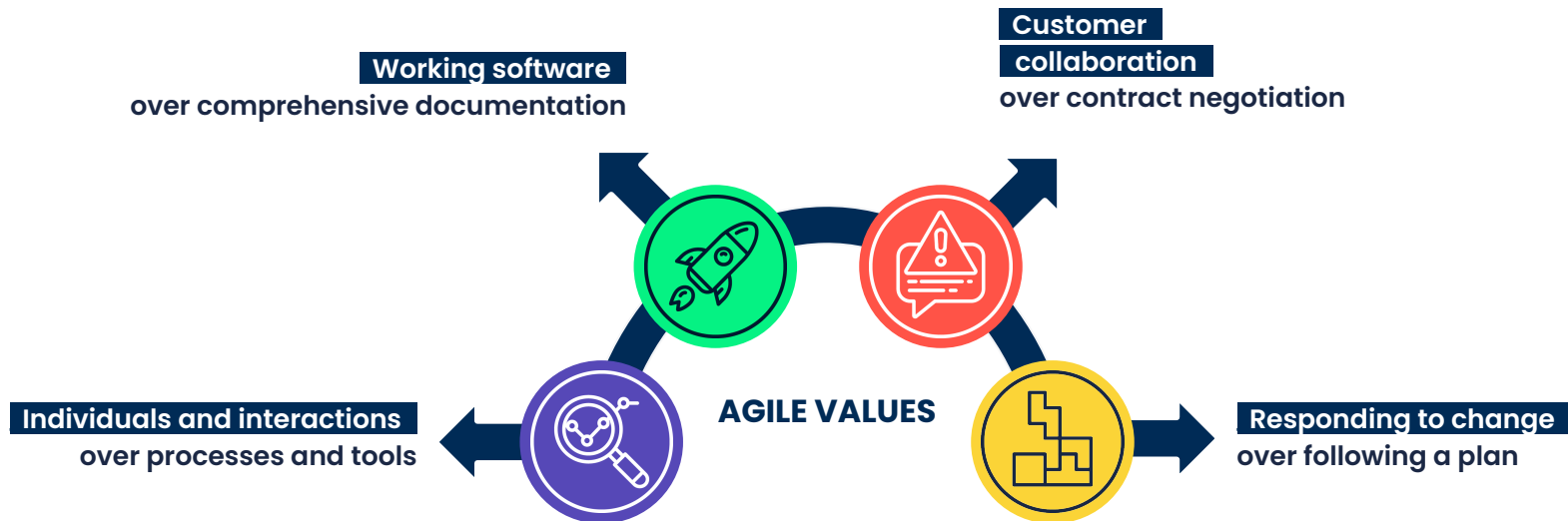
In the **dynamic** realm of **data science**, where **adaptability** and **collaboration** are key, **Agile** methodologies offer a **structured approach** to **managing projects** efficiently.

Agile is an **iterative framework** prioritizing **adaptive planning** and **continuous improvement** for more efficient and valuable **project delivery**.

Agile teams respond well to **change**, **collaborate**, and quickly and effectively **achieve project outcomes** by focusing on delivering **incremental** and **data-driven results**.

# Agile values

The **four Agile values**, originating from a software development approach, are centered around **quality**. While Agile initially focused on software, **various industries** now adopt it to design products and services that effectively meet consumers' needs and expectations.



# The Scrum framework for project teams

**Scrum** is an **Agile project management framework** that helps teams structure and manage their work through a set of values, principles, and practices.

## Iterative development

Scrum uses **iterative cycles** called **sprints**, usually one to three weeks long. Each sprint delivers a potentially **shippable product** increment for ongoing evaluation and adjustment.

## Collaborative roles

Scrum defines **specific roles** that work **collaboratively** to ensure efficient communication, alignment, and the delivery of valuable outcomes.

## Transparent ceremonies

Scrum **ceremonies** provide opportunities for **regular communication** and **improvement**, ensuring **transparency** throughout the project.

