December 8, 2015 v2.02

Team C

PostCardBuddy

Project Experiences

Authors of this document:
Emma Albertz
Caroline Brandberg
Linnéa Claesson
Billy Johansson
Johan Ju
Jacob Mejvik
Carl Rynegardh

Contents

1 Introduction

This document aims to describe how the work has been conducted during the project. It also contains the group's reflections on the work process and the difficulties with different parts of the project.

2 Methods and Techniques

A description of the methods and techniques will be presented in this section, along with a short motivation for why the specific technique was chosen. In section 3 can evaluations of the used methods and techniques be found.

2.1 Elicitation

To find relevant elicitation techniques, *Software Requirements - Styles and Techniques* by Soren Lauesen has been used as a guidance[1].

The initial identification of relevant stakeholders emanated from a discussion about the product and who interacts with it. From this discussion the stakeholders that were the most important for this product could be identified. Since it was important to quickly get going with the project this method was considered appropriate. The group is well aware that this approach could cause important stakeholders to be left out. In order to reduce this risk, identification of other stakeholders have been a top priority through out the project. The product is somewhat limited in scope and hence the number of additional stakeholders that have been considered significant have been few.

The following elicitation techniques were used:

Brainstorming Used as a first step within the team to come up with basic ideas and functions of the product. This is a quick method to get some initial ideas and starting points. During the brainstorming session the functions specified by the key customer, from their initial order of the product, were also considered.

Questionnaire The questionnaire was sent out to people within the end user group. Questions from the brainstorming session were used to form the questions. People answering were asked to grade functions from zero to five, where zero stood for not interesting and five for very interesting. An age field was added to see if there was a difference in interest of various functions between ages. This is also an easy method to get some ideas of what the intended users of the product might want (or not want).

Interviews In order to improve the understanding of the kind of product envisioned by the key customer, an interview session was conducted early in the elicitation process.

Prototypes Three team members created one prototype each, independently of each other so as not to affect each other's ideas. It was decided to do this right away due to the time constraint upon this project. The prototypes are meant to be used for ideas to the graphical interface of the application. The use of prototypes is considered a suitable technique for this project since there are many easy to use and free programs available to create them. Additionally, it gives not only the stakeholders but also the authors of the requirements a good idea of what it should look like and be able to do. They were specifically used when eliciting requirements from prospective end users.

Document study There is already a similar existing application on the market and it was used to further elicit functionality not already thought of and also to perhaps eliminate functionality that intervenes with the user experience. This was done *after* the initial brainstorming session, to avoid making an identical application or interfere with the creativity of the team.

Data model

Data dictionary/Virtual windows

Summary In the elicitation process several different elicitation techniques were used. The group has continuously reflected on the choices and experiences from the various techniques. In some cases the reflections has highlighted the need for more information, resulting in that additional elicitation has been conducted. However, there are still some areas that require further elicitation as the development of the product commences. As an example, a specific area that deliberately has been kept somewhat limited is the exact technical specifications. The elicitation process has also been heavily geared towards going beyond the initial stakeholders and challenge the domain borders. In doing so, several potential stakeholders, such as traveling companies and advertisement agencies, have been considered. The stakeholders that have been left out of the final project has been deemed to have little relevance to current structure of the project. In some cases, e.g traveling agencies, potential stakeholders could have been attributed with a higher relevance given a different business model for the product. Furthermore, in order to broaden and deepen the understanding of the domain stakeholders have been contacted. Although, this has given valuable feedback in most cases it also underlined some difficulties. For example, when contacting the postal services it proved difficult to get in touch with relevant staff to answer our rather technical questions.

2.2 Specification

Context diagram A context diagram was used since it is easy to make and is helpful when time comes for validation and verification. The diagram gives a good over-view of the system, both for the use of the client but also for the developers.

Data model

Data dictionary/Virtual windows

Task descriptions

Prototypes

Quality grid A quality grid was created by brainstorming what quality aspects was important for PostCardBuddy. This was done by starting out from Lauesens quality grid on page 227 [1]. Lauesen's quality grid was stripped down to the parts, the group thought, was worth mentioning. Doing the strip down, a few questions guided us:

- What quality aspects are more important for PostCardBuddy in comparison to other applications/software.
- Is their something we would like to describe more to the developers.
- How do we rank each of the quality factors.

- How do we set the quality factors and their ranking into context. Meaning, how do explain why this is important to the developers.

Many "As usual" quality factors where left out because the group thought that there was no use to explain them.

