**Agile research**

Student: Jorn Kersten

Student number: 483331

Date: 05-01-2023

Semester: S-DB-IPS3-S3-DB03

Table of contents

[My use of agile 3](#_Toc123841157)

[Scrum 3](#_Toc123841158)

[What is Agile 4](#_Toc123841159)

[Agile manifesto 5](#_Toc123841160)

[The 12 principles 5](#_Toc123841161)

[Scrum 5](#_Toc123841162)

[Sprint 5](#_Toc123841163)

[Agile requirements 6](#_Toc123841164)

[User stories 6](#_Toc123841165)

[Functional requirements 6](#_Toc123841166)

[Non-Functional requirements 6](#_Toc123841167)

[Acceptance tests 6](#_Toc123841168)

# My use of agile

In semester 3, both my individual and group project I used Agile. Agile is a method that uses multiple tools to simplify the process of creating an application for you and your team. With it, you break down the whole project into smaller easier tasks.

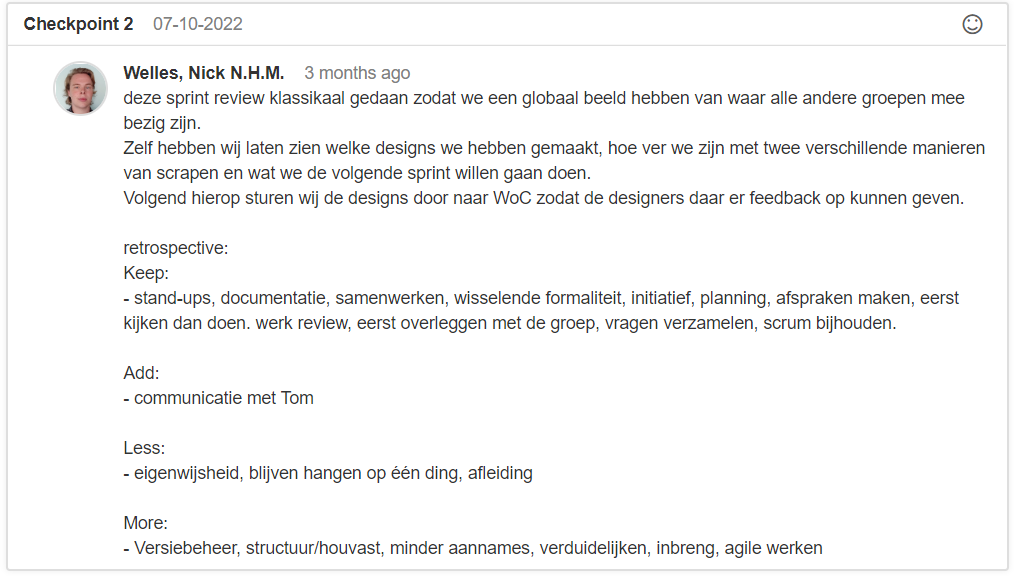
## Scrum

To apply agile in the my project, I used the Scrum method. I already worked with this a bit in semester 2, so I was keen to build on it. You can find the scrum boards of my individual and group project in the links below.

[Scrum-board individual project](https://github.com/users/Jorn-Kersten/projects/2/views/4)

[Scrum-board group project](https://github.com/orgs/wjjcn/projects/1)

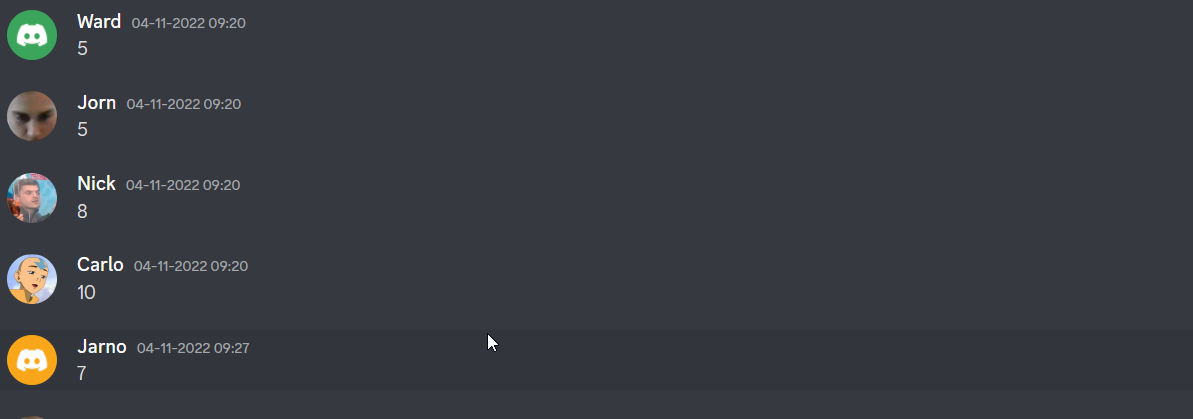
Working through the Agile methodology with scrum was different with the group project. Here, we work more with sprints; one sprint lasts three weeks. After those three weeks, we hold a delivery moment to show the agreed work with the stakeholder. We then show what we have finished or how far we are with things that did not quite manage to be completed. Afterwards, we hold a retrospective to go over what went well and what didn't that sprint. We then incorporated all this into a Feedpulse.



