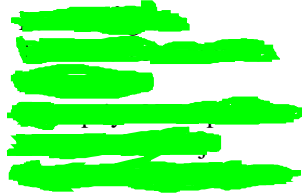


Experience Report

Requirements specification for Student Accommodation Matcher System



student.chalmers.se
student.chalmers.se
student.gu.se
student.gu.se
chalmers.se
student.chalmers.se

Abstract – Requirements engineering is an important part of software engineering that enables to define, document and maintain requirements of a software. Elicitation is a technique in requirements engineering used to research and collect requirements from customers, users and stakeholders. In this project, interviews and brainstorming elicitation techniques are used to collect the requirements of the system. The requirements collected through elicitation are presented using context diagram, task description, stakeholder map, data model and data dictionary, which are types of specification styles. Prioritization techniques were later used to define the priorities of the functional and non-functional requirements of the system. A mockup was created to show the realistic representation of what the system could look like. The team has, for instance, learned that specification styles are used at different levels of abstraction and that mockups can show the design choices and to identify problems of the system at an early stage.

Keywords – Elicitation, requirements prioritization, mockup, brainstorming, data model, data dictionary and interview

I. INTRODUCTION

The high demand for accommodation is a worldwide issue nowadays, and Gothenburg is not an exception. Gothenburg is the second biggest city in Sweden where students come to study every year. However, due to the high demand for accommodation students are facing difficulties to find accommodation for the period of their study. Existing agencies often use queue systems that are long and frustrating, which may lead to students not getting accommodation in time or not getting it at all. Students have to look for accommodation in several places where other people already have been registered and collected queue points for a long time. Because of this students want a system where they can find accommodation as soon as they have been admitted to the university without the need to be in a queue for a long time before their studies.

II. TECHNIQUES

The team used the following methods to document the requirements for the project idea selected.

A. Elicitation

Eliciting information from related stakeholders regarding their needs and expectations is one of the key initial tasks that need to be done. Four of the elicitation methods used are explained below.

a) Interviews

Lauesen [1] lists interviews as one of the methods to involve the stakeholders and suppliers in the requirements gathering process. He suggests interviewing both expert and ordinary users alike. Students and landlords were selected as the primary stakeholders to interview. The team created a list of questions that it wanted to be answered. The team used a semi-selective approach and tried to make the questions as open-ended as possible. 21 people (16 students and 5 landlords) were interviewed in the process. The interviews provided different perspectives regarding the needs of the stakeholders. Even the same type of stakeholders have different needs and expectations depending on their situation. For instance, the needs of an international student differs from those of a domestic student even though they are of the same stakeholder type. Most of the responses were as anticipated by the team. Since the team members had some prior knowledge of the domain, there was some bias regarding anticipation of the responses. After the interviews, the team gathered the requirements and documented them in the specification document. When all of the requirements were shortlisted, the team decided not to do further interviews since the focus was aimed at refining and specifying the ones that were already collected. If the project had included an additional release, the team would have found value in doing follow-up interviews in order to validate the requirements.

b) Brainstorming

Since the team members have prior knowledge in addition to the knowledge they gathered through interviews, brainstorming is an appropriate technique to come up with new ideas. In this technique, the team worked in two phases; first, a moderator gathered ideas from each team member, and then each idea was discussed and analyzed.

The team then combined various responses from the interviews and brainstormed the possible requirements for the system. The team devised additional requirements besides the requirements obtained through the interviews. The members were motivated to stimulate new ideas with respect to the problems that they gathered from their interview sessions. Any new ideas raised by the team members were noted without any criticism. This motivated every team member to come up with his/her own new ideas without being shy or afraid.

The new ideas and requirements raised through brainstorming were discussed and analyzed by the team members to filter the ideas and requirements that are appropriate to the project at hand. After thorough discussion and analysis, the team members selected the most important ideas and requirements that pertain to the proposed idea.

c) Hall of Fame

Hall of Fame [4] is a technique that produces ideas and insights by formulating ideas by thinking of someone's (fictional or real) traits and behaviours. The process involves creating a 'Hall of Fame,' which is a physical representation (either on a board or paper) of those traits and behaviours. The team selected three personas - Steve Jobs, Pele, and Sherlock Holmes. Any idea that came to discussion was noted on the board. After choosing the personas, the team decided to find requirements by thinking of the ideas as these personalities would have thought. The team used a list of questions to think of ideas for gathering requirements. The team deliberated on how each of the characters would have thought of the requirements for the project. After the team had gathered the requirements, the requirements were noted and prioritized to be included in the project.

