CHALMERS
UNIVERSITY OF TECHNOLOGY

COCO GUIDE

Development of a Cooperation Framework for Cross-Functional Teams

Master's Thesis in Interaction Design and Technologies

Julia Friberg and Therese Johansson

Department of Applied IT
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2017

Master's Thesis 2017:7

Development of a Cooperation Framework for Cross-Functional Teams

JULIA FRIBERG
THERESE JOHANSSON

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Applied IT
Chalmers University of Technology
Gothenburg, Sweden 2017

Development of a Cooperation Framework for Cross-Functional Teams
JULIA FRIBERG, THERESE JOHANSSON

Supervisors:
Jon Kindell, Varvet
Pontus Wallgren, Design & Human Factors, Product and Production Development
Examiner: Staffan Björk, Department of Computer Science and Engineering

Master's Thesis 2017:7
Department of Applied IT
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: The logo for the web page where the result of the project is presented.

Gothenburg, Sweden 2017

Abstract

This master's thesis explores how designers and developers collaborate and communicate in cross-functional teams when developing software within the IT industry. The work includes a social study of the current situation and mapping of the current problem areas, which later served as the base for the development of guidelines on how to best work within cross-functional teams.

Later, the result is presented as a collaborative web page containing guidelines that provide an opportunity for others within the field to share their experiences of working in cross-functional teams. The guidelines are then followed by suggestions for further work within this area.

Keywords: cross-functional teams, collaboration, communication, design, development, design process, remote work, work process, guidelines, team setup.

Acknowledgements

We would like to thank our supervisor at Chalmers, Pontus Wallgren for his support and enthusiasm during this project. Pontus' expertise in how to include the user during interviews and workshops has been of great value to us. Additionally we would like to thank Varvet for providing us with the resources needed to complete this project. A special thanks to Jon Kindell who has been our supervisor at Varvet.

Thank you to the Got UX community for inviting us to present our work and for participating in the workshop. Finally, thanks to all designers and developers that shared your knowledge and time with us.

An extra shout out to all companies who lent your employees to us: Creuna, Forza Football, Intunio, Uptive, Varvet and Wint.

Julia Friberg and Therese Johansson, Gothenburg, May 2017

Table of contents

1

Introduction

A diversity of thoughts within a project often leads to innovation (Maaike Kleinsmann 2010). To develop great advanced software it is therefore common to construct cross-functional teams of individuals with different expertise. Cross-functional teams may include both designers and developers in different constellations, with the most common of them being one or a few designers and a majority of developers.

Communicating with other disciplines is not an easy task (Maaike Kleinsmann 2010). Because of different backgrounds, knowledge and prioritizing, communication problems such as misunderstandings may occur within these teams. Despite this, the team has to be able to work together in order to produce high quality software. This sets high demands on the team and their communication skills (Maaike Kleinsmann 2010).

This report is purposed to highlight common problems that can occur within cross-functional teams. As a result of this thesis, suggested guidelines for how to solve these problems will later on be presented as well as plans for further work within this field.

1.1 Background

This section describes theories of working in cross-functional teams as well as the interpersonal communication between team members.

1.1.1 Definition of a Cross-Functional Team

A cross-functional team consists of individuals with a diversity of backgrounds and knowledge (Inc.com 2017).

The Business Dictionary (2017) defines a cross-functional team as a *"Group of people with different functional specialties or multidisciplinary skills, responsible for carrying out all phases of a program or project from start to finish."*

1.1.2 Working in Cross-Functional Teams

Many design tasks are collaborative acts that set a high demand on effective interaction within teams. This interaction affects both the relationships as well as the strength between members in an often cross-functional team. This will later affect the teams ability to work effectively (Otter and Emmitt 2008). A lack of shared understanding within the team will ultimately lead to a lack of quality in the final product, since not all problems have been solved in the end of development (Maaike Kleinsmann 2010). Furthermore, it is found that the highest quality products are developed by a team with a high level of shared understanding (Maaike Kleinsmann 2010).

Cockburn (2002) states that "Software development is a cooperative game of invention and communication". Here, a team constructed of individuals with talent work towards a common goal. New knowledge is constantly being generated within the group by collecting, sharing and transforming important information (Otter and Emmitt 2008). This can be done by interpersonal communication, written communication or by sketches or mock-ups (Maaike Kleinsmann 2010).

Studies have shown that a project's management and organization also influence the process of reaching a shared understanding within the team (Maaike Kleinsmann 2010). The sharing of knowledge is an essential part of the organizational performance. Despite this, there is no need for all team members to become experts in each others fields by sharing all knowledge. Instead of being transferred between members, knowledge should be integrated during the entire project (Maaike Kleinsmann 2010).

A designer normally works both individually as well as being a part of a team. It is common for designers to work individually on a design task before returning to the process of the working network (Otter and Emmitt 2008). During the implementation phase, it is common to involve other actors than during the design phase (Maaike Kleinsmann 2010). This can be a problem since there is a mutual need for information between the designer and the developers (Otter and Emmitt 2008). Furthermore, this hampers on the vision of a shared understanding between all parts of the team and eventually the quality of the final product (Maaike Kleinsmann 2010). Thus both designers and developers need to work together. All team members should be included in the project and share both success and failure together (Monteiro 2012).

1.1.3 Communication in Cross-Functional Teams

Communication within design teams can be defined as *"the compilation of all processes for sending and receiving messages between team members individually and collectively, using all the available means of communication."* (Otter and Emmitt 2008) or more specifically *"the sharing of meaning to reach a mutual understanding and to gain a response: this involves some form of interaction between a sender and receiver of the message"* (Otter and Emmitt 2008).

Synchronous, face-to-face, communication is essential when developing and implementing design tasks. Research reveals that design teams prefer dialogues over asynchronous communication, despite the growing use of web based communication. By sharing and discussing feedback within the group one will contribute towards the goal of a shared understanding within the group (Otter and Emmitt 2008).

Communication is never perfect, nor complete (Cockburn 2002). The more different two individuals are, the more details are requested during communication. But no matter how much basics and details are included, some will misunderstand each other (Cockburn 2002).

Design requirements can never be fully specified, one must therefore assume that the receiving part has a certain level of experience. If it is possible to assume that the other part has the required experience, then we can describe less. But if the two individuals engaged in the conversation do not have enough experience, then the demands are higher set on their communication skills (Cockburn 2002). In other words; it is easier to receive and give feedback once you know the one you are talking to (Monteiro 2012).

1.2 Research Question

As mentioned in the background, working in cross-functional teams can be challenging, but very rewarding when working well. Communicating can be hard when it is done with those that have the same experience, but it is even harder when talking to someone having another area of expertise. This leads to the research question for this project.

What is needed in terms of recommendations and methods in order to facilitate communication and teamwork within cross-functional teams in the IT sector?

With this question in mind, social research was conducted to map the current problems in cross-functional teams.

2

Theory

By finding out what had already been studied within this field, the work could be focused on collecting new information. The theory describes design, social research, communication as well as two agile development processes for cross-functional teams.

2.1 What is Design?

Design can be described as a wicked problem. A wicked problem is defined as a problem with no clear formulation, no definitive solution nor an absolute stopping rule. The result can either be bad or good, but never perfect. All wicked problems are also a part of a higher, bigger problem and they are all unique (Buchanan 1992).

Since design is a wicked problem is very hard to define. The British Design Council (Moggridge 2007) published a book called Definitions of Design in 1995 with the personal definitions of design from fifty people. The result was fifty definitions, which could not really be seen as definitions. They were all too vague to be satisfying. They stated that this is something that has been accepted when it comes to talking about design, and the vagueness is something designers have to live and work with. The British Design Council themselves have answered the question What is design? with the following statement: "Design is everywhere—and that's why looking for a definition may not help you grasp what it is." (Moggridge 2007).

Moggridge (2007) lists five core skills of design as a way of explaining what design is. These five skills can be done in the order mentioned below, but they are often done as an iterative process. The process does not have to be a circular iterative process, but rather a process where the designer jumps back and forth rapidly and sometimes unexpectedly (Moggridge 2007).

1. Synthesize a solution with the understanding of what may affect the result
2. Formulate the problem
3. Create possible alternatives for a solution
4. Select the best approach from the alternatives
5. Prototype the solution

Moggridge (2007) continues by explaining that the designer, when the design is good, is often the one to get the appreciation from the users, and stated that this is because the design is what the users see, and therefore also a big part of what makes the product as good as it is. For example clothes are more appreciated when they seem to be designed for the person wearing them. Of course, the designer is not solely responsible for the success of a product. In fact, the designer has to rely on everything else to be done well for the design to have a chance at success, since the design can please the user, but only if all other requirements are already fulfilled (Moggridge 2007).

2.2 Social Research

Research starts with a reason to ask questions and a desire to find the answers to how and why for those questions. This happens in everyday life for all people, for example when researching what food to eat, which clothes to buy or how to get from one place to another. This is called everyday research. When it comes to more scientific research, some are more successful than others. What is then good research? A common view of good research is that it should be objective, unbiased, scientific and based on facts. This view of good research does not take the human way of seeing things into consideration. For example feelings, values and interpretations are not included (Wadsworth 2011).

Yolanda Wadsworth (Wadsworth 2011) argues that facts are, in a sense, the perceptions that people share more or less of the world. One person's subjective opinion can become an objective fact if all others agree. For example if a person thinks it is hot outside, and others agree, then it is objectively hot outside. This fact can be proven to someone who thinks it is cold by showing a thermometer as evidence. A

thermometer however is only a proof because people have agreed that it is a way of measuring temperature. For someone who has never seen a thermometer before, it does not provide a strong evidence of whether it is hot or cold. She also mentions three grounds for agreement, which are faith and hope, a couple of examples and personal observations and experience. These are the grounds people have for agreeing on certain things, and not on others. These are important to consider when conducting social research, to help understand why people act and think in a certain way.

Wadsworth continues by describing the rules of the research game, which are important aspects to consider when researching within a social field. Firstly, the researcher should establish what is already known. It is important to know what has been done and where to start. Secondly, a goal for what to achieve should be set, where should the research go and how should it be used? When researching, it is important to question what is being observed and know which questions to seek answers to.

There are four conceptual groups when it comes to social research, which are the researchers, those the research is for, the researched and those that need to be convinced to make something happen, i.e. funding or resources. The researchers should know which of these groups, apart from the first, they belong to, in order for the research to be successful (Wadsworth 2011).

When it comes to social research, the fundamental task is to research how and why people see the world as they do. Therefore it is essential to talk to people and get their views. People see things differently, or see things that others do not. Why certain people see things in a certain way, as well as who sees it what way, are important aspects within this field. The researcher should not jump to conclusions quickly, but rather be skeptical for a long time. As long as there is skepticism, there are always more questions to ask. In social research, it is crucial to involve members from the research group (those the research is for) at every stage in the research process. Research is to generate ideas and then provide evidence for them. In social research it is also important to link these to people and their perceptions (Wadsworth 2011).

2.3 Communication

As mentioned in section 2.1, design is a wicked problem. Discussions between designers and other scientist about design tend to leave little to no room for reflections about the nature of design problems. This leads to confusion and a breakdown of communication between fields. The communication problems between fields are affected by participants finding it hard to understand each other (Buchanan 1992).

We should however not aim for complete communication but instead learn to deal with the incompleteness of it. There is no need for perfect communication, it is rather ineffective to deliver more information than needed. The aim is to get close enough as often as possible (Cockburn 2002).

2.3.1 Interpersonal Communication

Interpersonal communication is mainly divided into two sections, the first being communication between two to three individuals, so called dialogues. The second part of interpersonal communication concerns communication in groups containing three or more individuals (Otter and Emmitt 2008).

It is shown that synchronous, face-to-face, communication trumps asynchronous communication. Dialogues are normally used more often depending on distance between the parties and how available tools of asynchronous communication are as well as skills within the field. Dialogues are often performed face-to-face, complemented with sketches to visualize the spoken language (Otter and Emmitt 2008). Unspoken communication, as body language, is an important part of a dialogue and can sometimes be more important than the spoken word itself (Ahrnell and Wildhuss 1999).

An additional advantage of dialogues is their informal nature. Dialogues can take place outside of meetings such as in a corridor or by the coffee machine. When comparing dialogues and group meetings, dialogues are much easier to organize (Otter and Emmitt 2008). These advantages leads to dialogues having a high potential of being the most understandable media of communication (Otter and Emmitt 2008).

Different phases in a design project demands different levels of communication. Group meetings occur with a higher frequency during the early planning stages of a project. Meeting face-to-face is important when most of the design is still unspoken, when the majority of the design is in the designers mind (Otter and Emmitt 2008).

When reviewing design the interpersonal communication is preferred to e-mails and drawings because it is easier for the receiver to ask questions about the message, which makes it possible to explore issues with the design in depth. Research by Otter and Emmitt (2008) results in a clear link between meetings and a team's performance.

Different roles in a development team have different vocabularies (Monterio 2014). In the early stages of a project the insecurity between the participants is high, the team mates may feel insecure due to not knowing the other members or the context of the project. During this stage they tend to use interpersonal, face-to-face, communication more often. When communicating verbally the insecurity will eventually decrease. This will lead to a decreased use of verbal communication in time. During the early stages non-verbal communication is more seldom used. When the non-verbal communication increases in teams, the insecurity decreases with it (Jensen 2015).

A high level of insecurity will lead to an increased search for information within individuals. This insecurity will also lead to less intimate information being shared within the group. When sharing information in this stage the receiver will deliver information of the same sort back to the person who delivered information. This is less common when communication with a lower level of insecurity. Similarities between participants can reduce insecurities and differences can increase them. When participants feel secure they also tend to like the other participants more than when feeling insecure (Jensen 2015).

Interpersonal communication is built around turn taking. The participants in a conversation will take turns when talking, they will both take the change of speaking as well as giving away the word to others. When having the turn to speak, others are expected to be quiet. It happens that many participants talk at the same time, even if for a short period of time (Jensen 2015).

If a conversation stops in favor of silence the one that did not speak before is expected to speak. If this does not occur the one that most recently spoke is expected to speak up (Jensen 2015).

However, communicating is not all about what is spoken. When participating in a conversation and listening to others it is common for the one not leading the conversation to deliver short interjections, not to take the turn but to confirm the one speaking. These interjections are called feedback. The feedback is short and work as a lubricant for the conversation to keep it moving forward. When both parts in a conversation are active it interprets on them both being involved in the conversation (Jensen 2015).

2.3.2 Interfaces

In this context, interface is something in between people or something to unify them. This is a broader take than the common understanding of an interface being something between a user and a computer (Jensen 2015).

An interface in this construction can be a part of the environment or a linked area in time that enables information sharing between individuals (Jensen 2015).

Examples of interfaces in communication are face-to-face communication, mail writing, sending e-mails or by calling to name a few. Since all these alternatives are available, one has to make a decision on which interface to use when communicating. They will affect the communication in different ways. Interfaces may differ in space, size and even time. We differ between synchronous and asynchronous communication where a synchronous interface provides both parts with information in real time. An asynchronous interface will store messages from earlier for others to read. The time difference can last from seconds to months or years (Jensen 2015).

Interfaces can offer parallel or serial communication. During parallel communications both parts can deliver messages at the same time instead of serial communication which only handles one message at a time (Jensen 2015). Additionally an interface can have high or low level of closeness, feedback, intensity and affect different senses. This will include the communication within the given interface (Jensen 2015).

2.4 Agile Software Development

The agile software development process is an iterative method for cross-functional teams to work together during a development process. This process focuses on achieving personal, organizational and technical success (Shore 2007). Extreme programming (Shore 2007) and Scrum (Schwaber and Sutherland 2016) are two different ways to work agile.

2.4.1 Extreme Programming

Extreme programming emphasizes face-to-face communication by working in iterations with frequent feedback (Shore 2007). Between meetings, this method lacks guidelines on how the development should be performed. It encourages self organization. Anyhow, Shore mentions that the team should work together in an open working space. He also suggests that all team members should act as programmers by combining design, test and coding into one activity (Shore 2007).

2.4.2 Scrum

Scrum is an agile framework for working in complex projects (Schwaber and Sutherland 2016). A product backlog with tasks regarding the product is divided into so called sprints, each with certain tasks which should be finished after the sprint is over, usually in 2-4 weeks. A daily stand up meeting with all team members is held to check the progress, as well as a retrospective meeting after a sprint is done. When a sprint is finished, the next one begins until the product backlog is empty.

3

Methodology

During the project, a lot of different methods were used, which are presented in this chapter. No overarching framework was used.

3.1 Data Gathering

The purpose during the data gathering phase is to collect information about the users' needs and goals to produce a design of a product that best suits the user and aids the user in the way the product is supposed to (Preece 2002a). Preece (2002a) explains that the data gathering often leads to a set of requirements which are helpful for the continued development of the product. The information to gather in this stage is often related to how the user performs the task, what the goals are, in which context it is done and why it is done in that way (Preece 2002a). There are a number of techniques for gathering information, and below interviews and observations are presented.

3.1.1 Interviews

There are several types of interviewing techniques, and they are usually divided into two categories, structured and unstructured (Preece 2002a).

Structured interviews consist of the same set of questions for each interviewee and do not take into consideration the answers for further follow up questions (Karlsson 2007). Furthermore, Karlsson (2007) explains that these types of interviews are often used to get an idea of what clients think of a product and can be done over time to see how the perception of the product differs. Preece (2002a) describes structured interviews as quick and easy to conduct and that they can reach a lot of people without using a lot of resources. There are of course disadvantages with using this technique as well. For example, the questions have to be well formulated and thought through to get useful responses (Preece 2002a). Karlsson (2007) describes the difficulties with structured interviews as arising because they are often done as

questionnaires or phone interviews, which lead to less engagement and less elaborate answers from the interviewees. This hinders the interviewer from asking follow up questions to an answer, which in turn leads to less emotions an depth being captured from the interview. These types of interviews are typically done to get an overview of how a product was received or the general attitude towards something (Karlsson 2007).

Unstructured, or semi-structured interviews are more traditional interviews in the sense that they are often done in person and the interviewer is in control of the interview and what questions are asked during it (Karlsson 2007). Karlsson (2007) mentions the advantage that the interviewees can express the answers in their own words, which leads to more elaborate answers. The answers from these types of interviews are often used to generate a hypothesis as well as surfacing new information (Karlsson 2007). They are good when exploring issues, but they are very time consuming, therefore these types of interviews generate mostly qualitative data (Preece 2002a). Since the interviewer can control the interview, the interviewer has to be experienced and aware of the problem with being biased when interviewing, to make sure this does not influence the interview (Karlsson 2007). These types of interviews can be used as a base for structured interviews (Karlsson 2007). These interviews can be done both individually and in group. When done in group, it is important to note that some characters might be dominant in the discussion and sway the other participants (Preece 2002a).

Another important factor for all types of interviews is how to select the participants. There are many ways to do this, and how to choose depends on the situation. Karlsson (2007) describes the process of choosing participants, and explains that there are two main decisions to make when choosing participants, regarding quantity and quality. Quantity is the decision about how many to include. The general rule is that more participants are always better. Karlsson (2007) continues by explaining that in reality, there are two ways to decide the number of participants. One way is to choose as many as it takes to make sure the result is statistically significant. The other way is to find the saturation point, when more interviews do not contribute anything new to the result. When it comes to quality, there are three ways to choose from what criteria the participants should be chosen (Karlsson 2007). They can be chosen to represent the population statistically, they can be chosen from certain criteria to represent the target group, or they can be chosen to be the most critical users of the product (Karlsson 2007). If the product satisfies the most critical users, it satisfies everyone else as well (Karlsson 2007).

3.1.2 Observations

Observations are useful to see how users act in their natural environment. They can be done with a person observing or with the help of technical equipment. For example, video is often used when observing, which also opens up the opportunity

to re-watch and see new things that might have been missed the first time (Karlsson 2007). During an observation one can find behaviours that an interview might miss because the interviewee might not be aware of it (Karlsson 2007). Observations are more objective and are not as biased by the interviewee or the interviewer as interviews are (Karlsson 2007). They are not dependant on whether the interviewee is resistant to answering questions or not, and can fill in details that the interviewee might have forgotten (Karlsson 2007). However, feelings, attitudes and preferences are hard to discover during observations (Karlsson 2007).

Another way to observe is to let users keep a diary for a certain amount of time, where the user documents what and how they do a certain thing. This is more reliable way to document data than interviews, but the information about why certain events occur or certain decisions are made is lost, and the process of writing everything down might affect the behaviour (Karlsson 2007). This kind of observation is called self observation.