# Scrum roles

## Product owner

The product owner **represents the business interest.**

They are the **voice** of the **client**, set the **vision**, and **translate requirements into work packages** for the development team.

## Scrum master

The Scrum master or delivery lead **oversees** and **manages** the development process.

They **set the pace**, **facilitate** improvements, and make sure team members have everything they need to get the job done.

## Technical lead

The technical lead is a contributing member of the development team.

They help their team with the **technical aspects** of the job. They perform **technical reviews** to ensure the work is technically sound and may control what **makes it into the final product.**

## Development team

The team works **autonomously** with each member contributing and collaborating to **deliver the product.**

They are **transparent** about the work they are doing and **vocal** about where they need help from the technical or delivery leads.

# Scrum ceremonies

Scrum ceremonies are **crucial milestones** within Agile project management. They play a pivotal role in structuring work, promoting collaboration, and ensuring consistent progress.

There are **four key** Scrum ceremonies:

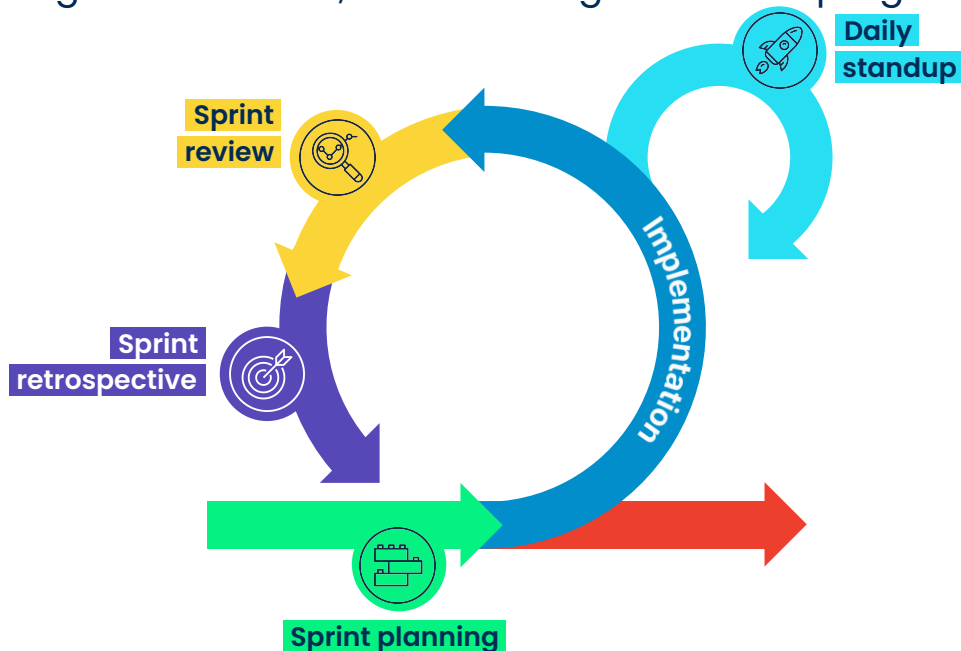
**Sprint planning:** Goal-setting and task selection.

**Daily standup:** Daily sync and progress update.

**Sprint review:** Feedback and presentation.

**Sprint retrospective:** Process reflection and improvement.

Next, let's see **how** the Scrum ceremonies fit into the **complete Scrum process**.



