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

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. The underlying idea is to transform the routine act of task management into an engaging experience that motivates users to maintain consistency and develop self-discipline.

This project provides a context for applying requirements engineering methods, including the elicitation, specification, validation, and prioritization of requirements. Through this process, we aimed to capture and define user needs accurately while ensuring the requirements remained feasible within the project scope. Additionally, the project provided an opportunity to strengthen teamwork and collaborative problem solving skills, as this was the first time most members worked together on a requirements-focused development task.

II. Requirements Engineering Work

Our requirements engineering work started with understanding the problem domain and identifying the needs of potential users. We questioned whether similar products already exist, what impact they have achieved, how users are drawn to similar applications and such an application is still needed in the current socioeconomic climate.

We concluded that this kind of 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 poor time management and neglected daily routines, ranging from drinking water to knowing when to start working again. These shared experiences confirmed that a gamified productivity tool could be beneficial for us during these trying times; thus its usefulness is validated by the team's own needs.

Our teamwork evolved throughout the project. Initially, collaboration was challenging because it was our first time working together, and we had to establish communication norms, align expectations, and build trust. The abstract nature of the project also made it difficult to translate ideas into concrete requirements at first. However, adopting a hands-on approach, regular meetings, brainstorming sessions, and iterative feedback, helped the team gain a clearer understanding of the requirements engineering process. These experiences are further detailed 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?

For the second revision of out project we made a questionnaire that was answered by 78 people. The purpose of the questionnaire was to get a better understanding of who the potential users would be and what features they would want in the app. It was divided into 4 sections, each of which had a purpose.

1) Who are you?
2) Assessing your daily life

3) Organizational methods
4) Usability expectations

The first section of the questionnaire was used to determine who was filling in the questionnaire. So we asked for their occupation, age, gender and if they use a smartphone or task management apps. In the second section the questions were more aimed at finding out the habits of the person, for example how much time they spend on their phone or if they procrastinate a lot. The third section was mainly used to find what kind of organizational methods were used by the person and if they would be receptive to this kind of app. The final section only opened to those who answered ”yes” or ”maybe” when asked if they would use a gamefied app to help them organize tasks. This section was mainly focused on what actual features they would like to see in the app. One example was what kind of reward they would like for completing tasks.

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. Validation

To ensure that our requirements were correct, complete, and aligned with user expectations, we used several validation techniques. The main approaches were walkthroughs and feedback analysis from the questionnaire results.

We held walkthrough meetings where we went through each use case and functional requirement to ensure that they matched the intended user goals. Since we did not have access to real end-users during this phase, we used the questionnaire data as a validation reference to verify that our requirements corresponded with user expectations.

In addition, we checked that each requirement could be traced to at least one elicitation source, which strengthened the credibility and traceability of our documentation.

D. 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.

The questionnaire worked relatively well, as it gave us a lot of different useful data and helped us specify a few potential stakeholders for the application. Furthermore, it also gave us more information to figure out user stories. However, there was a lot of data and it was difficult to actually sort through it and find something useful from 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. Validation

Validation was both successful and challenging. It helped us confirm that our requirements were logically consistent and addressed the users’ main needs identified from the questionnaire. However, since we lacked direct user interaction, some assumptions about user behavior remained unverified.

The walkthrough sessions enhanced our shared understanding as a team, which reduced misinterpretations during later revisions.

One challenge was that our validation relied mostly on our own perspective as potential users, which introduced bias. Inn a real-world project, we would have conducted validation workshops or prototype testing sessions with actual users to gain more reliable feedback.

This experience demonstrated how verification and validation connect directly to project quality. Verification ensured our documentation was consistent and complete, while validation confirmed its relevance and usefulness. We also realized that contextual risks, such as limited time, team inexperience, and lack of user access, can affect validation quality. Despite these risks, our validation techniques improved the overall success of the project by helping use produce clearer, more realistic, and user-focused requirements.

D. 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.

Overall, the combination of brainstorming, questionnaires, specification methods, and verification/validation activities contributed significantly to the project's success. We learned that maintaining quality in requirements engineering depends not only on the techniques used but also on team coordination, traceability, and early user involvement. If we were to do this as a real project, we would plan more structured validation sessions with external users and integrate risk assessment earlier in the process.

References

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

[2] https://finchcare.com