3.1.3 KJ Analysis

When working with limited resources and a large amount of data from a pre-study, priorities and decisions have to be made about future work. Priorities are made to make sure that the group is working on the biggest problems (Spool 2004). To find out which parts of the gathered problem areas that are most important, a KJ analysis can be used.

The KJ analysis, also known as an affinity diagram, can be used when faced with a large amount of complex data. Thus it is a useful tool when analyzing verbal data, such as survey and interview results (Tague 2004).

This is an example workflow of how a KJ analysis can be constructed.

1. Write statements of facts onto post-it notes
    - The first step is to collect facts and quotes from the data gathering. These facts and quotes are then cut out or rewritten onto sticky notes (Preece 2002b).
2. Arrange facts
    - One by one the notes are placed in groups where the relationships between facts are illustrated by distance in physical space. Notes that are similar on one or many ways are placed close together (Preece 2002b).
3. Create groups
    - The groups are not predefined. Once the sticky notes are placed groups can be emerged from their relationships (Preece 2002b).

3.2 Ideation

The ideation phase is the phase in which ideas are generated, based on the problems and data gathered in the previous phase. There are several ideation methods to use, and in this section brainstorming, "how might we" and workshops are described separately below.

3.2.1 Brainstorming

Brainstorming is a set of methods in which individuals or a group generate ideas to find solutions to stated problems (Chauncey 2013).

The session starts with a clear problem or question being stated for the group to generate ideas around. The second step is the divergent phase where ideas are generated without any criticism from yourself or anyone else in the group. Here it is important not to limit the amounts of ideas but rather to encourage the members to think without censorship and to aim for sheer quantity. Wild and new ideas are encouraged, even if the idea might be too wild they may work as a trigger for relevant ideas that are potentially useful. Later on it is time to sort, discuss and critique the ideas generated. This step is known as the convergent phase and the goal is to narrow the ideas down to one or a few that are the most applicable to the problem (Chauncey 2013).

3.2.2 How Might We

IDEO.org presents the method "How Might We", which is a brainstorming method, at their website for methods called Design Kit (Design Kit 2017). This method is based on problem areas that already have been defined. These problem areas are then reformulated as questions starting with "how might we...?". This method suggests that there are opportunities to solve the problems instead of them being challenges. It also provides the opportunity to answer these questions in a number of ways, and is therefore a method for ideating. The best "How Might We"-questions are narrow enough to ideate around, but wide enough to still generate wild ideas.

3.2.3 Workshop

A workshop can be used to include several participants when collaborating towards a solution (Koloski 2012). Having a brainstorming session is a a good way of facilitating a workshop, since many participants love to think about how to solve problems (Koloski 2012). It is preferable to limit the problems to solve during a workshop

to one or two. The workshop crowd will then be broken down into smaller teams (Koloski 2012), where if possible, the most dominant individuals should be placed in the same team (Toxboe 2016). The participants should be provided with sketching materials (Koloski 2012) and post-it notes (Toxboe 2016).

3.3 Prototyping

A prototype is a way to interact with and reflect upon the design of an envisioned product. It can be everything from a sketch on a piece of paper to a fully developed software prototype. The shape of a prototype often depends on where in the process and what the issue to test is at the moment. Certain prototypes fit certain needs.

Prototyping in itself is an iterative process, where each iteration gives more insight to what works and what does not work, contributing to a better final result (Preece 2002a). A prototype can be discussed with stakeholders, used as communication within the team or for reflecting and trying ideas for yourself (Preece 2002a).

However, prototyping does not only come with advantages, allthough the advantages of prototyping are many. Preece (2002a) describes that prototyping is about compromising. Making a prototype should not take too much time, which means the prototype will be limited. If a prototype looks too much like a real product, users will believe it is the actual product and either skip commenting on things they want to comment on since they feel like it is too late, or comment on things that are irrelevant. For example that the system is slow, which might only be because of the prototype's quality. This is why it is important to know when to compromise on what and which prototype to use for which purpose (Preece 2002a). A few different methods of prototyping are described below.

3.3.1 Low Fidelity

A low fidelity prototype is a prototype typically done quickly and out of a material far from the final product's material (Preece 2002a). It is a way to test out ideas since they are quick to do and therefore also quick to modify and throw away. These prototypes are a cheap way of communicating within the team and can be used as a proof of concept (Preece 2002a). It lacks the look and feel of the final product and is often not useful for user tests (Preece 2002a). This makes low fidelity prototypes useful at the beginning of the iterative process of prototyping (Preece 2002a). There are a number of different methods when producing a low fidelity prototype and described below is sketching.

Sketching is a natural part of most low fidelity prototypes and is an easy way to share thoughts and ideas. In a low fidelity prototype, the sketches does not have to be perfect, as long as they convey the message (Preece 2002a).

3.3.2 High Fidelity

A high fidelity prototype differs from the low fidelity prototype on a number of aspects. A high fidelity prototype is created to get a sense of the look and feel, and is often very similar to the final product(Preece 2002a). Furthermore, it is often done at the end of the prototyping process. It is user-driven and can be used when testing since it includes all the functionality that the final product will have (Preece 2002a). It can be used when marketing, but it is very inefficient as a proof of concept, since it takes a long time to create. Because it takes time, it is more expensive and harder to modify (Preece 2002a). The developer of the prototype will also get more attached to it because of the amount of work put into it, which can contribute to a reluctance to changing the prototype (Preece 2002a). Developers that are developing the actual product can gain a lot by having a high fidelity prototype to develop after. It works as a complete specification (Preece 2002a).

3.4 Crowdsourcing

Grier (2013) explains crowdsourcing to be the process of involving people and collecting the result of their work, often through the Internet. The kind of work asked for can be different, everything from funding to a contribution. In return, something is often offered to the people included in the crowdsourcing. It can be money, a membership or simply gratitude. Even if no money is involved, the person contributing can still get something in return. Satisfaction for contributing and showing their skills and being a part of a group are often feelings that appear from being involved in crowdsourcing.

To crowdsource, there are certain elements that need to exist. A crowdsourcer, which is the person who controls the process, a group of people (the crowd), who contributes with their work, a market or a platform where the work is managed and a means of communication, which often is the Internet due to the possibility of involving a larger group (Grier 2013).

4

Process

This chapter will describe the process of the thesis, starting with a planning phase. After that a data gathering phase was done, followed by an ideation phase. This leads up to a final concept which was prototyped and developed. All this was done to learn and find guidelines for designers and developers to work together in cross-functional teams.

4.1 Planning

The planning of the project consisted of first constructing a rough time plan followed by the execution of the planning phase. These are described below.

4.1.1 Time Plan

This master's theisis project is 20 weeks long and divided into three parts: planning (week 1-4), project work (week 5-16) and report and presentation (week 17-20). These sections can be seen at the first rows in the Gantt chart in figure 4.1. Each part is then divided further for a more detailed plan.

The project work includes interviews, which means that they were booked in advance of the pre-study. A goal during the planning phase was to contact chosen companies, and have a start up meeting with these before the planning phase was over.

The project work has two main parts, design and development. In the design process, a field study is performed to verify the problem and gather information about the subject. After that, an ideation is planned, followed by prototyping and evaluation of the prototype. After that the development should start with both frontend and backend development. Lastly a final evaluation are made. These steps are not meant to be followed strictly, but more to be seen as suggested deadlines for when different stages should be done. The process itself is planned to be rather agile, going back and forth between the different parts of this phase.

The report was to be be written in parallel with the project work, with one week dedicated to only report writing after the practical project work is done. The presentation and opposition each had three weeks in the end of the project.

Gantt

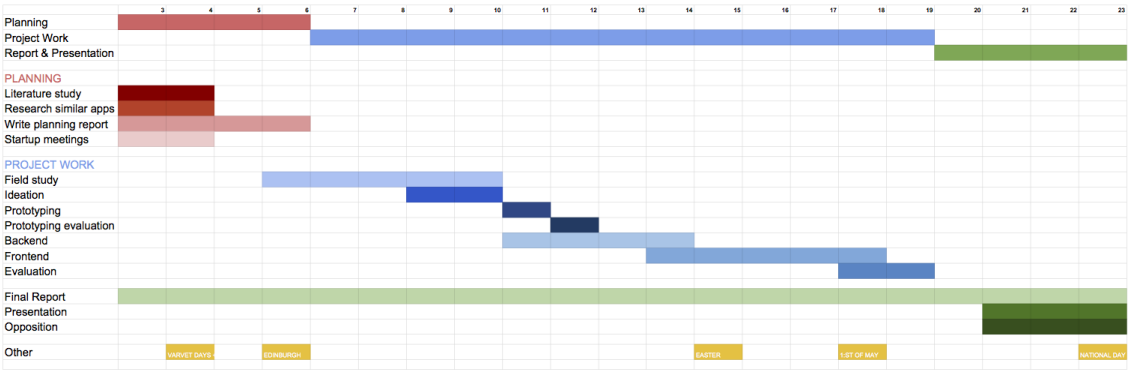

Figure 4.1: The original time plan for the entire project.

4.1.2 Planning Phase

The planning phase was nearly performed according to plan. The planning report was written within the first four weeks and the literature study started. The research of similar applications, which in this case were applications that facilitated the communication between designers, was done briefly, meaning that similar applications were found and described, but not evaluated. This was a decision based on lack of time during this phase. Instead time was spent to verify the problem and need for an application before testing similar ones. During this phase, a preparation for the interviews was performed, which is described further in the next sections.

4.2 Data Gathering

The goal during this phase was to gather as much information as possible about initially, how designers work together. This goal later changed to collecting information about how designers and developers work together, with a focus on communication. This change of focus was done after learning that these relationships were a bigger source of problems than the ones between designers. This is change of focus is explained further in section 4.2.2.

To learn as much as possible, interviews were prepared and held with developers and designers at a few selected companies. After the interviews the findings were structured and problem areas were defined. The different phases of the data gathering and decisions made during them are described below.

4.2.1 Preparing for Interviews

In order to get as much useful data as possible out of the interviews, some preparation before the actual interviews was done. This included choosing who to interview, contacting those companies and preparing questions. The questions that were prepared for designers and developers differed a bit.

4.2.1.1 Who To Interview

The preparation work was started by discussing who to interview. At this point the research was focused on the communication between designers. The research group had to be narrowed down because of the time frame. A decision was made to only talk to companies with offices in Gothenburg because travelling to other places would have taken unnecessary time that could be spent on taking the project further. This, together with the fact that there are enough companies in Gothenburg to get information about the subject was the ground for the location limitation. The next limitation discussed was that of the size of the company. The size influences the work and communication, and to deliver most value to out host company Varvet, there was a motivation to talk to companies of about the same size. Therefore it was decided that the companies should not have more than 35 employees at their Gothenburg office. Since the focus was designers, there had to be designers working at the offices as well. This did not include design resources outsourced to other companies.

One more limitation was made, which was the field of work in which the companies operated. Only companies that worked within the IT sector was included, and they had to work with development of software. The requirements for the companies were therefore to have an office in Gothenburg with less than 35 people and at least one designer and one developer employed at that office.

The people to be interviewed at this stage was designers, and since there was a limitation saying that the company should work within the IT area, so should the designers. Otherwise, no other limitation to the people interviewed was set.

4.2.1.2 Contacting Companies

Knowing the requirements for the companies, a research for which companies fulfilled these was started. Varvet had a lot of contacts in the Gothenburg area that were contacted. Additionally the teams own network was used to find suitable companies. An email explaining the project was sent to all the companies that had been found. An overwhelming amount of companies wanted to participate, which contributed to the luxury of having to choose those that were found most interesting. A variation of

companies that worked within different areas of software development were chosen; a few consultant bureaus, as well as a few product companies.

A start up meeting was held with a contact person at all of the chosen companies. At this meeting the project plan and purpose was explained in more detail, as well as our expectations on the participating companies. Additionally it was explained to them what they gained by participating. The meetings resulted in contact information to a set of designers and/or developers that could be interviewed.

4.2.1.3 Participating Companies

A total of six companies were included during the project. Each company is briefly described below.

Varvet, our host company with designers and developers that develop mobile and web applications as well as other services. The office in Gothenburg has 21 employees (Varvet 2017).

Intunio specializes in smart, mobile and connected devices. They work with combining business, design and development to build intuitive interfaces and lasting products. They are nine employees and have their only office in Gothenburg (Intunio 2017).

Wint works with accounting and has their own product which automates the process of accounting for other companies (Wint 2017b). They have seven offices around Sweden, and the office in Gothenburg is responsible for the development (Wint 2017a).

Creuna describes themselves as a leading customer experience agency. They work as consultants and develop online experiences. They have offices in Sweden, Norway, Finland and Denmark (Creuna 2017b). In their Gothenburg office they are 21 employees (Creuna 2017a).

Uptive is an IT consulting firm with focus on UX design and full stack development. They have offices in both Stockholm and Gothenburg and are eight employees at their Gothenburg office (Uptive 2017).

Forza Football is directed towards all kinds of football fans, building an application that lets the fans of different football teams to interact with the players. Their product is both easy to use and customizable. (Football Addicts 2017). They currently have 22 employees and their office is located in Gothenburg (allabolag.se 2017).

4.2.1.4 Questions

When starting to decide on which questions to ask, the first thing to do was to decide on a focus for the interviews. The aim was to gain a clear picture of their everyday work situation, how their work was perceived by them and what could be improved. This included questions about the work process, team setup, physical placement, meetings, remote work, work tools and how to handle conflicts and decisions. During this stage, the focus was on the communication between designers but was later on edited to fit developers as well. Later on more questions regarding communication at the company was added as well.

When having done a number of interviews, the focus of the project shifted and developers were included as well. Additional interview questions were written for interviews with developers. The interview questions for designers can be found in appendix A and the questions for developers can be found in appendix B. Note that these are in Swedish. Since the interviews were held with Swedish-speaking participants and Swedish was all the interviewees' native language, it seemed like more information could be retrieved from the interviews if they were held in Swedish.

4.2.2 Interviews

The interviews were semi-structured, which are explained in 3.1.1, meaning that the questions were a support and a starting point for the interviews. Follow-up questions were then added during the interview. This way each interview had the same foundation, where interesting answers could be explored further. If a follow-up question were asked multiple times with good results, this question got included in the original interview guide. In the same way, questions got excluded if they did not contribute to valuable data being gained.

Each interview took about one hour, but there was no set time limit. One person was asking the questions while using voice recording and the other took notes. All interviews were held at the interviewee's office. The following table shows who was interviewed and at which company.

Role	Company
Designer	Forza Football
Designer	Forza Football
Developer	Forza Football
Developer	Varvet
Developer	Varvet
Designer	Varvet
Designer	Varvet
Designer	Intunio
Designer	Creuna
Project coordinator	Creuna
Designer	Uptive
Developer	Wint
Designer	Wint

After a couple of interviews, it was clear that even though the communication between designers was a problem, the communication between designers and developers was an even bigger problem area. Therefore a change of focus of the project was made. From researching communication between designers the thesis now focused on the collaboration between designers and developers. This resulted in a change of questions and an involvement of developers.

The first interviews were transcribed. However, this turned out to be very time consuming. Due to the fact that detailed notes were taken during each interview, the time spent transcribing could be spent on other things. All interviews are nevertheless recorded and kept for reference.

Later in the process, interviews were done during lunch and over a coffee to make sure that there was time to meet as many as possible. A contributing factor to this was that the interest for the thesis that increased drastically, leading to more participants wanting to be included. During these interviews, the highlights were written down to be used when analyzing the data from the interviews.

4.2.3 Observations

When planning the data gathering phase, observations of those interviewed was planned to be performed. When interviewing participants, it was discovered that interviews gave a lot of information. It was also clear that to get the information that was set as a goal for the data gathering phase, observations would have to be made during a long period of time. An afternoon, which was what had been planned for each observation, would not have given a good view of their work process. It was noted that even though information would probably have been gained during these observations, it would not have been enough new information to justify the time spent. This kind of observation was therefore removed from the data gathering process.

4.2.3.1 Self Observations

When interviewing two designers that had been working very closely together, it was discovered that one of them was about to work remotely. A digital diary was created, to see how their work changed when one of them was remote. The instructions were that they should note everything that they felt was a difference from when they had worked together. For example if they contacted each other when they would have if they were in the same room. The diary was given to both designers. However, this turned out to be a hard way to record things and the diary was unfortunately forgotten. Instead, a follow up interview was held with the designer who did not work remotely to capture the challenges, advantages and differences that was experienced during this time.

4.2.4 Similar Applications

From the beginning it was intended to create an application that facilitated the communication and the process of giving and receiving feedback between designers. Due to the change of focus of the project, it was decided to not take the research of similar applications further. This because the new project focus did not include any specific application, rather a mapping of problems and solutions. At this stage, it was not clear what the final result should be, and therefore there were no similar applications to research.

4.2.5 Structuring Findings

To structure the findings from the data gathering and to identify the problem areas a KJ Analysis was performed. A photo of the work in progress can be seen in figure 4.2. First, each of the interviews was read through and important or interesting quotes and findings were marked and transferred to post-its or cut out from the printed interview material. This step was done individually to ensure that the opinion of the other team member did not influence when marking. Because of this a finding or a quote could appear twice when going through all of the post-its, thus leading to this quote being seen as interesting by both parties. When all interviews had been checked for facts, they were grouped according to resemblance and relation to each other. This KJ analysis resulted in the identification of different problem areas, which are presented in chapter 5.1.

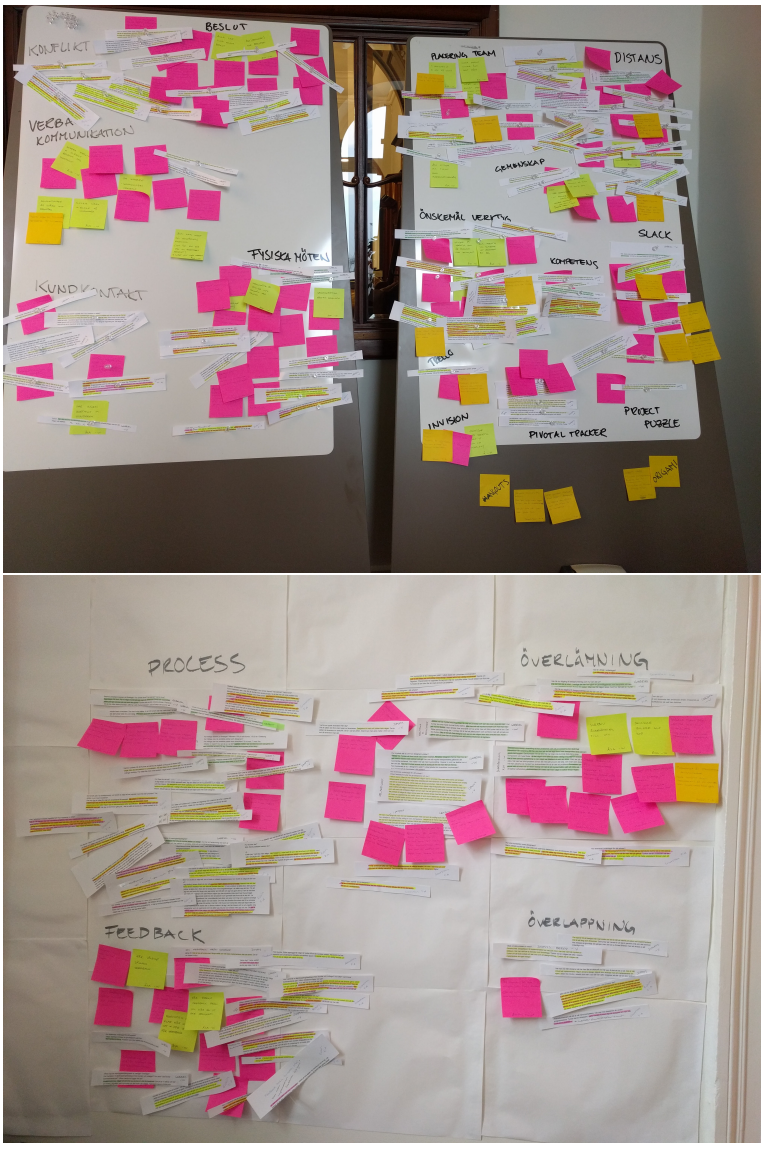

Figure 4.2: Pictures showing the groups created when performing the KJ analysis

4.3 Ideation

This phase of the project consisted of one internal ideation which was done in the project group and two workshops, one at Varvet and one session together with the meetup group Got UX. How these sessions were performed are described below.