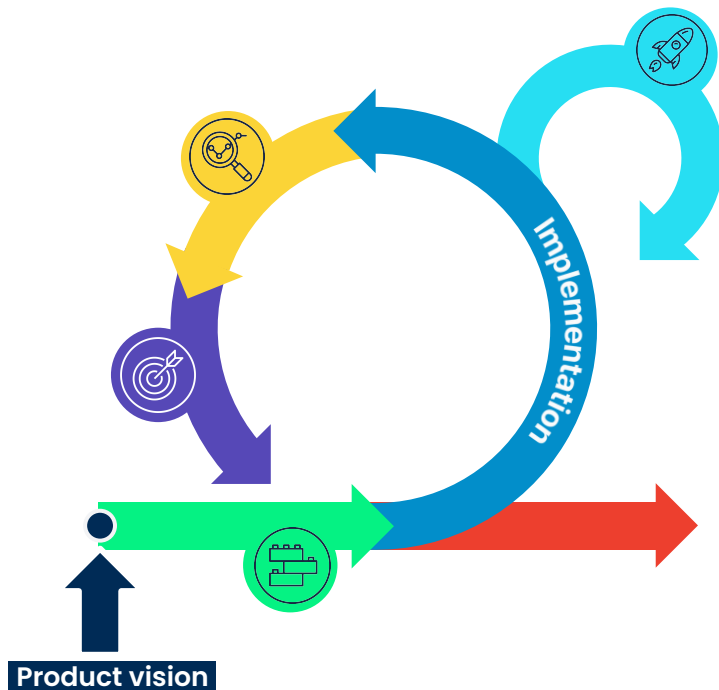
# Scrum ceremonies in context

## Product vision

At the start of the project the **product vision, strategy, direction, and objectives** are set.

It is important to understand and align on:

- Why are we building this product?



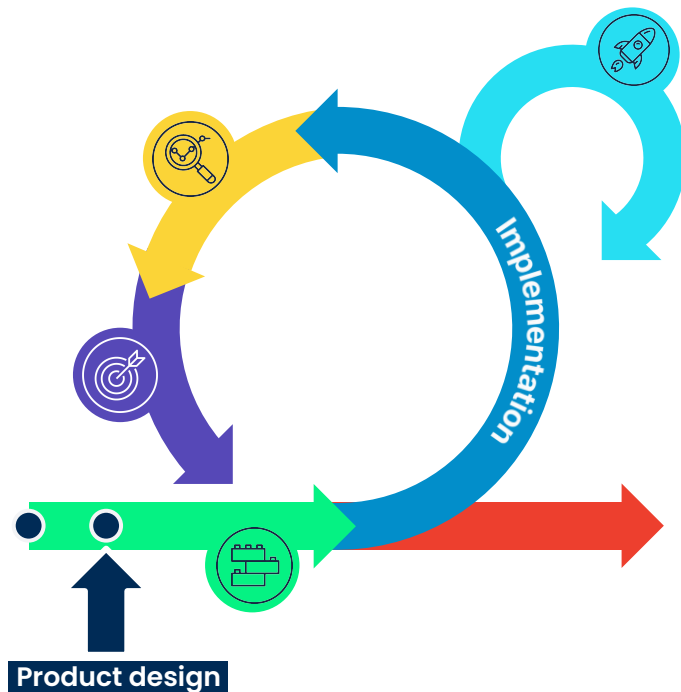
# Scrum ceremonies in context

## Product design

The **problem statement** and **design** are refined. We need to **understand** the various **features** and **design parameters**.

Here, it is important that we clearly understand:

- What are the user stories?
- Who will use this?
- What will they do with it?
- What is the product roadmap?





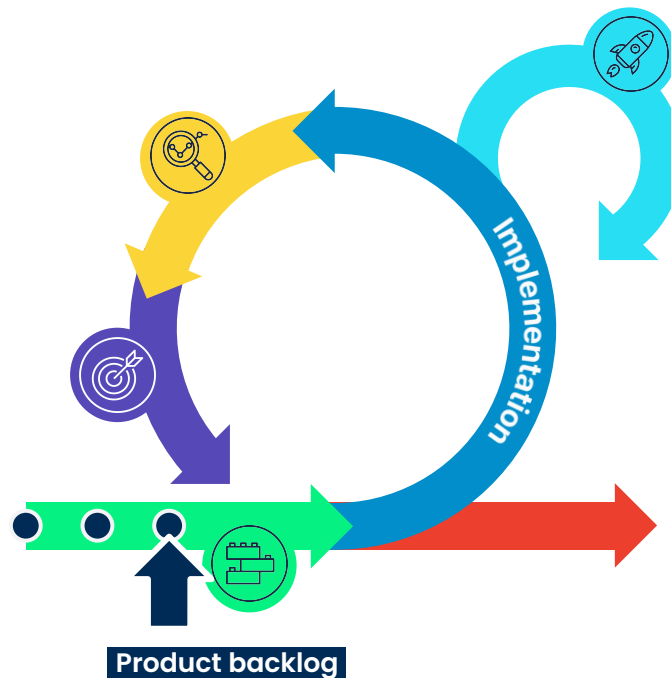
# Scrum ceremonies in context

## Product backlog

An **initiative** or **task backlog** is created to **address specific design features** and **user stories**.