Afbeelding met tekst

Automatisch gegenereerde beschrijvingBesides holding the retrospective, we also wrote down in Feedpulse for each group member what each group member did right or wrong that sprint.

Then we did sprint poker with the group using the chosen user stories. In this, we all make an estimate for how long we think the task will take to create.



In the group project, we went through which requirements we wanted to handle before each sprint. We also discussed these with the stakeholder during delivery and then implemented them.

Then the following sprint we did this again. During this sprint, we showed what we made. When creating, we looked at the prepared user stories to derive what we needed to create.

[User stories group project](https://github.com/orgs/WJJCN/projects/1/views/7)

# What is Agile

With agile, the aim is to maintain a process in a clear way. With this, the tricky thing is that not everything can be predicted, some things may go wrong or not be finished on time. And with agile, you can keep this more in line.

Source from agilealliance: “The authors of the Agile Manifesto chose "Agile" as the label for this whole idea because that word represents the adaptability and responsiveness to change that was so important to their approach.”

Srouce: <https://www.agilealliance.org/agile101/>

## Agile manifesto

The Agile Manifesto was founded in February 2001 at a ski resort in Snowbird, Utah. 17 representatives from various programming and development backgrounds gathered to discuss Agile methodologies and find a solution to the older processes with which they had become frustrated.

More than two decades have passed since the birth of the Agile Manifesto. In that time, teams across the world have embraced the four values and 12 principles contained within the document. According to the 14th Annual State of Agile Report, more than 95% of respondents confirmed that their organizations practiced Agile development methods.

## The 12 principles

1. Early and Continuous Delivery of Valuable Software
2. Embrace Change
3. Frequent Delivery
4. Business and Developers Together
5. Motivated Individuals
6. Face-to-Face Conversation
7. Working Software
8. Sustainable Development
9. Technical Excellence
10. Simplicity
11. Self-Organizing Teams
12. Regular Reflection and Adjustment

Source: <https://www.plutora.com/blog/12-agile-principles>

## Scrum

Scrum is a way of creating a project piece by piece. This involves continuous experimentation, development consultation and implementation. There is always feedback on how things can be done better or faster, this runs continuously.

With scrum, you divide a very large task into several small tasks. Each task is experimented with and receives feedback. This feedback is used to develop something, which in turn receives feedback. Then, when it is finished, this is implemented into the whole. That process is done with each small task.

In this process, teamwork is very important. Mainly giving feedback on the work done but also receiving and processing feedback.

### Sprint

A sprint is the basis of scrum. The duration of a sprint is determined in advance of a project, which is usually three weeks.

At the beginning of a sprint, a number of tasks are chosen; these tasks are expected to be completed by the end of the sprint. The expected time of a task is then often estimated with sprint poker. Work is then done on these tasks during the sprint.

When the sprint is over, the team and the stakeholder meet to discuss the chosen tasks and give feedback on them. Another agreement is then made as to which tasks will be created, and so the project continues until it is finished.

## Agile requirements

Recognising Agile requirements helps a team prior to a project to understand the features and functions of a the project. This includes preperformance benchmarks, this also helps in understanding the features.

### User stories

A user story helps in understanding what an end user wants to be able to see and do in a project. It helps to have a discussion about a user story, new points often emerge which in turn are a separate user story. Or just a modification to the user story. Examples of a user story are:

* As a user, I would like to be able to use the "remember password" button when I log in.
* As a user, I would like to have a clear overview of which products are in my shopping basket.
* As a user, I don't want to have to wait longer than one second for the page to load.

### Functional requirements

The functional requirements describe what the end result should contain and what it should look like. Users will find this helps to list all tasks and also to base designs and other technical elements on. Examples might include:

* The homepage should contain a contact form.
* A place where users can see a purchase history.
* A search bar that makes it easy to find products.

### Non-Functional requirements

Non-functional requirements describe how well a product should perform. Non-functional requirements depict the overall qualities of an item or framework. These are often things that can be measured. These can be things like security, page load time, product quality, etc.

### Acceptance tests

Acceptance testing is a scenario that a user has to go through to get to a certain place, or be able to control something. This can be used to experience how a feature works and whether the user can find it easily.

Source: <https://www.indeed.com/career-advice/career-development/agile-requirements>