4.3.1 Internal Ideation

Firstly, an internal ideation was performed where the problem areas detected during the data gathering phase were used as a base to ideate around. To do this, the method "How might we" was used. With the problem areas found during the KJ analysis in mind, which are presented in section 5.1, questions around these problems were formulated starting with "How might we". Each question was then answered by a brainstorming session where ideas of solutions were written down on post-its and placed next to the question. Each question was gone through twice to make sure that ideas that came up later in the ideation could influence the ideas of the previous questions as well. An idea was added next to the question even if that idea had been mentioned before at another question to emphasise the ideas that solve more than one problem. A picture showing some of the questions and solutions to those can be seen in figure 4.3.

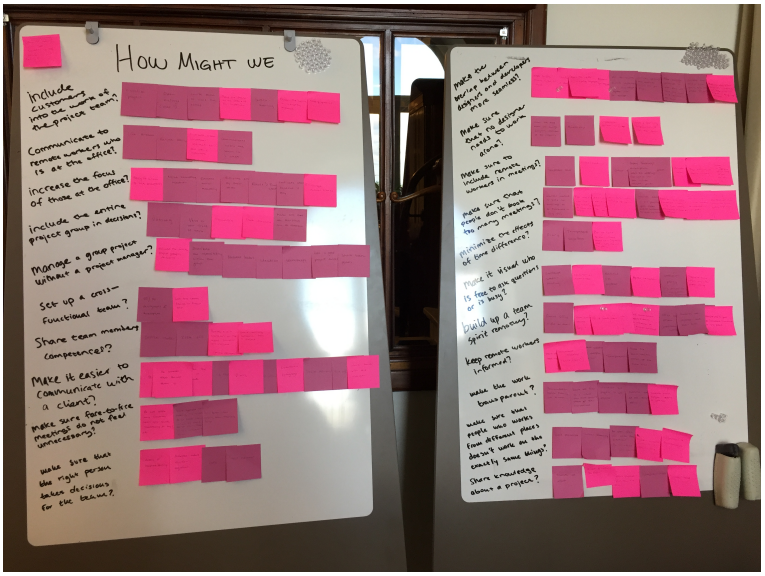

Figure 4.3: A picture from the ideation showing how the questions and their solutions were organized.

4.3.2 Collaboration With the UX Community

The 6th of April, a collaboration workshop was performed together with 30 designers and developers. These were representatives from various companies as well as students from different schools. The meetup group Got UX invited us to perform a presentation about the thesis work so far. After the presentation a 30 minutes ideation workshop was performed together with the attending designers and developers. The participants later presented their workshop result to the rest of the participants.

4.3.2.1 Planning

When planning the workshop, a question was chosen for the participants to answer as a result from the workshop. The question was based on the findings in the prestudy, and was chosen together with our supervisor Pontus Wallgren. The chosen question for the workshop states as follows:

> *How might we create a shared process for designers and developers and what could that process look like?*

Sheets of A2 papers were prepared with the question as a headline and blank space for ideation. An example can be seen in figure 4.4. Together with the prepared sheets, the participants were equipped with pens and post-it notes in order to visualize their concept.

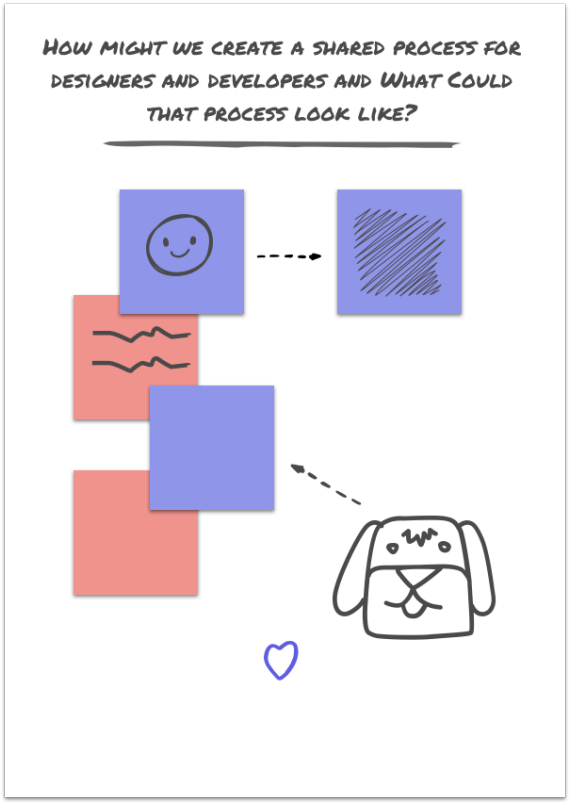

Figure 4.4: An example of a poster for the workshop.

4.3.2.2 Presentation

In the beginning of the workshop day, a 15 minutes presentation about the master thesis was held. As a part of this presentation, the biggest problem areas located during the pre-study were presented to the participants. The workshop instructions were also presented to the groups.

4.3.2.3 Workshop

After the presentation, the workshop was executed. The workshop was limited to 30 minutes, followed by presentations by each group. There were eight groups of 3-4 participants. These groups were spread out at the Varvet office with only the workshop question, equipment and a time limit. Since the discussion in many ways was more interesting than the short presentations of highlights, notes had to be taken during the workshop. To be able to manage this while administrating the workshop itself help was gathered from Varvet. Hans Rundin, Peter Hogen and Pontus Levén from Varvet helped to take notes in one group each, as well as during the presentations.

During the presentations the highlights of the process were supposed to be presented in front of the other groups. This plan was not followed, the participants rather presented their entire process and discussion to the rest of the attendants, taking longer and being more extensive than planned. Since the meetup group had no time preferences, the presentation was allowed to continue until all groups had presented their work in the same manor.

4.3.3 Workshop at Varvet

After the workshop with the UX community, which included a few employees at Varvet, Varvet asked for a similar workshop to be performed internally with the rest of the employees. This contributed to getting even more help on ideating around the area of the work process. Additionally, this workshop had a majority of developers instead of a majority of designers as the previous workshop. The workshop was divided into two parts, where each part had the same structure as the workshop at Got UX, but with different focuses. Part one was about mapping out how the process looks today and what problems there are, while part 2 had the same focus as the workshop with the UX community, namely to decide how they wanted the process to be. Each workshop was one hour long, with 30 minutes workshop in groups and then 30 minutes presentation of the result in front of all groups.

4.4 Framing of the Final Idea

When having ideated and performed the workshops, there were a lot of ideas on how to solve the different problems stated during the pre-study. However, there was not one clear solution to the bigger problem; how to improve the collaboration between designers and developers and their work process. Instead, many of these smaller solutions to parts of the problem together could help improve the overall collaboration. These smaller solutions to sub-problems were important and seen as information that deserved to be shared with others to help solve the bigger problem. It was therefore decided that the final result for this project should not be one single solution, rather a collection of solutions. Thus, a collection of guidelines for how to collaborate in cross-functional teams with designers and developers was decided to be the way to share all the information and solutions that had been generated in the previous phases of the project.

The next aspect to discuss was how to share these guidelines. For as many designers and developers as possible to have access to these guidelines, it should be accessible through the internet. Either the web or through a mobile application were the two most likely alternatives. Since there was a lot of information to convey, a web page seemed to be fitting for this project. Also, the fact that there were a lot of limitations made when deciding who to interview, made it feel important to let others add their view and insights to these guidelines as well. Writing and reading information is easier on a computer than on a phone, which was why a web page was chosen.

4.4.1 New Time Plan

The original time plan was not followed, and with the new idea it felt necessary to restructure and plan the rest of the project. A new Gantt-chart was made for the last nine weeks of the project to make sure the project was finished on time. This time, the final deadlines were added first and then the time was planned backwards. This Gantt chart can be seen in figure 4.5.

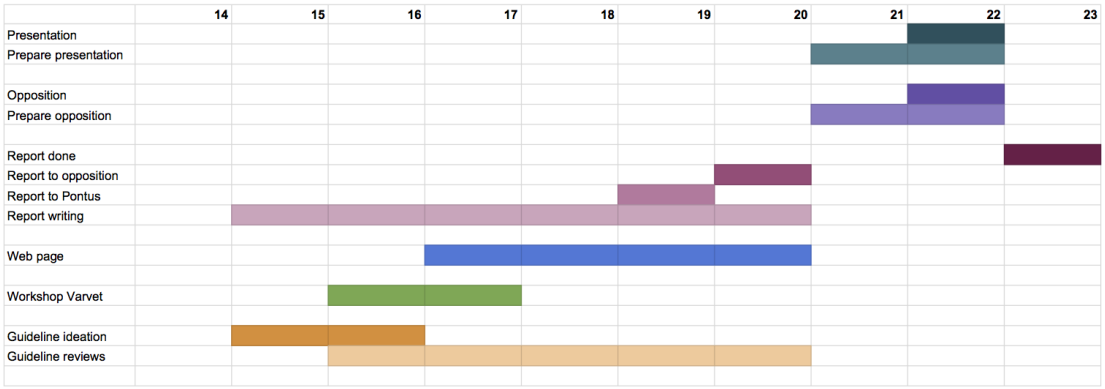

Figure 4.5: The new time plan.

4.4.2 Defining Guidelines

When going through the ideas from the ideation, these were translated into guidelines. Since the ideation was done by asking questions on how to solve a certain problem, translating the answers into guidelines came naturally. Each guideline was assigned a title and a short descriptive text. Monteiro (2012) wrote in Design is a Job that guidelines should be long enough to be helpful and short enough to not be considered additional work. The guidelines were read through and improved iteratively. They were then divided into categories to make it easier to structure them.

The guidelines themselves were not shared with others to get evaluated before the web page was published, since this kind of evaluation would come naturally on the web page with voting and commenting features. It was decided that this was enough and that that feedback would be easier to get an overview of since everything on a specific guideline would be collected in the same place, and not spread across several messages or emails.

4.4.3 Requirements

Due to time constraint, at this point in time it was not clear whether or not a web page with all the functionality desired could be developed. Therefore, both to make the development of the web page easier, and also to ensure that the idea could be developed further after this project, a list of requirements was constructed.

This was done by listing all the features that was deemed interesting for the web page, and then sorting them into prioritization order, with three levels of prioritization. The first level, Prio 1, was the minimum viable product. Prio 2 was features that were important, but not necessary for publishing a first version of the web page. Prio 3 was additional functionality that could be nice to have.

4.5 Prototyping and Development

This section covers the process of prototyping the web page with the guidelines, and later also the implementation of the web page.

4.5.1 Development of a Prototype

When the requirements were set, a few low fidelity prototypes were created. These were done by sketching on paper to get an idea of how the web page should look, and then discussing and comparing these until a common understanding was reached. A few of these sketches can be seen in figure 4.6. The sketches were used to decide on the flow and structure of the web page.

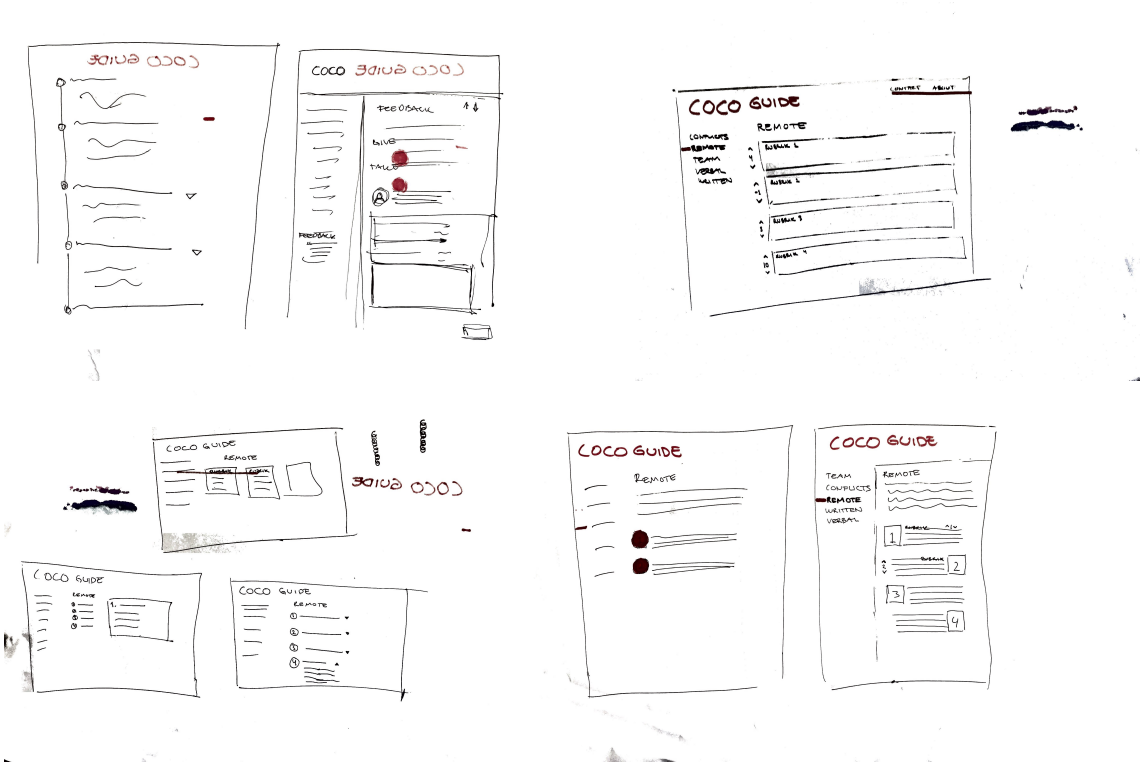

Figure 4.6: A few of the initial sketches that were done as low fidelity prototypes.

When the structure had been defined, a high fidelity prototype was made in Sketch. This was done as a guide to use when implementing the web page, as well as to reach an understanding on how the look and feel of the application should be on beforehand. The finalized high fidelity prototype can be seen in figure 4.7. A decision was made to have a top menu to choose what content should be visible. When the guide alternative was chosen, all the guidelines should be visible. The left menu is a way to sort through the guidelines quickly for the user to find their own problem area and see those guidelines. For each guideline category, there should be an introductory text to the category, as well as a list of guidelines related to that category. Each guideline should have a number to make it easier to talk about and refer to guidelines. A voting system was introduced as a way to moderate and make it easier to sort out those guidelines that are most relevant or important. The option to comment on a specific guideline was also introduced for the users themselves to share their views and experiences to make the web page even more useful for those visiting it.

Figure 4.7: The high fidelity prototype of the guide page.

The comments section can be seen in more detail in the high fidelity prototype in figure 4.8. Here the previous comments are shown, as well as the option to add a new comment. To ensure that the comments are valid, or to let the user decide if the comment is valid, the work title and the company of the author was added. This view should be shown when the user clicks on a guideline.

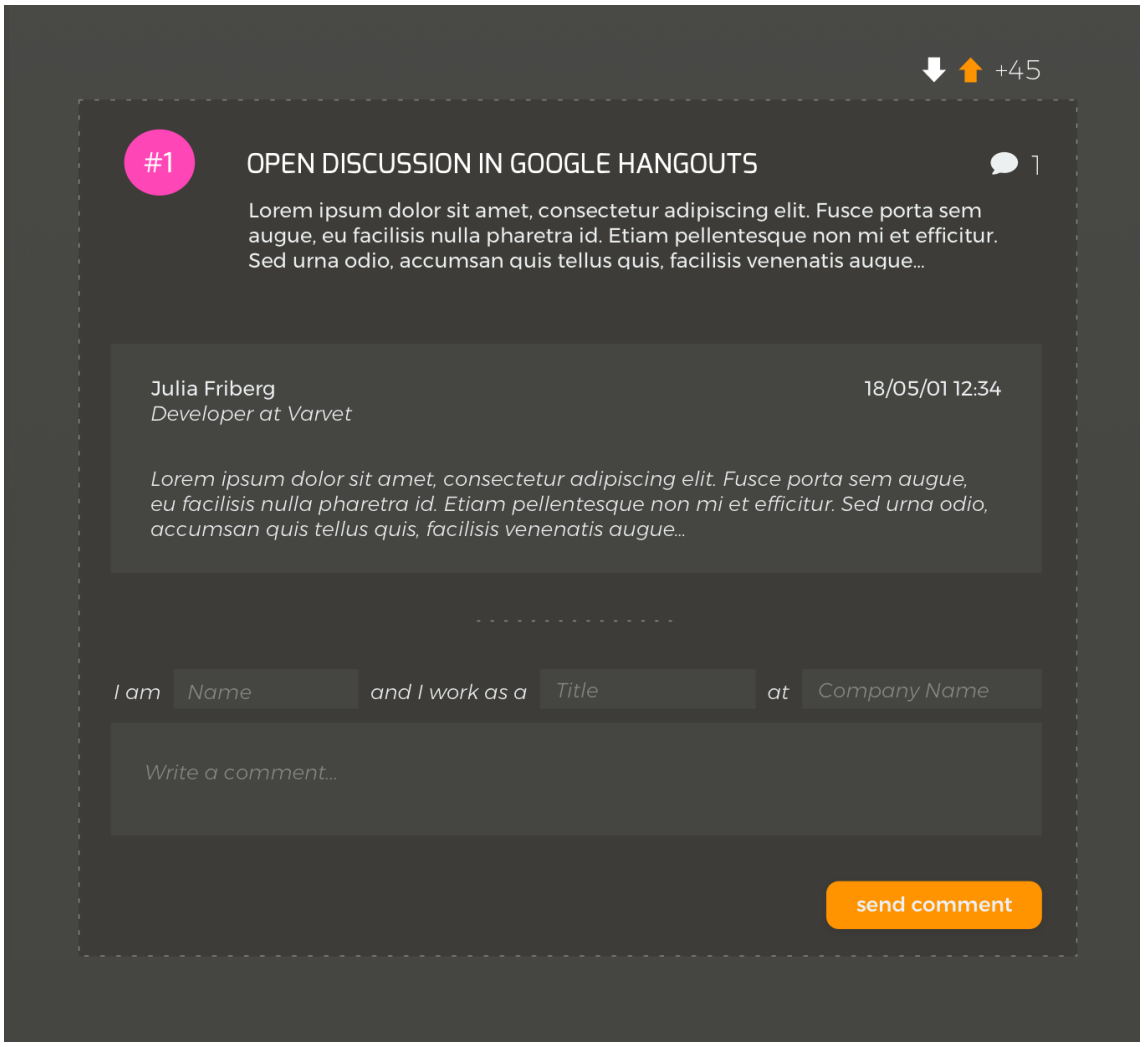

Figure 4.8: A more detailed high fidelity prototype of the comment feature.

4.5.2 Development of the Web Page

The first decision that was made when it came to developing the web page was which technology to use. Because of the time constraint, a familiar technology was chosen. The choice landed on using React (Facebook Inc 2017) to build the web page. To satisfy most of the requirements in the first prioritization level, there was no need for connecting to a database or a backend. This lead to the decision of starting with the look and feel and create the components needed, but using dummy data to fill them. During the project, the code was stored in a repository on Github (2017), which made it easy to share.

When the interface had been created, there was a need for adding real data which was deemed to be most convenient through a database. Here the choice for technology was made the same way as for the entire web page. The fact that Firebase (2017) had been used before made it the clear choice for the database in the project as well. A schema for the database was created to structure the data and then the data was added to the project, with the help of implementing Redux (2017) to keep track of the application state and data transfer. When the data had been added to the web page, it was published.

5

Result

This chapter will present the results of the thesis. Starting off with the problem areas that were discovered in the data gathering phase, followed by the workshop result and finally the resulting guidelines and the web page to present them.

5.1 Problem Areas

In this chapter, the results from the interviews will be presented. The findings have been divided into eleven problem areas. In each of those, a description of the current situations and the problems within will be presented. When referring to the interviewees, their title and a letter, as well as a number for the company will be used. As an example, Designer A at Company 1 refers to a specific person. This to make sure the interviewees and their companies are kept anonymous.

In total, 13 interviews were held at five different companies, which have been presented in section 4.2.2. The companies were chosen based on their size, the number of designers and developers as well as their field of work. The companies were chosen to get input from both consultant bureaus and companies with their own products.

5.1.1 Teams

There are a lot of different ways to work in teams. How they are placed, how they are constructed and how they work together are all aspects that are discussed below.

5.1.1.1 Physical Placement of the Team

How the teams were placed differed between the participating companies. There were two common ways to place employees at the office, either by team or by profession. At Company 3, the developers were placed together in a room, and the

designer in another room. The designer was then rarely seen in the room with the developers.

At some companies, there were individuals that worked remote and were therefore not placed with the others. Designer B at Company 1 worked remote towards a customer with the rest of the team at another location and knew that they wanted him to sit with them.