The backlog is a **prioritized list of work** with **detailed requirements** that the development team needs to complete.

The most **important tasks** are at the **top of the list** so the team can pull the most valuable work into scope when there is capacity for it.



# Scrum ceremonies in context

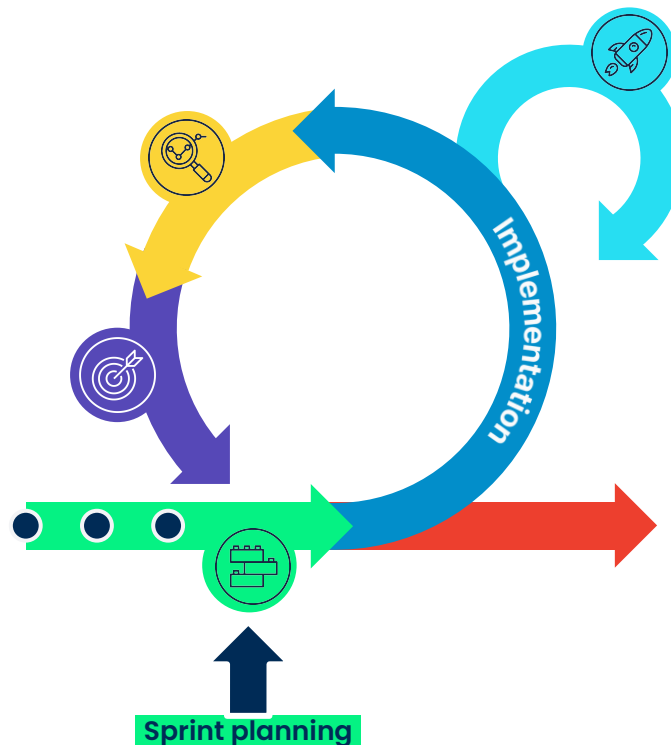
## Sprint planning

Sprint planning **initiates** the typically short **delivery cycles** between one and four weeks.

The team discusses the most **important backlog items** and how they **contribute** to the **overall objective**.

The team aligns on:

- Why is this sprint valuable?
- What can be done in this sprint?
- How will the work get done?

