**Mushroom++ Process Report**

**IoT group: Daria Maria Popa (293087), Natali Munk-Jakobsen (293132) and Mihai Anghelus (293169)**

**Data group: Levente Nagy (293115), Audrius Sauciunas (293156), Samantha Nettesheim (293089), Shaoyu Liu (294404)**

**Android group: Kristóf Lénárd (293110), Bogdan Mezei (293137), Uldis Alksnis (293168)**

**Supervisors: Erland Ketil Larsen, Astrid Hanghøj and Kasper Knop Rasmussen**

**VIA University College**

**[Number of characters]**

**Software Technology Engineering**

**4th Semester**

**4th June 2021**

Version: June, 2021

*This page intentionally left blank*

**Table of content**

[1 Introduction 1](#_Toc73006663)

[2 Group Description 3](#_Toc73006664)

[3 Project Initiation 4](#_Toc73006665)

[4 Project Description 5](#_Toc73006666)

[5 Project Execution 7](#_Toc73006667)

[6 Personal Reflections 8](#_Toc73006668)

[7 Supervision 9](#_Toc73006669)

[8 Conclusions 10](#_Toc73006670)

Appendices

# Introduction

During this project, two methodologies were used to execute tasks and proceed toward the predefined goal. These were Agile Unified Process (AUP) and the Scrum framework. Unified Process consists of four distinct phases: Inception, Elaboration, Construction and Transition.

During the first phase, Inception, the team defined the base technical requirements of the project, agreed on a common interpretation of facts concerning both the problem domain and the proposed solution, and established the core non-technical requirements, such as solution functionality, targeted process and timeline, and feasibility. This phase ended on the 17th of March 2021, beginning the next phase.

During the second phase, Elaboration, the team established the groundwork for all future work. Common design elements, such as the connectivity between different parts of the system, and general architecture were established, with supporting materials such as descriptions, core documentation and a working proof of concept.

During the second half of this phase (from the 7th of April 2021), the Scrum framework was also used to support the development. This was due to a variety of factors which will be further detailed in the Project Execution section.

For Scrum, Levente Nagy served as Product Owner and Kristóf Lénárd served as Scrum Master. All other team members served as members of the development team. This team was organized into smaller departments, consisting of three or four members, working on a distinct part of the system, led by the agile role of technical leads. Scrum was organized regularly, with the usual set of meetings, ie. sprint planning, daily scrum, sprint review and sprint retrospective. Further details can be found in the Project Execution section. Scrum was used, as previously mentioned, during the second half of the Elaboration phase and during the Construction phase.

During the Construction phase, the main focus was on the development of the software code, along with all required supplementary materials such as diagrams, reports, and other documentation. This was the longest phase of the project, stretching from the 28th of April all the way to the 2nd of June, with more frequent working days than before.

On the 3rd of June 2021, the project reached Transition phase. This phase was the shortest, even though, in theory, it could possibly be indefinitely long, since this phase consists of post-release support. This was also when the system was deployed to its currently final Production Release iteration- This phase also included finalization of supplementary materials such as user guides, and the final hand-in of the project.

# Group Description

## Android team

The Android team had three group members by the end of the project: developer Uldis Alksnis from Latvia, developer Bogdan Mezei from Romania and technical lead and Scrum master Kristóf Lénárd from Hungary.

In our group, tasks were divided fairly evenly, with all members working on coding, Kristóf working on the process report, along with handling the compilation of materials for the hand-in, while Uldis and Bogdan worked on the project report and technical materials.

We all had some experience, of course, this being our fourth project at VIA. This we feel helped us in many ways. One was that we have had more experience with software engineering and development in general, enabling more individual work, which is one of the core tenets of both Scrum and AUP. We have also known more about the issues we might have ran into during the project, which helped us in estimating what we need, and what we can accomplish in a given timeframe. Both of these were vital to the successful completion of the project.

In the group, we have emphasized working with the other philosophies of the Agile Unified Process, besides the aforementioned individual work.

We have not based our work on huge amounts of documentation, but planned connectivity and other things only in advance, the ones where substantial modification would have needed the other teams to adjust as well. Obviously, by using Scrum, we have worked in an agile context. We, from start to finish, prioritized (after completing general architectural duties) working on the most high-value requirements, only adding others after we had known that what we needed earlier worked.

All of these were core issues that we considered as such; therefore, we feel that our groupwork was successful in this project.

# Project Initiation

Before any group starts a project, there must be something that generates the group, a metaphorical force that gets the members of the group together. Of course, in our case, one aspect of this was the fact that the project is a mandatory part of our studies here at VIA. There were other factors, though.