Most agreed that being placed together with the team came with a lot of advantages. Discussing ideas, problems and questions were done quickly. Additionally, the overall communicating got easier. It also created a shared understanding of the project and how others worked and collaborated within the team. The work became more goal oriented and the team could work on solving one problem at a time. However, being placed together with the team is not always preferred. The designer at Company 4 wanted to be placed within different constellations depending on the progress of the project. In the beginning of the project, during the pre study, being placed with other designers was preferred, but during the everyday work being placed with developers was preferred instead. Designer B at Company 2 also mentioned a need to work with different people than those sitting close by. Because of the need to be quiet while in the open work space, temporary groups were formed that moved to another place for for example meetings, workshops and discussions.

The designers at Company 2, who worked in the same project, had tried to work in their own room closely together for a period of time, which turned out to be a great success. They called it their room of intelligence. The advantages they experienced during this time was that they could write relevant decisions, thoughts and documentation on the whiteboards and have them close by at all times. The closeness to pointing at something they had thought about a couple of days ago helped them remember their insights and contributed to them being more effective than before.

At Company 5 they had previously been placed according to project-team, because of the advantages that brought to the communication. However, the developers felt that they got more value out of being placed with other developers. They felt that the communication around technical problems was more valuable to have close.

5.1.1.2 Working Individually

Several of those participating in the study mentioned that they worked individually. They could still be a part of a team, but the only one in that team with their profession and skill set. The designer at Company 3 was the only designer at the company and worked alone most of the time. Working individually had the effect of being blind to flaws since there was no one to talk to or discuss ideas with. It was also perceived as hard to get relevant feedback. At Company 2, one designer said that they traditionally had worked mostly individually, but that he himself

wanted to have others as a resource when designing to get feedback. When working individually as a developer, it creates a dependence on others for code reviews and while making decisions that were too important to make alone.

5.1.1.3 Groups

The structure of the teams varied. Some had project leaders or product owners included, some had not. Generally each team consisted of both designers and developers. Usually these teams were composed by more developers than designers. Company 2 had two different teams for two different products. In one of these, a council with representatives from each profession was the deciding organ and then that representative communicated with the rest of the group within that profession. At Company 1, the whole company felt like a team, although they did not work with the same projects. There the developers tried to sit and work in pairs to ensure the quality of the code.

Working closely together for a long time comes with certain advantages. The group gets to know each other professionally, and by that knows who to ask in different situations. If unaware of which areas the individuals in the group are skilled in, you rely on their title and assume they only know what they were hired to do. The roles get locked and a lot of skill can be wasted since no one knew about the others field of knowledge. This was something that several people pointed out as negative. They instead wanted to know their co-workers abilities and skills to be able to know what was expected from both themselves and others. These things are important to have in mind when constructing teams. Otherwise there is a risk that a team might be constructed of people with the exact same abilities, instead of spreading the abilities among different roles, which work in opposite of the thought behind cross-functional teams. People are different and need different things to excel in a group. This is also something that should be considered when constructing groups. The designer at Company 6 thought that a group should phase in one new person at a time to really work at its best. This is often not the case at a company that has grown quickly.

5.1.1.4 Team Spirit

The relationship between the members of a team is mainly built up when meeting face to face. When meeting in real life the team spirit is enhanced, leading to an easier communication in the future as well as a climate where team members dare to share opinions.

At many workplaces the coworkers met during the coffee breaks a few times a day. This brought a sense of solidarity within the team as well as across team boundaries. These breaks are in favor of relations with coworkers not included in the same project team, who otherwise would not communicate outside of their team.

5.1.2 Process

When asking people about their project process, the answers varied a lot. Overall, the processes were flawed and several times they were nonexistent, or at least not known to the individuals at the same company. Often the project process was split between designers and developers. Some communication occurred at the delivery between the designer and the developer once the design phase ended and the implementation started.

5.1.2.1 General Process

Generally, the process was not well-defined. Some liked it that way, while some did not. One designer said that the process should be a support if needed, anyhow it should not be forced on the group. The same designer stated that at that company they had worked together for a long time and had an unspoken process, which worked great at the moment. Unfortunately, this could become a problem when hiring new people.

This process stated that the design should be finished first, and then to be developed iteratively with feedback from the developers. This way of working seemed to be quite usual, with more or less questions from the developers.

One designer experienced the process to be development driven and that the design did not get the time needed to be iterated on. They had more focus on the functionality than the user experience, which he found uninspiring. When working with customers, the general opinion was that if the customer was interested in the design, the design process became iterative, otherwise it was not. Many developers and designers experienced the process to move quickly. There was no time to go back and iterate, if it worked, it was done.

Developers who moved from one project to another sometimes experienced that they did not know enough about the new project when starting to work on it. They found that uninspiring and confusing. They also felt that it was a lot more fun to be a part of a project if it involved coming up with a solution, not only implementing one. It was also found to be more inspiring to do something that they knew would be used by someone when it was done.

Furthermore, one developer liked when there was a plan for the week with common goals to ensure everyone knew what to do that week, and when that goal was reached, everyone could be satisfied with their work. The plan was discussed and establshed together for everyone to get their opinion heard and to feel like they did something together.

5.1.2.2 Design Process

Just like the general process, the design process differed a lot between the companies, and even within the companies. Each designer seemed to have their own process when designing. Some sketched on paper, while others designed digitally and experimented with code from the beginning. What was common was that there most often was no user testing before ending the project when the design had been implemented.

One designer mentioned that he needed to have goals for the design to be evaluated against. The same company worked with surveys to collect user data and to get to know the user, which they then used as a foundation when making design decisions. They sometimes also did user stories before sketching and making prototypes that then were tested by conducting user tests. However, this process was not always followed, most often they only designed with the knowledge they had gained from the surveys. When the design was finished, it was passed on to the developers for them to implement. Another company used workshops with the customer to create user stories, and this was very appreciated by both developers and designers. The project became better when this had been done, and when there was continuous contact with the customer throughout the project.

A developer with design experience talked about component-based design and that there is a correlation between design that is easy to implement and design that is thought through.

5.1.2.3 Development Process

When talking about the development process, most participants used the word agile when describing it. They were working in iterations with stories from a backlog. The process was mostly separate from the designers and the developers rarely had any contact with the users, since this was taken care of by the designers. These developers agreed that having a designer they trusted was key when not talking directly to the users. Having an iterative process helped in getting a natural ending to a project, even though everything had not been implemented.

5.1.2.4 Project Planning Tools

Since everyone had an agile work process when developing, they also had a tool for planning their projects. Trello (2017) was a commonly used tool which was used as a task manager and to specify user stories together with the customer. Some participants said that to be able to accept and reject would have been a nice feature to have for tasks in their Trello board. Having a Trello board helped to keep track of what had been done. This was mainly used for development, and there was rarely anything design related included in the tasks. One designer felt that they did not take advantage of all the potential in Trello.

Another tool that was used for stories was Pivotal Tracker (2017). Here stories were added and could be prioritized and accepted. Each story has their own thread where things about that story can be discussed. It works like an archive where everyone can go in and see what is happening at any given time. When working with a customer it seemed to be important that the customer was involved in the planning in Pivotal Tracker as well for the project to be successful. A developer said it was hard to know when something was done and that, as with all communication, there could be misunderstandings. One developer avoided Pivotal Tracker since he experienced the program as hard to communicate within.

Furthermore, at one company where the employees worked at different projects, a planning tool called ProjectPuzzle (2017) was used to get an overview of who worked at what project. This was appreciated and the schedule for a person could be synced to their calender. However, no notifications were sent to the person when something changed.

Working with several of these tools, together with other communication tools, made it hard for employees at a company to know where to communicate what information.

5.1.2.5 Problems With the Process

Several participants mentioned that they wanted a more structured process, a common view on how to do things and how to tackle a problem, instead of everyone having their own process. A designer mentioned that having stated goals and steps in a process would help towards having a common process and a way to structure the work. Although a more structured process was a request from most participants, they also wanted it to be flexible. A good thing about design was that designing something in the wrong way is easily fixed by redoing it, especially when found during the early stages of low-fi prototyping. As long as a bad design is discovered, a quick flexible process works as well.

Where in the process most problems where encountered differed. Some felt that it was hard to get the project started and that they felt as if they missed some kind of description or document to help them get started. One developer said that things got stuck in the middle, and specifically when stories should be tested. Another person said that he never felt done and that there were things to improve even when the project was deemed as finished. Knowing when a project is done was something several people mentioned as a problem, for example one designer said that the celebration of being done came too early. A celebration was done when releasing a new feature, instead of celebrating two weeks later when a lesson could be learned from how the feature worked. These learning outcomes were lost since everyone had moved on to the next feature. Other things that were mentioned as hard was that a design could be finished long before the developers started to implement it, and when they had questions about the design and the decisions that had been made when designing it, it was hard to remember the reasons behind it. Thus, there was a lack of documentation about design decisions. Estimating time was also mentioned as something that was really hard. Something that appeared easy turned out to be very complex instead, leading to the plan not being followed. That lead to other things having to be compromised, which caused a strain on the communication in both the team and to customers. To remove features instead of adding more features was also something that people missed, as well as better planning and more personal development.

5.1.2.6 Communication

Overall, more communication was something people asked for. Communication about the project and the process and what had been learned during it was missing and seldom shared with others outside of the project. Closer communication between all roles would lead to better decision making and a better final result. One designer said that communicating throughout the project with the key individuals created a common understanding and team spirit. A developer said that a lacking communication was like digging a tunnel from two different sides and then realizing that they had not dug in the same direction. Last but not least, how much communication that is needed depends on where in the project. Different phases demands different types of communication and collaboration.

5.1.3 Sharing Information Within Teams

The most common sharing of information was when the designer shared a finished design with the developer, who while implementing it gave more or less feedback on that design. How involved the designer and the developer were in each others processes and how the delivery between them looked was very different at the different companies. Most developers wanted to be more involved in the design process, but Developer A at Company 3 did not want to get involved at all. He said that he wanted as much as possible of the design to be finished before receiving the design material. He simply did not want to think about the design.

5.1.3.0.1 Separate Processes

The most common methodology when producing software was that the designer started to develop the design followed by implementation from the developers with limited or no discussions in between. The discussions taken place was often far apart, sometimes because of a high workload where the developers needed to focus on their own implementation. At Company 3, the developer stated that he trusted the designer and did not question the received design material. A designer at Company 2 also spoke about how trust is needed within this process.

A common problem between designers and developers were the lack of knowledge about each other's work. The designer was rarely integrated in the developers process and vice versa, which many participants agreed on was not working well. They stated that the process and result would be better if all coworkers were integrated during the entire process.

5.1.3.0.2 Joined Processes

Developer A at Company 1 would like to be included earlier in the design process to be able to feedback on the design. Earlier inclusion would lead towards the goal of a shared understanding about which parts of the design that could be compromised, and why some could not. Another advantage of being involved early on was to be able to mentally prepare and to plan the work before implementation.

Designers and developers agreed that the designer should be included during the entire process as a part of the development team. For designers and developers to sit in the same room was appreciated by both designers and developers. They agreed that there should not only be deliverables between designers and developers, but rather a tighter collaboration between them. The design work should be transparent and easily accessible within the team.

The designer could then also ask the developers questions when having technical problems to understand or to get help when implementing their own design.

5.1.3.1 Delivery of Design

Today, many developers receive design material when the design has already been set. It works as a delivery, where the designers deliver the design for the developers to implement. This was especially clear when the designers and the developers worked at different companies. The method of working apart was not preferred by the teams. Developer A at Company 2 stated that the times when a project had been the most successful, designers and developers had worked together during the entire process.

The deliverable from the designers differed. The most common items received by developers were screenshots, mock-ups, prototypes, images or code. These items were often received via chat, by email or on a physical meeting. When working remote a lot of screenshots are sent back and forth between the design and developer instead of sitting together.

The developer at Company 2 would prefer if the designers instead pushed their own design changes to Github. He also wished for the files to be delivered in the correct format, five sizes in a .zip-file. Design material that has been delivered should be clear and usable for the rest of the team, Developer A at Company 1 said. Furthermore, he wanted to be able to make technical requests before the delivery. Having the designer deliver to the developer was also preferred, however, sometimes it was the other way around. The developer could create a component, which the designer then had to complete with design. Sometimes the developers created this component from a rough sketch from the designer and then the designer could add a final touch at the end.

A surprising fact was that at Company 2, one developer had to implement a design by memory and conversation. This was because the developer did not receive any design material from the designer, which was not discovered by the designer at the time since the developer never asked for any material.

5.1.3.2 Difficulties

Many problems occurred when the design and development team was divided spatially into two parts. This lead to less communication, which caused problems.

Another thing that can cause problems was when the knowledge gap was too large between the designers and developers. A designer that is too focused on look and feel could have a hard time understanding the technology behind it, thus not taking implementation into consideration. When the designer did not back up their design with arguments, a vexation could emerge.

The delivery of information between parts could also cause problems. Designer B at Company 2 learned that presenting a large task for the development team resulted in confusion, stress and demotivation. Also interactions and animations was experienced as hard to communicate.

Additionally, deciding when a project is ready for release was experienced as hard. Since developers and designers may not think alike concerning design, problems may occur during implementation. Designers stated that when the work was split up between designers and developers without further communication, the result would not mirror the sketch delivered from the designer.

5.1.4 Feedback

All participants agreed that feedback was important when working together in a cross-functional team. All feedback is good feedback, as Designer A at Company 4 said.

5.1.4.1 Feedback Subjects

Feedback about design was often subjective when concerning the look and feel. This type of feedback is agreed, by most participating designers, that it does not always have to be considered. Feedback about functions and flow were more appreciated and is also more well received. The level of details of the feedback varied. A developer at Company 1 stated that he did not feedback on details of the design but rather on the interactions, which tend to be the hardest parts of the design.

Designers needed input from others to review their design. Some developers found joy in giving feedback about the design to be implemented. Designers at Company 1 and 6 valued when developers questioned their design before implementing and the developers valued when the designer had well thought through arguments to support their design.

5.1.4.2 When feedback was received

A majority of designers in the study received feedback once new material was released to the developers. How often the designer released new material would therefore affect how often feedback was being received. The feedback cycle tended to be shorter within internal projects, resulting in a smaller amount of feedback being received with a higher frequency.

It was pointed out to be important to always ask colleagues for feedback in order to receive any, especially when working remote. A developer at Company 1 said that they needed to work at giving each other more feedback than they do today. The feedback given today was sometimes carried out via planned retroperspectives. At Company 1, a developer always asked for feedback before anything was perceived as done.

Designer B at Company 1 felt he had a better overview of the whole product than the developers. He therefore had an opportunity to criticize the product as a whole in a way that he did not think the developers could when working on smaller features. He preferred to discuss details with developers.

Therefore, designers often received feedback from developers during implementation. The feedback was then delivered as a dialogue between designers and developers. All designers appreciated relevant feedback from their coworkers, but Designer A at Company 4 thought that it could be hard to stand by your opinion when receiving unwarranted feedback from colleagues. A designer at Company 1 encouraged feedback. He said that all coworkers should feel safe about delivering critical feedback to each other.

When feedback was received after a release, it was hard to fix the problems detected. This was the situation at one company, where the feedback only came from the users that used the product after release.

When it comes to feedback for developers, many developers at different companies involved in the study received feedback from each other through code reviews, with the purpose to improve the code quality.

5.1.5 Working Remote

What was found out when asking questions about working remote was that people generally thought there are more negative aspects than there are positive. These different aspects are described below.

5.1.5.1 Time Difference

When working remote with a time difference, several people said that it naturally became harder to have a two-way communication. The communication then became restricted to mailing, where the mail was sent during the day to be answered during the next day. The time difference contributed to less outspoken communication, but also more of a silent communication where they looked at the progress in files instead of talking to each other. It became harder to sync things and if there was a need for a video call, then one of the participants had to change their work hours in order for it to be performed. Nevertheless, Designer A at Company 1 mentioned that the lack of real time communication contributed to things having time to breathe, which was positive.

5.1.5.2 Remote Meetings

When having meetings and having people remotely connecting through video the people sitting remote were experienced as harmed by their situation, if the meeting had several participants. They then had a harder time getting attention and contributing with their opinions. They were perceived less valuable by the people at the actual meeting as well. The ones that were remote experienced the meeting as boring and a lot longer than it actually was. They lost their motivation and felt like they could not contribute and that if they had something to say, they had to commit a lot more to it than they would have had if they had not been remote. The remote workers also felt like it was uncomfortable to have a monologue to a computer and that they did not want to engage in important discussions since they could not read the body language of their fellow participants. They lost their motivation and that all meetings seamed to be the same and hard to differentiate between afterwards. Designer B at Company 2 suggested that this was because they had less senses to connect to the meeting.

5.1.5.3 Responsibility

Several people experienced that the person working remote has a certain responsibility to make it work. This includes asking for information and making sure that the work gets done, but also to be responsive in communication. Developer A at Company 2 noticed that if you do not ask for information, you are not getting any.

All kind of feedback is a lot more important if you are remote. Attending meetings while being remote is also something that is very important to not miss any information. You have to be transparent in what you were doing to make sure people see your work so that they do not work on the same thing that you are.

Developer A at Company 1 experienced that a hard thing while being remote was to differentiate between work and spare time. Often when working remote, the people we talked to had not had any office space, but rather worked from home. This lead to a responsibility of working during work hours, and also to not work when outside of work hours. Since they did not have the goal to finish and go home, they felt like they were working 24 hours a day.

5.1.5.4 Changes in Work Process

The main complaint about what happened to the work process while working remote was that it got a lot slower. The process was not as effective and did not proceed in the same speed as it would have if everyone was in the same place. The communication issues seemed to be the biggest factor to why the process got slower. Waiting for feedback took longer, which lead to more individual work before asking for feedback, and therefore also more feedback when actually getting it. People felt more isolated which made the work less flexible. Additionally, the communication during the work process transferred from being verbal to written. Some felt like this contributed to more misunderstandings. Others felt that it helped to write things down to understand the problem better, as well as to save the documentation which if taken verbally does not exist. Also, they experienced that there was less communication when someone was remote. Not knowing what the person remote was working on at the moment contributed to people feeling hindered to contact that person, due to them not wanting to disturb. Furthermore, actually writing or calling someone was more of an effort than to go to that person when in the same place. An agreement had to be made to talk to someone when they are remote, and that extra step necessary to contact someone was damaging for the communication. For example, Designer A at Company 2 said "There should not be a need for an agreement because then it has become too hard. It is a too big step, and unfortunately, that is how most tools work today".

When actually talking to someone remote, mostly video calls were made. These were used during meetings, as mentioned before, but also as a tool in the work process. Developer A at Company 2 said that when working together with someone and being remote, a video call almost made it feel like being at the office. The video call was used to talk freely and anyone could join in while working.

5.1.5.5 Social Interactions

When being remote, people felt that they missed out on a lot of the social aspects of working together. They missed the coffee breaks, and therefore also the ideas and topics that were not important enough for meetings. A simple thing as eye-contact when talking to someone was something that they missed. They did not know who was at the office or not. Developer A at Company 2 pointed out that since he only talked to a few of their colleagues while working, it felt like the rest of them were not even there. He also missed getting to know the new people at the office as well. Generally all social interactions, and also all personal communication felt lost.

5.1.5.6 Advantages

Although people felt like there were a lot of disadvantages to working remote, they could find some things they did like about it. What was mentioned most was that they could focus a lot better when they did not get interrupted while working, leading them to being able focus on the task at hand. They also felt it was easier to multitask during for example meetings, when there were things that did not really concern them. People working remote could take breaks more naturally when it fit, instead of when everyone else did it. Some even said it did not affect their work at all. Laptops and tools like chat clients and video calls made it possible to work remote. Some had the philosophy that you should work from where you need to be, and to not be bound to a certain geographical place. Some did say that it worked better to be remote than they had thought it would, but they still think it is better to be at the office.

5.1.6 Physical Meetings

Meetings can be both planned and unplanned. A few companies described that they had more spontaneous meetings than booked ones. Often they stood by each other's tables while performing these meetings.

Another company chose to have one planned stand-up meeting every morning to keep the team informed about each other's progress. And they were not alone in their number of planned meetings every week. Many companies had fixed meetings

several times a week. What determined how often a meeting was called seemed to be the culture of the company. Some were better than others at determining when a meeting is needed and when it should be avoided.

The general opinion was that meetings should not be planned often, only when there was enough motivation for a physical meeting. The companies with fewer meetings booked stated that they were pleased with this situation.

Weekly meetings were used at some companies to set up goals for the following week as well as for following up progress. Daily meetings were being used to follow up progress, what had been done since yesterday as well as what is going on at the moment. Other topics of these meetings were clients, personnel and the companies progress. Physical meetings can also be planned for prioritizing and help with problem solving.