QUPER A Quper diagram was created to specify a Quality requirement, or perhaps target is a better word. Quper is documented in ref [1]. Document studies of the competitor "Riktiga Vykort" showed that requesting images from the competitor's, "Riktiga Vykort", image library into creating a postcard was a painful process. If PostCardBuddy could be better in this aspect it would be highly valuable. Quper as in ref [1] contains advanced features requiring a lot of technical knowledge in the domain. Not having this knowledge, a lot parts where left out so just the basics were left. To get a feel for what would be a reasonable target, in case of user experience, testing was done with "Riktiga Vykort". As the group did not have enough knowledge for the technical aspect this was, more or less, left out.

Summary When specifying the requirements numerous different techniques have been used. Furthermore, the requirements have been specified based on type and abstraction level used to describe the requirement. A very important aspect of this specification has been to construct a precise understanding of the context of the product. There has been many discussions and revisions of the systems boundaries before arriving at the final context diagram. In general the overall experience from specification is that it is impossible to specify everything with a great level of detail. The solution to this problem has been to combine different degrees of completeness and different abstraction levels. It has also been helpful to use several different specification techniques, tailored to the situation at hand, combined with examples. In addition, the group has paid special attention to tracing the requirements to the goals while simultaneously maintaining different hierarchies and requirement relations. When it comes to establishing a desired level for the target quality the QUPER analysis has been a great help, since it provides a good framework and scale to benchmark against competitors. Perhaps the most challenging aspect of the process has been to prioritize and decide on which aspects need special attention. Since one of the over arching goals of the product is simplicity special attention has been given to achieving this. In practice this means that it has been important to specify exactly which data needs to be added and special attention has been given to this aspect.

2.3 Validation

Prototypes The prototype gives the customer a unique opportunity to validate how the product matches their expectations. The prototypes will be continuously adapted to the customer's needs and wants and new features will be added (or others removed) so that it becomes a good reflection on where the project is going.

Validation checklists and validation report (as developers) A validation checklist was constructed and given to the customer. They used the checklist to write a validation, to evaluate the System Requirement. This report was then used to go over the requirements and improve on them.

Validation checklists and validation report (as customers) Validation checklists provided by the developer group were used to validate the product initially ordered as customers. A validation report was then written based on the checklists.

Informal review An informal review within the group was conducted to make sure the System Requirements was complete, no ambiguity etc. Everyone read the report prior to the review. At the time of the review, needs for changes were discussed and a protocol was created listing everything that needed to be changed.

2.4 Prioritization

Stakeholders The prioritization for each stakeholder was stated through discussions within the group.

3 Reflections

This section aims to evaluate the methods and techniques used, as described in section 2

3.1 Elicitation

Brainstorming The reason for selecting the method to collect our stakeholders was because it is a fast method which meant that is was possible to start working, such as contact some of the stakeholders.

Questionnaire Figure 1 presents the result of the questionnaire, which 38 people answered. To get answers from that amount of people was no problem and it gave a first idea of what the users were interested in. The result of this is that the functionality "Share postcard on social media" was not important and "Suggestion for GPS-based images" was appreciated. The result also showed that the desired functionality did not change that much depending on the age. Using a questionnaire was interesting since it gave a good idea of the functions people are interested in. However, as the questionnaire was created it was desirable that it was quick to answer. Therefore, only ten questions were used to maximize the number of respondents and the quality of the replies. Afterwards it was realized that some interesting functionalities were missing. Knowing the interest of these functionalities as well could have been of interest and might be investigated further prior future releases.

Interviews Although the interview provided valuable insights the main impression was ambiguity, both in terms of the role the key customer would have and exactly what the product should do. Given more time it would have been beneficial to invest in achieving a better consensus within the customer group before conducting the interview. Additional interviews will be conducted during the requirements specification process.

Furthermore, two separate companies in the postal service business have been contacted with the intention to conduct interviews. However, it has proved difficult to get past the first line support and get a hold of an appropriate contact. A possible explanation for this is that the postcard business is only a minor part of the postal services market and there is probably nobody with a clear responsibility for this area.

Prototypes A program was used for constructing the prototypes that worked very well. It also proved to be of use for brainstorming new ideas and features, since the program itself offered a lot different options on how to do things.

From discussions with the costumer team, new ideas for features emerged when the costumer tried the prototypes. The prototype helped the costumer to verify that the application conformed to their requirements and also gave them an opportunity to see if something was missing or wrong.

