Experience Report

Alexandros Gonatas
Department of Computer Science and Engineering
University of Gothenburg
alexthecipher@gmail.com

Alexander Spetz
Department of Computer Science and Engineering
Chalmers University of Technology
spetzal@student.chalmers.se

Haoning Ma
Department of Computer Science and Engineering
University of Gothenburg
mahaoning21@outlook.com

Xingrong Zong
Department of Computer Science and Engineering
Chalmers University of Technology
xingrong@student.chalmers.se

***Abstract*—**

I. Introduction

The project focuses on developing a gamified task management application aims to help users overcome procrastination and improve productivity. The application allows users to log daily tasks and, upon completion, receive rewards that can be spent within the game component. This project provides a context for applying requirements engineering methods, including the elicitation, specification, validation, and prioritization of requirements, while also promoting teamwork and reflective learning.

II. Requirements Engineering Work

Our requirements engineering work started with understanding the problem domain and user needs. We questioned whether similar products exist, what impact they have made, how users are drawn to similar applications and if this kind of application is needed during the current socioeconomic climate.

We concluded that this application is needed, since all members of our team would benefit from it if we were users. As students, we have faced issues with procrastination, loss of motivation, anxiety, feelings of overwhelming pressure- all which would lead to forgetting daily needs, from drinking water to knowing when to start working again. An application like this would have been beneficial for us during these trying times; thus its usefulness is validated by the team's own needs.

Our work together has been complicated, since it was our first time working together and we needed to introduce ourselves as well as assess how well our bond had to be in order for our team to be successful. A common problem was the abstract concept of the project, since most of us had not worked in a requirements engineering project before. However, when we used a hands-on approach to our problems, we started to get the gist of it, which is described in the following sections.

III. Methods and Techniques

A. Elicitation

For the first revision of our project, we conducted brainstorming sessions, document analysis and the 'Hall of Fame' [1] activity.

The brainstorming sessions were a natural part of our meetings, since most of the user requirements we included in the revision were derived from our personal experiences and needs as full-time students. During the sessions, we came up with our own ideas, wrote them down, and selected the best from each pile.

The next step we followed was document analysis, where we picked applications that have similar concepts and analyzed their characteristics: functionality, user rating, and profit. More precisely, these applications are: Finch [2] and Pokemon Go, with the latter not being researched yet.

Lastly, we worked with the 'Hall of Fame' activity during the creativity workshop provided by the course. We chose 'The Joker' as our persona and analyzed different aspects of his needs:

- What are the needs of people living in the same town as him?
- What is his personality like?
- What are his needs, generally?
- What would he need from an application like this?

B. Specification

During specification, we wrote our requirements in a clear and simple way. First, we separated functional and non-functional requirements. Functional requirements describe what the system must do, for example: create a new task, edit a task, give rewards when a task is finished, and use rewards to buy game items. Non-functional requirements describe how the system should work, for example: the app should respond in less than 2 seconds, and the interface should be simple and smooth.

Every requirement has an ID, a title, and sometimes a short explanation. We also made a basic use case diagram to show the relation between the user and the functions of the

application. It was sometimes hard to know the right level of detail, so we kept it high level and are planning to add more detail in the next revisions.

C. Prioritization

After we listed all requirements, we needed to know which are most important for the first revision. We used the MSCW method (Must, Should, Could, Won't). We discussed and put every requirement into one of these four groups. For example, "create and finish a task" is Must, "change theme color" is Could, and "multi-language support" is Won't. This helped us focus on and plan what can be implemented in the short project time.

IV. Reflection

A. Elicitation

The brainstorming sessions were the most useful tool we used during the elicitation process. Initially, it acted as a mediator in our team, as it allowed us to understand each other, how each individual works, and structures their ideas. We would evaluate it as a team building tool,as it allowed us to understand everyone's mindset.

Besides this unexpected outcome, most of our requirements (functional and non-functional) sprouted from these sessions, which helped our project initially.

The 'Hall of Fame' activity was awkward and confusing while it lasted, but in the end, we came up with unconventional and creative features for our application. Using this activity proved difficult for our coordination. However, picking 'difficult' personas, such as The Joker, proved fruitful for the expectations we must have for 'difficult' users in our case. For some of us, it felt like the most creative technique we have used thus far and the team seemed happier when using it.

B. Specification

Writing the requirements was not always easy. Sometimes we wrote requirements that were either too detailed or too abstract. It was also not easy to uphold a consistent style between team members. But working together helped us to see changes and keep track. Next time we would start with a fixed template earlier and decide the detail level in advance.

C. Prioritization

Choosing the priority was easier than we expected. The MoSCoW method was simple and worked well. Sometimes we had different opinions about which group a requirement belongs to, but discussion solved it fast. Next time we may also add a simple scoring (like 1–5) for risk and value to help decide better.

References

[1] More on the Hall of Fame creativity activity: https://becreative.city.ac.uk/details.php?id=14

[2] https://finchcare.com