Several people mentioned that they wanted meetings to be kept short and to the point. If a question or subject that only involved a few of the participants the discussion should be continued between those concerned after the end of the meeting.

Furthermore, physical meetings can be performed both in groups, but also in pairs. One company frequently sat together in pairs while developing to increase the code quality and to be able to discuss problems and solutions.

5.1.6.1 Advantages of Physical Meetings

The main reason for meeting face-to-face was the efficiency that followed. When meeting face-to-face the team members could ask questions and get instant feedback from the team. One designer stated that physical meetings was a good way to get small discussions and problems off the table. Some participants said that they would always prefer to have physical meetings.

Relationships are built when meeting face-to-face, not only within the company but with the clients as well. Meetings could also be used as an opportunity to check on the general well being of the team. Many participants stated that it was a positive thing to see each other in person on a regular basis. Additionally, regular meetings kept coworkers informed about the status of the company's work.

Another advantage of meeting face-to-face was the opportunity to easier solve conflicts. The possibility to explain a problem directly to a coworker was a contributing factor to this.

Having the routine of weekly and/or daily meetings was very convenient to some as well as it was a way to make sure that some meetings were performed and not forgotten.

5.1.6.2 Disadvantages of Physical Meetings

In contrast, participants also stated that not everything was more efficient when working face-to-face. This was especially experienced when planning a meeting without a clear agenda.

Another thing contributing to the lack of efficiency was the shortage of documentation. Many interpersonal meetings were not well documented in notes, thus making it hard for others to follow up decisions that were made on a physical meeting. Even when taking notes it was impossible to catch every phrase that was said during a physical meeting.

Having too many meeting was generally not experienced as something positive. When planning more meetings than necessary both designers and developers felt that they could use their time in a more efficient way. This felt stressful and counterproductive. An example of this was at a company with daily stand-up meetings, where people often said exactly the same thing day after day.

Another problem with meetings was that people did not know who to invite. To invite someone that did not need to participate in the meeting brought the same problems as having too many meetings. This lowered the motivation and brought focus from more important tasks.

5.1.7 Written Communication

Communication that was written was generally perceived as more important. The fact that people had to think about what they wanted to communicate and formulate it in a good way made it feel more official. This however differed a bit depending on where it was written. Another reason for using text as communication was that it was not dependent on neither time nor place. Questions not urgent were therefore more common in text. What was experienced as a problem with using written communication was that people interpreted what was written in different ways. This could lead to misunderstandings and some mentioned that having a common view was important before discussing things in text. Developer A at Company 2 pointed out that brainstorming and other similar activities were less successful in text since they felt too serious.

5.1.7.1 Email

Email was mostly used to communicate with external colleagues or clients. The reason for this was that it is a simple way to contact whomever, since no invite was needed. This makes it more efficient contributing to one less obstacle when contacting someone. Files can be sent through email, but often with restrictions on size. Email was also used to deliver reports in some cases.

5.1.7.2 Chat

Everyone asked had some kind of chat client for communication. It was used for quick questions, especially between people in different places, or simply to not disturb the other person while working, in a way that spoken communication might. Several different things seemed to be discussed in chat clients. For example code, feedback, everyday communication and when sharing something interesting. It was also used to share files since there is no limit to the size of the files and because several people can get access to the same file at once. Designer A at Company 2 mentioned that the chat client demanded low effort from the user and was very quick, but that a common view between the participants was necessary for it to work properly. Discussing something without a common view in a chat channel did rarely lead to anything being solved. Several participants experienced a lack of possibility to easily go back in the chat history to find things. They experienced that things got lost. They also talked about the problem with documentation and what should be discussed where. Important things could have been discussed in a chat channel and later the reasoning and documentation from that discussion was hard to find when it was needed. It was also perceived as hard to make decisions since there was no way to really say when a discussion was over and it was hard to know when everyone had seen and approved of something. People did not have the same idea of what should be discussed in a chat and what should be discussed during a real meeting. Although there were no limitations or rules to what should be discussed, which lead to people having their own opinion about this, most people liked having this kind of communication tool and felt it made their work easier.

When it came to chat clients, the one that was most common among the interviewees was Slack (2017). The general opinion about Slack was positive. They liked the possibility to use different channels and threads to discuss different subjects. They also liked that people could be tagged by their name. The problems they experienced were mostly because of people not using enough threads and things getting messy and hard to follow. They thought it was hard to know what was relevant and they felt they had to read everything to know that, which they did not like. They missed some kind of version control and archive for files to keep track of files and which file was the current one. Developer A at Company 2 appreciated the fact that it is web based and the user does not need to install anything to make it work.

5.1.8 Customer Contact

This section is about the contact with the customer. The type of contact varies if the company is a consult bureau or a product company. For a consult bureau, the customer refers to the one ordering the work, and paying for it. For a product company, the customer often is the user of the product. However, consult bureaus also have users for their products, but these are not included here.

The amount of contact with a customer, and who contacted the customer varied a lot. At Company 1, both designers and developers had customer contact, while at Company 3 the designer had no contact and the developer had a lot of contact with the customer. Some developers mentioned that the amount of contact depended on the task. Some tasks demanded a lot of contact and discussion, while some were more of an individual work and in no need of input from the customer. Designer A at Company 1 thought the contact got easier and the project was more interesting if the customer was interested in the design. Similarly a developer at the same company mentioned that the customer was not at all interested in looking at the actual code. They agreed that if the customer showed interest, the communication was easier. It was experienced that a lack of customer contact had a negative impact on the personal interest and engagement for the project, since the feeling of doing something important and also the appreciation from both customers and users got lost.

A lot of the communication with a customer was through meetings, both face-to-face and through video, depending on the distance. Designer A at Company 4 worked a lot with the customer and often sat near the customer when working, as a part of a team at the customer's office. The contact was therefore daily and when remote the communication was through email or phone instead of in person. They all mentioned the importance of understanding what the customer meant. Often the customer came with a solution, and then they had to step back to find out what the actual problem was, and solve that problem in a way that probably differed from what the customer suggested. At Company 2, Developer A tried to really find the problems the users experienced. The way this developer communicated with the users was by daily going through the reviews they had left in the application store. This was a decent way to communicate, but it could be improved. He wanted to get closer to the users and see how they really used the application.

5.1.8.1 Difficulties

At Company 1, most of the interviewees experienced difficulties when communicating with a customer. They felt it was easy to assume the customer understood, when they did not. This often lead to misunderstandings, and was something to always have in mind when communicating with a customer. They also felt that the customer often had the wrong idea about what was time consuming. This was

solved by a lot of communication, but since the developers did not know exactly how long things took either, this was a problem. Others mentioned a problem with communication with the person deciding the budget. It felt like the one with the money wanted to work the way they thought they saved most money, while both designers and developers wanted to work in a way that delivered the best product in the long term. The customer thought in short-term, which lead to conflicts and compromises.

5.1.9 Decision Making

Most decisions made had one thing in common; the one making the decisions needed to be able to motivate why. This motivation should be valid with a clear background, often with user feedback or other well supported arguments to back up the decision.

5.1.9.1 Group Decisions

The most common answer to who makes the decisions in a cross-functional team is that the team makes them together. One company explained how they all sat together when planning a project, with no project manger included. Another developer explained how the development group was in charge of the majority of the process with some help from the managers. The managers then handled some of the costumer contact but the team made the major decisions. At one company, the development team worked close to the users, thus knowing a lot about how they think and act. This also spoke for the team being the main decision maker during the development process.

In some cases only a few members of the team made decisions, since it was experienced as more effective than involving everyone. This was how they did even if others may have had an interest in participating.

For the group to make decisions, the team needed to take responsibility and care about the product. One or many of the group members also needed to be interested in taking responsibility for the costumer contact. The one with the closest contact to the users and costumers was experienced to have more authority when making decisions. The authority could also be based on the knowledge within the team. Designers talked about how their knowledge was recognized even in cases where the designer was new to that specific workplace. This sense of freedom and trust brought joy to the group.

Another factor that impacted who made the decisions was the feeling of ownership. Developers stated that they felt ownership in a project once they had contributed to that project.

5.1.9.2 Project Manager

The lack of a project manager was mentioned by some individuals. It was mentioned that it would be positive if a project leader could help by making the hard decisions, but it was also mentioned that this individual did not have to be a project manager but rather someone with a talent for communicating with others.

A problem could occur when having a selected project manager, especially in the cases where this individual's only task was to be a project manager, thus not knowing any inside information about the project. This could cause problems when the team members felt that the project manager knew too little to make good decisions.

In some organizations the designer was the one making a majority of the decisions as well as keeping in contact with the costumers and the end users.

5.1.9.3 Project Owner

Another role that was mentioned by some as an important one was the product owner. Some mentioned the product owner to be from the customer and others had them in the development team.

The product owner was often a person with a business or marketing background. This person then had to know or learn the technical skills needed to be a good product owner, but not many did. This was a problem since the product owner needed to be able to have a dialogue with the entire development team, without the proper knowledge of how to do that.

The general impression of a product owner was that that person should have an overview of the whole project and able to make decisions on all levels of the product. This person was also responsible for all feedback from the team, the board and the users. This feedback should then be converted into something useful that would bring the project further.

5.1.10 Conflicts

In a cross-functional team, sometimes there are conflicts. These can be because of misunderstandings or differences of opinions. What they were about and how they were solved was handled differently at the different companies.

5.1.10.1 Topics of Conflict

Most conflicts were regarding design decisions. They were mostly about interactions and almost never about details of the design. Often a discussion started and everyone could support their arguments in a good way. A designer at Company 1 said that when people did not get along, it was often because somehow, things had to be prioritized, which lead to compromises, and they did not agree on what to compromise. At Company 2, a designer said that conflicts arose because they did not agree on what were requirements and what were not.

5.1.10.2 Solving Conflicts

Most agreed that solving a conflict was done by having a democratic discussion where everyone got to have an opinion and express their arguments for why. At Company 1 they said that everyone was entitled to their opinion, but it still felt like everyone could change their mind and accept another solution than their own. They could agree to disagree if the arguments were reasonable. If they could not agree, the customer could make the decision. At Company 3 decisions were made based on advantages and disadvantages for the customer. Company 2 had the ambition to reach a consensus and if they did not, then they would have to look at it again. However, Designer A said that he tried to convince others about what he thought was best by showing mockups and argue for goals. He also said that he sometimes did not even care what other people thought and decided on his own. He did however also mention that if someone had a strong opinion about something and they could not agree, they had to step back, maybe narrow the scope and see if it really was a problem or not. A developer at the same company said that it was all about the arguments and that it was a process of giving and taking.

5.2 Workshop at Got UX

The workshop at the Got UX meetup group resulted in eight groups' ideas represented by a poster, a presentation and some notes. The ideas concerned what they thought was important to include in a joined process between designers and developers, and also how that process could look like. The question that was answered by the groups during the workshop was:

> *How might we create a shared process for designers and developers and what could that process look like?*

The ideas from the groups are presented below.

5.2.1 Group 1

Group 1 consisted of all designers with development experience. They divided the process into four different areas, which can be seen in figure 5.1. First they stated that there should be a project kick-off, where the product manager started the project. They felt that it was important to have a common ground with someone to lead the project. Designers, developers and researchers should all be involved in the planning of the project in this phase. The next phase was a conceptualizing phase. In this phase it was important that everyone was involved as well to create an understanding of the product and to make sure the ideas did not seem unplanned. After the product was conceptualized, a list of tasks and their prioritization should be finalized by the designers as well as the product manager and the developers. These tasks were then divided into sprints. Each sprint had an introduction, a design phase, testing and releases done by the developers. They pointed out that they wanted tests to be run as quickly as possible.

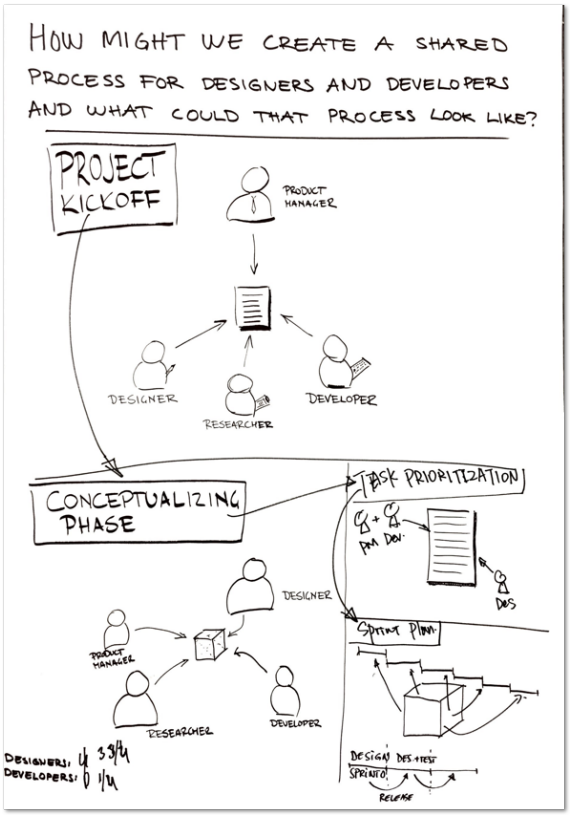

Figure 5.1: The poster made by group 1.

5.2.2 Group 2

In group 2 all three group members were designers. Their process had three phases. The first one was about gathering information. They wanted everyone to gather their own information, for example about constraints and the background of the project. This was also the phase where their previous experiences should be shared within the team. The next phase was a functional walk-through, where everyone should sit together and go through the gathered information. The final phase was the actual work phase, which should be iterative. In this phase it was important to update others on the status by stand ups or other forms of meetings. They stated that everyone should feel like a product owner and be responsible for their own actions. Everyone should also feel like a part of the team.

Their suggested process should be solution oriented and developers and designers should have separate processes, but keep each other updated. They also talked about the importance of documentation and experienced that people were better at documenting when they were remote. They wanted the documentation to be shared and to have an iterative documentation to keep track of decisions and versions. Their poster can be seen in figure 5.2

Figure 5.2: The poster made by group 2.

5.2.3 Group 3

Group 3 had two designers, one developer and one IT consultant included in the group. This group had a long process with focus on testing, which can be seen in figure 5.3. They wanted developers and designers to be involved a lot during the process and to have a continuous communication. The process started with a problem, which was followed by an idea from the designer. The idea was then transformed into a specification, where the developer should have an opinion about technical limitations. After that the designer should research, followed by an analysis with conclusions. These conclusions should be used later on when creating wireframes, which then should be tested with the developer involved in the process. This should be an iterative process with remaking of the wireframes, depending on the test results. The same process was repeated for high-fidelity prototypes. After this the developer started the implementation phase, with a lot of software tests where the designers should be involved. When this phase was finished, a demo should be performed, followed by a delivery.

Figure 5.3: The poster made by group 3.

5.2.4 Group 4

Group 4 consisted of two designers and one developer. They highlighted the importance of having a common ground when collaborating. They also talked about how transparency helps when working together, and illustrated this with a glass table below which the overlapping work of designers and developers could be seen. They mentioned that the process was dependant on the organisation. Additionally they emphasized that working together at the office was not always a good thing, they felt that they could focus more remote and did not get as distracted as when working at the office. They stated that the key when collaborating was to get to know each other and have fun together, both while working and outside of work. Their poster can be seen in figure 5.4

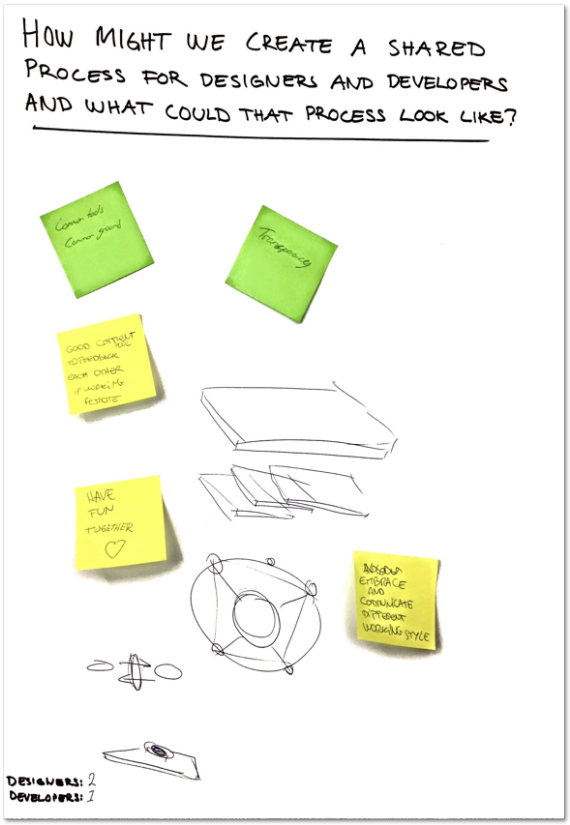

Figure 5.4: The poster made by group 4.

5.2.5 Group 5

Group 5 had two designers, one developer and one that identified as both developer and designer. Their poster can be seen in figure 5.5. Group 5 did not come up with a process, instead they argued that there is no specific process that works for everyone. Each process for each team should be individualized to work for that team specifically. They also said that the key to a successful collaboration was communication. Communication about expectations, their fields and what they were doing. They also said that an understanding of each other's fields was important and that being double-roled was an advantage. They talked about how designers wanted feedback to improve their design and that there should be some contract or guidelines for whom that was responsible for what in the different stages of the process. They also mentioned that it was hard to communicate directly with each other and they wanted a coordinator between the designer and the developer to solve that problem.

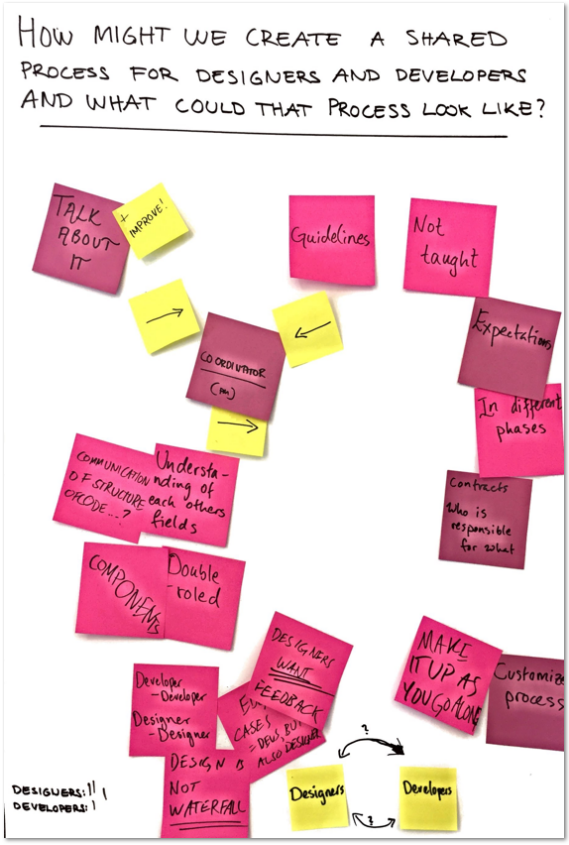

Figure 5.5: The poster made by group 5.

5.2.6 Group 6

This group had two designers and one developer. Their poster can be seen in figure 5.6. The lines at the top of the paper are two different process flows illustrated. The first one was the correct one, according to them, where the process of the designer and the developer overlap. The second was a process built on deliveries, and this was not preferred. Working together was important. Also learning from each other and having a common understanding and goal was something they valued highly. They also talked about the importance of showing vulnerability and admitting when something was not good, and to throw that away, even if a lot of time has been spent on it. Spending time on something does not automatically make it good. When starting of the project as a designer, they said that doing a minimal viable design that the developer could start implementing was something they recommended. Start with a simple design structure that is a minimal viable design, and then iterate the rest of the design. They also said that it is easy to have an opinion about design, and that people should have an opinion and ask a lot of questions. When it came to documentation they felt that it was a waste of time in most cases, and that they wanted to create a natural history of the project by photographs and screen shots to make people see the design without having to document it.

Figure 5.6: The poster made by group 6.

5.2.7 Group 7

Group 7 identified themselves as 2.5 designers and 0.5 developers. They had a more abstract concept than the rest of the groups and made up a metaphor for the collaboration between designers and developers. They said it was like a dance where the designer was the one leading. Sometimes the designer could send the developer away to do a turn on their own, but then they always returned to the dance together. They said there had to be trust between the designer and the developer and that the designer should inspire the developer to become even better. They mentioned the one responsible for the project as a cheerleader, cheering for the dance between the designer and the developer to be as good as possible. They also pointed out that neither the designer, nor the developer should give the impression of being a good dancer to each other, to make the other one ask questions that might otherwise seem too stupid. This illustration can be seen in figure 5.7, where D is the designer, U is the developer and A is the project owner.