Document study The existing application, and competitor, "Riktiga Vykort" was easy to use and rather slim. It did not contain a lot of functions but it felt as there were enough. Most of the basic functions we have specified are already implemented in "Riktiga Vykort". However, there are definitely some functionalities that could be of use that are not implemented. Also, "Riktiga Vykort"'s image gallery was not very big, and GPS based images depending on your localization only works in Sweden and Denmark. The group also noticed that it took a lot of time to request access to an image in "Riktiga Vykort"'s image gallery in to creating a postcard. The group started thinking that if PostCardBuddy would be able to do this better, it would be valuable giving competitive advantage. Therefore this was used in Quper.

Data model Creating the data model for the data requirements was in itself an exercise in elicitation. While gradually developing the ER-diagram new entities and relationships that were not easy to spot in the beginning started to emerge. This was largely due to dependencies between between different types of required data.

Data dictionary/Virtual windows Creating a data dictionary or virtual windows is a lot similar to doing a data model, from an elicitation point of view. Doing all three might be unnecessary (if elicitation is the purpose), but doing two of them would definitely be of value. The virtual windows technique especially is good for realizing what types of data might be missing from a certain feature.

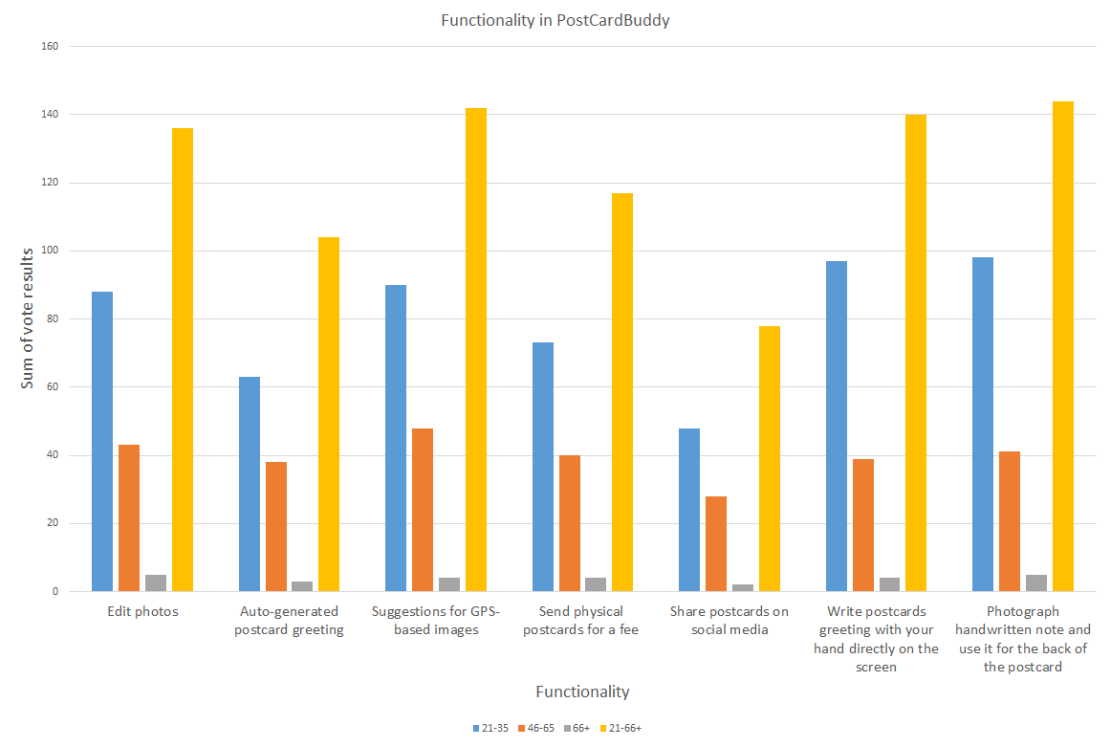

Figure 1: Result of the questionnaire on the desired functionality in PostCardBuddy

3.2 Specification

Context diagram The first context diagram created was presented in PMv2. The first diagram was very limited and contained too little information to understand the system. The updated diagram was then presented in release 1 of the report System Requirements. The biggest problem creating a context diagram was that it should be big enough to present important details, but small enough to be able to get an overview of the system. Therefore it is very important to think through which components it should contain, and which should be left out. This difference is often personal, which was noticed during the creation of release 1, which led to some discussion. Most of the discussion were spent talking about if the back-end should be presented and how the functionality that is used within the mobile should be presented.

The changes of the context diagram between release 1 and release 2 were mostly added descriptions of the parts. These descriptions were easily added without any problem. Also the contacts were added, which was a part that was missing in the previous version. It is very easy to miss parts of the diagram. To find out that every part is within, it is very good to try to describe each chain and see if it is easy from that description to follow in the context diagram.