d) Creativity Triggers

Creativity triggers [5] provide guidance to stakeholders to discover new requirements associated with a particular quality of a product, service or system – a quality that is believed to be associated with innovative solutions. One example of a creativity trigger is convenience, which guides the stakeholders to construct a hassle-free product, service or a system. While using this technique, the team used six triggers to work on and found new requirements according to them. The triggers used are adaptable, completeness, economical, light, entertaining and durable.

The team found Creativity triggers as a more successful technique, mostly because of the fact that Creativity Triggers produced more useful ideas compared to the Hall of Fame technique. So the team decided to use creativity triggers for further requirements generation. Instead of being a source of creativity, the team found the Hall of Fame limiting, since it was challenging to relate the characters of the Hall of Fame to the project. The creativity triggers, on the other hand, were found less limiting since the team found it easier to come up with ideas based on a specific word, rather than characteristics of a

persona. The team sees value in trying other techniques as well since the success of a certain technique might be dependent on the context. By using the techniques, the team learned to gauge their effectiveness and use it to add and improve the project's requirements.

e) Reflection on Elicitation Techniques

The team could have done more interviews with other stakeholders such as universities and payment services and could get more insights and information about the system. But since interviews are time-consuming and stakeholders are busy, the team only focused on the main stakeholders (i.e landlords and students) to save time. It could also have been possible to conduct questionnaires, but since questionnaires also are time-consuming, the team found the interviews more value-adding. This tradeoff was necessary with respect to the limited time of the project.

B. Specification

a) Context Diagram

There are many ways to represent the working of a system but the team chose to use a context diagram. The context diagram gives a high level abstraction of how the system works. This was found by the team to be a great way to represent the system from the customer point of view, who is not aware or concerned by technical detail but at the same time covers all the important components involved in the working of the system.

b) Task description

The tasks which need to be implemented by the system are described using subtasks and work areas. Most of the tasks include information about the start and the end of the task process. They also show the frequency of a user performing a specific task and the frequency of task being used. The tasks that the system will handle are divided among six general work areas along with subtasks to show the working of the system. Task description was used to show the full working system in a simple manner. It provides the overall working and concept of the system.

The team found this technique challenging since it was difficult to decide upon the level of abstraction. The team faced it a challenge to decide how much detail was necessary to reduce the risk for misinterpretation. What was successful though, was that when working with the task descriptions, the team realized that additional tasks could be added.

c) Stakeholder map

With the use of a stakeholder map, the team was able to identify all of the stakeholders that have an impact on and/or influence the system. The stakeholder map is divided in such a way that those to the right of the map have more impact and those on the left have less impact. Similarly, the stakeholders on the top have high influence and those at the bottom have less influence. After a few iterations of the stakeholder map, the team has

derived a version that best resembles the impact and the influence of each stakeholder. The team also prioritized the requirements according to the impact and the influence of the stakeholders.

d) Data model & data dictionary

Data are described at different levels of abstraction. In order to describe what data to include in the system, the team chose to use both data models and data dictionaries for the specification. For the core functional requirements, the data tables and the class diagram provide granular details. The data tables describe the type of required data and give additional information and examples of values in order to minimize uncertainty regarding the purpose of the data. The class diagram illustrates cardinality relationships and how one type of data is related to another one.

The team used the data dictionary technique to specify the intended source and use of data. The source describes in what context the data is created, and the use of data both describes the purpose of the data and how it is related to the tasks to support. The team found the data dictionary as a suitable technique since it provides a good overview of why certain data is necessary for the system without going into too technical level of detail, as for the data models. Descriptions of data are included to minimize the risk of misunderstandings.

e) Mockup

In Release 2, based on the feedback given from the supervisor, the team designed a mock-up. Mock-up is a representation of the system that will be developed. It can be used to show the design choices for the system. It is helpful to create an intuition on what the final product would look like in the early stage instead of waiting until the development is over. Moreover, mock-ups are important to identify problems that are not visible in the requirements documentation. It also makes revision on the design of the system easier compared to making revision at the final stage of development. Hence, the team created a mock-up that shows what the core functional requirements of the SAM system design will look like.