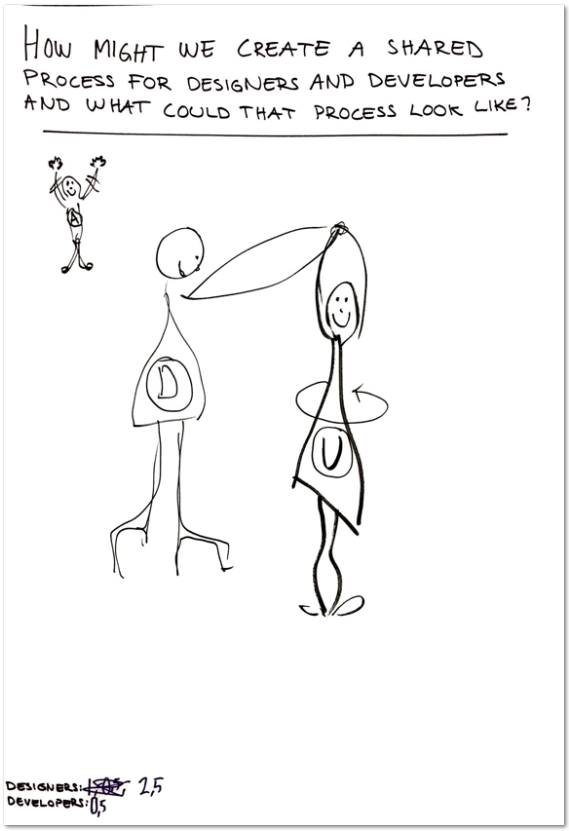

Figure 5.7: The poster made by group 7.

5.2.8 Group 8

Group 8 had illustrated a well-defined process on their poster, which can be seen in figure 5.8. The process started with trying to understand the problem and what it is that they are trying to solve. At this stage, it was important that the problem was a real problem and that it should be solved for real people. The next phase of the project was testing out ideas on how to solve the problem. This was done by a highly collaborative process where a design should not be entirely done for it to be shared with others. They mentioned that it is good to share the design as early as possible to start a conversation around it. They then started to build and deploy the design and followed by a new iteration, taking into account the outcomes and the gained knowledge from the previous iterations. They talked about the importance of communicating throughout the process and to share both ideas and knowledge between team members. Thus they should communicate the expectations on each other and feedback a lot.

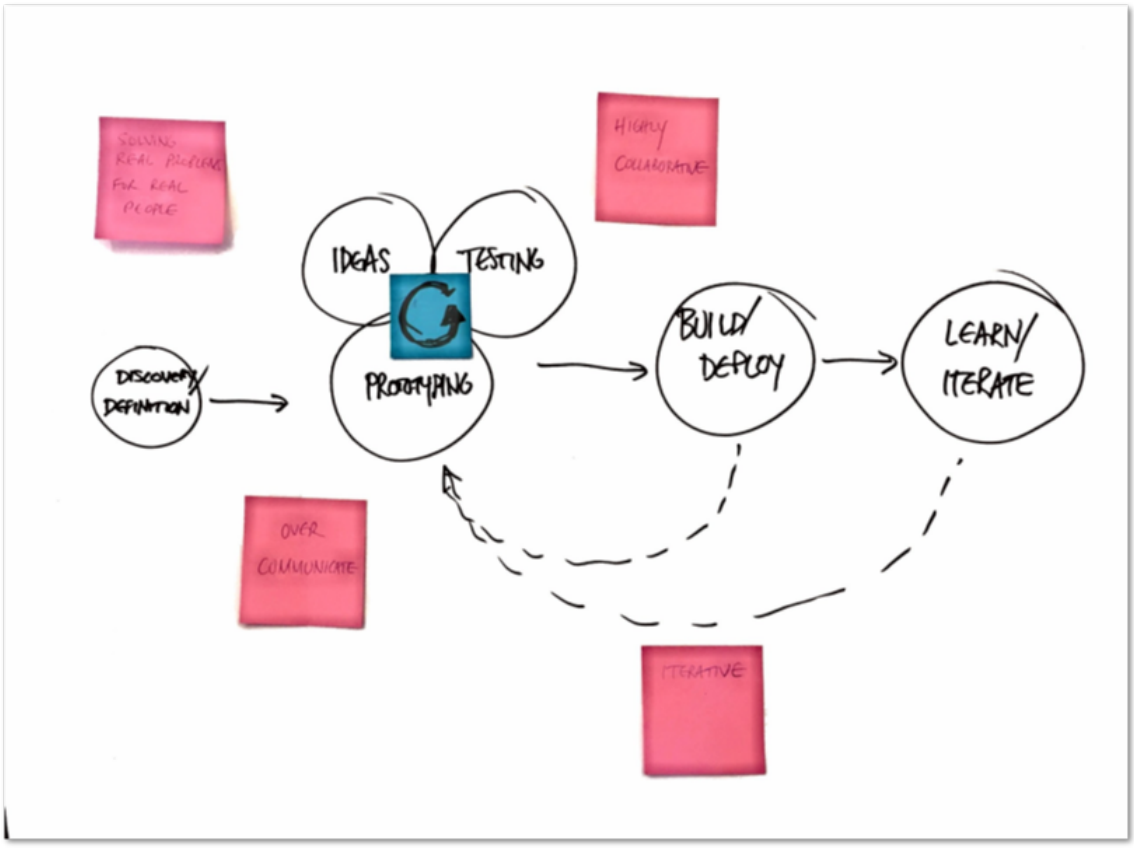

Figure 5.8: The poster made by group 8.

5.3 Workshop at Varvet

The workshop at Varvet was an internal workshop for all employees where they would first define the process as it was today and during the second part define how the process should be in the future. A total of 20 employees were involved in both workshops and the groups were divided in four groups where all professions where represented in every groups. This refers to both designers, developers and managers.

5.3.1 Part 1 - Mapping the Current Process

The mapping of the current process resulted in the following information being brought up.

Today the work process at Varvet was transforming from being driven by development to being more and more design driven.

It was discussed which responsibility Varvet had towards costumers and how that responsibility can be divided internally. It was also mentioned that Varvet should help the customer make well thought through decisions and make the customer smarter, but not do the customer's job.

Many projects at Varvet today are different and it should therefore not be the goal to have the same pipeline for all. The characteristics of the project influence how the collaboration should look.

Most collaboration between roles arise when there is some common ground to stand on, for example a common code base. Generally the collaboration was experienced to work well. Problems occured when the designer moved on to another project while the developers kept working on the first project without the designer and without guidelines. A suggestion was brought up about creating a style guide when the designer leaves the project that the developers can use to make decisions later in the project. It is also experienced that there is a lack of time to iterate during the process. One project was mentioned as an example where a pre-study had been conducted, but that study would have been needed to be followed up on when more decisions were made later in the project. Everyone in the team wants to know what is going on in the project, the purpose of it as well as how it should be done.

There is a strong sense of trust between the designers and the developers at Varvet today. Designers and developers sit near each other and this was an advantage when working together. There is an overlap in knowledge since sometimes the designers implement their own design and sometimes the developers design components.

User studies were often done at the beginning of a project and may be missing in the end of the project when the product should be tested or when new features should be implemented. There exists uncertainty regarding when in the process the different roles should be included. Today there is no clear line between designers and developers, not when it comes to responsibility, nor when it comes to work tasks.

User Experience was considered at certain times, but that kind of material was not always present when making decisions later in a project. The UX designer should not work as a filter between the customer and the developers. Instead the UX designer should be a support who knows a lot about the customer. Nevertheless, there was a request that developers as well should meet the customer and contribute with their expertise early on in the project.

Timing and budget makes it hard to do the project the way Varvet would have wanted to. Communication is a big part of it, and if the communication with the customer is lacking, or hard to maintain, the work suffers.

Ad hoc is an expression that is associated to the process of today. This is seen as an advantage in most cases.

5.3.2 Part 2 - Future Process

This part of the workshop was to determine how the work process at Varvet should be performed in the future.

There are a lot of different kinds of projects and one process would not work for all of them. A suggestion to create three sample projects and how to work within those was brought up. This would lead to having material to show to customers when starting projects and increase the understanding of the work not only within the team, but also towards the customer.

The general opinion was that everyone wanted the process to be more of a collaboration between the different roles. Both UX designers, graphical designers and developers should work continuously together and not be afraid to involve each other and ask for input. All input was considered valuable. Additionally, it was suggested that all roles should be present when conducting user stories, user interviews etc.

The UX designer should be more involved in the backlog and be a representative for the customer. The backlog should be the focus when a project had been started. Sprints and more workshops were suggested. Also that the project group sometimes should take a step back from the backlog to get an overview of the project.

Some said that they felt that they wanted to follow up on the design at a later stage in the project and update it to solve more problems and interactions. This also included having more user tests later on in projects.

Another suggestion was to look over the ratio of designers and developers in the projects to make sure no project was lacking a designer when one was needed. To create a style guide for the project was brought up as a solution for projects with a minority of designers. The documentation of the design seemed important and there seemed to be a need for a tool to share design and structure.

Varvet employees wanted to sit in the same room when working on a project, but then it was important to not forget to document the work for others not spatially included.

5.4 Guidelines

The results presented in the previous sections were analyzed and together with the theory learned during the project, they resulted in a set of guidelines. These guidelines are the main result of the project. They are based on the problems discovered during the project, and are suggestions on how to solve them.

They are created to help cross-functional teams work better and more efficiently together. Note that they are guidelines, not rules that have to be strictly followed. Every team is different and have to find their own way of working that works best for them. These guidelines are a way to help the teams find that way of working. There are eight categories of guidelines, listed below.

5.4.1 Collaboration

When working in cross-functional teams, it is important to work well together and to take advantage of the fact that the team members have different knowledge and expertise.

A working collaboration in a cross-functional team does not mean that each person only is involved in their own area of work, it means that all knowledge is shared at each stage of the process. People know more than their work title, and combining knowledge from different people with different views is a great way to create a working collaboration.

Collaborate Between Companies

We are stronger together. Taking advantage of the knowledge within the company is often very helpful, but sometimes we need to take a step back, see the bigger picture and ask for help outside of the office walls.

Document Your Work

For work to be shared over a long period of time it must be documented. This is not only valuable for your manager or the customer, but for yourself and your team members to remember what you have done and why.

Include Developers During the Pre-study

Not only designers and project managers have a say in the beginning of a project. Developers have important feedback on technical constraints and whether or not the product is possible to develop.

Perform Workshops

Include all parties in a workshop where all input can be put together. This sets a foundation for further work and creates a common understanding.

Ideate Together

Not only designers have good ideas. Try to bend the imaginary walls that are the work titles and involve developers in typical design work.

Plan Your Work Together

Set up goals together and make sure everyone has an understanding of what should be done when. This makes it easier to communicate about the work and creates a common view among the persons involved.

5.4.2 Communication

To communicate in a team is very important. A working communication reduces misunderstanding and makes working together much easier. One problem is that not enough is communicated, which leads to people doing their own thing that then has to be matched together with everyone else's. Even the things that do get communicated can lead to problems since everyone interpret both written and spoken communication in their own way. Communication is tricky, but when it works it is a great tool in helping the project move forward.

Communicate More

To avoid misunderstandings one must communicate often and a lot. This can contribute to problems being found before escalating. You can also avoid misunderstanding by asking the question; "did you mean .... ?" to confirm that you have understood what the other wanted to communicate.

Think Twice...

...before asking a question. You may know the answer just by structuring your problem, and if so, time has been saved for both yourself and your colleague. If not, you have probably gained a better understanding of the problem and can therefore explain it better to your colleague when asking the question.

Communicate Your Design

Not all design can be explained in a sketch, especially when it comes to animations and interactions. Therefore all designers need to be able to verbally communicate their design to developers and customers.

Hold a Presentation at the End of Your Project

Share all learning outcomes from your project, both good and bad. Make sure to learn from your's and others' mistakes and victories.

Dare to Ask Questions

Teams are cross-functional for a reason. There are other people with competencies outside of your own. Take advantage of this and ask questions.

5.4.3 Process

When it comes to the working process in cross-functional teams, it is important to note that each team has their own individuals that work best in a certain way. It is hard to create a process that works for everyone since everyone is different.

Plan Breaks

Working constantly for a long period of time without breaks is something most people find hard. If you are one of those who have a hard time taking breaks; try to include them in your planned schedule.

Encourage Natural Breaks

No one can work for too long without being ineffective. For example, a designer may find it hard to be creative for 8 hours. Create an environment where it is okay to take breaks whenever you need to.

The Result Is More Important Than the Time Spent

Some people work better when they are allowed to work focused over a short amount of time instead of working a full day with more breaks. Allow your coworkers to plan their own schedule in order to get the most out of every day.

Test Early in the Project

Test both the design and the code as early as possible to make sure problems are caught early on to save time and effort at a later stage in the process.

5.4.4 Decisions

Decisions can be hard. It is preferable to know why decisions are made and most importantly; who is responsible for making certain decisions? These are questions that can cause confusion within a team. It is important that the process of decision making is clear within the group to avoid unnecessary conflicts and misunderstandings.

Group Decisions

A project manager is not the only way to control a group. Many groups work best when they themselves are in charge of decisions and planning. This works well when the team members know their own project and because of this can make well thought out decisions.

Areas of Responsibility

A cross functional team is made up of individuals with different skillsets, which can be used beneficially. Create areas of responsibility where the one with the most knowledge in a field makes the decisions affecting it.

Get a Good Product Owner

If possible, get a product owner with knowledge within the field of the project. This person is less likely to set up impossible conditions for the group to work within.

Vote

Whenever a discussion can not solve your problem, try to conduct a vote. When voting, every opinion is equally valued, not only the opinion of the loudest person in the room.

Make Decisions Reversible

If you are unsure about a decision, make one that you know is reversible. That way you can try a solution and later on change your mind.

Track Decisions

When a new coworker enters the team, all former documentation is valuable. As is to be able to track who made decisions and why as well. However, it is not only for those not present at the time the decisions were made, even those present could after a while forget why a certain decisions was made. Track your decisions and the reasons for making them.

5.4.5 Meetings

Meetings are an important and natural part of working. However, the policy of meetings seems to differ a lot between companies, teams and even individuals. How do you maximize the advantages of having meetings without them taking unnecessary time and resources? Also, how do you make sure that those that missed the meeting also know what has been said and decided? These are some problems that need to be solved in order to have an efficient meeting culture.

Have a Start-up Meeting at the Beginning of Every Project

Set a good foundation to all projects by talking it through together. Take time to plan the project and to set up goals and expectations together.

Document Your Meetings

Many coworkers may be interested in what was said at your meeting, without necessarily having to be a part of it. When documenting meetings more people can be involved without making the meeting itself crowded or lengthy. To do this someone has to be responsible for taking notes or in any other way document the meeting. This documentation then has to be shared for everyone to access.

Keep Your Meetings Short and to the Point

Both designers and developers complain about how meetings extract time from their practical work. Keeping the meetings short and to the point will free up time for other tasks.

Communicate Within the Team

Having a working communication within the team, and seeing it as an essential part in working together can reduce the amount of meetings needed.

Time Budget for Meetings

Long and ineffective meetings can be shortened by setting up a budget for how much time per week every individual can spend on meetings. When the budget is gone, the colleague may not attend any more meetings. Because of this, one has to plan their meetings wisely to make them short and valuable.

Meeting Representatives

Choose representatives for your team to attend the meeting. These representatives will then communicate the most important information to the rest of their team.

Have an Agenda

This tip is easier said than done. Many meetings lose track of the important subject and instead is diluted by small talk. Have an agenda for the meeting. An agenda can be used for many things, as you will read in later tips. By making an agenda one has to think through the meeting, thus planning it beforehand. If the agenda is shared to the other participants before the meeting they have a possibility to also be prepared.

Only Book a Meeting If You Have Something to Say

This may sound obvious but you may be surprised if we told you how many meetings that are performed without cause. When the reason for your meeting is "we always have a meeting on Monday morning" then you know that this is one of those. Do not do this to your colleagues.

Invite the Right Participants

Having too many participants in a meeting will make it crowded, long and unfocused. It is also harder for every participant to share their opinion on the subject. Aim to make the participants as few as possible without excluding important parties. Keeping track of how every participant contributes to a meeting may be a way of singling out the necessary parties.

Prepare Feedback Before the Meeting

For everyone to get their chance to speak up during a meeting, feedback can be prepared. This list of feedback is then sent to the moderator of the meeting to make sure that all feedback is brought up during the meeting.

Queue of Speakers

If you ever find yourself in a meeting with someone who tends to take over the entire meeting, then this is a good one. When implementing a queue of speakers everyone gets to speak their mind.

Decide If You Want To Attend Or Not

This tip sets a high demand on the culture of the company. If you have a culture that allows for a colleague to turn down a meeting then there is a lesser chance for you to invite participants that will not get any value out of the meeting. To make this as efficient as possible, share the agenda beforehand. By doing this the participants can make the decisions of however they will participate based on the agenda of the meeting.

Open Invitation

Book a meeting and share the agenda with your colleagues. Then let them announce who will join.

Open Schedule

Not every part of the meeting is relevant for every participant. A solution to this is to schedule open meetings, where colleagues can attend the part of the agenda that they find relevant.

5.4.6 Working Remote

The opportunity to sit anywhere when working can bring a sense of freedom. In other cases it is necessary due to family situations or similar occasions. However, it brings a lot of challenges when working in a team. First and foremost, communication gets harder. It is harder to get hold of people and to contact someone to ask a question feels like a bigger commitment than asking someone in the same room. Reduced communication and communication of lower quality is are risk that are important to be aware of when working remotely.

Avoid Working Remote For Long Periods of Time

In many aspects, working remote can be a bad idea. It can however be useful for charging up creativity and energy by getting away from your usual environment. But do not do this for too long. When working remotely over a long period of time, communication suffers. Written communication can not cover all aspects of a face-to-face conversation. Relationships are better built in real life.

Have Face-to-Face Meetings As Well

Whenever suited or necessary, try to invite your remote workers to participate in real life meetings and events. Make sure that they have met all coworkers in real life at least once to get a sense of who they are working with.

Ask for Feedback

When working remote it is hard to get feedback from others in your project team. To get feedback you need to ask for it.

Use a Good Chat Client

When working remote, communication moves from spoken to being written. Text becomes the main source of information. This must be supported by a good chat client. Today many teams successfully use Slack (2017).

Open Voice Call

The biggest reason for communication being written rather than spoken is the threshold of starting a voice call. Keep an open voice channel in order to lower this threshold.

Include More Senses When Working Remote

When attending meetings through video, make sure to include more of your senses. For example have a good cup of hot tea or the same coffee that they have at the real meeting as this will help you remember the meeting and what was said during it.

Office Live Stream

To feel like you are participating even when working remote, you can stream the activities going on at the working site. This can make remote workers feel included and give them a sense of fellowship.

5.4.7 Team

The team itself is of course very important when it comes to working in cross-functional teams. How do you place the team spatially, which individuals should be in the team and how do you utilize everyone's knowledge to make them reach their full potential?

Sit Together

When sitting together, communication flows. Developers and designers at many different companies have stated that it is both easier to ask questions and to understand each other when sitting closer together. It is when looking at another team members screen or work that we can tell whether they are in the mood for questions or not.

Include Everyone Spatially

During the entire project, from start to finish. Include everyone in the physical space. Even if you sit in the same room, some people may be excluded. You must plan the office in a way that makes it easier to communicate when necessary.

Work in Pairs

Working in pairs creates a natural opportunity to discuss problems and together find solutions. Two heads are often better than one.

Keep a good ratio between designers and developers

Find out which ratio works best in your team, maybe it is more designers than developers, or vice versa. However, make sure no one works alone. Working alone you may find it hard to discuss your field and knowledge with others.

Display Competencies

It is a common mistake to not know about the different competencies that exist within the group. The work titles can work as blinders where it is easy to think that a person with a developer title only knows code. But the thing is, we all know much more than our title displays! In order to take advantage of all the competencies, they must reach the surface to be displayed. Set up a company wiki or talk within the group about each and every team member, their interests and skills.

Portrait All Colleagues on the Wall

Frame a photo together with a small interview where you get to know the person displayed. Show off their skills and interests to make it easier to get to know each other.