Data model The data model is a good tool to easily visualize dependencies of different systems and stakeholders. If done thoroughly it could be used as a good starting point for developers, and in particular database developers. But the more complicated the data model becomes, the harder it gets for non-technical personnel to understand it. And in the same way it loses some of its value for developers if it is not thorough enough. This is why it was combined with a data dictionary and virtual windows, to adequately satisfy technical as well as non-technical personnel.

Data dictionary/Virtual windows The data dictionary is probably the simplest tool for specifications. It is easy to write but can become tedious and it is hard to see relationships between data. As a complement to a data model it is very good for properly communicating a specification. Virtual windows are very helpful for non-technical personnel and is a very efficient way of presenting an overview of what types of data are needed for a specific feature.

Quality grid Doing a quality grid was a very good idea. While creating the quality grid many ideas came to life. Especially ideas for different requirements. The quality grid should probably have been created earlier in the project phase. Maybe even for release one. Creating the quality grid together with stakeholders, perhaps using questionnaires or interviews, could have been beneficial. Creating the quality grid forces the creator to think of different aspects for the application. The creator gets an overview over what is important, and with that ideas can come to life. Therefore, a quality grid could be a way of elicitating, mostly quality requirements, but maybe even functional requirements. The line between quality requirements and functional requirements can be very thin. A negative aspect with the Quality grid could be that it does not specify any requirements. It mostly highlights what is important for the application and describes to the developer why this is important. If it really is a negative aspect could be discussed, as it is a very useful way of communicating quality factors to the developers. As the Quality grid was created rather late in the project, we did not base any quality requirements on it. If the quality grid would have been created earlier, quality requirements could also been based on the grid. This would have complimented the quality grid.

QUPER It is interesting to compare the quality grid with Quper. The quality grid gives a better overview over quality factors as a whole while Quper serves as a way to specify a target or requirement. Normally, Quper demands a lot of knowledge which the group did not have. Therefore, it was not very useful in our case. It did highlight the problem with the competitor "Riktiga Vykort"'s application and specified how long time it should take for PostCardBuddy, but this could have been done in different ways. If we would have more knowledge in the domain, Quper would probably have been more useful.

3.3 Validation

Prototypes

Validation checklists and validation report (as developers) It was very useful to receive input from the key customer. They noticed things previously missed and had very clear and concrete comments that were easily fixed.

Validation check lists and validation report (as customers) The use of checklists was very helpful as it is easy to check the quality of existing requirements. However, it is also limiting in the sense that it makes it difficult to see if there are any missing requirements. It is easy to "get stuck inside the box" that the checklist actually is. Checklists are very quantitative and need explaining comments in addition to get qualitative information from them.

Informal review This was very important in the process. Since everyone has been working on separate parts it was easy for contradictions to appear. Additionally, it gave a good overview of the state of the report such as gaps that needed to be filled etc.

3.4 Prioritization

Stakeholders This was both a very hard but also a very easy task. The hardest part was that when stating the stakeholders, the focus was on the stakeholders that would in some way interact with the system. Therefore all of the stakeholders were very important. Furthermore, in the projects development, it became clear on which stakeholders that meant more and who does less. That meant that suddenly the prioritization was more or less done.

Summary Although prioritization has been done in previous courses the formalized approach in this course was a new experience that turned out to be rather challenging. Several different techniques were used but despite that they provided a framework, there was still a need to take a lot of decisions within these techniques. For example when combining prioritizations from different stakeholders resulted in a need to go beyond the ordinal scale in the sense that a low prioritized stakeholder at times had very strong opinions of certain aspects. In spite of these challenges a release plan based on the prioritization was created. The release plan has been developed iteratively taking into account both scheduling and precedence constraints. The process of finding the right priorities has also been very helpful in directing attention to areas that needed special attention in order to improve the quality of the specification.

4 Personal Statements

4.1 Emma Albertz

4.2 Caroline Brandberg

4.3 Linnéa Claesson

4.4 Billy Johansson

4.5 Johan Ju

The main responsibility I had was the quality requirements and prototypes. I also contributed to the formulation of some functional requirements. Under the elicitation phase I helped with the questionnaire and had discussions with the key customer about the prototypes. I have also actively participated in the inspection and amendment of the specification.

4.6 Jacob Mejvik

4.7 Carl Rynegardh

Contributed in/with: Questionnaire, functional requirements, quality grid, quper, validating another groups requirements, informal reviewing.

References

[1] Soren Lauesen, *Software Requirements - Styles and Techniques*, Pearson Education Limited, 2002

[2] Björn Regnell et al, *Supporting Roadmapping of Quality Requirements*, published in IEEE Software, 2008