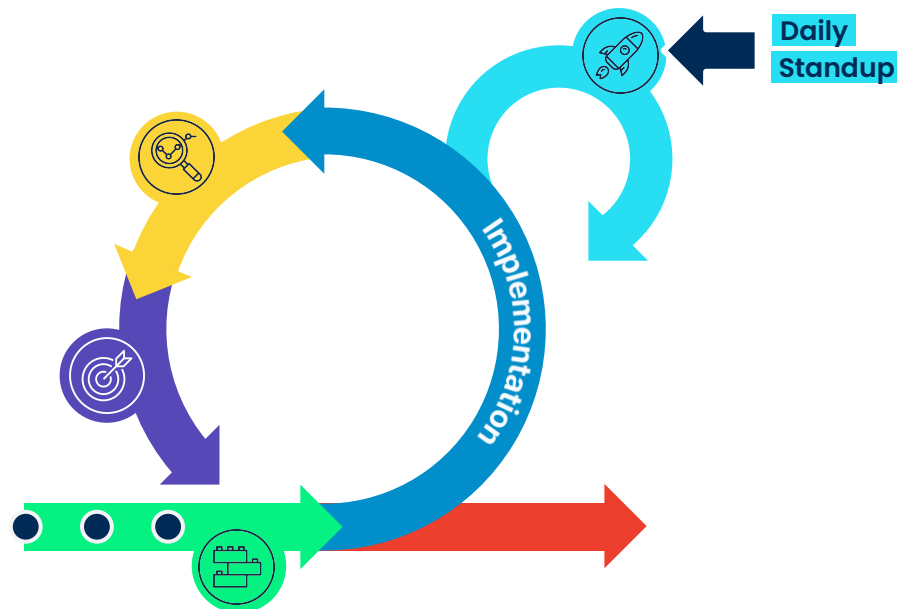


## Daily standup

Standups may be daily or at some other **regular** cadence set by the team. These **short, focused meetings** highlight progress and raise flags so the team can **address issues quickly**.

Each team member should be able to answer:

- What did I do yesterday?
- What am I working on today?
- What issues are blocking my progress?



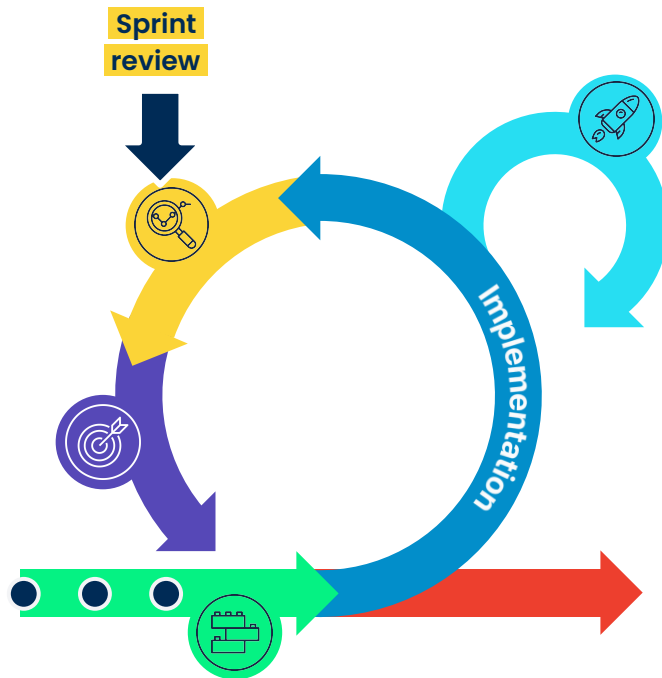
# Scrum ceremonies in context

## Sprint review

Sprint reviews are about **demonstrating** and **celebrating** the **hard work** of the entire team.

Great reviews:

- Have **data-driven discussions** around success and missing, meeting, or exceeding the sprint goal.
- Build morale and team motivation.



