## 2.4. Development Method

At a high-level overview the software development process can be separated into two main methodologies:

* Plan Driven: All the steps are carefully planned, variables such as budget and features are fixed early on in development and overall progress is compared to these predefined criteria.
* Agile: All the steps are done in small but frequent increments with minimal initial planning and are modified according to the constant feedback and changes in requirements

### 2.4.1. Bohem and Turner chart

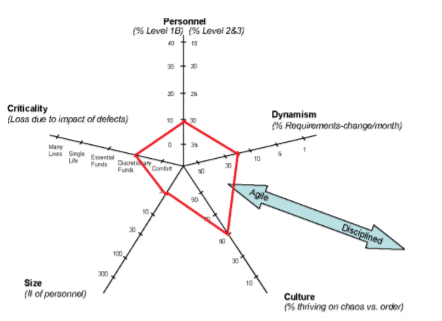


Figure 1 Bohem and Turner diagram depicting our team

Choosing the right development method is key to a successful product. One way of choosing the right development method is by assessing the project, the team, the project scale and the customer of the final product.

The following diagram, a Bohem and Turner chart, helped us in picking a development methodology. According to it, we can conclude the fact that our team needed an agile method of sorts, given that most criteria point towards the center of the star with slight tendencies towards the exterior.

### 2.4.2. Considered Methods

Following the Bohem and Turner chart, we have decided to take an in depth look at three of the most popular Agile development methods.

#### a) Extreme Programming (XP)

Extreme Programming (XP) is an agile development methodology which shifts its main focus on the overall quality of the software. XP is based on four main values:

1. The first and arguably the most important value is communication. Communication is crucial in any methodology, but especially in Agile as the requirements can change rapidly.
2. Simplicity, the simpler the solution the less chance something fails. Development is to be started with a simple solution and improved with continuous refactoring (also one of the key XP practices).
3. Feedback is the 3rd value which is a must for high product quality.
4. The final value is courage. It encourages making decisions without the need of a formal process.

This is the complete opposite with Plan Driven methods like UP where every significant change is accompanied by an abundance of documentation. XP lists 12 key principles for creating a high-quality software. Among the 12 principles is test driven development or TDD for short. TDD requires writing the tests before developing the product. This ensures the number of errors/bugs is minimized in the later stages of production. XP takes the Agile incremental philosophy into programming with small sized releases and continuous integration. The small releases allows the end users to test the system in early stages of the development process, compared to Plan Driven products, even though they only include part of the intended functionality.

#### b) SCRUM

Scrum is one of the Agile development methods most suitable for small development teams working in sprints, ranging from a single week up to a month. At the end of every sprint, value should be presented to the customer by being one step closer to completion. Unlike XP (Extreme Programming), the scrum methodology focuses only on the development process. It defines three roles:

* Product owner, who represents the product stakeholders (ex. the company receiving the final product).
* The scrum master (usually a single person, but it can change over sprints) who is responsible for the scrum process and removing any distractions that the team might encounter.
* The team, which represents all the development team who design, create, implement and test the system (the order is not necessarily fixed).

At the start of each sprint, a sprint planning meeting is held. In this meeting the product owner selects the tasks that should go into the sprint backlog based on the current state of the application and the feedback given by the final users/ customers.

Each sprint is closed with a sprint review where the product owner shows off the accomplishments to the stakeholders. An internal meeting is also held to discuss and improve upon the development process itself.

Scrum also includes daily standup meetings where the members discuss what they have done since the last meeting, what they will be working on and what problems did they run into. These problems are to be resolved by the scrum master.

Besides the usual sprints there is also a special initialization sprint called “sprint zero”. Sprint zero is used to set up the work space for the project to begin. Investigate any problems which can potentially slow down the process (such as unknown technology), set up the development environment, etc. Unlike UPs inception and elaboration phases, the requirements set in sprint zero are not final and are expected to change over the course of the development.

#### c) Kanban

As with the previous two agile development methodologies, Kanban aims to work with the chaotic nature of variable requirements and the need to shift focus between priorities quickly. Kanban also focuses on making the development as smooth as possible by making transparency, respect and agreement as its values (there are a more values which are equally as important, but these are the ones we aimed to retain when considering the method).

Leadership is an integral part and is encouraged at every level (again circling back to its values) and the changes are smaller, more evolutionary rather than revolutionary. Frequent deliveries in small sizes is a must, and the quality must be up to scratch with the rest of the teams work. The scope and progress of the development work is visualized using a Kanban board, where teams agree to which tasks to choose. If a task happens to last longer than the intended time, it will be put back to the backlog and can be chosen again. Making policies explicit is a must. Policies, such as the definition of” ready”, or the definition of ”done”, should be used rarely for maximum impact. They’re used to ensure a common ground across both the team and the stakeholders.

Feedback loops, to ensure the upmost relevant tasks, are implemented; and are a good way to minimize time wasted on less important tasks. Those feedback loops include: Strategy Review (Quarterly), Risk Review (Monthly), Replenishment Meeting (Weekly), Kanban Meeting (Daily) and Delivery Planning Meeting (variable)

### 2.4.3. Chosen Method

Since all of the considered methodologies contained traits that we considered both useful and impractical, we decided to create a unique combination of all the three major Agile development methods. One that would contain the following guidelines:

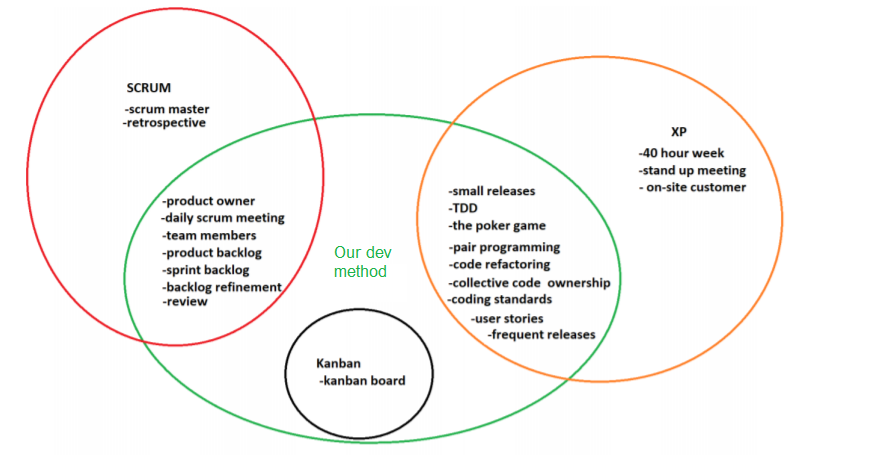


Figure 2 Detailed overview of all elements, of the dev method we used

Our aim was to first build a Minimal Viable Product (MVP). After the MVP has been finished, with small increments we added functionality without braking the core application.

We also put a heavy emphasis on communication, as we noticed early on the importance of keeping a shared and up to date understanding of the developments state. Alongside communication, Pair programming, and TDD were used extensively. As expected TDD slowed us down in the early stages of development, but it proved to be a major contributor to the quality of the overall application and saved us from the headaches of debugging in the later stages where the application was becoming more complex.