Kick Off

When starting off a new project with a new team, plan a kick off! This will help the team get to know each other and to ease future communication. This is a good time for inviting the project owner as well.

Have Breakfast or Lunch Together

Start off your week or day together with your colleagues with a breakfast and use your lunch break for team building. Here you can small talk to build team spirit or talk about the week to come. This is a good foundation when asking for help with problem solving or to get input for further work.

Use the Same Playground

Make sure everyone has the same foundation when working together. For example, decide that all communication is through a special chat client, or that all planning is done in a special planning tool.

Conduct an Online Experience Together

Even colleagues that are working remote need to be included when building up a team spirit. Try to get together in an online experiences where everyone is online and equal. This could for example be an online game.

It Should Be Okay to Say No

In a climate where it is okay to ask a lot of questions, it should also be okay to say no. If you do not have time to answer a question right now, say so.

Keep Track of How People Feel

Your team is constructed out of people with feelings. Take care of your team by asking how they are and be observant when people are down. As Mike Monteiro writes in his book "Design Is A Job"; it takes listening, empathy and understanding of what drives people to learn how to communicate with your coworkers and clients (Monteiro 2012).

Have Enough Personnel

To avoid unnecessary stress, do not have more work than personnel. Team conflicts are enhanced during stressful situations.

Cross-Field Knowledge

All designers should know what is possible to implement within their design, and it is not a disadvantage if being able to implement it by themselves. That way, the designer can sit next to the developers when developing and keep the conversation going about the design. Important design decisions will be communicated and the designer will not design anything that is impossible to build.

Talk About Other Than Work

To really get to know your teammates you need to talk about other than work, because our personal life will influence our work. This creates an important understanding of your teammates and their behaviour.

Indicate when someone is busy

If you do not want to be disturb by someone when you are busy or in a flow, make sure to communicate this for others to know. At some workplaces, this is indicated by that person wearing headphones.

Have a Policy for When It Is Okay to Disturb Others

If you have a problem with questions being asked so much that they disturb the workflow, set a policy for when it is okay to ask questions. For examples, have the afternoon open for questions, but before lunch restricted to working uninterrupted.

Offer Noise Cancelling Headsets

If people at your office have a problem with there being too much noise, make sure they have access to noise cancelling headsets. This also opens up the possibility of people discussing in an open office when they need to, without disturbing everyone else.

Explain Your Work In Everyday Language

In order for all team members as well as the customer to feel included, the communication must be performed in a language that everyone can understand. This does not only refer to an actual speaking language, but also to the terms used when talking.

Mentorships

Having an assigned mentor within, or outside of, the company to talk to about your problems with and to ask for feedback from. This could make sure that feedback are discussed more often.

Be Honest When Things Screw Up

In order to get help from your colleagues, they must know that you need it. If saying that everything is okay when it is not is not going to solve anything. Be honest about the situation and ask for help when things screw up!

Intro to Other Fields

To communicate across fields could be easier if you have an insight into those fields. Getting an introduction by for example shadowing for a day could achieve this.

5.4.8 Transparency

To know what the other team members are up to and what problems and tasks they are facing right now is not always easy. Making your work more transparent to others can be a way of including others in your work. This makes it easier when discussing problems and when making decisions, since everyone is more informed.

Screen Sharing

It is easier to communicate if everyone can see the same view to discuss around. This can be achieved by either standing next to the other colleague's computer or by a screen sharing software.

Share Knowledge Across Teams

We want to include everyone in our projects, even those who do not currently work within this project. Even if they do not work in the cross-functional team, they can have valuable input. When performing a stand up within the company, knowledge can be shared across project boundaries.

Share All Files

To work together during the entire project, everyone needs access to all important information. Set up a way for the entire team to reach all resources, preferably online to include remote workers as well.

5.5 Web Page

To make the guidelines public for everyone to read, a web page was created.

The resulting web page and the requirements behind it is presented below. The web page can be found at www.thecocoguide.com.

5.5.1 Requirements

The requirements were divided into three categories depending on their prioritization, and these are presented below. The guidelines are both regarding technical requirements and functionality for the web page.

5.5.1.1 Prio 1

- Decide on a name for the web page
- Publish the web page
- Create a database to store the guidelines in
- Show the guidelines on the web page
- Show information about the project
- Show contact information
- Show problem areas

5.5.1.2 Prio 2

- Let the user comment on a guideline
- Let the user vote on a guideline
- Sort guidelines according to number of votes
- Let the user choose how to sort the guidelines
- Show the guidelines in different categories

5.5.1.3 Prio 3

- Let the users add their own guidelines
- Create admin accounts
- Let admins delete guidelines
- Let admins edit guidelines
- Let the user search for a guideline
- Show the companies that have contributed to the project
- Add the possibility to upload images for guidelines

5.5.2 First Version

The first version of the web page, and also the current version when writing this report, is presented with functionality and images within this section.

All requirements in Prio 1 and all except "Let the user choose what to sort the guidelines on" in Prio 2 was fulfilled in the first version of the web page.

5.5.2.1 Domain and Database

The name was decided to be COCO Guide, which stands for Communication and Cooperation Guide. The domain for the web page is thecocoguide.com.

The database is located on Firebase, which is a solution from Google to help develop applications. In this project, only the realtime database is used. The database is a NoSQL database which uses JSON data and syncs in realtime (Firebase 2017).

The data in the database is divided into three main parts: Guidelines, Categories and Comments, which can be seen in figure 5.9. In Guidelines all the guidelines are stored. They have a number as the key, which is the same number as their number on the website. Each guideline also has a title, a text, number of votes and authors, consisting of name, work title and company. In Comments, each guideline number is the key to all the comments for that guideline. A comment has an author, with the same attributes as an author for a guideline, a text and a date. In Category, all categories are stored. A category has their category name as the key and then has an introduction text and a list of all the guideline numbers included in that category.

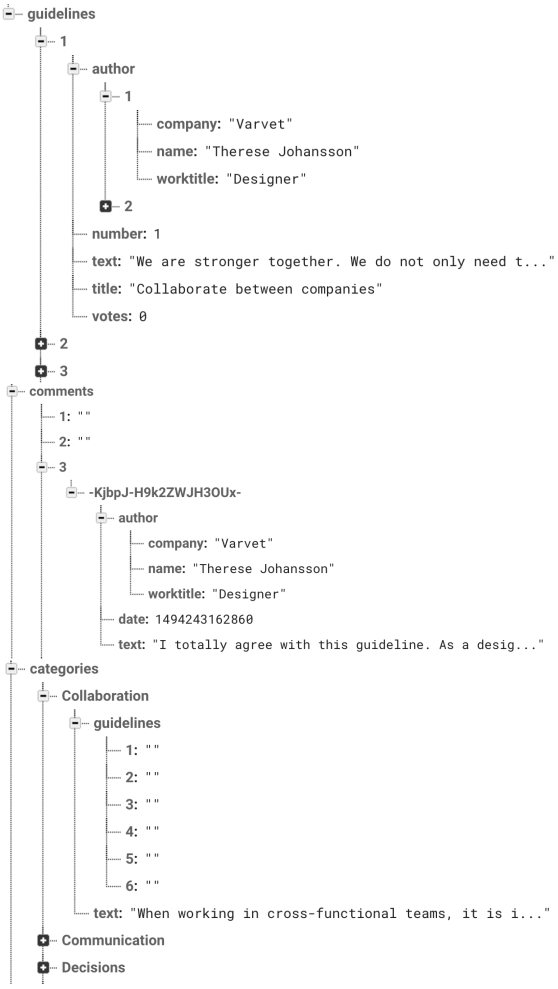

Figure 5.9: A visualization of the database structure.

5.5.2.2 Web Interface

The application is written using React and Redux. React is a component-based JavaScript framework created by Facebook to make it easier to build user interfaces for the web (Facebook Inc 2017). Redux is often paired with React and it is a state container to help handle the state of the application (Redux 2017).

The use of these frameworks lead to the web interface consisting of several components and a common state for the entire application. When something in the state updates, only the components affected by this update are re-rendered, creating a smoother running application.

The interface was based on the prototype described in the section 4.5.1. When the user first enters the web page, the About page is shown. Here, the project is described, as well as the purpose of the web page. This can be seen in figure 5.10.

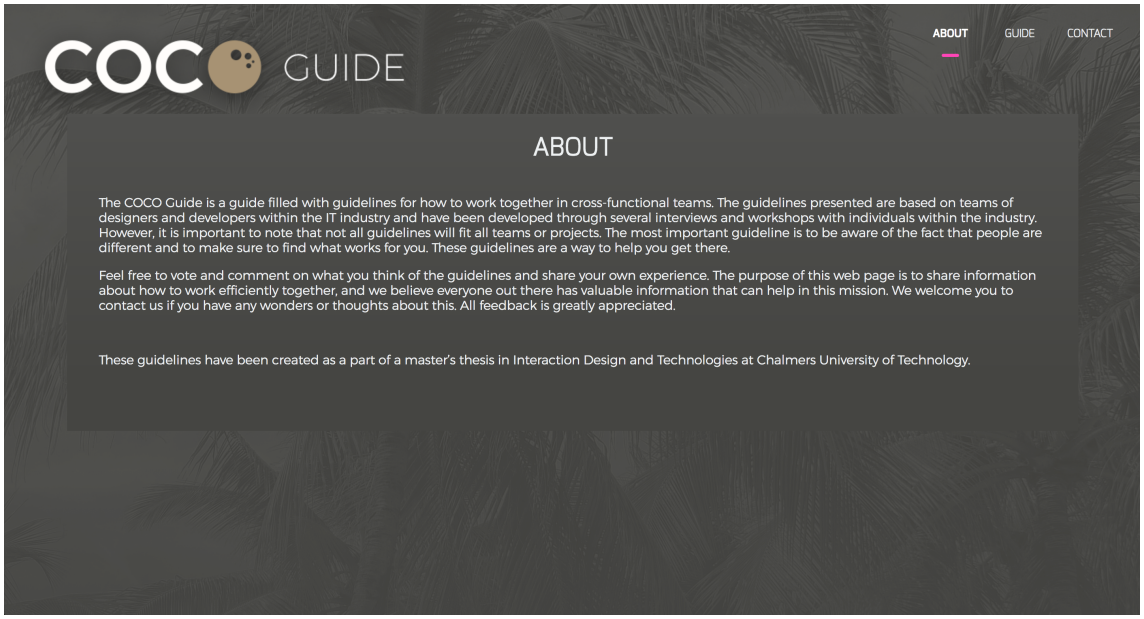

Figure 5.10: The page describing the project.

The user can then choose to go to either the Guide page or the Contact page. When entering the Contact page, the page seen in figure 5.11 is shown. Here the user can see the contact information for the authors of this project.

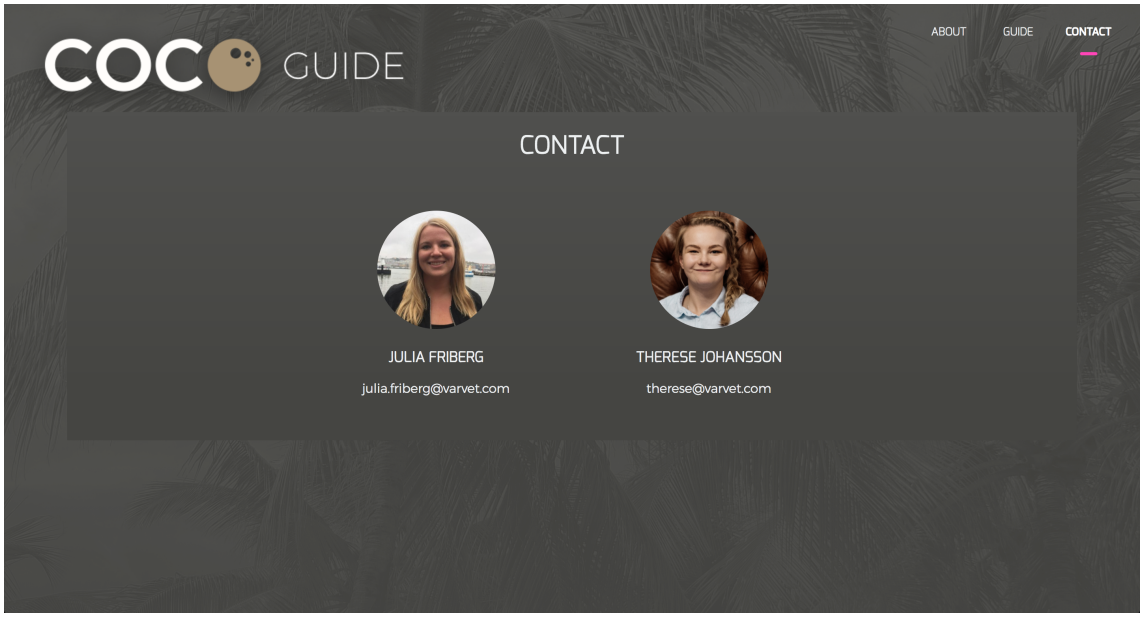

Figure 5.11: The page showing the contact information.

The Guide page is the part of the web page where the guidelines are presented. This can be seen in figure 5.12. To the left, there is a menu with all the different categories for the guidelines. The chosen category is indicated in the menu and fills the content

area that takes up the rest of the space. The category title is at the top of this area, and is followed by an introductory text explaining the area and the problems within this area. After that all guidelines are presented in the order of most votes to ensure the most important guidelines are seen first. Each guideline has a number to make it easier to refer to guidelines and to go back and find guidelines again. Each guideline also has a title and a text describing the guideline. The number of votes and the number of comments can also be seen. To view the comments, the guideline can be clicked. This expands the guideline and displays all the comments, in order of the date they were added, followed by a form where the user can comment themselves.

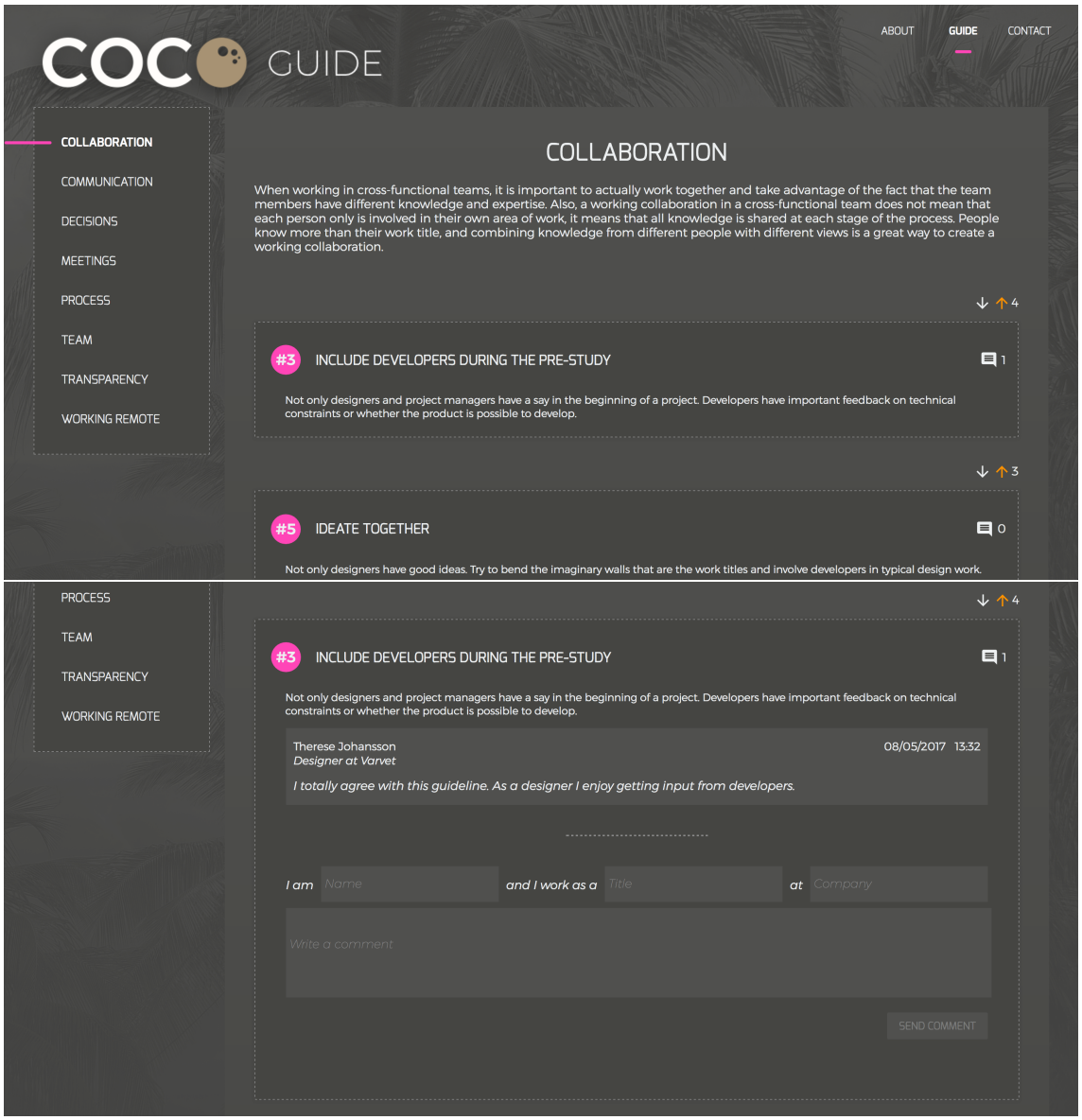

Figure 5.12: The page showing all the guidelines. The image on top is when no guideline has been pressed, while the lower image illustrates this feature.

6

Discussion

6.1 Discussion of the Result

The purpose of the project was to answer the following research question:

What is needed in terms of recommendations and methods in order to facilitate communication and teamwork within cross-functional teams in the IT sector?

Has this question been answered by this project? Since the main part of the result are guidelines regarding communication and teamwork within cross-functional teams, these can serve as a recommendation. If exactly these guidelines are what is needed can not be determined without studying the effects they have in projects. Since this has not been done, it is not certain that these guidelines would improve the communication. However, the data discovered by research and interviews suggests that following these guidelines could help solve many of the problems that that the companies that we have talked to have experienced. It is important to be aware of that following each of these guidelines to the letter will probably not help. As the name suggests, they are only guidelines. Since each person is individual and each team works in a different way, not all guidelines are suited for all teams. What was clear during this project was that creating a process and rules for all to follow would not work. Each team needs to find their own way of working, and the guidelines presented in this project is a way to help them on their way to doing that.

To follow all guidelines is not an option since many of them are contradicting. For example the guideline regarding taking natural breaks is contradicted by the guideline suggesting to plan breaks in your work. These two guidelines suit different individuals and it is therefore important to learn what suits yourself and the individuals within your team best. Some guidelines might not work for everyone, and some might be hard to realize. Having a transparent work flow might be a really good idea for involving people in each other's work, but it could be hard to actually get people to open up their work for everyone to see at all times. As important as transparency is when working in a team, privacy is just as important. People might feel watched or experience an increase in pressure if everything they do can be seen by everyone else within the team. Some are not comfortable with sharing

their work when it has not yet been finalized. Therefore, it is important to stress that the guidelines are not to be followed strictly. Although they are meant to ease the communication in cross-functional teams, but they should be individualized by the team itself for them to work at their best.

The purpose of presenting the guidelines on a web page is to spread the information to a bigger crowd, both to help them and to let them share their own experiences and thoughts on the subject to help others. That way the guidelines are always improved and kept up to date by those working and experiencing the problems in cross-functional teams everyday. The possibility for this kind of collaboration between people within the industry could bring a lot of knowledge and experience into the guidelines that would not be possible to collect within the time limit of this project. It is a desire that this project will carry on once this thesis is done.

However, bringing together a lot of people and letting them freely contribute to a web page is not all about advantages. There is a possibility that the web page could be used in the wrong way. Since there is no way to moderate the comments or change the votes without actually editing the database, people can write whatever they want. This means that the web page could be used to spread unwanted information or to subject certain people to bullying. This is not uncommon on the internet and is therefore a problem that could be very real. If this web page is to be taken further, this problem needs to be addressed.

Since the web page only has been released as a first version, there are some problems that might occur once the usage of it increases. All data is retrieved from the database when the web page is loaded, which could be a problem if the amount of data increases. If for example the guidelines would explode in numbers, a limit on how many to retrieve from the beginning must be set. The remaining guidelines can be retrieved when needed. The decision to retrieve all data at the same time was taken due to the fact that there is not that much data at the time of writing this report, and that the user experience is improved by not letting the user wait while the data is loaded. This is also the reason for the change of start page at the web page to the "About"-page. In the prototype, the guide was the starting point, while in the first version the "About"-page is the first thing the user encounters. This is done to let the data load in the background, but also to introduce the user to the project and the guidelines before actually reading the guidelines. This is better for novice users, but not for users visiting the page often. In the future, once the project is known by the users, it makes more sense to redirect the start page to the "Guide"-page instead.