C. Prioritization

To prioritize both functional and non-functional requirements, the team members discussed about which requirements have higher priority, and which requirements require less focus. The team ended up using two methods for that: MoSCoW and Scorecard.

a) MoSCoW

MoSCoW, a numerical assignment technique [2] is used because it is easy to apply when managing requirements. It gives an overview to the team of what the team expects as a customer.. The team grouped the requirements by priority that the system must have in order to satisfy its core value. To achieve this, the team created a list of requirements that have to be delivered. For each requirement listed, the team decided its importance by using MoSCOW

methodology. Categories used for prioritization are: Must, Should, Could, Won't. For every requirement listed, each team member put their own estimation of importance. The final importance for each requirement was then set to the most dominant estimation since an actual average is not possible for categorical classifications.

In Release 2, additional sections and requirements were added, leading to a need for reprioritization. The team realized for example, that having a focus on security as a non-functional requirement is of high importance for the system. Therefore this requirement was grouped in the prioritization list as a standalone requirement, in comparison with other non-functional requirements which were grouped as "G2 - G4. Non-functional requirements."

b) Priority Scorecard

As a second prioritization technique, Priority Scorecard method was used [3]. This technique is more detailed technique and the team found it as a good technique to prioritize requirements from different perspectives. By introducing various factors applied on the system, the team had a deeper understanding of what requirements will need more financial support and what areas are affected by each requirement. The results of prioritization using this methodology indicates whether requirements prioritized using MoSCoW actually meet the stakeholders' needs. When using this methodology, the stakeholders decided the most important criteria that impact the cost-value relationship. The goal was to have the most beneficial requirements on top, followed by the less beneficial ones.

The team felt that MoSCOW technique did not give the trust about prioritization correctness, but rather it reduces the scope of importance for each requirement, while using Priority Scorecard technique, the team identified the most valuable requirements. Constrained with time and costs, Priority Scorecard selected an optimal set of requirements to meet business goals and estimates expectations of the team. The team liked both methods because each method has a different viewpoint on the requirement priority. One reduces the scope of importance, another one analyses that importance from a low level.

III. VALIDATION AND RISKS

A. Validation

The team did validation of the specification document using Content Check and Reviews. The team analyzed the content of the specification. The team checked if the specification contains the following sections:

- Introduction: explanation about customers and stakeholders, background and reasoning to get the system and project type.
- System goals: business goals and evidence of tracing from goals to the requirements.

- Data requirements: description of data in database, input/output data formats
- Functional requirements: system boundaries, domain-level requirements (tasks), product-level events, design-level requirements (prototype)
- Handling of special cases: cases that stress the system to its limits (core functionalities described in separate sections)
- Quality requirements: detailed focus on the most important quality requirements for the system

Reviews of the specification are used as another approach to validate the document. For the purpose of this project, the reviews are done by another project group. They provide overall feedback on the document. The feedback helped the team to improve the document. The team's teaching assistant provided feedback and opinions as well.

The team discussed the issues and reformulated or deleted the problematic statements. The team has continuously reviewed the document and improved it.

B. Risks

There are various risks involved in building a requirement document. Described here are a few of what the team faced. Clearly defining every requirement is very important to avoid contextual risks. If requirements are not specified clearly, the project may not meet the desired outcome. So the team found it important to go back and reflect on all of the requirements for every release and fine tune them.

Using two different techniques of prioritization may assign contrasting priorities to a requirement, which put the team in a dilemma in deciding the correct priority for a requirement. While using various techniques to prioritize requirements, there is a risk of misjudging the importance of a requirement. The team felt the need to either use one technique for prioritization or choose techniques that complement each other for best results.

Another risk was the assignment of priorities to inter-dependent requirements. When one high priority requirement cannot satisfy the outcome without being dependent on a low priority requirement, both requirements may have to be given the same level of priority.

Perspectives of all stakeholders should be considered while writing down requirements and prioritizing them. The group ensured that this was done to avoid the risk of disagreement between the stakeholders which could hamper the project.