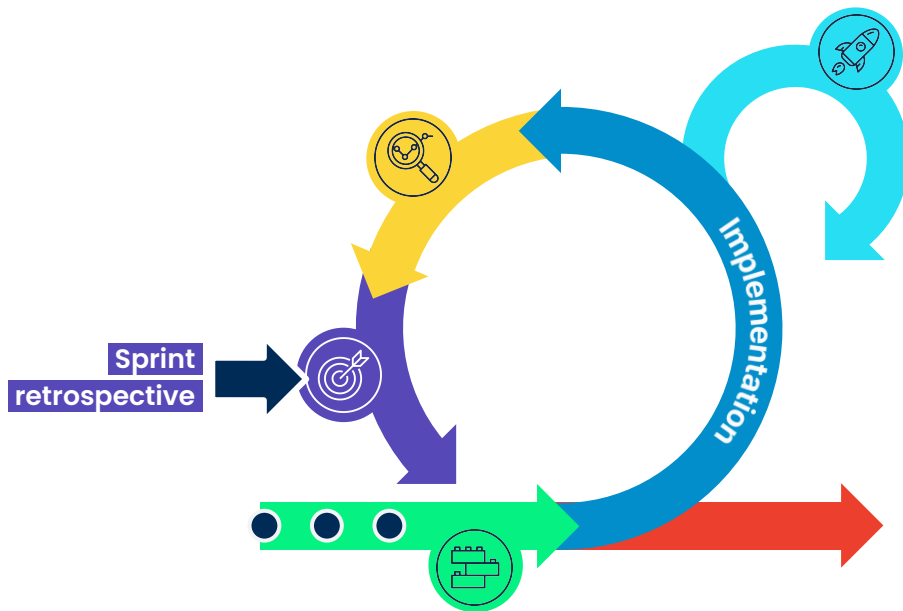
# Scrum ceremonies in context

## Sprint retrospective

Sprint retrospectives are a safe space to **share feedback** and nurture a culture of **continuous learning** and **incremental improvement**.

It is essentially focused on discussion and experiences around:

- What did we do well?
- What can we do better?

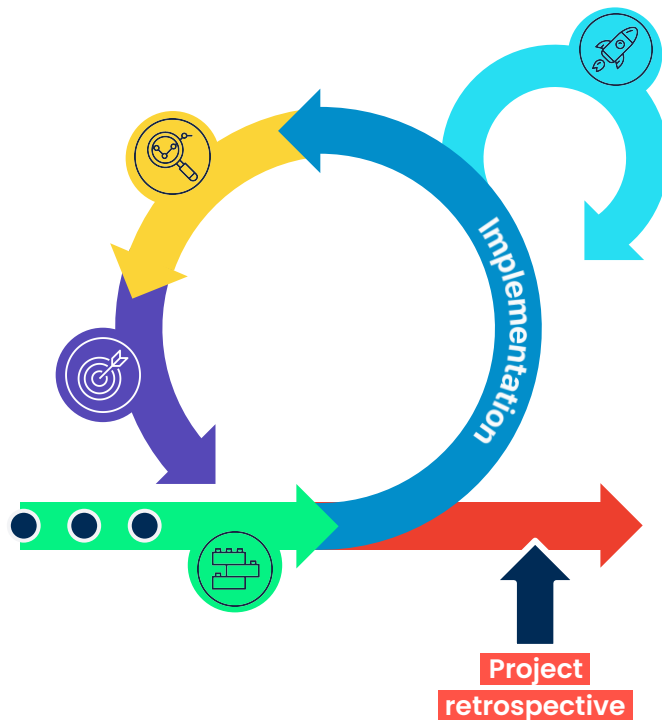


# Scrum ceremonies in context

## Project retrospective

Once the **minimum viable product** (MVP), product iteration, or final product is delivered, the team can **reflect** on the project as a whole.

Use team, business, and client feedback to determine ways in which the team and process may be **improved** for future projects.



# The twelve agile principles

1

## **Customer satisfaction**

By early and continuous delivery of valuable software.

2

## **Welcome change**

Welcome changing requirements, even in late development.

3

## **Deliver frequently**

Deliver working software frequently (weeks rather than months). Focus on the minimum viable product (MVP) and iterations.

4

## **Working together**

Close, daily cooperation between business people and developers.

5

## **Motivated team**

Projects are built around motivated individuals, who should be trusted.

6

## **Face to face**

Face to face is the best form of communication.

# The twelve agile principles

7

## **Working software**

Working software is the primary measure of progress.

8

## **Constant pace**

Sustainable development, able to maintain a constant pace.

9

## **Good design**

Continuous attention to technical excellence and good design.

10

## **Simplicity**

Simplicity – the art of maximizing the amount of work not done – is essential.

11

## **Self-organisation**

The best architectures, requirements, and designs emerge from self-organizing teams.

12

## **Reflect and adjust**

The team regularly reflects on how to become more effective and adjusts accordingly.



# Scrum ≠ Agile

Scrum is just **one of the many Agile methodologies**, it is not the only one.

Following Scrum will not necessarily lead to agility. Agile is **more than** just a set of **principles** or **methodologies** – it's a **mindset**.

You won't necessarily follow a Scrum process in an Agile team, it could be **Kanban** or a **Lean Software Development** approach, for example. However it is important to remember, at the **heart of Agile** lies **collaboration, delivery, reflection, and improvement**.