At the beginning, some of us have tentatively formed two groups, each consisting of only a few people, expecting a similar project as we have experienced in previous years. This, as it later turned out, was not the case. When the project description meeting was held and the full scope of the project detailed, we had known that we needed more people. Soon after a group was generated – the original group of seven. This was still too few, so we sought to attract more people, ones that we knew we could trust to accomplish their tasks. Thus, through friends and contacts, we came into contact with another small group. This group then joined us, bringing the count of our members to eleven. While this number would later be reduced to ten by the unfortunate departure of Eva Nikolaeva, this was still a number that was sufficient – and thus, the group came to be.

So now we had a group. We asked the question: “What next?”. The answer was obvious – we need to have something as our project. For that we needed an idea, and so we sat down (unfortunately, due to the corona-situation, not in person, but online, which we kept on using through the whole project). We have discussed many ideas, iterating through the ones that we thought of as interesting or otherwise appropriate. The results of this can be seen in Appendix C. In the end, we reduced the final round of ideas to two, and voted on the better one – thus, we selected our topic.

We then knew what should be our next step – forming a consensual understanding of the problem, and detailing our way of solving it.

# Project Description

The first challenge when writing the project description came from the fact that most of us were unfamiliar with the problem domain, that is, the processes and inner workings of fungiculture. This problem was tackled by intense research, and with the most experienced member of our group, Audrius Sauciunas, working to secure more data from personnel who were then working in the industry. This gave us a significant amount of data on which we could base our background description. This helped us define the problems within the entire problem domain more precisely, and have allowed us to select a realistic goal.

Many of the following parts of the project description phase were derived from this information, as per usual procedure. These sections were no less important, however, since, for example, the supporting questions of the problem statement, formed the basis of our functional requirements later.

These parts were also critical procedurally. Before, we have worked on either single-issue problem resolutions (such as the decision of the idea), or on heavily factual texts (such as the background description). These sections, however, taken all that and added the problem of balancing the derivation of certain elements from others, and negotiating which aspects should we focus on. Both of these are critical to the process of software development. Therefore, we have, with the benefit of hindsight, have seen this as the final rehearsal of these before starting the more technical work on the project. During these, all members demonstrated such conduct and integrity that it was obvious to all members what to expect if and when the group again had similar meetings. We have aimed at having similar discussion in both procedure and conduct for the entire remainder of the project, a goal that we hope we have achieved.

This phase was, however, perhaps the most important in forming a preliminary way of meetings. During this phase, we have agreed on how to hold meetings, both with regards to meeting issues and procedures, and the conduct of the person chairing the meeting. These we have sustained during the rest of the project. Some of these matters derive from the group contract – however, some are merely procedural, which occasionally required adjustment.

The following sections, however, were more technical in nature. For example, the time schedule, while without a doubt important, was a highly factual and technical document, with perhaps only one thing remarkable with regards to the process. This was the fact that the group showed admirable restraint and appropriacy in discussing these issues. There was nearly no time wasted, which proved to be of advantage later, as we have continued to show these traits in later procedural discussion, which have allowed us to spend more time on, and focus better on, the development of the project instead of having to spend additional time on resolving minor issues. We were still able to solve all issues – but the time and energy not expended on these were of great use in the project.

All in all, however, we view this phase as highly influential in conducting this project, and highly successful in enabling further work on the project.

# Project Execution

## General work

After getting the feedback on the project description document, and finalizing it based on that, we began working in earnest on the first phase of the project, that is, the analysis section. This, and the first part of the Design section belonged to the first half of the Elaboration phase of AUP, involving four disciplines: the main discipline of Modeling, and the supporting disciplines of Configuration Management, Project Management, and Environment.

Our execution here was very simple: we held regular meetings on every working day (ie. on Wednesdays), and talked about the things we needed.

First and foremost was the definition of the requirements, both functional and not. These we derived from two sources by consensus – the functional ones from the project description document, chiefly the Problem Statement and Definition of Purpose sections, while the non-functional ones were derived from the predefined mandatory requirements document.

From these, further documents were derived: the use cases, the supporting documentation for said use cases (such as activity diagrams and test cases), and other materials that composed the analysis of the problem domain.

All of these, however, had one and only one goal: to help define a document that can both be non-technical, and be the core of the technical implementation. This was none other than the domain model, which was the last document created for the Model discipline of AUP.