IV. LEARNING OUTCOMES

In this section, lessons learned from the project are presented. The team experienced how difficult it is to write requirements specification. Lauesen

describes that a good requirements specification, for example, is complete, unambiguous and verifiable [1]. These three examples are the ones that the team has discussed the most. The team found that it is difficult to know whether the level of detail is good enough to include all necessary requirements. It was hard to balance the level of detail. In other words, it was not easy to know when the specification is 'complete'.

The team experienced how useful different specification techniques are. A context diagram gives a good overview of the system. It is helpful in the way that it describes the system on a high level. For the more detailed levels, task description serves as a good technique for specifying what to include. The team regarded a task as something that should be done without interruptions and that the end of it was regarded as an achievement. This made it easier to keep the tasks on a similar level of detail. Many of the requirements that the team had in mind were difficult to verify. This was handled by creating event and function lists, which were easier to use as checklists for validation.

During the prioritization of requirements, the team faced issues when deciding on what technique to use. Since there are various applicable techniques, the team spent some time to understand which one will best suit the goals. The team learned that using only one technique is not enough to prove if the prioritization done is really what the team wants. Including one more prioritization technique on the same requirements gave a different point of view. The team also realized how important it is to include all of the stakeholders' perspectives during prioritization since different stakeholders have different priorities. imagine how different stakeholders' perspectives could influence the prioritization. The team realized the importance of good specification documentation when going through the identified requirements. When working with the requirements specification, the team realized that the requirements were initially ambiguous. This provided the insight that is important to work on the specification.

One challenge faced during the specification of data was the level of detail required to specify what is asked for without limiting the supplier's ability to come up with its own suggestions and solutions. The team approached the challenge by providing the supplier with example solutions in a data dictionary as examples that guide the supplier in the desired direction.

One of the key learnings was that it is very difficult to have a clear picture of all necessary requirements at the earlier stages of product specification. As the team worked on the requirements document, many additional requirements and features were added and reprioritization was done. It is clear that the team got a better understanding what the team wants the final product to be like.

To assess the quality of the requirements, every member of the team has thoroughly read and understood each requirement to ensure that the requirements are not ambiguous. The team has also reviewed the requirements ID to make sure they are consistent and unique throughout the document. The peer review provided the insight that the team had another view of the mock-up than the reviewing group had. Hence, the mock-up was modified so that different people could have a common understanding of it.

A. *Group dynamics*

Working on this project, the team did not face any bigger issues. The team had structured meetings as well as clear and open communication. The team ensured that each member understand the scope and goal with this project and that everybody has a clear picture of what to do and how to do it. Introducing internal deadlines ensured on time delivery for each release. Meetings were set up on a weekly basis with respect to the available time for each team member. Members who could not be physically present, the team ensured participation by using online conferencing. The workload was divided so that each member could actively contribute in writing both specification and experience report. Some team members put more effort into the work than others but all members contributed to the final product.

V. CONCLUSION

Requirement engineering is a difficult but an important part of software engineering. It is useful to know the customer's needs, verify and validate the requirements gathered for the supplier to develop. In this requirements specification project, the team has learned different techniques and methods used in requirements specification. Requirements specification styles such as context diagram and task description are a good way to present the high-level overview and the detailed description of the system respectively. Interview and brainstorming techniques are used to elicit the requirements of the system. Interview is a good way of gathering requirements and gain a thorough understanding of the domain knowledge. Moreover, brainstorming is also a useful elicitation technique when the analysers have domain knowledge about the proposed system. Prioritization techniques used in the project helped to hone the specification quality during the iterations.

The team went back-and-forth between requirements to determine their importance and specification. Ultimately, prioritizing the requirements improved them tremendously compared to how they were at the beginning of the project. In conclusion, the team understood that the choice of the different techniques of requirements specification depends on the nature of the project.

BIBLIOGRAPHY

- [1] Lauesen, S., Guide to Requirements SL-07: Problem-oriented requirements v5, 2017.
- [2] P. Achimugu, A. Selamat, R. Ibrahim and M. Naz'ri Mahrin, "Information and Software Technology," in Elsevier, Skudai, Johor, Malaysia, 2014.
- [3] J. J. Phillips,, T. W. Bothell and G. . L. Snead, The project management scorecard, London/ New York: Routledge Taylor & Francis Group, 2011.
- [4] Thinkertoys by Michael Michalko 2006 edition
- [5] B. Hollis and N. Maiden, Extending Agile Processes with Creativity Techniques, 2013.