**Beet Seed 1**

| **Methodology** | **Pros** | **Cons** | **For which industry is appropriate** |
| --- | --- | --- | --- |
| Waterfall | * Any errors detected can be fixed before the next phase * It’s easier to measure progress, since one stage can only start when the previous is completed | * Each phase is dependent of the previous to start, leading to delays in the project * Any changes in requirements are harder to implement, since it implies starting all over again | Companies with smaller projects or projects well defined that are unlikely going to change, since with this methodology, any changes imply starting all over again |
| V-model | * Focus on validation and verification activities through the various phases helps to detect and fixes defects * Emphasis on testing through all phases allows to ensure quality * Better link between the requirements and the product | * Only works with simpler projects were there’s less likely for the requirements to change * Any changes in requirements are harder to implement, since it implies starting all over again | Companies with larger teams and projects where the requirements are well set and are unlikely to change. This methodology requires human resources and time and changes in the requirements can lead to the starting over of the project |
| Iterative Model | * It is easier to implement requirement’s changes * Its iteration has design and testing changes, allowing the product to change and improve | * It works better for smaller projects * It can lead to too many design changes due to changes in the requirements * More resources are needed | Companies with products whose requirements are likely to change. Since there’s iterations on the project, it easily accommodates requirement’s changes |
| Incremental Model | * Can be used regardless the team size * Allows improvement to the product at each increment * Allows for a risk management, since it allows to identify and fix issues when they are detected | * Planning at the beginning is necessary to ensure that each increment is correctly performed * It can leads to numerous and unnecessary changes since they are easily implemented | Companies that want to quickly launch their product, because there’s always a possibly to change the product, if needed, at any increment |
| Spiral Model | * Changes of requirements can easily be implemented * Clients connect with the product at early stages * Risk management is important to minimize the risk for the product | * Complex methodology not suitable for smaller projects * Risk management requires experienced workers * The various analysis can take too much time | Companies that are developing project with high risk levels, since the client is involved and there is a risk analysis through the process |
| Scrum | * Teams work more closely, sharing their ideas what can lead to innovative approaches * Teams are less micromanaged which can make the works feel more responsible for their tasks, increasing their motivation * Daily meeting allow to address day-to-day problems and fix them quickly | * It’s more difficult for inexperienced workers to join, since it requires a lot of experience and knowledge of how this methodology works * It only works with small teams, since the daily meeting can take up too much time with big teams * If the tasks are not completed in the sprint they are supposed to be done, it can leads to project delays | Companies with large projects that can be divided into small goals since it is the sprints main focus |
| Kanban | * Allows to quickly identify which tasks are in each stage (to do, on progress, completed) * Better planning of tasks allows the focus to be on finishing the assigned tasks * Improvement of focus, since each team members knows exactly what needs to be done | * If a tasks is not correctly identified it can lead to tasks not being performed or for a task to be done again - wasting time * There are no time frame to perform each tasks, that can lead to delays in the project | Companies with complex projects that can be divided into smaller tasks, allowing to quickly assess what is done and needs to be done to finish the project |

**Beet Sprout 2**

* **In your opinion, why did the Agile manifesto appear?**

The Agile Manifesto was the result of a need to simplify and make software development methodologies more functional. Even though the methodologies used, at the time, were delivering results, they were too complicated and time consuming.

The Agile manifesto was created as a way to give guidance to software development teams by defining priorities between different aspects of the software development work.

* **What problems did it have to solve and did it succeed?**

The Agile Manifesto allowed for a number of problems being solved, especially through the implementation of its 4 Principles.

Valorizing individuals and their interaction over processes and tools allows for better communication and quicker problem solving, since it is people who are behind every software development.

The functioning of the software is prioritized over documentation, because what the client wants is a software that functions. Even though the documentation is important to know how the software works, without a functioning software the documentation has very little use.

Evolving the customer in the software development process allows the client to understand what is being done and to participate actively in the various stages. With that, If any changes are necessary, the software development team can implement them more quickly.

With a well established plan it is more difficult to accommodate changes necessary. With an Agile methodology it is more easy to implement needed changes without compromising the entire product or delaying the project, because these methodologies are more open and responsive to change.