6.2 Discussion of the Process

At the start of the project, the focus was to develop a tool for designers to share their design with other designers to receive feedback on their work. This was the task that the project was planned for, and since the focus changed, the planned process differ a lot from the actual process.

The planned process had two main phases. The first half of the project was planning, data gathering and defining the product, while the second phase was the development and evaluation of the product. When starting with the data gathering and the interviews it was discovered that even though the collaboration between designers was a problem, the even bigger problem was the collaboration between designers and developers. This caused the focus to change, and therefore also the process. A lot of the initial plans could still be kept, but at this point it was not clear what the end result should be, which made it hard to plan the last part of the process without further studies.

The more data that was gathered, the clearer it got that it was not a product that would solve all the problems discovered and this caused the process to change further. Even though a web page actually was created, the web page itself was not the important result, but merely a tool to present and share the findings. The second half of the project that previously had been planned for development was exchanged in favor of a much longer data gathering and ideation phase. This had both advantages and disadvantages. Having a longer data gathering phase made sure more relevant data could be gathered, which was very helpful when moving into the ideation phase. The longer ideation phase was also successful since it brought the possibility of involving more people by doing workshops. However, spending more time on these aspects of the project made the prototyping and development suffer. Having more time on these would probably have resulted in an earlier release of the web page, which would have brought the valuable possibility of evaluating both the web page and the guidelines presented on it by crowdsourcing.

6.2.1 Planning

The planning of the project was performed during four weeks. Having such a long planning phase contributed to a well thought out plan and background for the project. However, some of the time spent on planning felt a bit wasted once the focus changed, and with it the plan as well. It would probably have been better to get out and verify the problem before planning the entire project around a question that later on changed. With that being said, the time spent on planning was still valuable and created an understanding of the project and a verified scope.

6.2.2 Data Gathering

This phase included many interviews. The first interviews were only held with designers because of the initial focus of the project. Quite quickly it was realized that the focus had to be changed, which was done directly. The quick change was good because it meant that only a little time was lost focusing on the wrong thing.

Having performed a few interviews with only designers probably influenced the project towards a more biased view towards designers, and there were more interviews done with designers than with developers. This was not experienced as a problem, but it would probably have been better to have an even distribution between designers and developers. The ratio between designers and developers during this study therefore did not represent the reality at many companies. To represent reality, more developers than designers should instead have been included in the study.

The interview questions transformed during this phase, which means everyone did not get the exact same questions. Questions were swapped during the interview process based on the answers, questions who gave valuable answers were added and kept and those not contributing to new valuable data were removed. The interviews were semi-structured, and this was a way to get as much as possible out of the interviewees since the goal was to get their opinion on the collaboration and what their experience was. This was a successful tactic, but it might have been good to go back and ask follow up questions to those in the beginning when new things were discovered later on to get their view on those subjects as well.

It would have been better to have done more interviews at additional companies, but also to do more interviews at the companies already included. However, the last interviews did not give enough new information, rather only confirmed the problems already found. Thus a saturation of data was reached. With the time constraint of this project, maybe this phase should have been finished even earlier to make room for focusing on other parts of the project.

The first interviews were transcribed, which took a lot of time. This process could have been skipped entirely since there already existed notes from the interviews, which themselves were enough to remember what had been said. When needing quotes, the interviews were taped and could be listened to to find the exact quote if it was not in the notes.

Structuring the data in a KJ analysis felt like a really good way to get clear problem areas, which also made it easier when moving on to ideation. However, finding the important data in the interviews was very time consuming and could have been done right after each interview instead of a few weeks later.

6.2.3 Ideation

The use of the "How might we"-method turned out to work very well during this project. It had not been used by either of us before, but the trick of phrasing the problems as questions of how to solve them instead of only stating them really helped to enhance the creativity. The answers to these questions were helpful when writing the guidelines since the answers then were naturally formulated as solutions, and therefore also guidelines.

It was difficult to write questions that were open enough to generate many ideas, but slim enough to generate valuable solutions. More time could have been spent on developing the questions to get the maximum result out of this method. Also, since there were many stated problems, there were also many questions. Even though each question did not take long to answer, the process of ideating through all of them had to be divided in to sessions over a few days. This was good since it was hard to be creative for a longer period of time. Unfortunately it did also contribute to us forgetting what had been said the last time and therefore it took some time starting up the ideation at each session.

The workshops that were held with the Got UX meetup group were helpful in getting input and ideas from a lot of people in a short time. Additionally some of the participants during the workshop appreciated to reflect on their situation and how they wanted to work in the future. This was something that felt important, since the project is meant to help that exact user group.

6.2.4 Prototyping and Development

The development phase was a bit rushed. Prototyping was done by discussing and sketching quickly. No other users were involved during development and no evaluation of the interface was done, which of course would have been better. However, the interface is not that complicated and there is not much the user can interact with.

The process of developing the web page was not long and no special methods were used. The code was shared through git, but since only one of us was developing the code, that did not mattered much.

6.2.5 Evaluation

The evaluation phase was not directly included during the project work. Nevertheless, the project work prepared for a future evaluation phase through crowdsourcing. By voting on valuable guidelines, editing them and adding new guidelines the COCO-guide is a collaborative space for designers and developers. This however depends on the usage. Only when used, the guide can be developed further.

Evaluation could also be done by letting companies adopt the guidelines and evaluate how they work for them. This would take a lot of time since only one guideline at a time should be tested to make sure to evaluate that certain guideline.

6.3 Future Work

Future work for this project includes evaluating the guidelines through crowdsourcing, evaluating and possibly redesign the web page and of course to fulfill all the requirements for the web page.

Evaluating the guidelines will come naturally if the web page is spread and people start to share their opinions. Therefore an important part of the future work is to make sure that users get aware of this guide. As a starting point in doing this, the guide will be presented at the development conference Ship It! in June 2017 (Varvet AB 2017).

Once all the requirements have been fulfilled, the feature for a user to add their own guideline will exist, which also will lead to developing the guidelines further. For the remaining requirements, a prototype needs to be developed as well since those existing only cover the fulfilled requirements for Prio 2. The design of the web page has not been tested by anyone, which should be done as future work for it to be as intuitive as possible.

One feature that has not yet been implemented is the one regarding admin accounts. The thought with admin accounts was for them to have the ability to change and delete guidelines, and also to delete comments and function as moderators. This requires that the people with access to these accounts are trustworthy and that they do not abuse the responsibility they have been given. Under the right circumstances, this could lower the risk of inappropriate material being published within the guide, such as comments that could hurt users.

Right now, there is no way for regular users to create accounts. This possibility is something that can be looked into to see what advantages and what disadvantages that could bring. Possible advantages could be that the user could save information like name and work, and maybe even a picture to accompany these when comment-

ing or adding guidelines. When people can add their own guidelines, it might be reasonable to also show the author of the guideline, which today is saved but not displayed. Another possibility if users had accounts would be for them to build up a reputation, depending on votes. This is a system that can be seen at for example Stack Overflow (2017) where the reputation is a measurement of how much the community trusts your knowledge. A possibility for further developing the web page could be to make it become a community where people can ask questions and share their experience in a more relaxed environment. Thus making a good reputation more desirable.

For the project to become successful, the web page would have to be used by a lot of people. The purpose of sharing information and experience requires that you have someone to share it with. Users are important for this project to take off and become something bigger than what it is today. This can be done in several ways, and a start could be to share it on social media. Another way to go could be to start by spreading it to those that have been involved during the project. They have shown a lot of interest in the project, which is an indication of this being a hot topic for designers and developers.

6.4 Generalizability

The project restrictions were that only designers and developers at companies within the IT industry, with offices in Gothenburg with less than 35 employees, should be included. The first step towards generalizing this project would be to expand to other parts of Sweden and other countries. This would bring in other cultures, which could be very interesting and give a lot of new information and knowledge. This would add another dimension to the guide. Also, increasing the number of employees at the companies could be done. Since employees at bigger companies work in a different way than those working at smaller companies, resulting in them having other problems to solve, they could add new input to the guide as well.

A more interesting generalization would be to move outside of the IT industry and include other industries, since cross-functional teams exist in a lot of industries, such as for example the building industry. An architect designs a house, just like a designer designs a interface, while engineers and building workers are to implement it. Product development has the same structure in their teams as well. However, it is not clear if the way of sharing the guidelines would work as well in these industries. Letting people share their own experience to help others improve their work is appreciated and very common within the IT industry, through for example meetups and open source projects. This could be a result of the IT industry being fairly new, leading to people discussing and helping each other take the industry further, trying to "catch up" to the industries that have been around for much longer.

7

Conclusion

The goal of this project was to answer the question:

What is needed in terms of recommendations and methods in order to facilitate communication and teamwork within cross-functional teams in the IT sector?

This question has been answered through creating guidelines for how to work in cross-functional teams of designers and developers. If these guidelines really do help has not yet been determined. This will be shown over time once the usage of the web page where the guidelines are presented increases. Nevertheless, the gathering of information from how people in this industry work today and what problems they experience has been the main foundation on which these guidelines have been based on. There is still a long way to go when it comes to testing these guidelines in real projects, and by sharing them on a web page where people can vote and add their own experience of the guidelines it is believed that this could be done. Sharing the knowledge and reminding people of what problems can arise, and suggestions on how to solve them, will hopefully help them in their communication and teamwork within cross-functional teams. Because of the high interest shown to this project, the hopes are high for a valuable tool to grow strong over time.

Bibliography

Ahrnell, Britt-Marie and Rikard Wildhuss (1999). Swedish. Uppsala Publishing House AB. ISBN: 91-7005-168-2.

allabolag.se (2017). *Football Addicts AB*. URL: http://www.allabolag.se/5568808405/footballaddicts-ab.

Buchanan, Robert (1992). "Wicked Problems in Design Thinking". In: *Design Issues: Vol. VIII, Number 2*.

Business Dictionary (2017). "What Is Cross-Functional Teams?" In: URL: http://www.businessdictionary.com/definition/cross-functional-team.html.

Chauncey, Wilson (2013). *Brainstorming and Beyond*. Elsevier Inc. Chap. 1-3.

Cockburn, Alistair (2002). *Agile Software Development*. English. The Agile Software Development Series. Pearson Education Inc. ISBN: 0201699699.

Creuna (2017a). *Medarbetare*. URL: http://www.creuna.se/om-oss/medarbetare/kategori/goteborg/.

– (2017b). *Who we are*. URL: http://www.creuna.com/en/who-we-are/.

Design Kit (2017). *How Might We*. URL: http://www.designkit.org/methods/3.

Facebook Inc (2017). *React*. URL: https://facebook.github.io/react/.

Firebase (2017). *Firebase Realtime Database*. URL: https://firebase.google.com/products/database/.

Football Addicts (2017). *About us*. URL: https://www.footballaddicts.com/.

Github (2017). *Github*. URL: https://github.com/.

Grier, David Alan (2013). *Crowdsourcing for Dummies*. English. John Wiley Sons, Ltd. ISBN: 9781119943846.

Inc.com (2017). "Cross-Functional Teams". In: *inc.com*. URL: https://www.inc.com/encyclopedia/cross-functional-teams.html.

Intunio (2017). *What We Do*. URL: http://www.intunio.se/.

Jensen, Mikael (2015). Swedish. Studentlitteratur. ISBN: 978-91-44-10130-9.

Karlsson, MariAnne (2007). *Lyssna till kundens röst*.

Koloski, Beth (2012). "Don't Have a Meeting, Throw a Workshop". In: URL: http://uxmag.com/articles/don%25E2%2580%2599t-have-a-meeting-throw-a-workshop.

Maaike Kleinsmann, Rianne Valkenburg (2010). *Barriers and enablers for creating shared understanding in co-design projects*. Tech. rep. Delft University of Technology, Faculty of Industrial Design Engineering, Department of Product Innovation Management, Landbergstraat 15, 2628 CE Delft, The Netherlands.

Maaike Kleinsmann Jan Buijs, Rianne Valkenburg (2010). "Understanding the complexity of knowledge integration in collaborative new product development teams: A case study". In: *www.elsevier.com/locate/jengtecman*.

Moggridge, Bill (2007). *Designing Interactions*. English. MIT Press. Chap. 10. ISBN: 978-0262134743.

Monteiro, Mike (2012). *Design is a Job*. A Book Apart. ISBN: 9781937557041.

Monterio, Mike (2014). *You're My Favorite Client*. English. A Book Apart. A Book Apart. ISBN: 9781937557140.

Otter, Ad den and Stephen Emmitt (2008). "Design Team Communication and Design Task Complexity". In: *www.earthscanjournals.com*.

Pivotal Tracker (2017). *Pivotal Tracker*. URL: `https://www.pivotaltracker.com/`.

Preece, Jennifer (2002a). *Interaction Design: Beyond Human-Computer Interaction*. English. John Wiley & Sons, Inc. Chap. 8, pp. 240–250. ISBN: 0-471-49278-7.

– (2002b). *Interaction Design: Beyond Human-Computer Interaction*. English. John Wiley & Sons, Inc. Chap. 9, pp. 304–305. ISBN: 0-471-49278-7.

ProjectPuzzle (2017). *ProjectPuzzle*. URL: `https://www.projectpuzzle.com`.

Redux (2017). *Redux*. URL: `http://redux.js.org/`.

Schwaber, Ken and Jeff Sutherland (2016). "The Scrum GuideTM". In: URL: `http://www.scrumguides.org/docs/scrumguide/v2016/2016-Scrum-Guide-US.pdf#zoom=100`.

Shore, James (2007). *The Art of Agile Development*. English. O'Reilly Media, Inc. ISBN: 978-0-596-52767-9.

Slack (2017). URL: `https://slack.com/`.

Spool, Jared M. (2004). "The KJ-Technique: A Group Process for Establishing Priorities". In: *uie.com*. URL: `https://articles.uie.com/kj_technique/`.

Stack Overflow (2017). *What is reputation? How do I earn (and lose) it?* URL: `https://stackoverflow.com/help/whats-reputation`.

Tague, Nancy R (2004). *The Quality Toolbox*. English. WASQ Quality Press, pp. 96–99.

Toxboe, Anders (2016). *How to facilitate and plan a UX workshop*. URL: `http://ui-patterns.com/blog/how-to-facilitate-and-plan-a-ux-workshop`.

Trello (2017). *Trello*. URL: `https://trello.com`.

Uptive (2017). *We are Uptive*. URL: `http://www.uptive.se/#what-we-do/`.

Varvet (2017). *About*. URL: `https://www.varvet.com/about/`.

Varvet AB (2017). *Ship it! A conference about delivering great software*. URL: `https://shipitconf.org`.

Wadsworth, Yoland (2011). *Do It Yourself Social Research*. English. Taylor and Francis. Chap. 2. ISBN: 978-1-61132-108-1.

Wint (2017a). *Vad kan vi hjälpa till med?* URL: `https://wint.se/Kontakt`.

– (2017b). *Vi vill inte sortera dina papper*. URL: `https://wint.se/OmWint`.

A

Questions for Designers

Hur många arbetar på företaget?
Hur många utav de anställda jobbar som designers?
Hur många utav de anställda jobbar som utvecklare?
Arbetar ni i team? Hur ser de ut i så fall?
Vart sitter era kollegor? På plats? Remote?
Vilket område arbetar du med?

Hur sker kontakten med kund?
Hur sker kontakten med utvecklare?
Hur mycket skulle du föredra att prata med utvecklare/kund/uppdragsgivare?
Vart sitter de du arbetar med i förhållande till dig på kontoret? På plats? Remote?

Hur ser en vanlig arbetsdag ut för dig?
Beskriv designprocessen på företaget. Hur funkar denna? Vad saknas? Vad är bra?
Jobbar de iterativt eller enligt vattenfallsmetoden? Vad görs för tester?
Vad är en typisk leverabel ifrån dig?
Vart i designprocessen uppstår fel? Hur hanteras detta?

Vad har ni för kommunikationskanaler? Hur funkar dessa? Vad saknas? Vad är bra?
Vad diskuteras i vilken kanal? Hur funkar det?
Använder ni digitala verktyg för att samarbeta?
Vad är nyckeln till att kunna använda ett digitalt verktyg på ett effektivt sätt?
Vilka arbetsverktyg/program använder sig en designer av på företaget?
Vilka verktyg önskas?

Hur ofta har ni fysiska möten? Hur funkar de? Vad saknas? Vad är bra?
Vad diskuteras på mötena? Vad är det som gör att de är fysiska?
Vilket ansvar har en designer på Uptive?

Hur mycket vet du om ett projekt när det startar?
Vad är det första du får veta om ett projekt?
Vem ger dig denna information?
Är den tillräcklig? Vad saknas? Vad finns?

Vem startar projektet?
Hur styrs projektet?

Vem tar designbeslut? Med vilken bakgrund?
Hur hanterar ni meningsskiljaktighet?
Vilken typ av meningsskiljaktigheter är vanliga? Ovanliga?
Hur hanterar ni återkoppling/feedback från kunder och kollegor? Hur ofta? Vad brukar kommenteras på? Vilken detaljnivå ligger det på?
Vad brukar diskuteras mest under processen?

När lämnar du ifrån dig designunderlag och hur ser det ut?
Vilken feedback brukar du få från utvecklare och hur?
Vilka frågor uppstår kring designen?

Hur mycket vet du om hur era utvecklare jobbar?
Hur involverad är du i utvecklarens jobb? I vilket stadie blir utvecklare involverade?
Hur involverad är era utvecklare i ditt arbete?

Har du någon erfarenhet att arbeta på distans?
Hur ändras ditt arbetssätt? Vad tycker du om distansarbete?

B

Questions for Developers

Hur många arbetar på företaget?
Hur många utav de anställda jobbar som designers?
Hur många utav de anställda jobbar som utvecklare?
Arbetar ni i team? Hur ser de ut i så fall?
Vart sitter era kollegor? På plats? Remote?
Vilket område arbetar du med?

Hur sker kontakten med kund?
Hur sker kontakten med designers?
Hur mycket skulle du föredra att prata med designers/kund/uppdragsgivare?
Vart sitter de du arbetar med i förhållande till dig på kontoret? På plats? Remote?

Hur ser en vanlig arbetsdag ut för dig?
Beskriv arbetsprocessen på företaget. Hur funkar denna? Vad saknas? Vad är bra?
Jobbar de iterativt eller enligt vattenfallsmetoden?
Vad har ni för kommunikationskanaler? Hur funkar dessa? Vad saknas? Vad är bra?
Vad diskuteras i vilken kanal? Hur funkar det?
Använder ni digitala verktyg för att samarbeta?
Vad är nyckeln till att kunna använda ett digitalt verktyg på ett effektivt sätt?
Vilka arbetsverktyg/program använder sig en utvecklare av på företaget?
Vilka verktyg önskas?

Hur ofta har ni fysiska möten? Hur funkar de? Vad saknas? Vad är bra?
Vad diskuteras på mötena? Vad är det som gör att de är fysiska?

Hur mycket vet du om ett projekt när det startar?
Vad är det första du får veta om ett projekt?
Vem ger dig denna information?
Är den tillräcklig? Vad saknas? Vad finns?
Vem startar projektet?
Hur styrs projektet?

När får du tillgång till designunderlag och hur ser det ut?

Hur levereras underlaget för ditt arbete?
Vilken feedback brukar du ge designers och hur?
Vilka frågor uppstår kring designen?
Vad är viktigt i underlaget?
Vad måste finnas med i underlaget för att du ska kunna genomföra ditt arbete?

Hur mycket vet du om hur era designers jobbar?
Hur involverad är du i designers jobb? I vilket stadie blir utvecklare involverade?
Hur involverad är era designers i ditt arbete?

Vem tar designbeslut? Med vilken bakgrund?
Hur hanterar ni meningsskiljaktighet?
Vilken typ av meningsskiljaktigheter är vanliga? Ovanliga?
Hur hanterar ni återkoppling/feedback från kunder och kollegor? Hur ofta? Vad brukar kommenteras på? Vilken detaljnivå ligger det på?
Vad brukar diskuteras mest under processen?
Vart i arbetsprocessen uppstår fel? Hur hanterar ni detta?

Har du någon erfarenhet att arbeta på distans? Hur ändras ditt arbetssätt?
Hur ser du på att andra arbetar på distans?
Vad krävs för att du eller någon annan ska kunna arbeta på distans?