As this discipline Model, according to the official description of Agile Unified Process, is defined as “The goal of this discipline is to understand the business of the organization, the problem domain being addressed by the project, and to identify a viable solution to address the problem domain.”, the domain model and the other aforementioned documents fulfil this purpose. Therefore, we concluded that this discipline was completed for the time being, and the team could move on to the next main discipline – Implementation.

Here is where I ask others to step in, and help me detail what the reflections of each team are in this phase, where we, while not working together on day-to-day issues, still relied on each other to form a coherent system.

## Android team

The Implementation discipline began after the Model discipline was finished, and lasted for the second half of the Elaboration phase and the entirety of the Construction phase, running concurrently with the Test and, occasionally, the Deployment discipline.

This part was, naturally, the largest – this is where we had to develop the system, test it, and deploy the final version of it to gain documents that we needed for the hand-in.

To facilitate this, and support the AUP, we have employed another framework, both in-group and in-team: Scrum.

This, of course, had numerous advantages: experience, a clear and concise methodology of working, and a good way of tracking the supporting disciplines of the AUP.

The implementation of Scrum followed the constructed framework. Before each sprint, the project backlog was reviewed by the Scrum master and the product owner, and constructed or reconstructed in such a way that ensured continuous refinement of tasks, and facilitated the selection of sprint goals.

This was followed by a sprint planning meeting. During this meeting, the teams were assessed individually, with any member of the group able to ask questions and add comments to anything that was decided, with these being explicitly recommended in tasks involving multiple departments. After the tasks have been decided and estimated for all teams, the group clarified the acceptance criteria for that given sprint. When this, too, was concluded, the group voted to accept these resolutions as the sprint plan.

The daily scrum meetings were individual for each team. Usually, they started at 10, with the Android team’s meeting taking place first (this was so that the Android team could start working while the Android team’s technical leader Kristóf Lénárd, who was also Scrum master, could hold the meetings for the other teams). During this meeting, three things were usually discussed: the completed work, the work that the team members planned on doing that day, and whether there were any impediments to progress, either from a technical or a process standpoint. Occasionally, “after-party” meetings were conducted later, to discuss a problem in more detail.

At the end of the sprint, two meetings were conducted as a whole group, usually directly following each other. These were, of course, the sprint review and sprint retrospective meetings. The sprint review meeting, as usual, focused on the completed and incomplete tasks. We discussed the tasks that were completed, how they impacted both currently existing and future work and how they contributed to the sprint goal. Incomplete work was also discussed, along with reasons of delay, and how that impacted the future work. Finally, the product owner, as customer representative, made suggestions with regards to future work, which in turn impacted the next sprint planning meeting.

The other meeting was the sprint retrospective. Here we discussed what we thought of the past sprint, and what issues we had with the process. Sometimes, there were no issues, other times, however, the team resolved to have a quick brainstorming session to try and find solutions to any aforementioned issues. Usually these lasted for about 30 minutes with discussion, at the end of which the group resolved to adopt one or more of the proposed solutions. These were then employed by the teams, enabling better cross-team work.

The Android team feels that these methods were effective at driving the development forward. We feel that the group has avoided micromanagement, yet still was able to work together on their goals, even though the group was essentially working in three smaller pieces. We feel that out of the methods we are familiar with, this one was perhaps the most balanced with regards to working together in three small teams.

We also think that with these frameworks and methodologies, we have been able to work together efficiently, and were able to execute the tasks that we planned to. Therefore, we think this project was successful not only in a technical, but also in a procedural sense.

# Personal Reflections

For content see Appendix 2 “Process Report – VIA Engineering Guidelines”.

# Supervision

## Android team

The Android team was completely satisfied with the supervision. We, personally, have not needed to work with the supervisors a lot, except for a few questions about implementing certain things and clarifications to other small matters, but all of these questions were answered in a timely and helpful manner.

The other times we met with the supervisors, during feedback meetings, we received helpful information, tips and other guidance that helped us create, even though some of the suggested things were merely not scoped for that release, instead of not being thought of. However, one thing I must highlight is the amount of preplanned feedback we received, which was much greater than the previous two semesters. We feel that this is a welcome improvement.

Communication with the supervisors was easy – we could just send an email, or ask them in a Zoom chat, and while this was not as fast and quick as the in-person method (which is, unfortunately, still unusable for some of us), it was still fast enough for basically all things that we could need.

All in all, while we have not had much contact with our supervisors, when we had, their help was greatly appreciated.

# Conclusions

**Appendices**

* Appendix A – Scrum log book
* Appendix B – Azure DevOps log
* Appendix C – Ideation document